Robust Constrained Markov Games: Multi-Agent Decision-Making under Model Uncertainty and Constraints

Ningkang Chang[1,4] **Chenyu Xu**[2] **Ziying Jia**[4] **Yue Wang**[3] **Sihong He**[4]

[1]The University of Tokyo, Tokyo, Japan
[2]Iowa State University, Iowa, USA
[3]University of Central Florida, Orlando, USA
[4]University of Texas at Arlington, Texas, USA

Abstract

Multi-agent decision-making under both uncertainty and constraints is a central challenge in safety-critical domains such as autonomous driving, where agents must coordinate in uncertain environments while ensuring feasibility with respect to safety and resource limits. However, there is limited theoretical and practical work that addresses this challenge. Therefore, we introduce a *Robust Constrained Markov Game* (RCMG), the first multi-agent framework that simultaneously captures adversarial transition uncertainty and cost constraints. We further propose a solution concept, the *Robust and Feasible Nash Equilibrium* (RFNE), and a principled relaxation that yields a tractable surrogate upper bound to the intractable problem of computing an exact RFNE. This surrogate enables both a fixed-point existence proof via Kakutani's theorem and a decentralized algorithm that alternates robust dynamic programming with Lagrangian dual updates. In settings where the relaxation is tight, the method provably recovers an exact RFNE. A grid-world experiment illustrates consistency with the theoretical results and highlights the viability of RCMGs as a foundation for reliable multi-agent decision-making under uncertainty and constraints.

1 INTRODUCTION

Multi-agent decision-making in dynamic and uncertain environments has emerged as a key research challenge, particularly when safety and resource constraints must be simultaneously satisfied [9]. For instance, in autonomous vehicle fleet management, dozens to thousands of vehicles must jointly plan routes and maneuvers, enforce collision-avoidance and right-of-way rules, respect energy/battery and service-level constraints, and react to uncertain interactions with human drivers, sensing noise, and exogenous disturbances (e.g., weather and road closures). An equally demanding domain is multi-UAV emergency response and infrastructure inspection, where aerial robots coordinate coverage and handoffs under no-fly zones, communication limits, and tight flight-envelope and battery constraints, while facing uncertain winds, stochastic task arrivals, and rapidly evolving hazards. These deployments require decisions that are simultaneously robust to model misspecification and operationally feasible under hard constraints. Similar challenges arise in robotic coordination [37], intelligent transportation systems [20], and resource management [11], where agents must optimize objectives subject to constraints on cost, safety, or fairness.

Markov games (MGs) [31] provide a powerful framework for multi-agent decision-making. However, MGs often rely on strong assumptions, such as no uncertainties and unconstrained cost, which limit their applicability in real-world settings. To address these limitations, robust Markov games (RMGs) [18, 19, 23, 38] were proposed to handle uncertainties in transition dynamics, rewards, and observations, and constrained Markov games (CMGs) [1] were proposed to incorporate cost constraints. However, they fail to address the simultaneous presence of uncertainty and operational constraints, a critical requirement in real-world applications.

Therefore, we propose *Robust Constrained Markov Games* (RCMGs), a novel framework that unifies the robustness of RMGs and feasibility of CMGs into a cohesive model for decision-making under both uncertainty and constraints. We formally define RCMGs and introduce a generalized solution concept that simultaneously addresses robustness and feasibility. This integration raises several challenges. First, a naive combination of RMG and CMG leads to an intractable formulation involving worst-case transitions and auxiliary cost constraints, making standard equilibrium analysis difficult. Second, while Nash Equilibrium (NE) existence is well studied in RMG and CMG individually [1, 17, 33, 38], the structural changes in RCMG, such as altered Bellman

Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence (UAI 2026), PMLR 337:1003–1033.

operator properties, complicate its analysis. Finally, even if a tractable formulation is achieved, computing a convergent equilibrium solution remains non-trivial, and empirical validation adds further complexity. To address these challenges, this work makes the following contributions:

- We propose the first formal definition of Robust Constrained Markov Games (RCMGs), a unified framework that jointly captures robustness to transition uncertainty and feasibility under operational constraints. To overcome the intractability of directly combining RMG and CMG, we propose a two-step relaxation that yields a tractable surrogate objective (11) and the associated solution concept, the Upper Robust and Feasible Nash Equilibrium (Upper-RFNE), distinguishing our setting from RMG- or CMG-only formulations.
- We prove the existence of Upper-RFNE by constructing a composite set-valued operator that integrates robust best-response mappings with dual variable updates. We show that this operator is upper semicontinuous with convex and compact images (Lemma 5.6), which allows us to invoke Kakutani's fixed-point theorem to guarantee equilibrium existence (Theorem 5.7). Our analysis clarifies how the interplay of robustness and feasibility alters classical Bellman operator properties, providing new insights into equilibrium structure in constrained uncertain games.
- Building on the theoretical results, we develop a decentralized algorithm (Algorithm 1) that alternates between robust policy optimization and subgradient-based dual updates, directly targeting the surrogate objective. The algorithm is built from convergent components (robust dynamic programming, subgradient dual updates, and a Kakutani-based existence guarantee), so that any joint policy at which it stabilizes constitutes an Upper-RFNE. We further validate the method in a two-agent grid game where the surrogate is provably tight, demonstrating convergence to the theoretical equilibrium. These results, though based on a minimal grid-world, validate that the algorithm enforces feasibility through dual updates and recovers the analytically derived equilibrium values, thereby supporting the theoretical soundness of the RCMG framework.

2 RELATED WORK

Robust Markov Games (RMGs) RMGs enhance standard MGs by modeling uncertainty in transitions, rewards, or strategies, enabling robust performance in adversarial environments. Zhang et al. [38] introduced RMGs into MARL, developing robust Q-learning and actor-critic algorithms. Subramanian et al. [33] analyzed robustness in general-sum MGs, linking approximate equilibria between original and surrogate games. Ma et al. [27] proposed decentralized robust V-learning for computing correlated equilibria under uncertainty. Reinforcement learning in RMGs was further developed under different settings, including generative-model [32], offline [6, 25], and online [15, 40] settings. McMahan et al. [28] showed that s-rectangular RMGs admit regularized solution methods. Despite these advances, existing RMGs focus solely on robustness, neglecting crucial safety or resource constraints.

Constrained Markov Games (CMGs) CMGs introduce constraints into MGs to regulate agent behavior or outcomes. Altman and Shwartz [1] established the foundational theory of constrained NE. Subsequent works extended equilibrium concepts and algorithms: Hakami and Dehghan [17] proposed constrained no-regret Q-learning for correlated equilibria; Jiang et al. [21] addressed deterministic policies in continuous domains; Chen et al. [10] introduced the first primal-dual learning method for correlated equilibria in CMGs; Ding et al. [12] developed a safe online MARL algorithm with regret guarantees; and Jordan et al. [22] proposed decentralized learning in CMPGs. These approaches enforce feasibility but do not account for adversarial uncertainty, exposing learned policies to potential failure in real-world settings.

Robust Constrained MDPs (RCMDPs) RCMDPs unify robustness and constraints in single-agent settings by minimizing worst-case cost subject to safety or budget limits. Notable developments include Lagrangian-based solutions [30], robust primal-dual methods [36], and sample complexity analysis [16]. Other contributions span distributional robustness [39], epigraph reformulations [24], and policy gradient methods [7, 34]. While RCMDPs inspire our approach, they do not address multi-agent interactions, motivating our formulation of RCMGs, which extend robust and constrained decision-making to strategic environments.

3 PRELIMINARIES

Constrained Markov Games A CMG [1] is defined by a tuple $(\mathcal{N}, \mathcal{J}, \mathcal{S}, \{\mathcal{A}_i\}_{i\in\mathcal{N}}, P, \{R_i\}_{i\in\mathcal{N}}, \{C_{i,j}\}_{(i,j)\in\mathcal{N}\times\mathcal{J}}, \{c_{i,j}\}_{(i,j)\in\mathcal{N}\times\mathcal{J}}, \gamma, \rho)$, where $\mathcal{N} = \{i\}_{i=1}^{N}$ is the set of agents and $\mathcal{J} = \{j\}_{j=1}^{J}$ indexes cost constraints. Each agent i has an action space $\mathcal{A}_i$ and all agents share a state space $\mathcal{S}$. $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition kernel, where $\mathcal{A} = \prod_{i\in\mathcal{N}} \mathcal{A}_i$ is the joint action space of all agents, $\Delta(\mathcal{S})$ denotes the probability simplex supported on $\mathcal{S}$. The reward function for agent i is $R_i : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, and its j-th cost function is $C_{i,j} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$. The game is discounted by $\gamma \in (0, 1)$ with initial state distribution $\rho \in \Delta(\mathcal{S})$. A stationary agent policy is a map $\pi_i : \mathcal{S} \to \Delta(\mathcal{A}_i)$, where $\pi_i(a_i|s)$ denotes the probability of taking action a_i when agent i is at state s. Then a stationary joint policy of all agents is given by $\pi = \prod_{i\in\mathcal{N}} \pi_i$. We denote the set of all the stationary policies of agent i by Π_i and the set of stationary joint pol-

icy by $\Pi = \prod_{i\in\mathcal{N}} \Pi_i$. The non-robust value function of state s and joint policy π is defined as the expected accumulative discounted reward if the agents follow a joint policy π : $V_i^\pi(s) = \mathbf{E}_{\pi,P}\left[\sum_{t=0}^{\infty}\gamma^t R_i(s_t,\mathbf{a}_t,s_{t+1})\middle|s_0=s\right]$, where $\mathbf{E}_{\pi,P}$ denotes the expectation when the joint policy is π and the transition kernel is P. Similarly, the non-robust *cost* value function of state s and joint policy π is defined as $\Lambda_{i,j}^\pi(s) = \mathbf{E}_{\pi,P}\left[\sum_{t=0}^{\infty}\gamma^t C_{i,j}(s_t,\mathbf{a}_t,s_{t+1})\middle|s_0=s\right]$. We define the expected values under initial distribution ρ as $V_i^{\pi_i,\pi_{-i}}(\rho) = \mathbf{E}_{s\sim\rho}[V_i^{\pi_i,\pi_{-i}}(s)]$ and $\Lambda_{i,j}^{\pi_i,\pi_{-i}}(\rho) = \mathbf{E}_{s\sim\rho}[\Lambda_{i,j}^{\pi_i,\pi_{-i}}(s)]$, where $-i$ represents the indices of all agents except agent i. A feasible Nash equilibrium (FNE) policy $\pi^* = \prod_{i\in\mathcal{N}}\pi_i^*$ satisfies:

$$\begin{aligned} & V_i^{\pi_i^*,\pi_{-i}^*}(\rho) \geq V_i^{\pi_i,\pi_{-i}^*}(\rho), \quad \forall \pi_i \in \Pi_i \\ \text{subject to} \quad & \Lambda_{i,j}^{\pi^*}(\rho) \leq c_{i,j}, \quad \forall j \in \mathcal{J}, \end{aligned} \tag{1}$$

where $c_{i,j}$ are some thresholds. To address this optimization problem efficiently, a common assumption employed in the literature is Slater's condition [8], which states that for each agent i, there exists at least one policy π_i such that, for any fixed stationary product policy of other agents π_{-i}, it holds that $\Lambda_{i,j}^{\pi_i,\pi_{-i}}(\rho) < c_{i,j}$ for all $j \in \mathcal{J}$. This condition guarantees strong duality and enables solving the constrained optimization problem via its dual. Under this condition, the existence of an FNE is established in Theorem 2.1 of Altman and Shwartz [1].

Robust Markov Games An RMG [23] is defined by a tuple $(\mathcal{N},\mathcal{S},\{\mathcal{A}_i\}_{i\in\mathcal{N}},P,\{R_i\}_{i\in\mathcal{N}},\gamma,\rho,\mathcal{P})$, where $\mathcal{P} := \{P : \mathcal{S}\times\mathcal{A} \to \Delta(\mathcal{S})\}$ is an uncertainty set over transition kernels. Then, the robust value function of a joint policy π is defined as the worst-case expected discounted return over all transitions in the uncertainty set: $\bar{V}_i^\pi(s) = \min_{P\in\mathcal{P}} V_i^\pi(s) = \min_{P\in\mathcal{P}} \mathbf{E}_{\pi,P}[\sum_{t=0}^{\infty}\gamma^t R_i(s_t,\mathbf{a}_t,s_{t+1})|s_0=s]$. We focus on the (s,a)-rectangular uncertainty set [14], where the uncertainty decomposes as $\mathcal{P} := \bigotimes_{(s,\mathbf{a})\in\mathcal{S}\times\mathcal{A}}\mathcal{P}(s,\mathbf{a})$, allowing a robust Bellman equation [6]:

$$\bar{V}_i^\pi(s) = \min_{p\in\mathcal{P}(s,\mathbf{a})} \mathbf{E}_{\pi,p}\left[R_i(s,\mathbf{a},s') + \gamma\bar{V}_i^\pi(s')\right] \tag{2}$$

The goal of an RMG is to find a robust perfect Nash equilibrium (RPNE) policy $\pi^* = \prod_{i\in\mathcal{N}}\pi_i^*$ such that for each agent i and each $s \in \mathcal{S}$, we have $\pi_i^*(\cdot \mid s) \in \arg\max_{\pi_i\in\Pi_i} \bar{V}_i^{\pi_i,\pi_{-i}^*}(s)$.

4 ROBUST CONSTRAINED MARKOV GAME

The motivation for robust constrained Markov games is two-fold: (i) to design policies that optimize expected rewards under worst-case transition dynamics, ensuring robustness, and (ii) to enforce strict adherence to constraints even in the worst-case cost realizations. Unlike rewards, where robustness minimizes returns, robust cost handling seeks to guard against the maximum possible cumulative cost. We define the robust cost-value function as $\bar{\Lambda}_{i,j}^\pi(s) = \max_{P\in\mathcal{P}}\Lambda_{i,j}^\pi(s) = \max_{P\in\mathcal{P}}\mathbf{E}_{\pi,P}\left[\sum_{t=0}^{\infty}\gamma^t C_{i,j}(s_t,\mathbf{a}_t,s_{t+1})\middle|s_0=s\right]$. An RCMG naturally unifies the settings of CMGs and RMGs and is defined by the tuple: $(\mathcal{N},\mathcal{J},\mathcal{S},\{\mathcal{A}_i\}_{i\in\mathcal{N}},P,\{R_i\}_{i\in\mathcal{N}},\{C_{i,j}\}_{(i,j)\in\mathcal{N}\times\mathcal{J}},\{c_{i,j}\}_{(i,j)\in\mathcal{N}\times\mathcal{J}},\gamma,\rho,\mathcal{P})$ inheriting components from both frameworks. Based on this, we define the following solution concept:

Definition 4.1 (Robust and Feasible Nash Equilibrium)**.** A policy profile $\pi^* = (\pi_1^*,\ldots,\pi_N^*)$ is a Robust and Feasible Nash Equilibrium (RFNE) if, for each agent $i \in \mathcal{N}$, it satisfies:

$$\begin{aligned} & \bar{V}_i^{\pi_i^*,\pi_{-i}^*}(\rho) \geq \bar{V}_i^{\pi_i,\pi_{-i}^*}(\rho), \quad \forall \pi_i \in \Pi_i \\ \text{subject to} \quad & \bar{\Lambda}_{i,j}^{\pi^*}(\rho) \leq c_{i,j}, \quad \forall j \in \mathcal{J}, \end{aligned} \tag{3}$$

An RFNE ensures that no agent can improve its worst-case expected return unilaterally without violating its constraints, offering stability in adversarial and constrained environments. This equilibrium is essential in environments where agents must navigate competitive or cooperative interactions while adhering to practical constraints. For example, in multi-robot warehouse logistics, robots must optimize their movement strategies to avoid collisions, minimize travel time, and conserve battery life. RFNE policies guarantee optimality under mutual dependence while maintaining feasibility, supporting reliable multi-agent deployment in uncertain, resource-limited systems. Similar to the non-robust case, the RFNE condition (3) can be reformulated as a constrained optimization problem for each agent $i \in \mathcal{N}$,

$$\max_{\pi_i\in\Pi_i} \bar{V}_i^{\pi_i,\pi_{-i}}(\rho), \quad \text{s.t. } \bar{\Lambda}_{i,j}^{\pi_i,\pi_{-i}}(\rho) \leq c_{i,j}, \forall j \in \mathcal{J} \tag{4}$$

By introducing Lagrange multipliers, problem (4) can be transformed into an equivalent max-min optimization form:

$$\begin{aligned} \max_{\pi_i\in\Pi_i}\min_{\lambda_i\succeq 0} \bar{V}_i^L(\pi,\lambda) = \max_{\pi_i\in\Pi_i}\min_{\lambda_i\succeq 0}\Bigg[\bar{V}_i^{\pi_i,\pi_{-i}}(\rho) \\ - \sum_{j\in\mathcal{J}}\lambda_{i,j}\left(\bar{\Lambda}_{i,j}^{\pi_i,\pi_{-i}}(\rho) - c_{i,j}\right)\Bigg] \end{aligned} \tag{5}$$

where $\lambda_i = (\lambda_{i,j})_{j\in\mathcal{J}} \succeq 0$ denotes the Lagrange multipliers associated with the constraints in (4). Reversing the order of optimization yields the dual min-max problem:

$$\min_{\lambda_i\succeq 0}\max_{\pi_i\in\Pi_i} \bar{V}_i^L(\pi,\lambda) \tag{6}$$

The solution to problem (6) provides an upper bound to the original constrained optimization problem (4). This bound tightens under conditions such as Slater's condition, leading to strong duality. However, as demonstrated by Wang et al. [36], strong duality may fail to hold when

robustness considerations are introduced into constrained Markov decision processes (CMDPs). To analyze $\bar{V}_i^L(\pi, \lambda)$, let $R_i^t := R_i(s_t, a_t, s_{t+1})$ and $C_{i,j}^t := C_{i,j}(s_t, a_t, s_{t+1})$, yielding:

$$\begin{aligned}\bar{V}_i^L(\pi,\lambda) &= \bar{V}_i^{\pi_i,\pi_{-i}}(\rho) - \sum_{j\in\mathcal{J}} \lambda_{i,j}\left(\bar{\Lambda}_{i,j}^{\pi_i,\pi_{-i}}(\rho) - c_{i,j}\right) \\ &= \min_{P\in\mathcal{P}} \mathbf{E}_{\pi,P,\rho}\left[\sum_{t=0}^{\infty}\gamma^t R_i^t\right] \qquad (7) \\ &\quad - \sum_{j\in\mathcal{J}} \lambda_{i,j} \max_{P\in\mathcal{P}} \mathbf{E}_{\pi,P,\rho}\left[\sum_{t=0}^{\infty}\gamma^t C_{i,j}^t\right] + \lambda_i^\top c_i \\ &\le \min_{P\in\mathcal{P}} \mathbf{E}_{\pi,P,\rho}\left[\sum_{t=0}^{\infty}\gamma^t R_i^t - \sum_{j\in\mathcal{J}}\lambda_{i,j}\sum_{t=0}^{\infty}\gamma^t C_{i,j}^t\right] + \lambda_i^\top c_i\end{aligned}$$

where $\lambda_i^\top c_i = \sum_j \lambda_{i,j} c_{i,j}$. Inequality (7) results from consolidating a previously intractable nested minimization involving $J+1$ separate worst-case kernels (one per reward and each cost component) into a single unified kernel. Practically, rewards and costs evolve jointly under one transition kernel; hence, this unified kernel relaxation is realistic and analytically manageable. By adjusting the summation order, we define the surrogate Lagrangian:

$$\bar{V}_i^{SL}(\pi,\lambda) = \min_{P\in\mathcal{P}} \mathbf{E}_{\pi,P,\rho}\left[\sum_{t=0}^{\infty}\gamma^t(R_i^t - \lambda_i^\top C_i^t)\right] + \lambda_i^\top c_i \qquad (8)$$

where $\lambda_i^\top C_i^t = \sum_j \lambda_{i,j} C_{i,j}^t$. Since $\bar{V}_i^{SL}$ serves as an upper bound to $\bar{V}_i^L$ for any (π, λ), the surrogate dual problem provides a looser yet analytically tractable upper bound relative to problem (6). Summarizing the relationship, where every optimization is taken over $\pi_i \in \Pi_i$ and $\lambda_i \succeq 0$ and the arguments (π, λ) of $\bar{V}_i^L$ and $\bar{V}_i^{SL}$ are suppressed:

$$\max_{\pi_i}\min_{\lambda_i} \bar{V}_i^L \;\le\; \min_{\lambda_i}\max_{\pi_i} \bar{V}_i^L \;\le\; \min_{\lambda_i}\max_{\pi_i} \bar{V}_i^{SL} \qquad (9)$$

The first inequality tightens under strong duality conditions, while the second inequality becomes exact if there exists a single transition kernel that simultaneously realizes the worst case for both the reward and all cost components.

Before the formal definition, we highlight the intuition behind this relaxation. Robustness here acts *asymmetrically*: each agent maximizes its worst-case reward (minimized by nature) while keeping its worst-case cost (maximized by nature) below threshold, so the exact Lagrangian is governed by two distinct adversarial kernels. The surrogate (8) instead commits to the single kernel acting on the combined reward $\tilde{R}_i = R_i - \lambda_i^\top C_i$, which is what yields the upper bound that names the Upper-RFNE. Intuitively, it is the most conservative feasible equilibrium the agents can robustly guarantee when they cannot separately optimize against different adversarial kernels. Leveraging this upper bound formulation, we introduce a new equilibrium concept:

Definition 4.2 (Upper-RFNE)**.** A policy profile $\pi^* = (\pi_1^*, \ldots, \pi_N^*) \in \Pi$ is an Upper-RFNE if there exists a vector of dual variables $\lambda^* = (\lambda_1^*, \ldots, \lambda_N^*) \succeq 0$ such that, for each agent $i \in \mathcal{N}$ and all $\pi_i \in \Pi_i$, $\lambda_i \succeq 0$, it satisfies:

$$\bar{V}_i^{SL}(\pi_i, \pi_{-i}^*, \lambda^*) \le \bar{V}_i^{SL}(\pi^*, \lambda^*) \le \bar{V}_i^{SL}(\pi^*, \lambda_i, \lambda_{-i}^*) \qquad (10)$$

Remark 4.3. *An exact RFNE may not exist, due either to the failure of strong duality or to infeasibility under worst-case dynamics. The Upper-RFNE instead always exists under mild conditions and is best read as a solution concept in its own right, a principled relaxation permitting controlled constraint violations, rather than an approximation to a possibly non-existent RFNE. It coincides with the true RFNE whenever the surrogate is tight, i.e. strong duality holds and a single transition kernel simultaneously realizes the worst-case reward and cost. This tightness is automatic in the vanishing-uncertainty limit: as the uncertainty set shrinks to the nominal singleton $\{P_0\}$, the RCMG reduces to a CMG, so the same kernel attains both worst cases and the hierarchy degenerates, Upper-RFNE = RFNE = FNE* (1)*, the classical feasible Nash equilibrium whose existence holds under standard Slater conditions. The relaxation thus introduces no gap once uncertainty vanishes.*

Example 4.4 (Tight vs. non-tight regimes)**.** Consider a minimal two-agent RCMG on $\mathcal{S} = \{s_1, s_2\}$ (s_2 absorbing) in which the state moves toward absorption only when both agents advance, under an uncertain probability $p \in [0.6, 0.95]$; advancing earns more reward, and only Agent 2 pays a cost for it. The reward is worst when absorption is fast and the cost is worst when it is slow, so the two worst-case kernels sit at opposite ends of $[0.6, 0.95]$ unless the cost is tied to the same outcome as the reward. When the cost is charged on the absorbing transition the two coincide, the tightness condition of Remark 4.3 holds, and the Upper-RFNE equals the RFNE; when it is instead charged on Agent 2's advance within the non-absorbing state s_1, they separate and the two concepts become *distinct equilibria*, with the Upper-RFNE adopting a more permissive policy that violates Agent 2's constraint under the true worst case. Appendix G specifies the instance and derives both equilibria in closed form.

At this stage, the problem for each agent $i \in \mathcal{N}$ reduces to solving the following surrogate optimization:

$$\underbrace{\min_{\lambda_i\succeq 0}\left\{\overbrace{\max_{\pi_i\in\Pi_i}\min_{P\in\mathcal{P}} \mathbf{E}_{\pi_i,\pi_{-i},P,\rho}\left[\sum_{t=0}^{\infty}\gamma^t \tilde{R}_i^t(\lambda_i)\right]}^{\text{inner subproblem}}\right\} + \lambda_i^\top c_i}_{\text{outer problem}} \qquad (11)$$

where the surrogate reward is defined as $\tilde{R}_i^t(\lambda_i) = R_i^t - \lambda_i^\top C_i^t$. The inner problem computes a robust best response by maximizing the worst-case cumulative surrogate return,

while the outer problem updates the dual variables to guide constraint satisfaction. Since strict feasibility cannot be guaranteed without strong duality, the goal of the outer problem is to minimize violation and approximate the RFNE conditions as closely as possible.

5 METHODS

Despite the two-level relaxation (via the Lagrangian reformulation (6) and the surrogate objective (8)), solving (11) remains challenging. The core difficulty lies in its hierarchical structure and the coupling between policies and Lagrange multipliers: each agent's optimal policy depends on both its own multipliers and others' policies, and vice versa. This interdependence induces a nontrivial optimization landscape and complicates equilibrium computation, which requires implicit coordination for stability and fairness. As a result, even in its relaxed form, the problem is computationally intensive and analytically intractable in general.

Given these challenges, a decentralized approach is adopted as a practical means of computing an Upper-RFNE. The algorithm takes as input an initial joint policy $\pi(0)$ and Lagrange multipliers $\lambda(0)$, and proceeds through a nested iterative process. Let T_π denote the number of outer iterations and T_λ a cap on the inner dual iterations. At each outer iteration $t = 0, \dots, T_\pi - 1$, each agent i fixes the policies of the other agents, denoted $\pi_{-i}(t, 0)$, and runs an inner dual loop with these opponents held fixed throughout. In each inner iteration k, the agent solves its inner subproblem with the current dual variable $\lambda_i(t, k)$ and fixed opponent policy $\pi_{-i}(t, 0)$, yielding an updated policy $\pi_i(t, k + 1)$. Concretely, the inner subproblem is a robust MDP that we solve by *robust Q-value iteration* under the fixed opponent policy $\pi_{-i}(t, 0)$, iterating the robust Bellman backup (12) to its unique fixed point (Lemma 5.4), followed by greedy policy improvement. It then takes one subgradient step on its Lagrange multipliers based on the observed constraint violations; nonnegativity of the multipliers is maintained throughout by the lower-bound projection in Algorithm 1. This dual loop repeats until the multipliers converge, so that $\lambda_i(t, \cdot)$ approximates the dual optimum for the fixed opponents, or until the cap T_λ is reached; only then does each agent adopt the resulting policy $\pi_i(t + 1) := \pi_i(t, k)$, which is used in the next outer iteration. Each outer iteration thus realizes one application of the robust and feasible best-response map $\overline{\Psi}$ (formalized in (19) below), and the algorithm incrementally approaches an Upper-RFNE.

To solve the inner subproblem with fixed λ_i and π_{-i}, we first introduce the following assumptions:

Assumption 5.1. All reward functions $\{R_i\}_{i\in\mathcal{N}}$, cost functions $\{C_{i,j}\}_{(i,j)\in\mathcal{N}\times\mathcal{J}}$ and policies π_i are defined on compact subsets of some finite-dimensional space and take values in compact (i.e., bounded and closed) ranges.

Assumption 5.2. The state space $\mathcal{S}$ and each agent's action space $\mathcal{A}_i$ are finite.

Under these assumptions, the value space for agent i is defined as $\tilde{V}_i \in W_i = \mathbb{R}^{\mathcal{S}}$, and the joint value space is $W = \prod_{i\in\mathcal{N}} W_i$. We equip this space with the infinity norm, defined as $\|\tilde{V}\|_\infty := \max_{(i,s)\in\mathcal{N}\times\mathcal{S}} |\tilde{V}_i(s)|$, which will be used to analyze convergence and characterize fixed points. With λ_i and π_{-i} fixed, the inner subproblem reduces to a robust MDP, where λ_i and the opponents' policies effectively become part of the environment specification. To characterize worst-case dynamics in this robust setting, we build on the robust Bellman equation (2) and define the robust Bellman operator $\overline{T}_i^{\lambda_i,\pi} : W_i \times \mathcal{S} \to \mathbb{R}$ as follows:

$$\overline{T}_i^{\lambda_i,\pi}(\tilde{V}_i, s) = \min_{p\in\mathcal{P}(s,\mathbf{a})} \mathbf{E}_{\pi,p}\left[\tilde{R}_i(\lambda_i) + \gamma\tilde{V}_i(s')\right] \quad (12)$$

Using these per-agent operators for all agents, we define the joint-robust Bellman optimality operator $\mathbf{L}^{\lambda,\pi} : W \to W$ via

$$\left[\mathbf{L}^{\lambda,\pi}(\tilde{V})\right]_{i,s} = \max_{\pi_i\in\Pi_i} \overline{T}_i^{\lambda_i,\pi}(\tilde{V}_i, s) \quad (13)$$

which performs a one-step dynamic programming update: each agent i maximizes its local robust value given others' fixed policies π_{-i}, while the transition kernel adversarially minimizes the expected return. This operator satisfies the following properties (see Appendix B):

Proposition 5.3. *(1) $\mathbf{L}^{\lambda,\pi}$ is a contraction on $(W, \|\cdot\|_\infty)$; (2) $\mathbf{L}^{\lambda,\pi}$ is continuous with respect to (λ, π) on $\mathbb{R}_+^{N\times J} \times \Pi$; (3) The family $\{\mathbf{L}^{\lambda,\pi}(\tilde{V}) : \tilde{V}$ is bounded$\}$ is equicontinuous on the metric space $(\mathbb{R}_+^{N\times J} \times \Pi, \|\cdot\|_\infty)$.*

The contraction property in particular implies the existence and uniqueness of a fixed point. Specifically, since $\mathbf{L}^{\lambda,\pi}$ is a contraction on the complete metric space $(W, \|\cdot\|_\infty)$, Banach's Fixed Point Theorem (Theorem A.3) applies directly:

Lemma 5.4. *For every $(\lambda, \pi) \in \mathbb{R}_+^{N\times J} \times \Pi$, $\mathbf{L}^{\lambda,\pi}$ has exactly one fixed point.*

By Lemma 5.4, each (λ, π) determines a unique solution $\tilde{V} = \mathbf{L}^{\lambda,\pi}(\tilde{V})$ in W. We interpret this unique fixed point as the robust best response value. Formally, define $\varphi : \mathbb{R}_+^{N\times J} \times \Pi \to W$ by

$$\varphi(\lambda, \pi) = \left\{\tilde{V} \ : \ \tilde{V} = \mathbf{L}^{\lambda,\pi}(\tilde{V})\right\} \quad (14)$$

Repeatedly applying $\mathbf{L}^{\lambda,\pi}$ will converge to this fixed point, from which no agent i can further improve its robust value given π_{-i}. Specifically, for each (i, s), $[\varphi(\lambda, \pi)]_{i,s} = \tilde{V}_i^{\lambda_i,\pi_i^*,\pi_{-i}}(s) = \max_{\pi_i\in\Pi_i} \tilde{V}_i^{\lambda_i,\pi_i,\pi_{-i}}(s)$, indicating that π_i^* is a robust best response to π_{-i}. To complete the characterization, we turn to the set of policies that achieve these

Algorithm 1 Iterative Decentralized Algorithm for Computing an Upper-RFNE

Input: State space $\mathcal{S}$, action spaces $\{\mathcal{A}_i\}_{i\in\mathcal{N}}$, rewards $\{R_i\}_{i\in\mathcal{N}}$, costs $\{C_{i,j}\}_{(i,j)\in\mathcal{N}\times\mathcal{J}}$ and thresholds $\{c_{i,j}\}_{(i,j)\in\mathcal{N}\times\mathcal{J}}$, uncertainty set $\mathcal{P}$, initial policies $\{\pi_i(0)\}_{i\in\mathcal{N}}$ and multipliers $\{\lambda_{i,j}(0)\}_{(i,j)\in\mathcal{N}\times\mathcal{J}}$, discount γ, diminishing stepsizes $\{\alpha_k\}_{k\geq 0}$ satisfying (17), maximum iteration counts T_π, T_λ, multiplier floor $\underline{\lambda}\geq 0$.

Output: Joint policy $\pi(T_\pi)=\big(\pi_1(T_\pi),\dots,\pi_N(T_\pi)\big)$, an Upper-RFNE upon stabilization.

for $t=0$ **to** $T_\pi-1$ **do** ▷ outer best-response iteration ($\overline{\Psi}$)
  $\pi(t,0)\leftarrow\big(\pi_1(t),\dots,\pi_N(t)\big)$; $k\leftarrow 0$
  repeat ▷ dual loop: drive λ to the dual optimum $\Phi(\pi(t,0))$
    for all agents $i\in\mathcal{N}$ **do** ▷ robust best response $\Psi(\lambda(t,k),\cdot)$
      Fix opponents $\pi_{-i}(t,0)$ and multiplier $\lambda_i(t,k)$; set surrogate reward $\tilde{R}_i\leftarrow R_i-\lambda_i(t,k)^\top C_i$
      repeat ▷ robust Q-value iteration
        $Q_i(s,a_i)\leftarrow\min_{P\in\mathcal{P}(s,a)}\mathbf{E}_{a_{-i}\sim\pi_{-i}(t,0),\,s'\sim P}\Big[\tilde{R}_i+\gamma\max_{a_i'}Q_i(s',a_i')\Big],\quad\forall\,(s,a_i)\in\mathcal{S}\times\mathcal{A}_i$
      until Q_i converges
      $\pi_i(t,k+1)(\cdot\mid s)\leftarrow$ greedy policy w.r.t. $Q_i(s,\cdot)$
    end for
    for all $(i,j)\in\mathcal{N}\times\mathcal{J}$ **do** ▷ dual subgradient step
      Estimate worst-case discounted cost $\bar{\Lambda}_{i,j}\leftarrow\mathbf{E}_{\pi(t,k+1),\,P'}\Big[\sum_{n\geq 0}\gamma^n C_{i,j}^n\Big]$ under the worst-case kernel P'
      Projected multiplier update: $\lambda_{i,j}(t,k+1)\leftarrow\max\big(\underline{\lambda},\ \lambda_{i,j}(t,k)-\alpha_k(c_{i,j}-\bar{\Lambda}_{i,j})\big)$
    end for
    $k\leftarrow k+1$
  until $\lambda(t,\cdot)$ has converged **or** $k=T_\lambda$ ▷ $\lambda(t,\cdot)\approx\Phi(\pi(t,0))$
  $\pi(t+1)\leftarrow\pi(t,k)$ ▷ advance the profile only after the dual loop converges
end for

robust best response values. We define the robust best response mapping $\Psi:\mathbb{R}_+^{N\times J}\times\Pi\to 2^\Pi$ as follows:

$$\Psi(\lambda,\pi)=\left\{\pi'\in\Pi\;\middle|\;\begin{array}{l}\forall(i,s)\in\mathcal{N}\times\mathcal{S},\\ \pi_i'(\cdot\mid s)\in\arg\max_{\pi_i\in\Pi_i}\tilde{V}_i^{\lambda_i,\pi_i,\pi_{-i}}(s)\end{array}\right\}\tag{15}$$

Here, each $\pi'\in\Psi(\lambda,\pi)$ yields a robust best response for every agent i and each state s. The mapping Ψ satisfies the following properties (see Appendix C for details):

Proposition 5.5. *(1) Ψ is upper semicontinuous on $\mathbb{R}_+^{N\times J}\times\Pi$; (2) For every $(\lambda,\pi)\in\mathbb{R}_+^{N\times J}\times\Pi$, the set $\Psi(\lambda,\pi)$ is convex.*

Having characterized the robust best responses within the inner subproblem, we now address the outer problem (11). As discussed in Section 5.2 of Boyd and Vandenberghe [8], a dual problem is always convex, even if the primal problem may not be. Consequently, convex optimization techniques apply directly in this dual setting. A natural approach to solving the outer problem (11) in practice is through gradient-based methods on the surrogate Lagrangian function. Given a step size $\alpha_k>0$, the Lagrange multipliers of agent i are updated iteratively according to

$$\lambda_i(k+1)=\lambda_i(k)-\alpha_k g_i(k)$$

where $g_i(t)\in\partial_{\lambda_i}\bar{V}_i^{SL}(\pi',\lambda)$ is a subgradient of the surrogate Lagrangian evaluated at a robust best response $\pi'\in\Psi(\lambda,\pi)$ obtained from the inner iteration. To derive this subgradient, we observe that for any worst-case transition kernel $P'\in\mathcal{P}$, we have $\bar{V}_i^{SL}(\pi',\lambda)=V_i^{SL}(\pi',\lambda,P')$. For such fixed P', the function becomes differentiable in λ_i, and its gradient is given by:

$$\begin{aligned}&\nabla_{\lambda_i}V_i^{SL}(\pi',\lambda,P')\\&=\nabla_{\lambda_i}\left\{\mathbf{E}_{\pi',P',\rho}\left[\sum_{t=0}^{\infty}\gamma^t(R_i^t-\lambda_i^\top C_i^t)\right]+\lambda_i^\top c_i\right\}\\&=c_i-\mathbf{E}_{\pi',P',\rho}\left[\sum_{t=0}^{\infty}\gamma^t C_i^t\right]\end{aligned}\tag{16}$$

However, due to the potential non-uniqueness of the worst-case transition kernel P', the function $\bar{V}_i^{SL}(\pi',\lambda)$ may not be differentiable. In such cases, writing $\mathcal{P}^\star=\arg\min_{P\in\mathcal{P}}V_i^{SL}(\pi',\lambda,P)$ for the set of worst-case kernels, its subdifferential $\partial_{\lambda_i}\bar{V}_i^{SL}(\pi',\lambda)$ is the convex hull of the corresponding gradients:

$$\mathbf{Conv}\left\{c_i-\mathbf{E}_{\pi',P',\rho}\left[\sum_{t=0}^{\infty}\gamma^t C_i^t\right]\;\middle|\;P'\in\mathcal{P}^\star\right\}$$

According to Bertsekas et al. [5], since $\bar{V}_i^{SL}(\pi',\lambda)$ is convex, its subdifferential $\partial_{\lambda_i}\bar{V}_i^{SL}(\pi',\lambda)$ is always nonempty, convex, and compact for all $\lambda_i\in\mathbb{R}_+^J$. Thus, the convex hull characterization above always holds. Moreover, the optimality condition states that λ_i' is optimal for the outer problem if and only if the zero vector lies in the subdifferential of the surrogate objective, i.e., $0\in\partial_{\lambda_i}\bar{V}_i^{SL}(\pi',\lambda)$. The

convergence of the subgradient method is further ensured under standard step-size rules [4]. In this work, we adopt the widely used diminishing rule,

$$\sum_{k=0}^{\infty} \alpha_k = \infty \quad \text{and} \quad \sum_{k=0}^{\infty} \alpha_k^2 < \infty, \tag{17}$$

which guarantees convergence to an optimal dual solution. The lower-bound projection $\max(\underline{\lambda}, \cdot)$ used in Algorithm 1 keeps each multiplier nonnegative without forcing it to exactly zero, and induces a self-correcting feedback loop: when a constraint is slack, $\lambda_{i,j}$ decays toward $\underline{\lambda}$ and the cost penalty becomes negligible, so the agent pursues reward and cost rises; the rising cost then drives $\lambda_{i,j}$ back up to re-penalize the violation, maintaining approximate feasibility at equilibrium. To formally capture the set of such optimal dual variables for each agent, we define the dual problem optimal set mapping $\Phi : \Pi \to 2^{\mathbb{R}_+^{N \times J}}$:

$$\Phi(\pi) = \left\{ \lambda' \in \mathbb{R}_+^{N \times J} \;\middle|\; \begin{array}{l} \forall i \in \mathcal{N}, \\ \lambda_i' \in \arg \min\limits_{\lambda_i \in \mathbb{R}_+^J} \max\limits_{\pi_i \in \Pi_i} \bar{V}_i^{SL}(\pi, \lambda) \end{array} \right\} \tag{18}$$

Intuitively, $\Phi(\pi)$ collects all dual variables λ_i' such that, when agent i optimizes its policy against π_{-i}, the corresponding multiplier λ_i' minimizes the worst-case surrogate objective and thus ensures approximate constraint satisfaction. With the dual optimal set $\Phi(\pi)$ and the robust best response mapping $\Psi(\lambda, \pi)$ in hand, we can combine these to define the robust and feasible best response mapping $\overline{\Psi} : \Pi \to 2^{\Pi}$ as

$$\overline{\Psi}(\pi) = \bigcup_{\lambda \in \Phi(\pi)} \Psi(\lambda, \pi) \tag{19}$$

This composition captures how, for any given policy profile π, one may first determine a set of optimal Lagrange multipliers via $\Phi(\pi)$ and then compute the robust best response policies via $\Psi(\lambda, \pi)$. The mapping $\overline{\Psi}(\pi)$ satisfies the following regularity properties (a detailed proof is provided in Appendix D):

Lemma 5.6. *Under Assumptions 5.1 and 5.2, the set-valued map $\overline{\Psi}$ is upper semicontinuous on Π, and its image $\overline{\Psi}(\pi)$ is convex for every $\pi \in \Pi$.*

In each iteration, the mapping $\overline{\Psi}$ is used to update the joint policy. Given the current joint policy $\pi \in \Pi$, the map yields a set of next-step candidate policies $\pi' \in \overline{\Psi}(\pi)$. By iteratively applying $\pi' \in \overline{\Psi}$, i.e., solving for each agent's optimal Lagrange multiplier and corresponding robust best response, we refine the joint policy over time. If this iterative procedure converges to a fixed point $\pi^* = \prod_{i \in \mathcal{N}} \pi_i^*$, then each π_i^* is simultaneously a robust and feasible best response to π_{-i}^*. As a result, the joint policy π^* satisfies the conditions of an Upper-RFNE. The following theorem guarantees that such a fixed point always exists:

Theorem 5.7 (Existence of Upper-RFNE in RCMGs)**.** *For any RCMG satisfying Assumptions 5.1 and 5.2, there exists a stationary product policy π that constitutes an Upper-RFNE.*

Proof. Lemma 5.6 establishes that $\overline{\Psi}$ is upper semicontinuous on Π and that, for each $\pi \in \Pi$, the set $\overline{\Psi}(\pi)$ is convex. Thus, $\overline{\Psi}$ meets all of Kakutani's conditions, and by Kakutani's Fixed Point Theorem (Theorem A.4), $\overline{\Psi}$ admits at least one fixed point in Π, which corresponds to an Upper-RFNE. □

Building on the above components, we now characterize the behavior of Algorithm 1. For fixed (λ, π), the inner dynamic programming step converges to the unique robust best response value, which underlies the map $\Psi(\lambda, \pi)$ (Lemma 5.4). The updates of the Lagrange multipliers converge under the diminishing stepsize rule (17), characterizing the dual optimal set mapping $\Phi(\pi)$. Composing the two, the robust and feasible best response map $\overline{\Psi}(\pi)$ is guaranteed to possess a fixed point by Kakutani's theorem (Theorem A.4). Each subroutine of the procedure is therefore convergent, and any joint policy at which the outer iteration stabilizes is a fixed point of $\overline{\Psi}$ and hence an Upper-RFNE. We do not claim global convergence of the outer fixed-point iteration; establishing such a guarantee for the full decentralized process is left to future work.

Corollary 5.8 (Well-posedness and fixed-point characterization)**.** *The inner robust dynamic-programming step and the dual updates in Algorithm 1 each converge, and any joint policy at which its outer iteration stabilizes constitutes an Upper-RFNE.*

6 EXPERIMENTS

We empirically evaluate whether Algorithm 1 can accurately compute an equilibrium in RCMGs. Our experimental setup is constructed to satisfy strong duality and ensure that a single transition kernel realizes the worst case for both reward and all cost components. Under these conditions, both inequalities in (9) hold with equality, so the solution computed corresponds to a true RFNE, not merely an Upper-RFNE. Existing MARL benchmarks lack native support for both robustness and constraints, the defining features of RCMGs. We therefore design a minimal, analytically tractable environment for two purposes: (1) to clearly illustrate the game-theoretic structure of RCMGs, and (2) to verify the correctness of our dynamic programming-based algorithm in a setting with an analytically computable ground truth. As our goal is to validate the equilibrium concept rather than evaluate learning generalization, we focus on a simplified environment.

Robust Constrained Grid Chase We introduce a two-agent grid game played on a one-dimensional, three-cell

grid world. Each agent $i \in \mathcal{N} = \{1, 2\}$ selects an action from the set $\mathcal{A}_i = \{L, S, R\}$, representing a move left, staying in place, or moving right, respectively. Figure 1 shows the state $(1, 1)$ in which both agents occupy Cell 1.

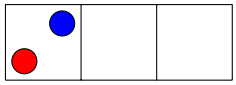

Figure 1: Example state $(1, 1)$ in the three-cell grid world. Cell indices run from left to right, and positions are labeled as $(\cdot, \cdot)$.

In this environment, Agent 1 aims to "catch" Agent 2, i.e., occupy the same cell, while Agent 2 seeks to "evade" Agent 1 by occupying a different cell. Formally, Agent 1 receives a reward of 1 when the resulting state is one of $(1, 1)$, $(2, 2)$, or $(3, 3)$, and 0 otherwise. Conversely, Agent 2 receives a reward of 1 if the resulting state is any other configuration, and 0 otherwise. Additionally, Agent 2 gets a penalty of 1 when it enters Cell 1. We set the penalty threshold as 0.5. The nominal transition kernel is deterministic, meaning agent actions translate directly to movement unless blocked. We introduce uncertainty via a kernel set $\mathcal{P} = \{P, P_1, P_2, P_3\}$ where P is the nominal kernel, and each P_k $(k = 1, 2, 3)$ overrides both agents' actions in Cell k with a uniform random assignment. If movement is blocked, the agent remains in place. A full description of the environment is provided in Appendix E.1.

RFNE Policy of the Game The RFNE in this game is for both agents to move to Cell 3 and remain there. A full justification is provided in Appendix E.1, and we briefly sketch the reasoning here. Under transition kernel P_2, Agent 2 incurs cost unless it stays in Cell 3, but doing so sacrifices potential reward by making it easier for Agent 1 to catch it. Hence, P_2 simultaneously maximizes Agent 2's cost and minimizes its reward, qualifying as its worst-case kernel. On the other hand, Agent 1 faces no cost and merely aims to match positions with Agent 2. Knowing that Agent 2 tries to stay in the Cell 3, Agent 1's worst-case kernel is P_3, which introduces randomness in Cell 3 and makes it harder to catch Agent 2. Thus, both agents act against their respective worst-case kernels, and this mutual anticipation leads to the equilibrium strategy. Under a discount factor of $\gamma = 0.9$, this RFNE yields a robust value of $\bar{V}_2^{\pi^*}(\rho) = \frac{85730}{28509} \approx 3.007$ for Agent 2 (and, by complementarity, $\bar{V}_1^{\pi^*}(\rho) \approx 6.993$), with a robust cost of $\bar{\Lambda}_2^{\pi^*}(\rho) = \frac{190}{387} \approx 0.491$ that respects the threshold 0.5. The full closed-form derivation is given in Appendix E.1.

Experimental Setup and Results Each agent's initial policy $\pi_i(0)$ samples an action uniformly from $\{L, S, R\}$ at every state, and the single multiplier is initialized from $\lambda(0) \sim \mathrm{Uniform}(0.01, 10)$. We use a discount factor $\gamma = 0.9$, allow up to $T_\lambda = 10000$ dual updates and $T_\pi = 1000$ outer policy-improvement steps, and solve each inner robust MDP by robust Q-value iteration with up to 10000 steps. All experiments are repeated over five fixed random seeds. A full list of hyperparameters is given in Appendix E.2.

All experiments were conducted on a MacBook Pro with an Apple M2 chip (10-core CPU, 16GB unified memory), using Python 3.11 with NumPy 1.26.4 and Gymnasium 0.29.1. In Figure 2, we report Agent 2's robust value and

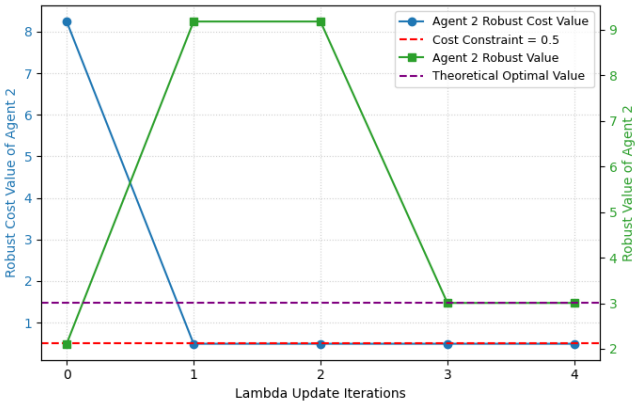

Figure 2: Convergence of Agent 2's robust cost value and robust value under the RFNE. The cost (blue) remains below the constraint, while the robust value (green) converges near the theoretical RFNE value of ≈ 3.007.

robust cost value over update iterations, demonstrating that our Algorithm 1 is consistent with the theoretical analysis. As the iterations progress, Agent 2's robust cost value (blue) settles below the specified constraint (0.5), while its robust value (green) converges toward the theoretically derived benchmark of $\frac{85730}{28509} \approx 3.007$. The final policy also closely aligns with the predicted equilibrium actions, confirming the soundness of our method and its ability to achieve the RFNE in practice. A constraint-threshold sweep and a stabilization study under loose constraints are reported in Appendices E.4 and E.5.

Baselines To isolate the role of each ingredient, we compare against a constrained-but-non-robust CMG variant and a robust-but-unconstrained RMG variant within the same environment. Under the worst-case kernel, the RMG policy leaves Agent 2's cost entirely unprotected ($\bar{\Lambda}_2^{\pi^*}(\rho) \approx 3.46 \gg 0.5$), while a CMG policy that is feasible under nominal dynamics becomes brittle once transitions are adversarial, its cost rising to 1.92 (violating the threshold) and Agent 1's value collapsing from 9.47 to 3.51. Only the full RCMG simultaneously stays feasible ($0.491 \leq 0.5$) and robust. We further verify on the minimal running example (Example 4.4) that the surrogate is exact when the reward and cost worst cases coincide and strictly overestimates the RFNE otherwise. Full tables and figures are in Appendices F and G.

7 DISCUSSIONS

This work addresses the challenge of multi-agent decision-making under adversarial uncertainty and cost constraints, an important yet underexplored problem. We introduced the Robust Constrained Markov Game (RCMG) as the first formal framework that incorporates robustness in both the objective and the constraint structure. To characterize equilibrium behavior, we proposed the Robust and Feasible Nash Equilibrium (RFNE) and established foundational theoretical results, including existence. To approximate RFNEs in general settings, we developed a tractable surrogate relaxation and a decentralized algorithm grounded in Lagrangian duality and robust dynamic programming. Empirically, when the relaxation is tight, our method recovers the RFNE and enforces feasibility, laying groundwork for robust and constrained multi-agent planning in domains such as autonomous driving and robotics.

Uncertainty Set Modeling Building on this foundation, it is important to reflect on the assumptions underlying our prototype algorithm and how they can be relaxed in more practical environments. In our experimental validation, we assumed a discrete uncertainty set consisting of a finite collection of candidate kernels, so that the inner minimization could be solved exactly by enumeration at each Bellman backup. While this assumption allows us to highlight the theoretical behavior of the algorithm, deploying robust solutions in large-scale domains typically requires additional structural assumptions on $\mathcal{P}$.

A slightly more general yet still tractable class of sets includes δ-contamination models and total variation (TV) balls [35, 36], where each $P \in \mathcal{P}$ is a convex combination of a nominal model $P^\star$ and an arbitrary distribution. These models admit closed-form expressions for the worst-case expectation, making robust planning efficient.

More expressive uncertainty sets such as KL-divergence or Wasserstein balls are widely used in practice. While they do not yield closed-form solutions, they are still tractable through convex optimization, dual reformulations, or robust dynamic programming. These methods incur higher computational costs but remain scalable in many scenarios.

For more general settings where no tractable uncertainty structure is known, a common approach is to model nature as an adversarial player [38] which selects transitions to minimize the agent's value. This leads to a two-player zero-sum game formulation, where the "nature" also has a policy that can be learned via standard algorithms such as robust policy iteration or adversarial reinforcement learning. While more computationally demanding, this framework provides a principled way to approximate robust solutions in otherwise intractable cases.

Scalability Algorithm 1 is a planning-oriented prototype whose primary purpose is to validate the theoretical framework rather than to scale to large environments. Its cost grows from three compounding factors: the joint state–action space grows exponentially with the number of agents N; each agent maintains J dual variables updated per iteration; and the worst-case expectation over $\mathcal{P}$ must be recomputed at every Bellman backup. In the grid world these operations are cheap because $|\mathcal{S}| = 9$ and $\mathcal{P}$ is finite, but in general the per-iteration cost scales as $O(|\mathcal{S}|^N |\mathcal{A}|^N |\mathcal{P}| J)$, which quickly becomes intractable and motivates the learning-based route discussed next.

From Planning to Learning Our analysis so far targets the planning setting. To move beyond tabular dynamic programming, we are extending the framework to a learning-based instantiation built on an MADDPG-style actor-critic [26] with twin delayed critics. Each agent keeps its own critic and policy network, and robustness is realized by a per-agent *nature actor*, a learned adversarial network that outputs a bounded perturbation of the agent's action within an ℓ_∞ ball around the nominal dynamics, so the uncertainty set $\mathcal{P}$ is represented implicitly rather than enumerated. The surrogate reward $\tilde{R}_i = R_i - \lambda_i^\top C_i$ and the projected dual update of Algorithm 1 carry over unchanged, with each nature actor trained to degrade its own agent's surrogate return. This setting is the natural continuous lift of the grid pursuit-evasion game of Section 6: the discrete cells become continuous positions in a square arena, the moves become bounded velocity actions, and the penalized cell becomes a continuous cost zone. On this task the learned variant qualitatively reproduces the equilibrium behavior predicted by our tabular analysis, with the learned policies adapting interpretably as the constraint levels vary. A full development and empirical study of this constrained robust MARL algorithm is left to future work; related adversarial-player constructions without feasibility constraints appear in Zhang et al. [38].

Acknowledgements

YW's work is supported by an Amazon Research Award, Fall 2025. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of Amazon.

References

[1] Eitan Altman and Adam Shwartz. Constrained Markov games: Nash equilibria. In Jerzy A. Filar, Vladimir Gaitsgory, and Koichi Mizukami, editors, *Advances in Dynamic Games and Applications*, pages 213–221, Boston, MA, 2000. Birkhäuser Boston. ISBN 978-1-4612-1336-9.

[2] Jean-Pierre Aubin and Arrigo Cellina. *Set-Valued Maps*, pages 37–92. Springer Berlin Heidelberg,

Berlin, Heidelberg, 1984. ISBN 978-3-642-69512-4. doi: 10.1007/978-3-642-69512-4_3.

[3] J.P. Aubin and H. Frankowska. *Set-Valued Analysis*. Modern Birkhäuser Classics. Birkhäuser Boston, 2009. ISBN 9780817648480.

[4] Dimitri Bertsekas. *Convex optimization algorithms*. Athena Scientific, 2015.

[5] Dimitri Bertsekas, Angelia Nedic, and Asuman Ozdaglar. *Convex analysis and optimization*, volume 1. Athena Scientific, 2003.

[6] Jose Blanchet, Miao Lu, Tong Zhang, and Han Zhong. Double pessimism is provably efficient for distributionally robust offline reinforcement learning: Generic algorithm and robust partial coverage. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 66845–66859. Curran Associates, Inc., 2023.

[7] David M. Bossens. Robust Lagrangian and adversarial policy gradient for robust constrained Markov decision processes. In *2024 IEEE Conference on Artificial Intelligence (CAI)*, pages 1227–1239, 2024. doi: 10.1109/CAI59869.2024.00219.

[8] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[9] Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008.

[10] Ziyi Chen, Shaocong Ma, and Yi Zhou. Finding correlated equilibrium of constrained Markov game: A primal-dual approach. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 25560–25572. Curran Associates, Inc., 2022.

[11] Jingjing Cui, Yuanwei Liu, and Arumugam Nallanathan. Multi-agent reinforcement learning-based resource allocation for UAV networks. *IEEE Transactions on Wireless Communications*, 19(2):729–743, 2019.

[12] Dongsheng Ding, Xiaohan Wei, Zhuoran Yang, Zhaoran Wang, and Mihailo R. Jovanović. Provably efficient generalized Lagrangian policy optimization for safe multi-agent reinforcement learning. In *Proceedings of the 5th Annual Learning for Dynamics and Control Conference (L4DC)*, volume 211 of *Proceedings of Machine Learning Research*, pages 315–332. PMLR, 2023.

[13] Asen L. Dontchev and R. Tyrrell Rockafellar. *Regularity properties of set-valued solution mappings*, pages 131–195. Springer New York, New York, NY, 2009. ISBN 978-0-387-87821-8. doi: 10.1007/978-0-387-87821-8_3.

[14] Larry G. Epstein and Martin Schneider. Recursive multiple-priors. *Journal of Economic Theory*, 113(1): 1–31, 2003. ISSN 0022-0531. doi: https://doi.org/10.1016/S0022-0531(03)00097-8.

[15] Zain Ulabedeen Farhat, Debamita Ghosh, George K Atia, and Yue Wang. Sample-efficient distributionally robust multi-agent reinforcement learning via online interaction. In *The Fourteenth International Conference on Learning Representations*, 2026.

[16] Arnob Ghosh. Sample complexity for obtaining suboptimality and violation bound for distributionally robust constrained MDP. In *First Reinforcement Learning Safety Workshop*, 2024. URL `https://openreview.net/forum?id=T2XyWqN2dw`.

[17] Vesal Hakami and Mehdi Dehghan. Learning stationary correlated equilibria in constrained general-sum stochastic games. *IEEE Transactions on Cybernetics*, 46(7):1640–1654, July 2016. ISSN 2168-2275. doi: 10.1109/tcyb.2015.2453165. URL `http://dx.doi.org/10.1109/TCYB.2015.2453165`.

[18] Songyang Han, Sanbao Su, Sihong He, Shuo Han, Haizhao Yang, Shaofeng Zou, and Fei Miao. What is the solution for state-adversarial multi-agent reinforcement learning? *Transactions on Machine Learning Research*, 2024.

[19] Sihong He, Songyang Han, Sanbao Su, Shuo Han, Shaofeng Zou, and Fei Miao. Robust multi-agent reinforcement learning with state uncertainty. *Transactions on Machine Learning Research*, 2023.

[20] Sihong He, Yue Wang, Shuo Han, Shaofeng Zou, and Fei Miao. A robust and constrained multi-agent reinforcement learning electric vehicle rebalancing method in AMoD systems. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5637–5644. IEEE, 2023.

[21] Xiaofeng Jiang, Shuangwu Chen, Jian Yang, Han Hu, and Zhenliang Zhang. Finding the equilibrium for continuous constrained Markov games under the average criteria. *IEEE Transactions on Automatic Control*, 65(12):5399–5406, 2020. doi: 10.1109/TAC.2020.2970153.

[22] Philip Jordan, Anas Barakat, and Niao He. Independent learning in constrained Markov potential games. In Sanjoy Dasgupta, Stephan Mandt, and

Yingzhen Li, editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 4024–4032. PMLR, 02–04 May 2024. URL https://proceedings.mlr.press/v238/jordan24a.html.

[23] Erim Kardes, Fernando Ordóñez, and Randolph W. Hall. Discounted robust stochastic games and an application to queueing control. *Operations Research*, 59:365–382, 2011.

[24] Toshinori Kitamura, Tadashi Kozuno, Wataru Kumagai, Kenta Hoshino, Yohei Hosoe, Kazumi Kasaura, Masashi Hamaya, Paavo Parmas, and Yutaka Matsuo. Near-optimal policy identification in robust constrained Markov decision processes via epigraph form. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

[25] Na Li, Zewu Zheng, Wei Ni, Hangguan Shan, Wenjie Zhang, and Xinyu Li. Sample-efficient tabular self-play for offline robust reinforcement learning. *Advances in Neural Information Processing Systems*, 38: 155562–155626, 2025.

[26] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

[27] Shaocong Ma, Ziyi Chen, Shaofeng Zou, and Yi Zhou. Decentralized robust V-learning for solving Markov games with model uncertainty. *Journal of Machine Learning Research*, 24(371):1–40, 2023. URL http://jmlr.org/papers/v24/23-0310.html.

[28] Jeremy McMahan, Giovanni Artiglio, and Qiaomin Xie. Roping in uncertainty: Robustness and regularization in Markov games. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pages 35267–35295. PMLR, 2024.

[29] H.L. Royden and P. Fitzpatrick. *Real Analysis*. Prentice Hall, 2010. ISBN 9780131437470.

[30] Reazul Hasan Russel, Mouhacine Benosman, Jeroen van Baar, and Radu Corcodel. Lyapunov robust constrained-MDPs for Sim2Real transfer learning. In Roozbeh Razavi-Far, Boyu Wang, Matthew E. Taylor, and Qiang Yang, editors, *Federated and Transfer Learning*, volume 27 of *Adaptation, Learning, and Optimization*, pages 307–328. Springer, Cham, 2022. doi: 10.1007/978-3-031-11748-0_13.

[31] L. S. Shapley. Stochastic games. *Proceedings of the National Academy of Sciences*, 39(10):1095–1100, 1953. doi: 10.1073/pnas.39.10.1095.

[32] Laixi Shi, Eric Mazumdar, Yuejie Chi, and Adam Wierman. Sample-efficient robust multi-agent reinforcement learning in the face of environmental uncertainty. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, 2024.

[33] Jayakumar Subramanian, Amit Sinha, and Aditya Mahajan. Robustness and sample complexity of Model-Based MARL for General-Sum Markov games. *Dynamic Games and Applications*, 13(1):56–88, March 2023.

[34] Zhongchang Sun, Sihong He, Fei Miao, and Shaofeng Zou. Constrained reinforcement learning under model mismatch. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, 2024.

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

[36] Yue Wang, Fei Miao, and Shaofeng Zou. Robust constrained reinforcement learning, 2022. URL https://arxiv.org/abs/2209.06866.

[37] Chao Yu, Xin Wang, and Zhanbo Feng. Coordinated multiagent reinforcement learning for teams of mobile sensing robots. In *Proceedings of the 18th international conference on autonomous agents and multiagent systems*, pages 2297–2299, 2019.

[38] K. Zhang, Tao Sun, Yunzhe Tao, Sahika Genc, S. Mallya, and Tamer Başar. Robust multi-agent reinforcement learning with model uncertainty. In *Neural Information Processing Systems*, 2020.

[39] Zhengfei Zhang, Kishan Panaganti, Laixi Shi, Yanan Sui, Adam Wierman, and Yisong Yue. Distributionally robust constrained reinforcement learning under strong duality. *Reinforcement Learning Journal*, 4: 1793–1821, 2024.

[40] Zewu Zheng and Yuanyuan Lin. Distributionally robust online markov game with linear function approximation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 28892–28900, 2026.

Supplementary Materials

A PRELIMINARIES

We begin by introducing some definitions and key theoretical results that will be utilized throughout the proof.

$$\|\tilde{V}\|_\infty := \max_{(i,s)\in\mathcal{N}\times\mathcal{S}} |\tilde{V}_i(s)| \tag{20}$$

Definition A.1 (Equicontinuity, [29]). A set $\mathcal{F}$ of real-valued functions on a metric space (X, d) is said to be equicontinuous at the point $x \in X$ provided that for each $\varepsilon > 0$, there is a $\delta > 0$ such that for every $f \in \mathcal{F}$ and $x' \in X$

$$\text{if } d(x', x) < \delta, \text{ then } |f(x') - f(x)| < \varepsilon. \tag{21}$$

The set $\mathcal{F}$ is said to be equicontinuous on X provided that it is equicontinuous at every point in X.

Definition A.2 (Contraction, [29]). A mapping T from a metric space (X, d) into itself is called a contraction provided that there is a real number $0 \leq k < 1$ such that

$$d(T(x), T(y)) \leq k\, d(x, y), \quad \forall\, x, y \in X \tag{22}$$

Theorem A.3 (Banach's Theorem, [29]). *Let (X, d) be a complete metric space and let $T : X \to X$ be a contraction. Then T has exactly one fixed point.*

Theorem A.4 (Kakutani's Fixed Point Theorem, [2]). *If X is a closed, bounded, and convex set in a Euclidean space, and Ψ is an upper semicontinuous (u.s.c.) set-valued map mapping X to the family of closed, convex subsets of X, then there exists a point $x \in X$ such that $x \in \Psi(x)$.*

Definition A.5 (Upper Semicontinuity, [3]). Let $\Psi : X \to 2^Y$ be a set-valued map. Let $x^n, x \in X$ and $\lim_{n\to\infty} x^n = x$. Let $y^n \in \Psi(x^n) \subseteq Y$ and $\lim_{n\to\infty} y^n = y$. Then Ψ is said to be upper semicontinuous (u.s.c.) at x if and only if $y \in \Psi(x)$.

Theorem A.6 (Upper Semicontinuity of Composition of Set-valued Mapping, [2]). *Let $\Psi : X \to 2^Y$ and $\Phi : Y \to 2^Z$ be two set-valued maps. Define their composition $\Phi \circ \Psi$ as*

$$\Phi \circ \Psi(x) = \bigcup_{y\in\Psi(x)} \Phi(y) \tag{23}$$

If both Ψ and Φ are u.s.c., then $\Phi \circ \Psi$ is also u.s.c.

Definition A.7 (Optimal Value and Set Mapping, [13]). The optimal value mapping is defined by

$$\Phi_{\text{val}} : p \to \inf_x \{f_0(p, x) \mid x \in \mathbb{R}^n\} \tag{24}$$

when the $\inf$ is finite, and the optimal set mapping is defined by

$$\Phi_{\text{opt}} : p \to \{x \in \mathbb{R}^n \mid f_0(p, x) = \Phi_{\text{val}}(p)\} \tag{25}$$

Theorem A.8 (Basic Continuity Properties of Optimal Value Mapping, [13]). *Let $p \in P$ be fixed with the feasible set nonempty and bounded, and suppose that the function f_0 is continuous relative to $P \times \mathbb{R}^n$ at (p, x) for every $x \in \mathbb{R}^n$. Then Φ_{val} is continuous at p, whereas Φ_{opt} is u.s.c. at p.*

Lemma A.9 (Necessity and Sufficiency Condition of Convex Set-valued Mapping, [3]). *A set-valued map $\Psi : X \to 2^Y$ is convex if and only if $\forall\, x_1, x_2 \in X$, $\eta \in [0, 1]$, we have:*

$$\eta\Psi(x_1) + (1 - \eta)\Psi(x_2) \subseteq \Psi(\eta x_1 + (1 - \eta)x_2) \tag{26}$$

B PROPERTIES OF JOINT-ROBUST BELLMAN OPTIMALITY OPERATOR

Proposition B.1. *Let $\mathbf{L}^{\lambda,\pi} : W \to W$ be the joint-robust Bellman optimality operator defined in Equation* (13)*. Then, the following properties hold:*

1. ***Contraction:*** *The operator $\mathbf{L}^{\lambda,\pi}$ is a contraction on W.*
2. ***Continuity:*** *The operator $\mathbf{L}^{\lambda,\pi}$ is continuous with respect to (λ, π) on $\mathbb{R}_+^{N\times J} \times \Pi$.*
3. ***Equicontinuity:*** *The family of functions $\{\mathbf{L}^{\lambda,\pi}(\tilde{V}) : \tilde{V} \text{ is bounded}\}$ is equicontinuous on the metric space $(\mathbb{R}_+^{N\times J} \times \Pi, \|\cdot\|_\infty)$.*

Proof. We prove these three properties in two stages. In Section B.1, we show that $\mathbf{L}^{\lambda,\pi}$ is a contraction (Property 1), relying on Lemma B.2. In Section B.2, we establish continuity and equicontinuity (Properties 2 and 3). Specifically, we will use Lemma B.3 and Lemma B.4 to handle the continuity of the key sub-operator, and then deduce the continuity and equicontinuity of $\mathbf{L}^{\lambda,\pi}$ itself. □

B.1 CONTRACTION

Lemma B.2. *The function $\mathbf{L}^{\lambda,\pi}$ is a contraction.*

Proof. Let $\tilde{V}, \tilde{U} \in W$. Fix π_{-i}, and for any $i \in \mathcal{N}$ and $s \in \mathcal{S}$, we have:

$$
\begin{aligned}
&\left[\mathbf{L}^{\lambda,\pi}(\tilde{V})\right]_{i,s} - \left[\mathbf{L}^{\lambda,\pi}(\tilde{U})\right]_{i,s} \\
=&\max_{\pi_i'\in\Pi_i} \overline{T}_i^{\lambda_i,\pi_i',\pi_{-i}}(\tilde{V}_i, s) - \max_{\pi_i'\in\Pi_i} \overline{T}_i^{\lambda_i,\pi_i',\pi_{-i}}(\tilde{U}_i, s) \\
=&\overline{T}_i^{\lambda_i,\pi_i^*,\pi_{-i}}(\tilde{V}_i, s) - \overline{T}_i^{\lambda_i,\pi_i^\dagger,\pi_{-i}}(\tilde{U}_i, s) \\
\leq&\overline{T}_i^{\lambda_i,\pi_i^*,\pi_{-i}}(\tilde{V}_i, s) - \overline{T}_i^{\lambda_i,\pi_i^*,\pi_{-i}}(\tilde{U}_i, s) \\
=&\min_{p\in\mathcal{P}(s,\mathbf{a})} \mathbf{E}_{a_i\sim\pi_i^*(\cdot|s),\mathbf{a}_{-i}\sim\pi_{-i}(\cdot|s),s'\sim p(\cdot)}\left[\tilde{R}_i(\lambda_i) + \gamma\tilde{V}_i(s')\right] - \min_{p\in\mathcal{P}(s,\mathbf{a})} \mathbf{E}_{a_i\sim\pi_i^*(\cdot|s),\mathbf{a}_{-i}\sim\pi_{-i}(\cdot|s),s'\sim p(\cdot)}\left[\tilde{R}_i(\lambda_i) + \gamma\tilde{U}_i(s')\right] \\
=&\mathbf{E}_{a_i\sim\pi_i^*(\cdot|s),\mathbf{a}_{-i}\sim\pi_{-i}(\cdot|s),s'\sim p^*(\cdot)}\left[\tilde{R}_i(\lambda_i) + \gamma\tilde{V}_i(s')\right] - \mathbf{E}_{a_i\sim\pi_i^*(\cdot|s),\mathbf{a}_{-i}\sim\pi_{-i}(\cdot|s),s'\sim p^\dagger(\cdot)}\left[\tilde{R}_i(\lambda_i) + \gamma\tilde{U}_i(s')\right] \\
\leq&\mathbf{E}_{a_i\sim\pi_i^*(\cdot|s),\mathbf{a}_{-i}\sim\pi_{-i}(\cdot|s),s'\sim p^\dagger(\cdot)}\left[\tilde{R}_i(\lambda_i) + \gamma\tilde{V}_i(s')\right] - \mathbf{E}_{a_i\sim\pi_i^*(\cdot|s),\mathbf{a}_{-i}\sim\pi_{-i}(\cdot|s),s'\sim p^\dagger(\cdot)}\left[\tilde{R}_i(\lambda_i) + \gamma\tilde{U}_i(s')\right] \\
=&\mathbf{E}_{a_i\sim\pi_i^*(\cdot|s),\mathbf{a}_{-i}\sim\pi_{-i}(\cdot|s),s'\sim p^\dagger(\cdot)}\left[\tilde{R}_i(\lambda_i) + \gamma\tilde{V}_i(s') - \left(\tilde{R}_i(\lambda_i) + \gamma\tilde{U}_i(s')\right)\right] \\
=&\gamma\mathbf{E}_{a_i\sim\pi_i^*(\cdot|s),\mathbf{a}_{-i}\sim\pi_{-i}(\cdot|s),s'\sim p^\dagger(\cdot)}\left[\tilde{V}_i(s') - \tilde{U}_i(s')\right] \\
\leq&\gamma\mathbf{E}_{a_i\sim\pi_i^*(\cdot|s),\mathbf{a}_{-i}\sim\pi_{-i}(\cdot|s),s'\sim p^\dagger(\cdot)}\|\tilde{V} - \tilde{U}\|_\infty \\
=&\gamma\|\tilde{V} - \tilde{U}\|_\infty
\end{aligned} \tag{27}
$$

Similarly, we can show:

$$
\left[\mathbf{L}^{\lambda,\pi}(\tilde{U})\right]_{i,s} - \left[\mathbf{L}^{\lambda,\pi}(\tilde{V})\right]_{i,s} \leq \gamma\|\tilde{V} - \tilde{U}\|_\infty \tag{28}
$$

Combining these results, we obtain:

$$
\|\mathbf{L}^{\lambda,\pi}(\tilde{V}) - \mathbf{L}^{\lambda,\pi}(\tilde{U})\|_\infty = \max_{(i,s)\in\mathcal{N}\times\mathcal{S}}\left|\left[\mathbf{L}^{\lambda,\pi}(\tilde{V})\right]_{i,s} - \left[\mathbf{L}^{\lambda,\pi}(\tilde{U})\right]_{i,s}\right| \leq \gamma\|\tilde{V} - \tilde{U}\|_\infty \tag{29}
$$

Therefore, $\mathbf{L}^{\lambda,\pi}$ is a contraction. □

B.2 CONTINUITY

As mentioned earlier, to establish the continuity of the joint-robust Bellman optimality operator, we need to start by analyzing the continuity of the robust Bellman operator (Lemma B.3).

Lemma B.3. *The function $\overline{T}_i^{\lambda_i,\pi}$ is continuous on $\mathbb{R}_+^J \times \Pi$, $\forall (i,s) \in \mathcal{N} \times \mathcal{S}$.*

Proof. For convenience, let us first define the (non-robust) Bellman operator $T_i^{\lambda_i,\pi} : W_i \times \mathcal{S} \times \mathcal{P} \to \mathbb{R}$ by

$$T_i^{\lambda_i,\pi}(\tilde{V}_i, s, p) = \mathbf{E}_{\mathbf{a}\sim\pi(\cdot|s), s'\sim p(\cdot)}\left[\tilde{R}_i(\lambda_i) + \gamma \tilde{V}_i(s')\right] \tag{30}$$

We first show that, for any state $s \in \mathcal{S}$ and agent $i \in \mathcal{N}$, the set $\left\{T_i^{\lambda_i,\pi}(\tilde{V}_i, s, p) \;:\; \tilde{V}_i \in W_i,\; p \in \mathcal{P}\right\}$ is equicontinuous on the metric space $(\mathbb{R}_+^J \times \Pi, \|\cdot\|_\infty)$. Concretely, we want to show that for any $\varepsilon > 0$, there exists $\delta > 0$ such that if for any $(\lambda_i, \pi), (\lambda_i', \pi') \in \mathbb{R}_+^J \times \Pi$:

$$d\big((\lambda_i,\pi),(\lambda_i',\pi')\big) = \big\|\lambda_i - \lambda_i'\big\|_\infty + \big\|\pi - \pi'\big\|_\infty = \max_{j\in\mathcal{J}} \big|\lambda_{i,j} - \lambda_{i,j}'\big| + \max_{(i,\mathbf{a})\in\mathcal{N}\times\mathcal{A}} |\pi_i(\mathbf{a} \mid s) - \pi_i'(\mathbf{a} \mid s)| < \delta \tag{31}$$

then, $\forall p \in \mathcal{P}(s, \mathbf{a})$, we have

$$\left|T_i^{\lambda_i,\pi}(\tilde{V}_i, s, p) - T_i^{\lambda_i',\pi'}(\tilde{V}_i, s, p)\right| < \varepsilon \tag{32}$$

Since the robust value function and the reward/cost function are bounded, the Lagrange multipliers are also bounded, then there exist constants $H, K, L < \infty$ such that $\forall i \in \mathcal{N}, (s, \mathbf{a}, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}, \pi \in \Pi$:

$$\left|\tilde{V}_i(s)\right| \le H, \quad \left|\tilde{R}_i(\lambda_i)\right| \le K \quad \text{and} \quad |\lambda_{i,j}| \le L \tag{33}$$

Then,

$$\begin{aligned}
\left|T_i^{\lambda_i,\pi}(\tilde{V}_i, s, p) - T_i^{\lambda_i',\pi'}(\tilde{V}_i, s, p)\right| &= \left|\mathbf{E}_{\mathbf{a}\sim\pi(\cdot|s), s'\sim p(\cdot)}\left[\tilde{R}_i(\lambda_i) + \gamma \tilde{V}_i(s')\right] - \mathbf{E}_{\mathbf{a}\sim\pi'(\cdot|s), s'\sim p(\cdot)}\left[\tilde{R}_i^{\lambda_i'}(s,\mathbf{a},s') + \gamma \tilde{V}_i(s')\right]\right| \\
&= \left|\sum_{\mathbf{a}\in\mathcal{A}}\sum_{s'\in\mathcal{S}} p(s' \mid s, \mathbf{a})\Big[\pi(\mathbf{a} \mid s)\big(\tilde{R}_i(\lambda_i) + \gamma\tilde{V}_i(s')\big) - \pi'(\mathbf{a} \mid s)\big(\tilde{R}_i^{\lambda_i'}(s,\mathbf{a},s') + \gamma\tilde{V}_i(s')\big)\Big]\right| \\
&\le \left|\sum_{\mathbf{a}\in\mathcal{A}}\sum_{s'\in\mathcal{S}} p(s' \mid s, \mathbf{a})\Big[\big(\pi(\mathbf{a} \mid s) - \pi'(\mathbf{a} \mid s)\big)\big(R_i(s,\mathbf{a},s') + \gamma\tilde{V}_i(s')\big)\Big]\right| \\
&\qquad + \left|\sum_{\mathbf{a}\in\mathcal{A}}\sum_{s'\in\mathcal{S}} p(s' \mid s, \mathbf{a})\Big[\pi(\mathbf{a} \mid s)\sum_{j\in\mathcal{J}}\lambda_{i,j}C_{i,j}(s,\mathbf{a},s') - \pi'(\mathbf{a} \mid s)\sum_{j\in\mathcal{J}}\lambda_{i,j}'C_{i,j}(s,\mathbf{a},s')\Big]\right| \\
&\le \sum_{\mathbf{a}\in\mathcal{A}}\sum_{s'\in\mathcal{S}}\Big|p(s' \mid s, \mathbf{a})\Big|\Big|\pi(\mathbf{a} \mid s) - \pi'(\mathbf{a} \mid s)\Big|\Big|R_i(s,\mathbf{a},s') + \gamma\tilde{V}_i(s')\Big| \\
&\qquad + \left|\sum_{\mathbf{a}\in\mathcal{A}}\sum_{s'\in\mathcal{S}}\sum_{j\in\mathcal{J}} p(s' \mid s, \mathbf{a})C_{i,j}(s,\mathbf{a},s')\Big[\pi(\mathbf{a} \mid s)\lambda_{i,j} - \pi'(\mathbf{a} \mid s)\lambda_{i,j}'\Big]\right| \\
&\le (K + \gamma H)|\mathcal{S}|\sum_{\mathbf{a}\in\mathcal{A}}\Big|\pi(\mathbf{a} \mid s) - \pi'(\mathbf{a} \mid s)\Big| + K|\mathcal{S}|\sum_{\mathbf{a}\in\mathcal{A}}\sum_{j\in\mathcal{J}}\Big|\pi(\mathbf{a} \mid s)\lambda_{i,j} - \pi'(\mathbf{a} \mid s)\lambda_{i,j}'\Big|
\end{aligned}$$

Let us first consider the first term of the expression.

$$\begin{aligned}
\Big|\pi(\mathbf{a} \mid s) - \pi'(\mathbf{a} \mid s)\Big| &= \left|\prod_{i\in\mathcal{N}}\pi_i(a_i \mid s) - \prod_{i\in\mathcal{N}}\pi_i'(a_i \mid s)\right| \\
&= \left|\prod_{i\in\mathcal{N}}\Big[\pi_i'(a_i \mid s) + \big(\pi_i(a_i \mid s) - \pi_i'(a_i \mid s)\big)\Big] - \prod_{i\in\mathcal{N}}\pi_i'(a_i \mid s)\right| \\
&= \left|\sum_{\Omega\subseteq\mathcal{N}}\Big[\prod_{i\in\Omega}\big(\pi_i(a_i \mid s) - \pi_i'(a_i \mid s)\big)\prod_{i\in\Omega^c}\pi_i'(a_i \mid s)\Big] - \prod_{i\in\mathcal{N}}\pi_i'(a_i \mid s)\right|
\end{aligned}$$

$$
\begin{aligned}
&= \left| \sum_{\Omega \subseteq \mathcal{N}, \Omega \neq \varnothing} \left[\prod_{i \in \Omega} \left(\pi_i(a_i \mid s) - \pi_i'(a_i \mid s) \right) \prod_{i \in \Omega^c} \pi_i'(a_i \mid s) \right] \right| \\
&\leq \sum_{\Omega \subseteq \mathcal{N}, \Omega \neq \varnothing} \left| \prod_{i \in \Omega} \left(\pi_i(a_i \mid s) - \pi_i'(a_i \mid s) \right) \right| \left| \prod_{i \in \Omega^c} \pi_i'(a_i \mid s) \right| \\
&\leq \sum_{\Omega \subseteq \mathcal{N}, \Omega \neq \varnothing} \prod_{i \in \Omega} \left| \pi_i(a_i \mid s) - \pi_i'(a_i \mid s) \right| \\
&< \sum_{\Omega \subseteq \mathcal{N}, \Omega \neq \varnothing} \delta_1^{|\Omega|}
\end{aligned} \tag{34}
$$

where $\Omega^c = \mathcal{N} \setminus \Omega$, and the last inequality is because $\max_{(i,\mathbf{a}) \in \mathcal{N} \times \mathcal{A}} |\pi_i(\mathbf{a} \mid s) - \pi_i'(\mathbf{a} \mid s)| < \delta$. Next, we move on to the second term,

$$
\begin{aligned}
&\left| \pi(\mathbf{a} \mid s) \lambda_{i,j} - \pi'(\mathbf{a} \mid s) \lambda_{i,j}' \right| \\
=&\left| \lambda_{i,j} \prod_{k \in \mathcal{N}} \pi(a_k \mid s) - \lambda_{i,j}' \prod_{k \in \mathcal{N}} \pi'(a_k \mid s) \right| \\
=&\left| \lambda_{i,j} \prod_{k \in \mathcal{N}} \left[\pi'(a_k \mid s) + \left(\pi(a_k \mid s) - \pi'(a_k \mid s) \right) \right] - \lambda_{i,j}' \prod_{k \in \mathcal{N}} \pi'(a_k \mid s) \right| \\
=&\left| \lambda_{i,j} \sum_{\Omega \subseteq \mathcal{N}} \left[\prod_{k \in \Omega} \left(\pi(a_k \mid s) - \pi'(a_k \mid s) \right) \prod_{k \in \Omega^c} \pi'(a_k \mid s) \right] - \lambda_{i,j}' \prod_{k \in \mathcal{N}} \pi'(a_k \mid s) \right| \\
=&\left| \lambda_{i,j} \left\{ \prod_{k \in \mathcal{N}} \pi'(a_k \mid s) + \sum_{\Omega \subseteq \mathcal{N}, \Omega \neq \varnothing} \left[\prod_{k \in \Omega} \left(\pi(a_k \mid s) - \pi'(a_k \mid s) \right) \prod_{k \in \Omega^c} \pi'(a_k \mid s) \right] \right\} - \lambda_{i,j}' \prod_{k \in \mathcal{N}} \pi'(a_k \mid s) \right| \\
=&\left| \left(\lambda_{i,j} - \lambda_{i,j}' \right) \prod_{k \in \mathcal{N}} \pi'(a_k \mid s) + \lambda_{i,j} \sum_{\Omega \subseteq \mathcal{N}, \Omega \neq \varnothing} \left[\prod_{k \in \Omega} \left(\pi(a_k \mid s) - \pi'(a_k \mid s) \right) \prod_{k \in \Omega^c} \pi'(a_k \mid s) \right] \right| \\
\leq& \left| \lambda_{i,j} - \lambda_{i,j}' \right| \left| \prod_{k \in \mathcal{N}} \pi'(a_k \mid s) \right| + |\lambda_{i,j}| \sum_{\Omega \subseteq \mathcal{N}, \Omega \neq \varnothing} \left| \prod_{k \in \Omega} \left(\pi(a_k \mid s) - \pi'(a_k \mid s) \right) \right| \left| \prod_{k \in \Omega^c} \pi'(a_k \mid s) \right| \\
<& \delta_2 + L \sum_{\Omega \subseteq \mathcal{N}, \Omega \neq \varnothing} \delta_3^{|\Omega|}
\end{aligned}
$$

Let $\delta = \min\{\delta_1, \delta_2, \delta_3\}$, where

$$
\delta_1 \leq \frac{\min\{1, \varepsilon\}}{3(K + \gamma H)|\mathcal{A}||\mathcal{S}|\left(2^N - 1\right)}, \quad \delta_2 \leq \frac{\min\{1, \varepsilon\}}{3KJ|\mathcal{S}||\mathcal{A}|}, \quad \delta_3 \leq \frac{\min\{1, \varepsilon\}}{3KJL|\mathcal{S}||\mathcal{A}|\left(2^N - 1\right)} \tag{35}
$$

Then

$$
\begin{aligned}
\left| T_i^{\lambda_i, \pi}(\tilde{V}_i, s, p) - T_i^{\lambda_i', \pi'}(\tilde{V}_i, s, p) \right| \leq& (K + \gamma H)|\mathcal{S}| \sum_{\mathbf{a} \in \mathcal{A}} \sum_{\Omega \subseteq \mathcal{N}, \Omega \neq \varnothing} \delta_1^{|\Omega|} + K|\mathcal{S}| \sum_{\mathbf{a} \in \mathcal{A}} \sum_{j \in \mathcal{J}} \left(\delta_2 + L \sum_{\Omega \subseteq \mathcal{N}, \Omega \neq \varnothing} \delta_3^{|\Omega|} \right) \\
<& (K + \gamma H)|\mathcal{S}||\mathcal{A}| \sum_{\Omega \subseteq \mathcal{N}, \Omega \neq \varnothing} \delta_1^{|\Omega|} + KJ|\mathcal{S}||\mathcal{A}| \left(\delta_2 + L \sum_{\Omega \subseteq \mathcal{N}, \Omega \neq \varnothing} \delta_3^{|\Omega|} \right) \\
\leq& \frac{\min\{1, \varepsilon\}}{3} + \frac{\min\{1, \varepsilon\}}{3} + \frac{\min\{1, \varepsilon\}}{3} \\
=& \min\{1, \varepsilon\}
\end{aligned}
$$

Thus

$$
\left| T_i^{\lambda_i, \pi}(\tilde{V}_i, s, p) - T_i^{\lambda_i', \pi'}(\tilde{V}_i, s, p) \right| < \varepsilon \tag{36}
$$

Hence, the collection $\left\{T_i^{\lambda_i,\pi}(\tilde{V}_i,s,p)\ :\ \tilde{V}_i\in W_i,\ p\in\mathcal{P}\right\}$ is equicontinuous on the metric space $(\mathbb{R}_+^J\times\Pi,\|\cdot\|_\infty)$. Since Equation (36) is equivalent to

$$T_i^{\lambda_i,\pi}(\tilde{V}_i,s,p)-\varepsilon<T_i^{\lambda_i',\pi'}(\tilde{V}_i,s,p)<T_i^{\lambda_i,\pi}(\tilde{V}_i,s,p)+\varepsilon \tag{37}$$

By the definition of $\overline{T}_i^{\lambda_i,\pi}$ (see (12)),

$$\overline{T}_i^{\lambda_i,\pi}(\tilde{V}_i,s)=\min_{p'\in\mathcal{P}(s,\mathbf{a})}T_i^{\lambda_i,\pi}(\tilde{V}_i,s,p')\le T_i^{\lambda_i,\pi}(\tilde{V}_i,s,p) \tag{38}$$

which implies that for all $p\in\mathcal{P}(s,\mathbf{a})$,

$$\overline{T}_i^{\lambda_i,\pi}(\tilde{V}_i,s)-\varepsilon<T_i^{\lambda_i',\pi'}(\tilde{V}_i,s,p) \tag{39}$$

Combining inequalities, we obtain

$$\overline{T}_i^{\lambda_i',\pi'}(\tilde{V}_i,s)\ge\min_{p'\in\mathcal{P}(s,\mathbf{a})}T_i^{\lambda_i',\pi'}(\tilde{V}_i,s,p')>\overline{T}_i^{\lambda_i,\pi}(\tilde{V}_i,s)-\varepsilon \tag{40}$$

For the other direction, we can find a $p^\dagger\in\mathcal{P}(s,\mathbf{a})$ such that

$$T_i^{\lambda_i,\pi}(\tilde{V}_i,s,p^\dagger)=\overline{T}_i^{\lambda_i,\pi}(\tilde{V}_i,s) \tag{41}$$

Then, we have

$$\overline{T}_i^{\lambda_i',\pi'}(\tilde{V}_i,s)\le T_i^{\lambda_i',\pi'}(\tilde{V}_i,s,p^\dagger)<T_i^{\lambda_i,\pi}(\tilde{V}_i,s,p^\dagger)+\varepsilon=\overline{T}_i^{\lambda_i,\pi}(\tilde{V}_i,s)+\varepsilon \tag{42}$$

Combining the above results, we obtain

$$\overline{T}_i^{\lambda_i,\pi}(\tilde{V}_i,s)-\varepsilon<\overline{T}_i^{\lambda_i',\pi'}(\tilde{V}_i,s)<\overline{T}_i^{\lambda_i,\pi}(\tilde{V}_i,s)+\varepsilon \tag{43}$$

which implies that

$$\left|\overline{T}_i^{\lambda_i',\pi'}(\tilde{V}_i,s)-\overline{T}_i^{\lambda_i,\pi}(\tilde{V}_i,s)\right|<\varepsilon \tag{44}$$

Thus, for any $\varepsilon>0$, there exists $\delta>0$ such that if for any $(\lambda_i,\pi),(\lambda_i',\pi')\in\mathbb{R}_+^J\times\Pi$, $d\big((\lambda_i,\pi),(\lambda_i',\pi')\big)<\delta$. Then, we have $\left|\overline{T}_i^{\lambda_i',\pi'}(\tilde{V}_i,s)-\overline{T}_i^{\lambda_i,\pi}(\tilde{V}_i,s)\right|<\varepsilon$. Therefore, we conclude that $\overline{T}_i^{\lambda_i,\pi}$ is continuous on $\mathbb{R}_+^J\times\Pi$. □

Building on Lemma B.3, we derive the following result, which ensures the continuity and equicontinuity of $\mathbf{L}^{\lambda,\pi}$.

Lemma B.4. *The function $\mathbf{L}^{\lambda,\pi}$ is continuous on $\mathbb{R}_+^{N\times J}\times\Pi$. Furthermore, the set $\{\mathbf{L}^{\lambda,\pi}(\tilde{V}):\tilde{V}$ is bounded$\}$ is equicontinuous on the metric space $(\mathbb{R}_+^{N\times J}\times\Pi,\|\cdot\|_\infty)$.*

Proof. For any $(\lambda_i,\pi_{-i}),(\lambda_i',\pi_{-i}')\in\mathbb{R}_+^J\times\Pi_{-i}$, we can find $\pi_i^*,\pi_i^\dagger\in\Pi_i$, such that:

$$\left[\mathbf{L}^{\lambda,\pi}(\tilde{V})\right]_{i,s}=\overline{T}_i^{\lambda_i,\pi_i^*,\pi_{-i}}(\tilde{V}_i,s)\ge\overline{T}_i^{\lambda_i,\pi_i^\dagger,\pi_{-i}}(\tilde{V}_i,s)$$
$$\left[\mathbf{L}^{\lambda',\pi'}(\tilde{V})\right]_{i,s}=\overline{T}_i^{\lambda_i',\pi_i^\dagger,\pi_{-i}'}(\tilde{V}_i,s)\ge\overline{T}_i^{\lambda_i',\pi_i^*,\pi_{-i}'}(\tilde{V}_i,s)$$

Then,

$$\left[\mathbf{L}^{\lambda,\pi}(\tilde{V})\right]_{i,s}-\left[\mathbf{L}^{\lambda',\pi'}(\tilde{V})\right]_{i,s}\le\overline{T}_i^{\lambda_i,\pi_i^\dagger,\pi_{-i}}(\tilde{V}_i,s)-\overline{T}_i^{\lambda_i',\pi_i^\dagger,\pi_{-i}'}(\tilde{V}_i,s)$$
$$\left[\mathbf{L}^{\lambda',\pi'}(\tilde{V})\right]_{i,s}-\left[\mathbf{L}^{\lambda,\pi}(\tilde{V})\right]_{i,s}\le\overline{T}_i^{\lambda_i',\pi_i^*,\pi_{-i}'}(\tilde{V}_i,s)-\overline{T}_i^{\lambda_i,\pi_i^*,\pi_{-i}}(\tilde{V}_i,s)$$

we have

$$\left|\left[\mathbf{L}^{\lambda,\pi}(\tilde{V})\right]_{i,s}-\left[\mathbf{L}^{\lambda',\pi'}(\tilde{V})\right]_{i,s}\right|\le\min\left\{\left|\overline{T}_i^{\lambda_i,\pi_i^\dagger,\pi_{-i}}(\tilde{V}_i,s)-\overline{T}_i^{\lambda_i',\pi_i^\dagger,\pi_{-i}'}(\tilde{V}_i,s)\right|,\left|\overline{T}_i^{\lambda_i',\pi_i^*,\pi_{-i}'}(\tilde{V}_i,s)-\overline{T}_i^{\lambda_i,\pi_i^*,\pi_{-i}}(\tilde{V}_i,s)\right|\right\}$$

From Lemma B.3, for each $\varepsilon > 0$, there exists a $\delta > 0$ such that, if for any $s \in \mathcal{S}$, $(\lambda_i, \pi_{-i}), (\lambda_i', \pi_{-i}') \in \mathbb{R}_+^J \times \Pi_{-i}$, $\pi_i^*, \pi_i^\dagger \in \Pi_i$ satisfy

$$\begin{aligned}
&\max\Big\{d\big((\lambda_i, \pi_i^\dagger, \pi_{-i}), (\lambda_i', \pi_i^\dagger, \pi_{-i}')\big), d\big((\lambda_i', \pi_i^*, \pi_{-i}'), (\lambda_i, \pi_i^*, \pi_{-i})\big)\Big\}\\
=&\max\Big\{\big\|(\lambda_i, \pi_i^\dagger, \pi_{-i}) - (\lambda_i', \pi_i^\dagger, \pi_{-i}')\big\|_\infty, \big\|(\lambda_i', \pi_i^*, \pi_{-i}') - (\lambda_i, \pi_i^*, \pi_{-i})\big\|_\infty\Big\}\\
\leq&\max\Big\{\big\|\lambda_i - \lambda_i'\big\|_\infty + \big\|\pi_i^\dagger\big\|_\infty\big\|\pi_{-i} - \pi_{-i}'\big\|_\infty, \big\|\lambda_i - \lambda_i'\big\|_\infty + \big\|\pi_i^*\big\|_\infty\big\|\pi_{-i}' - \pi_{-i}\big\|_\infty\Big\}\\
=&\big\|\lambda_i - \lambda_i'\big\|_\infty + \big\|\pi_{-i}' - \pi_{-i}\big\|_\infty \max\Big\{\big\|\pi_i^\dagger\big\|_\infty, \big\|\pi_i^*\big\|_\infty\Big\}\\
\leq&\big\|\lambda_i - \lambda_i'\big\|_\infty + \big\|\pi_{-i}' - \pi_{-i}\big\|_\infty < \delta
\end{aligned}$$

then

$$\max\left\{\left|\overline{T}_i^{\lambda_i, \pi_i^\dagger, \pi_{-i}}(\tilde{V}_i, s) - \overline{T}_i^{\lambda_i', \pi_i^\dagger, \pi_{-i}'}(\tilde{V}_i, s)\right|, \left|\overline{T}_i^{\lambda_i', \pi_i^*, \pi_{-i}'}(\tilde{V}_i, s) - \overline{T}_i^{\lambda_i, \pi_i^*, \pi_{-i}}(\tilde{V}_i, s)\right|\right\} < \varepsilon$$

Thus

$$\left|\Big[\mathbf{L}^{\lambda,\pi}(\tilde{V})\Big]_{i,s} - \Big[\mathbf{L}^{\lambda',\pi'}(\tilde{V})\Big]_{i,s}\right| < \varepsilon$$

Then if for any $(\lambda, \pi), (\lambda', \pi') \in \mathbb{R}_+^{N\times J} \times \Pi$,

$$\begin{aligned}
d\big((\lambda, \pi), (\lambda', \pi')\big) =&\big\|\lambda - \lambda'\big\|_\infty + \big\|\pi' - \pi\big\|_\infty\\
=&\big\|\lambda - \lambda'\big\|_\infty + \big\|\pi_i\big\|_\infty\big\|\pi_{-i}' - \pi_{-i}\big\|_\infty\\
\leq&\big\|\lambda - \lambda'\big\|_\infty + \min_{i\in\mathcal{N}}\big\{\big\|\pi_{-i}' - \pi_{-i}\big\|_\infty\big\}\\
\leq&\max_{i\in\mathcal{N}}\big\{\big\|\lambda_i - \lambda_i'\big\|_\infty + \big\|\pi_{-i}' - \pi_{-i}\big\|_\infty\big\} < \delta
\end{aligned}$$

we have

$$\forall (i,s) \in \mathcal{N}\times\mathcal{S} \quad \left|\Big[\mathbf{L}^{\lambda,\pi}(\tilde{V})\Big]_{i,s} - \Big[\mathbf{L}^{\lambda',\pi'}(\tilde{V})\Big]_{i,s}\right| < \varepsilon$$

then

$$\left\|\mathbf{L}^{\lambda',\pi'}(\tilde{V}) - \mathbf{L}^{\lambda,\pi}(\tilde{V})\right\|_\infty = \max_{(i,s)\in\mathcal{N}\times\mathcal{S}}\left|\Big[\mathbf{L}^{\lambda,\pi}(\tilde{V})\Big]_{i,s} - \Big[\mathbf{L}^{\lambda',\pi'}(\tilde{V})\Big]_{i,s}\right| < \varepsilon \tag{45}$$

Thus, $\mathbf{L}^{\lambda,\pi}$ is continuous on $\mathbb{R}_+^{N\times J} \times \Pi$ and because of the uniform continuity of $\overline{T}_i^{\lambda_i,\pi}$, the set $\{\mathbf{L}^{\lambda,\pi}(\tilde{V}) : \tilde{V} \text{ is bounded}\}$ is equicontinuous on the metric space $(\mathbb{R}_+^{N\times J} \times \Pi, \|\cdot\|_\infty)$. □

C PROPERTIES OF ROBUST BEST RESPONSE MAPPING

Proposition C.1. *Let $\Psi : \mathbb{R}_+^{N\times J} \times \Pi \to 2^\Pi$ be the robust best response mapping defined in Equation* (15)*. Then, the following properties hold:*

1. ***Upper Semicontinuity**: Ψ is u.s.c. on $\mathbb{R}_+^{N\times J} \times \Pi$;*
2. ***Convexity**: For any $(\lambda, \pi) \in \mathbb{R}_+^{N\times J} \times \Pi$, the set $\Psi(\lambda, \pi)$ is convex.*

Proof. To analyze the two properties of Ψ, we rely on theoretical results related to the joint-robust Bellman optimality operator established in Section B. An equivalent way of expressing $\Psi(\lambda, \pi)$, which will be crucial to our subsequent analysis, is:

$$\Psi(\lambda, \pi) = \Big\{\pi' \in \Pi \,:\, \forall (i,s) \in \mathcal{N}\times\mathcal{S},\ [\varphi(\lambda, \pi)]_{i,s} = \overline{T}_i^{\lambda_i, \pi_i', \pi_{-i}}\left([\varphi(\lambda, \pi)]_i, s\right)\Big\} \tag{46}$$

The proof is divided into two sections. In Section C.1, we examine the properties of φ using Lemma C.2 and Corollary C.3, which then lead to the upper semicontinuity of Ψ as established in Lemma C.4. In Section C.2, through the analysis of $\overline{T}_i^{\lambda_i,\pi}$ in Lemma C.5, we establish the convexity of Ψ as stated in Lemma C.6. □

C.1 UPPER SEMICONTINUITY

Lemma C.2. *The range of φ is bounded.*

Proof. By Lemma B.2, the sequence $\{\tilde{V}^0, \tilde{V}^1, \cdots\}$, where $\tilde{V}^0 = \mathbf{0}$ and $\tilde{V}^{n+1} = \mathbf{L}^{\lambda,\pi}(\tilde{V}^n)$, converges to the unique fixed point of $\mathbf{L}^{\lambda,\pi}$. Furthermore, we have:

$$\lim_{n\to\infty} \mathbf{L}^{\lambda,\pi}(\tilde{V}^n) = \varphi(\lambda,\pi)$$

and

$$\|\tilde{V}^{n+1} - \tilde{V}^n\|_\infty = \|\mathbf{L}^{\lambda,\pi}(\tilde{V}^n) - \mathbf{L}^{\lambda,\pi}(\tilde{V}^{n-1})\|_\infty \le \gamma\|\tilde{V}^n - \tilde{V}^{n-1}\|_\infty$$

Then

$$\|\tilde{V}^n - \tilde{V}^0\|_\infty \le \sum_{k=1}^{n} \|\tilde{V}^k - \tilde{V}^{k-1}\|_\infty \le \sum_{k=1}^{n} \gamma^{k-1}\|\tilde{V}^1 - \tilde{V}^0\|_\infty = \frac{1-\gamma^n}{1-\gamma}\|\tilde{V}^1 - \tilde{V}^0\|_\infty \le \frac{1}{1-\gamma}\|\mathbf{L}^{\lambda,\pi}(\tilde{V}^n) - \tilde{V}^0\|_\infty$$

Thus

$$\begin{aligned}
\|\varphi(\lambda,\pi)\|_\infty &= \|\varphi(\lambda,\pi) - \mathbf{0}\|_\infty \\
&\le \frac{1}{1-\gamma}\|\mathbf{L}^{\lambda,\pi}(\mathbf{0}) - \mathbf{0}\|_\infty \\
&= \frac{1}{1-\gamma} \max_{(i,s)\in\mathcal{N}\times\mathcal{S}} \left| \max_{\pi_i\in\Pi_i} \overline{T}_i^{\lambda_i,\pi}(\mathbf{0}, s) \right| \\
&= \frac{1}{1-\gamma} \max_{(i,s)\in\mathcal{N}\times\mathcal{S}} \left| \max_{\pi_i\in\Pi_i} \min_{p\in\mathcal{P}(s,\mathbf{a})} \mathbf{E}_{\pi,p}\left[\tilde{R}_i(\lambda_i) + \gamma\mathbf{0}\right] \right| \\
&\le \frac{1}{1-\gamma} \max_{i\in\mathcal{N},(s,\mathbf{a},s')\in\mathcal{S}\times\mathcal{A}\times\mathcal{S}} \left|\tilde{R}_i(\lambda_i)\right|
\end{aligned}$$

Hence the range of φ is bounded. □

Corollary C.3. *The function φ is continuous on $\mathbb{R}_+^{N\times J} \times \Pi$.*

Proof. From Lemma B.4, the set $\{\mathbf{L}^{\lambda,\pi}(\tilde{V}) : \tilde{V} \text{ is bounded}\}$ is equicontinuous on the metric space $(\mathbb{R}_+^{N\times J} \times \Pi, \|\cdot\|_\infty)$. Then, for each $\varepsilon > 0$, there exists a $\delta > 0$ such that, if for any (λ,π), $(\lambda',\pi') \in \mathbb{R}_+^{N\times J} \times \Pi$ satisfy

$$d\big((\lambda,\pi),(\lambda',\pi'))\big) < \delta$$

then for any bounded $\tilde{V}, \tilde{U} \in W$, we have

$$\left|\mathbf{L}^{\lambda,\pi}(\tilde{V}) - \mathbf{L}^{\lambda,\pi}(\tilde{U})\right| \le \varepsilon$$

By the definition of φ (see (14)), for each (λ,π),

$$\varphi(\lambda,\pi) = \mathbf{L}^{\lambda,\pi}(\varphi(\lambda,\pi))$$

Moreover, by Lemma C.2, the range of φ is bounded. Consequently, if

$$d\big((\lambda,\pi),(\lambda',\pi'))\big) < \delta$$

then

$$|\varphi(\lambda,\pi) - \varphi(\lambda',\pi')| = \left|\mathbf{L}^{\lambda,\pi}(\varphi(\lambda,\pi)) - \mathbf{L}^{\lambda,\pi}(\varphi(\lambda',\pi'))\right| \le \varepsilon \tag{47}$$

Since ε was arbitrary, this shows that φ is continuous on $\mathbb{R}_+^{N\times J} \times \Pi$. □

Lemma C.4. *The set-valued map Ψ is u.s.c. on $\mathbb{R}_+^{N\times J} \times \Pi$.*

Proof. To prove that $\Psi(\lambda,\pi)$ is u.s.c. on $\mathbb{R}_+^{N\times J}\times\Pi$, we need to show that whenever $\lim_{n\to\infty}(\lambda^n,\pi^n)=(\lambda,\pi)$, $\lim_{n\to\infty}\pi'^n=\pi'$ and $\pi'^n\in\Psi(\lambda^n,\pi^n)$ for all n, then it follows that $\pi'\in\Psi(\lambda,\pi)$. Since φ is continuous on $\mathbb{R}_+^{N\times J}\times\Pi$ (see Corollary C.3) and its image is bounded (Lemma C.2), it follows that for any sequence $\lim_{n\to\infty}(\lambda^n,\pi^n)=(\lambda,\pi)$, we necessarily have $\lim_{n\to\infty}\varphi(\lambda^n,\pi^n)=\varphi(\lambda,\pi)$. Let us denote $\varphi(\lambda,\pi)=\tilde{V}$. Now take any $(i,s)\in\mathcal{N}\times\mathcal{S}$. By the definition of Ψ (see (15)), if $\pi'\in\Psi(\lambda,\pi)$ then,

$$\overline{T}_i^{\lambda_i,\pi_i',\pi_{-i}}\left(\tilde{V}_i,s\right)=\tilde{V}_i(s)$$

Observe that

$$\begin{aligned}&\left\|\overline{T}_i^{\lambda_i,\pi_i',\pi_{-i}}\left(\tilde{V}_i,s\right)-\tilde{V}_i(s)\right\|_\infty\\&\leq\left\|\overline{T}_i^{\lambda_i,\pi_i',\pi_{-i}}\left(\tilde{V}_i,s\right)-\overline{T}_i^{\lambda_i^n,\pi'^n_i,\pi^n_{-i}}\left(\left[\varphi(\lambda^n,\pi^n)\right]_i,s\right)\right\|_\infty+\left\|\overline{T}_i^{\lambda_i^n,\pi'^n_i,\pi^n_{-i}}\left(\left[\varphi(\lambda^n,\pi^n)\right]_i,s\right)-\tilde{V}_i(s)\right\|_\infty\end{aligned}$$

Since $\pi'^n\in\Psi(\lambda^n,\pi^n)$, we have:

$$\left[\varphi(\lambda^n,\pi^n)\right]_{i,s}=\overline{T}_i^{\lambda_i^n,\pi'^n_i,\pi^n_{-i}}\left(\left[\varphi(\lambda^n,\pi^n)\right]_i,s\right)$$

Because $\lim_{n\to\infty}(\lambda^n,\pi^n)=(\lambda,\pi)$, $\lim_{n\to\infty}\pi'^n=\pi'$, then:

$$\begin{aligned}&\lim_{n\to\infty}\left\{\left\|\overline{T}_i^{\lambda_i,\pi_i',\pi_{-i}}\left(\tilde{V}_i,s\right)-\overline{T}_i^{\lambda_i^n,\pi'^n_i,\pi^n_{-i}}\left(\left[\varphi(\lambda^n,\pi^n)\right]_i,s\right)\right\|_\infty+\left\|\overline{T}_i^{\lambda_i^n,\pi'^n_i,\pi^n_{-i}}\left(\left[\varphi(\lambda^n,\pi^n)\right]_i,s\right)-\tilde{V}_i(s)\right\|_\infty\right\}\\&=\lim_{n\to\infty}\left\{\left\|\overline{T}_i^{\lambda_i,\pi_i',\pi_{-i}}\left(\tilde{V}_i,s\right)-\overline{T}_i^{\lambda_i^n,\pi'^n_i,\pi^n_{-i}}\left(\left[\varphi(\lambda^n,\pi^n)\right]_i,s\right)\right\|_\infty+\left\|\left[\varphi(\lambda^n,\pi^n)\right]_{i,s}-\tilde{V}_i(s)\right\|_\infty\right\}\\&=\left\|\overline{T}_i^{\lambda_i,\pi_i',\pi_{-i}}\left(\tilde{V}_i,s\right)-\overline{T}_i^{\lambda_i,\pi_i',\pi_{-i}}\left(\tilde{V}_i,s\right)\right\|_\infty+\left\|\tilde{V}_i(s)-\tilde{V}_i(s)\right\|_\infty=0\end{aligned}$$

Thus for any $(i,s)\in\mathcal{N}\times\mathcal{S}$, $\overline{T}_i^{\lambda_i,\pi_i',\pi_{-i}}\left(\tilde{V}_i,s\right)=\tilde{V}_i(s)$ and hence

$$\forall(i,s)\in\mathcal{N}\times\mathcal{S},\quad\left[\varphi(\lambda,\pi)\right]_{i,s}=\overline{T}_i^{\lambda_i,\pi_i',\pi_{-i}}\left(\left[\varphi(\lambda,\pi)\right]_i,s\right)\tag{48}$$

Therefore, we conclude $\pi'\in\Psi(\lambda,\pi)$. This completes the proof of upper semicontinuity. □

C.2 CONVEXITY

Lemma C.5. *For any $s\in\mathcal{S}$, with π_{-i} and λ_i fixed, the function $\overline{T}_i^{\lambda_i,\pi_i,\pi_{-i}}(\tilde{V}_i,s)$ is concave on Π_i.*

Proof. Let $\pi_i,\pi_i'\in\Pi_i$, and let $\eta\in[0,1]$. We define a new policy $\pi_i^\dagger\in\Pi_i$ as a convex combination of π_i and π_i'. Formally,

$$\pi_i^\dagger(a_i\mid s)=\eta\pi_i(a_i\mid s)+(1-\eta)\pi_i'(a_i\mid s),\quad\forall(s,a_i)\in\mathcal{S}\times\mathcal{A}_i\tag{49}$$

By the definition of $\overline{T}_i^{\lambda_i,\pi_i,\pi_{-i}}$, there exists a transition probability $p^\dagger\in\mathcal{P}(s,\mathbf{a})$ such that

$$\overline{T}_i^{\lambda_i,\pi_i^\dagger,\pi_{-i}}(\tilde{V}_i,s)=\mathbf{E}_{a_i\sim\pi_i^\dagger(\cdot|s),\mathbf{a}_{-i}\sim\pi_{-i}(\cdot|s),s'\sim p^\dagger(\cdot)}\left[\tilde{R}_i(\lambda_i)+\gamma\tilde{V}_i(s')\right]\tag{50}$$

Now, consider the convex combination of $\overline{T}_i^{\lambda_i,\pi_i,\pi_{-i}}(\tilde{V}_i,s)$ and $\overline{T}_i^{\lambda_i,\pi_i',\pi_{-i}}(\tilde{V}_i,s)$:

$$\begin{aligned}&\eta\overline{T}_i^{\lambda_i,\pi_i,\pi_{-i}}(\tilde{V}_i,s)+(1-\eta)\overline{T}_i^{\lambda_i,\pi_i',\pi_{-i}}(\tilde{V}_i,s)\\&=\eta\min_{p\in\mathcal{P}(s,\mathbf{a})}\mathbf{E}_{a_i\sim\pi_i(\cdot|s),\mathbf{a}_{-i}\sim\pi_{-i}(\cdot|s),s'\sim p(\cdot)}\left[\tilde{R}_i(\lambda_i)+\gamma\tilde{V}_i(s')\right]+(1-\eta)\min_{p\in\mathcal{P}(s,\mathbf{a})}\mathbf{E}_{a_i\sim\pi_i'(\cdot|s),\mathbf{a}_{-i}\sim\pi_{-i}(\cdot|s),s'\sim p(\cdot)}\left[\tilde{R}_i(\lambda_i)+\gamma\tilde{V}_i(s')\right]\\&\leq\eta\mathbf{E}_{a_i\sim\pi_i(\cdot|s),\mathbf{a}_{-i}\sim\pi_{-i}(\cdot|s),s'\sim p^\dagger(\cdot)}\left[\tilde{R}_i(\lambda_i)+\gamma\tilde{V}_i(s')\right]+(1-\eta)\mathbf{E}_{a_i\sim\pi_i'(\cdot|s),\mathbf{a}_{-i}\sim\pi_{-i}(\cdot|s),s'\sim p^\dagger(\cdot)}\left[\tilde{R}_i(\lambda_i)+\gamma\tilde{V}_i(s')\right]\\&=\mathbf{E}_{\mathbf{a}_{-i}\sim\pi_{-i}(\cdot|s),s'\sim p^\dagger(\cdot)}\sum_{a_i\in\mathcal{A}_i}\left\{\eta\pi(a_i\mid s)\left[\tilde{R}_i(\lambda_i)+\gamma\tilde{V}_i(s')\right]+(1-\eta)\pi'(a_i\mid s)\left[\tilde{R}_i(\lambda_i)+\gamma\tilde{V}_i(s')\right]\right\}\end{aligned}$$

$$
\begin{aligned}
=&\mathbf{E}_{\mathbf{a}_{-i}\sim\pi_{-i}(\cdot|s),s'\sim p^\dagger(\cdot)}\sum_{a_i\in\mathcal{A}_i}\left\{\pi^\dagger(a_i\mid s)\left[\tilde{R}_i(\lambda_i)+\gamma\tilde{V}_i(s')\right]\right\}\\
=&\mathbf{E}_{a_i\sim\pi_i^\dagger(\cdot|s),\mathbf{a}_{-i}\sim\pi_{-i}(\cdot|s),s'\sim p^\dagger(\cdot)}\left[\tilde{R}_i(\lambda_i)+\gamma\tilde{V}_i(s')\right]\\
=&\overline{T}_i^{\lambda_i,\pi_i^\dagger,\pi_{-i}}(\tilde{V}_i,s)
\end{aligned}
\tag{51}
$$

Therefore, $\overline{T}_i^{\lambda_i,\pi_i,\pi_{-i}}(\tilde{V}_i,s)$ is concave in π_i. □

Lemma C.6. *For any $(\lambda,\pi)\in\mathbb{R}_+^{N\times J}\times\Pi$, the range of $\Psi(\lambda,\pi)$ is convex.*

Proof. Let $\pi^\dagger,\pi'\in\Psi(\lambda,\pi)$, Then for any $(i,s)\in\mathcal{N}\times\mathcal{S}$, we have

$$
\tilde{V}_i(s)=[\varphi(\lambda,\pi)]_{i,s}=\overline{T}_i^{\lambda_i,\pi_i',\pi_{-i}}\left([\varphi(\lambda,\pi)]_i,s\right)=\overline{T}_i^{\lambda_i,\pi_i^\dagger,\pi_{-i}}\left([\varphi(\lambda,\pi)]_i,s\right)
$$

and for any $\pi^\star\in\Pi$:

$$
\tilde{V}_i(s)\geq\overline{T}_i^{\lambda_i,\pi_i^\star,\pi_{-i}}\left([\varphi(\lambda,\pi)]_i,s\right)
$$

We can find a $\pi^*\in\Pi$ such that for any $\eta\in[0,1]$,

$$
\forall(i,s,a_i)\in\mathcal{N}\times\mathcal{S}\times\mathcal{A}_i,\quad \pi_i^*(a_i\mid s)=\eta\pi_i^\dagger(a_i\mid s)+(1-\eta)\pi_i'(a_i\mid s)
$$

From Lemma C.5, $\overline{T}_i^{\lambda_i,\pi_i,\pi_{-i}}(\tilde{V}_i,s)$ is concave on Π_i, then,

$$
\tilde{V}_i(s)=\eta\overline{T}_i^{\lambda_i,\pi_i',\pi_{-i}}\left([\varphi(\lambda,\pi)]_i,s\right)+(1-\eta)\overline{T}_i^{\lambda_i,\pi_i^\dagger,\pi_{-i}}\left([\varphi(\lambda,\pi)]_i,s\right)\leq\overline{T}_i^{\lambda_i,\pi_i^*,\pi_{-i}}\left([\varphi(\lambda,\pi)]_i,s\right)\leq\tilde{V}_i(s)
$$

Then, we have

$$
\eta\overline{T}_i^{\lambda_i,\pi_i',\pi_{-i}}\left([\varphi(\lambda,\pi)]_i,s\right)+(1-\eta)\overline{T}_i^{\lambda_i,\pi_i^\dagger,\pi_{-i}}\left([\varphi(\lambda,\pi)]_i,s\right)=\overline{T}_i^{\lambda_i,\pi_i^*,\pi_{-i}}\left([\varphi(\lambda,\pi)]_i,s\right)
\tag{52}
$$

Thus, the range of $\Psi(\lambda,\pi)$ is convex. □

D PROOF OF THE PROPERTIES OF ROBUST AND FEASIBLE BEST RESPONSE MAPPING

We establish the upper semicontinuity and convexity of robust and feasible best response mapping $\overline{\Psi}$ in Equation (19) via properties of Ψ and Φ.

Lemma D.1. *The set-valued map $\overline{\Psi}$ is upper semicontinuous (u.s.c.) on Π.*

Proof. We first establish the relationship between the mapping φ and the dual function $\max_{\pi_i\in\Pi_i}\bar{V}_i^{SL}(\pi,\lambda)$:

$$
[\varphi(\lambda,\pi)]_{i,\rho}+\lambda_i^\top c_i=\max_{\pi_i\in\Pi_i}\tilde{V}_i^{\lambda_i,\pi_i,\pi_{-i}}(\rho)+\lambda_i^\top c_i=\max_{\pi_i\in\Pi_i}\bar{V}_i^{SL}(\pi,\lambda)
\tag{53}
$$

Since φ is continuous on $\mathbb{R}_+^{N\times J}\times\Pi$ (see Corollary C.3), it follows that, for each agent $i\in\mathcal{N}$, the dual function $\max_{\pi_i\in\Pi_i}\bar{V}_i^{SL}(\pi,\lambda)$ is continuous on $\mathbb{R}_+^J\times\Pi_i$. Recall that Φ is defined as the optimal set mapping for the family of functions $\left(\max_{\pi_i\in\Pi_i}\bar{V}_i^{SL}(\pi,\lambda)\right)_{i\in\mathcal{N}}$. By Theorem A.8 (Basic Continuity Properties of Optimal Value Mapping), we know Φ is u.s.c. on Π. In Section C, we also show that Ψ is u.s.c. on $\mathbb{R}_+^{N\times J}\times\Pi$.

Finally, by Theorem A.6 (Upper Semicontinuity of Composition of Set-valued Mapping), the composition $\overline{\Psi}(\pi)=\bigcup_{\lambda\in\Phi(\pi)}\Psi(\lambda,\pi)$ is itself u.s.c. on Π. This concludes the proof. □

Lemma D.2. *For any $\pi\in\Pi$, the range of $\overline{\Psi}(\pi)$ is convex.*

Proof. As discussed in Section 5.2 of Boyd and Vandenberghe [8], regardless of whether the primal problem is convex, its dual problem is always convex. Consequently, the range of $\Phi(\pi)$ is convex. Let $\pi^\dagger \in \Psi(\lambda^\dagger, \pi) \subseteq \overline{\Psi}(\pi)$ and $\pi' \in \Psi(\lambda', \pi) \subseteq \overline{\Psi}(\pi)$. By the convexity of Φ, for any $\eta \in [0, 1]$, we have:

$$\lambda^* = \eta\lambda^\dagger + (1-\eta)\lambda' \in \Phi(\pi) \tag{54}$$

By the representation of $\overline{\Psi}$ in (19), it follows that $\Psi(\lambda^*, \pi) \subseteq \overline{\Psi}(\pi)$. Next, by Lemma A.9 (Necessity and Sufficiency Condition of Convex Set-valued Mapping) and the convexity of Ψ (see Section C),

$$\eta\Psi(\lambda^\dagger, \pi) + (1-\eta)\Psi(\lambda', \pi) \subseteq \Psi(\eta\lambda^\dagger + (1-\eta)\lambda', \pi) = \Psi(\lambda^*, \pi) \tag{55}$$

Thus, we can construct:

$$\pi^* = \eta\pi^\dagger + (1-\eta)\pi' \in \eta\Psi(\lambda^\dagger, \pi) + (1-\eta)\Psi(\lambda', \pi) \subseteq \Psi(\lambda^*, \pi) \subseteq \overline{\Psi}(\pi) \tag{56}$$

Therefore, the range of $\overline{\Psi}(\pi)$ is convex. □

E EXPERIMENTAL ENVIRONMENT AND EMPIRICAL RESULTS

E.1 ENVIRONMENT AND ANALYTICAL EQUILIBRIUM

We consider a grid world with three cells, denoted as Cell 1, Cell 2, and Cell 3. Two agents ($\mathcal{N} = \{1, 2\}$) simultaneously occupy the grid, and each cell can hold one or both agents at once. The state space $\mathcal{S}$ includes all possible combinations of the two agents' positions. Because each agent can be in any of the three cells independently, the state space has $|\mathcal{S}| = 9$ states. The 9 states are illustrated below, where red circles represent Agent 1, and blue circles represent Agent 2. The notation $(\cdot, \cdot)$ denotes the positions of Agent 1 and Agent 2, respectively:

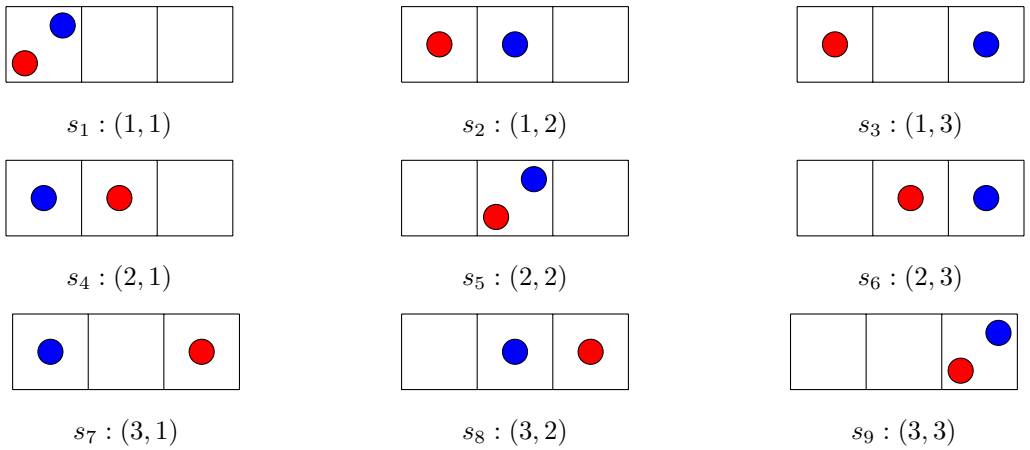

Figure 3: Visualization of the 9 states in the grid world. Red circles represent Agent 1, and blue circles represent Agent 2. The notation $(\cdot, \cdot)$ denotes the positions of Agent 1 and Agent 2, respectively.

The initial state distribution is assumed to be uniform over all 9 states:

$$\rho(s_i) = \frac{1}{9}, \quad i = 1, 2, \cdots, 9.$$

Each agent has the same action space $\mathcal{A}_i = \{L, S, R\}$, Here, L denotes moving one cell to the left, S represents staying in the current cell, and R indicates moving one cell to the right. The transition kernel P deterministically executes the agents' actions. For instance, $P(s' = s_3 \mid s = s_5, \mathbf{a} = (L, R)) = 1$ implies that starting from $s_5 = (2, 2)$, if Agent 1 moves one cell left and Agent 2 moves one cell right, the resulting state becomes $s_3 = (1, 3)$.

The reward function is defined such that Agent 1 receives a reward of 1 if it ends up in the same cell as Agent 2 after executing its action; otherwise, the reward is 0. Conversely, Agent 2 receives a reward of 1 if it ends up in a different cell from Agent 1 after executing its action; otherwise, the reward is 0. Specifically, this means that when the agents transition to states s_1, s_5, or s_9, Agent 1 is rewarded, while in all other states (s_2, s_3, s_4, s_6, s_7, s_8), Agent 2 is rewarded. Formally, the reward functions can be expressed as follows:

$$R_1(\cdot,\cdot,s') = \begin{cases} 1 & \text{if } s' \in \{s_1, s_5, s_9\}, \\ 0 & \text{otherwise} \end{cases} \qquad \text{and} \qquad R_2(\cdot,\cdot,s') = \begin{cases} 0 & \text{if } s' \in \{s_1, s_5, s_9\}, \\ 1 & \text{otherwise} \end{cases}$$

To maximize the expected cumulative rewards, Agent 1 must strive to catch Agent 2, while Agent 2 must avoid Agent 1 as much as possible. Consequently, the Nash equilibrium (π_1, π_2) of this Markov game is as follows:

s (Given State)	π_1 (Agent 1 Policy)	π_2 (Agent 2 Policy)	s' (Next State)
$s_1 = (1,1)$	$\frac{1}{2}(S/L), \frac{1}{2}R$	$\frac{1}{2}(S/L), \frac{1}{2}R$	$\frac{1}{4}s_1, \frac{1}{4}s_2, \frac{1}{4}s_4, \frac{1}{4}s_5$
$s_2 = (1,2)$	R	R	s_6
$s_3 = (1,3)$	R	S/R	s_6
$s_4 = (2,1)$	$\frac{1}{2}L, \frac{1}{2}S$	$\frac{1}{2}R, \frac{1}{2}(S/L)$	$\frac{1}{4}s_2, \frac{1}{4}s_1, \frac{1}{4}s_5, \frac{1}{4}s_4$
$s_5 = (2,2)$	$\frac{1}{3}L, \frac{1}{3}R, \frac{1}{3}S$	$\frac{1}{3}L, \frac{1}{3}R, \frac{1}{3}S$	$\frac{1}{9}$ for all states
$s_6 = (2,3)$	$\frac{1}{2}R, \frac{1}{2}S$	$\frac{1}{2}L, \frac{1}{2}(S/R)$	$\frac{1}{4}s_8, \frac{1}{4}s_9, \frac{1}{4}s_5, \frac{1}{4}s_6$
$s_7 = (3,1)$	L	S/L	s_4
$s_8 = (3,2)$	L	L	s_4
$s_9 = (3,3)$	$\frac{1}{2}L, \frac{1}{2}(S/R)$	$\frac{1}{2}L, \frac{1}{2}(S/R)$	$\frac{1}{4}s_8, \frac{1}{4}s_9, \frac{1}{4}s_5, \frac{1}{4}s_6$

Table 1: Nash equilibrium for Agent 1 and Agent 2, along with the next state (s'). The notation $(\cdot/\cdot)$ indicates that the two actions are indistinguishable in their effects on the state transition for the given state.

We extend the above Markov game to a robust constrained Markov game (RCMG) by introducing uncertainty in the transition dynamics and a cost function for Agent 2. The uncertainty set is defined as $\mathcal{P} = \{P, P_1, P_2, P_3\}$, where P represents the fully deterministic transition kernel, and P_k $(k = 1, 2, 3)$ represents the transition kernel in which the actions within the Cell k become fully stochastic. Specifically, under P_k, in the Cell k, the actions L, S, and R are executed with equal probability $(\frac{1}{3})$, regardless of the agents' choices. Additionally, Agent 2 now faces a cost function C_2. This cost is 1 whenever Agent 2 moves into the leftmost cell (Cell 1), and 0 otherwise:

$$C_2(\cdot,\cdot,s') = \begin{cases} 1 & \text{if } s' \in \{s_1, s_4, s_7\}, \\ 0 & \text{otherwise.} \end{cases}$$

Agent 2 aims to maximize its expected cumulative reward while ensuring that its expected cumulative cost does not exceed a predefined constraint. To account for the worst-case transition kernel for Agent 2, we focus on P_2. Introducing P_1 does not result in additional costs because, even if Agent 2 starts in the leftmost cell, it can move to the middle cell to avoid incurring a cost. Similarly, while P_1 may cause Agent 2 to remain in the leftmost cell with a probability of $\frac{2}{3}$, Agent 2 only needs to exit the leftmost cell once to reach safe regions. Moreover, the initial probability of being in the leftmost cell is only $\frac{1}{3}$, limiting its overall impact on the cost. In contrast, under P_2, the only way for Agent 2 to avoid incurring a cost is to remain in the rightmost cell. This scenario allows Agent 1 to position itself in the rightmost cell, thereby preventing Agent 2 from receiving rewards. Thus, P_2 simultaneously maximizes the expected cumulative cost and minimizes the expected cumulative reward for Agent 2, making it the worst-case transition kernel.

For Agent 1, there is no need to consider costs; its objective is simply to remain in the same cell as Agent 2 as much as possible to maximize its reward. Knowing that Agent 2 will attempt to stay in the rightmost cell, Agent 1's optimal strategy is also to remain in the rightmost cell. Thus, the worst-case transition kernel for Agent 1 is P_3. This increases the likelihood

of Agent 1 being forced out of the rightmost cell, thereby reducing its expected cumulative reward as it becomes less likely to stay in the same cell as Agent 2.

Based on the above analysis, the Robust and Feasible Nash Equilibrium for this RCMG is as follows:

s (Given State)	π_1 (Agent 1 Policy)	π_2 (Agent 2 Policy)	s' (Next State)
$s_1 = (1,1)$	R	R	s_5
$s_2 = (1,2)$	R	Uncertain	$\frac{1}{3}s_4, \frac{1}{3}s_5, \frac{1}{3}s_6$
$s_3 = (1,3)$	R	S/R	s_6
$s_4 = (2,1)$	S	R	s_5
$s_5 = (2,2)$	S	Uncertain	$\frac{1}{3}s_4, \frac{1}{3}s_5, \frac{1}{3}s_6$
$s_6 = (2,3)$	R	S/R	s_9
$s_7 = (3,1)$	Uncertain	R	$\frac{1}{3}s_5, \frac{2}{3}s_8$
$s_8 = (3,2)$	Uncertain	Uncertain	$\frac{1}{9}s_4, \frac{1}{9}s_5, \frac{1}{9}s_6, \frac{2}{9}s_7, \frac{2}{9}s_8, \frac{2}{9}s_9$
$s_9 = (3,3)$	Uncertain	S/R	$\frac{1}{3}s_6, \frac{2}{3}s_9$

Table 2: Robust and Feasible Nash Equilibrium for Agent 1 and Agent 2, along with the next state (s'). Policies labeled as "Uncertain" explicitly indicate that the actions L, S, and R are executed with equal probability ($\frac{1}{3}$). The notation $(\cdot/\cdot)$ indicates that the two actions are indistinguishable in their effects on the state transition for the given state.

According to the above table, we can derive Agent 2's value function under the optimal policy π^* by applying the Bellman equation. The value function $\bar{V}_2^{\pi^*}(s)$ for each state $s \in \mathcal{S}$ is determined by the immediate reward $R_2(\cdot,\cdot,s')$ plus the discounted value of the next state $\bar{V}_2^{\pi^*}(s')$. Formally, for each state s_i, we write:

$$\bar{V}_2^{\pi^*}(s_1) = R_2(s_1,(R,R),s_5) + \gamma \bar{V}_2^{\pi^*}(s_5)$$
$$\bar{V}_2^{\pi^*}(s_2) = \frac{1}{3}\left[R_2(s_2,(R,L),s_4) + \gamma \bar{V}_2^{\pi^*}(s_4)\right] + \frac{1}{3}\left[R_2(s_2,(R,S),s_5) + \gamma \bar{V}_2^{\pi^*}(s_5)\right] + \frac{1}{3}\left[R_2(s_2,(R,R),s_6) + \gamma \bar{V}_2^{\pi^*}(s_6)\right]$$
$$\bar{V}_2^{\pi^*}(s_3) = R_2(s_3,(R,S/R),s_6) + \gamma \bar{V}_2^{\pi^*}(s_6)$$
$$\bar{V}_2^{\pi^*}(s_4) = R_2(s_4,(S,R),s_5) + \gamma \bar{V}_2^{\pi^*}(s_5)$$
$$\bar{V}_2^{\pi^*}(s_5) = \frac{1}{3}\left[R_2(s_5,(S,L),s_4) + \gamma \bar{V}_2^{\pi^*}(s_4)\right] + \frac{1}{3}\left[R_2(s_5,(S,S),s_5) + \gamma \bar{V}_2^{\pi^*}(s_5)\right] + \frac{1}{3}\left[R_2(s_5,(S,R),s_6) + \gamma \bar{V}_2^{\pi^*}(s_6)\right]$$
$$\bar{V}_2^{\pi^*}(s_6) = R_2(s_6,(R,S/R),s_9) + \gamma \bar{V}_2^{\pi^*}(s_9)$$
$$\bar{V}_2^{\pi^*}(s_7) = \frac{1}{3}\left[R_2(s_7,(L,R),s_5) + \gamma \bar{V}_2^{\pi^*}(s_5)\right] + \frac{2}{3}\left[R_2(s_7,(S/R,R),s_8) + \gamma \bar{V}_2^{\pi^*}(s_8)\right]$$
$$\bar{V}_2^{\pi^*}(s_8) = \frac{1}{9}\left[R_2(s_8,(L,L),s_4) + \gamma \bar{V}_2^{\pi^*}(s_4)\right] + \frac{1}{9}\left[R_2(s_8,(S,L),s_5) + \gamma \bar{V}_2^{\pi^*}(s_5)\right] + \frac{1}{9}\left[R_2(s_8,(R,L),s_6) + \gamma \bar{V}_2^{\pi^*}(s_6)\right]$$
$$+ \frac{2}{9}\left[R_2(s_8,(L,S/R),s_7) + \gamma \bar{V}_2^{\pi^*}(s_7)\right] + \frac{2}{9}\left[R_2(s_8,(S,S/R),s_8) + \gamma \bar{V}_2^{\pi^*}(s_8)\right] + \frac{2}{9}\left[R_2(s_8,(R,S/R),s_9) + \gamma \bar{V}_2^{\pi^*}(s_9)\right]$$
$$\bar{V}_2^{\pi^*}(s_9) = \frac{1}{3}\left[R_2(s_9,(L,S/R),s_6) + \gamma \bar{V}_2^{\pi^*}(s_6)\right] + \frac{2}{3}\left[R_2(s_9,(S/R,S/R),s_9) + \gamma \bar{V}_2^{\pi^*}(s_9)\right]$$

Recall that the reward R_2 equals 1 in states $s_2, s_3, s_4, s_6, s_7, s_8$, and 0 otherwise. We choose the discount factor $\gamma = 0.9$ to balance immediate and future rewards. After solving the resulting system of linear equations, we obtain the values for $\bar{V}_2^{\pi^*}(s_1), \bar{V}_2^{\pi^*}(s_2), \ldots, \bar{V}_2^{\pi^*}(s_9)$:

$$\bar{V}_2^{\pi^*}(s_1) = \frac{1590}{559}, \quad \bar{V}_2^{\pi^*}(s_2) = \frac{5300}{1677}, \quad \bar{V}_2^{\pi^*}(s_3) = \frac{120}{39}, \quad \bar{V}_2^{\pi^*}(s_4) = \frac{1590}{559}, \quad \bar{V}_2^{\pi^*}(s_5) = \frac{5300}{1677},$$
$$\bar{V}_2^{\pi^*}(s_6) = \frac{90}{39}, \quad \bar{V}_2^{\pi^*}(s_7) = \frac{104740}{28509}, \quad \bar{V}_2^{\pi^*}(s_8) = \frac{97840}{28509}, \quad \bar{V}_2^{\pi^*}(s_9) = \frac{100}{39}$$

Finally, to compute Agent 2's overall expected return from the initial state distribution ρ, we take the average of $\bar{V}_2^{\pi^*}(s)$ over all s sampled from ρ, i.e.,

$$\bar{V}_2^{\pi^*}(\rho) = \mathbf{E}_{s\sim\rho}\bar{V}_2^{\pi^*}(s) = \frac{1}{9}\sum_{s_i\in\mathcal{S}} \bar{V}_2^{\pi^*}(s_i) = \frac{85730}{28509} \approx 3.007$$

In addition, we can similarly compute its cost value function, denoted $\bar{\Lambda}_2^{\pi^*}(s)$. The Bellman recursion for the cost follows exactly the same structure as the reward Bellman equation, except that the immediate reward R_2 is replaced by the immediate cost C_2. Recall that the cost C_2 equals 1 in states s_1, s_4, s_7, and 0 otherwise. Solving the resulting linear system yields the following cost values for $\bar{\Lambda}_2^{\pi^*}(s_1), \bar{\Lambda}_2^{\pi^*}(s_2), \ldots, \bar{\Lambda}_2^{\pi^*}(s_9)$:

$$\bar{\Lambda}_2^{\pi^*}(s_1) = \frac{30}{43}, \quad \bar{\Lambda}_2^{\pi^*}(s_2) = \frac{100}{129}, \quad \bar{\Lambda}_2^{\pi^*}(s_3) = \bar{\Lambda}_2^{\pi^*}(s_6) = \bar{\Lambda}_2^{\pi^*}(s_9) = 0,$$
$$\bar{\Lambda}_2^{\pi^*}(s_4) = \frac{30}{43}, \quad \bar{\Lambda}_2^{\pi^*}(s_5) = \frac{100}{129}, \quad \bar{\Lambda}_2^{\pi^*}(s_7) = \frac{30}{43}, \quad \bar{\Lambda}_2^{\pi^*}(s_8) = \frac{100}{129},$$

Averaging over the uniform initial distribution ρ gives

$$\bar{\Lambda}_2^{\pi^*}(\rho) = \frac{1}{9}\sum_{i=1}^{9} \bar{\Lambda}_2^{\pi^*}(s_i) = \frac{190}{387} \approx 0.490956.$$

Finally, we can also obtain Agent 1's value function under the same policy π^*. By construction, Agent 1's stage-wise reward is the complement of Agent 2's reward:

$$R_1(s, a, s') \;=\; 1 - R_2(s, a, s').$$

Hence, at each time step $R_1(s_t, \mathbf{a}_t, s_{t+1}) + R_2(s_t, \mathbf{a}_t, s_{t+1}) = 1$, and therefore the discounted returns satisfy

$$\bar{V}_1^{\pi^*}(s) + \bar{V}_2^{\pi^*}(s) = \mathbf{E}_{\pi^*,P}\left[\sum_{t=0}^{\infty}\gamma^t\big(R_1 + R_2\big)\,\Big|\, s_0 = s\right] = \sum_{t=0}^{\infty}\gamma^t = \frac{1}{1-\gamma}.$$

With $\gamma = 0.9$ this gives $\bar{V}_1^{\pi^*}(s) = 10 - \bar{V}_2^{\pi^*}(s)$ for all $s \in \mathcal{S}$, and averaging over the uniform initial distribution ρ yields

$$\bar{V}_1^{\pi^*}(\rho) = 10 - \bar{V}_2^{\pi^*}(\rho) = \frac{199360}{28509} \approx 6.992879.$$

E.2 EXPERIMENTAL SETUP

We summarize the hyperparameters and implementation details used in our experiments:

- **Initial policies.** Each agent's initial policy $\pi_i(0)$ is generated by sampling an action uniformly at random from $\{L, S, R\}$ for every state, avoiding bias toward any particular initial behavior.
- **Initial Lagrange multiplier.** The multiplier for the single constraint is initialized from a wide uniform distribution $\lambda(0) \sim \mathrm{Uniform}(0.01, 10)$, which improves robustness across different constraint thresholds.
- **Iteration limits.** The dual update loop allows up to $T_\lambda = 10000$ iterations; the policy-update loop allows up to $T_\pi = 1000$ iterations per outer update. Inside each λ update, robust Q-value iteration runs for a maximum of 10000 steps (terminating early upon convergence).
- **Step-size schedule.** At the beginning of each dual-update phase the step-size is reset to $\alpha_0 = 1$ and decayed via $\alpha_k = \alpha_0\,\eta^k$ with $\eta = 0.2$. Because the inner dual problem here is finite and low-dimensional, the multiplier reaches its optimum within a few iterations in all our runs, so we adopt this fast geometric schedule as a practical choice. It does not by itself meet the diminishing step-size condition (17) used for the asymptotic convergence guarantee; a schedule such as $\alpha_k = \alpha_0/(k+1)^{0.9}$ does satisfy (17) and converges to the same equilibrium when an asymptotic guarantee is desired.
- **Multiple runs with fixed seeds.** All experiments are repeated over five independent runs using the same fixed set of five random seeds to ensure consistency and fair comparison.

E.3 VALIDATION AT $c_2 = 0.5$

We first report performance under the strictest constraint level $c_2 = 0.5$, recording the runtime, the number of dual updates T_λ, the number of outer policy-iteration updates T_π, and the resulting values $\bar{V}_2^{\pi^*}(\rho)$ and $\bar{\Lambda}_2^{\pi^*}(\rho)$; the results across all five seeds are reported in Table 3. Because the grid dynamics are deterministic given the chosen kernel and the seeds randomize only the initial policy $\pi_i(0)$ and multiplier $\lambda(0)$, every seed converges to the same fixed point; the seeds therefore affect only the convergence speed (runtime, T_λ, T_π), not the converged value or cost.

Runtime (s)	T_λ	T_π	$\bar{V}_2^{\pi^*}(\rho)$	$\bar{\Lambda}_2^{\pi^*}(\rho)$
0.15	4	3	3.00712	0.490956
0.20	5	3	3.00712	0.490956
0.15	4	3	3.00712	0.490956
0.22	5	3	3.00712	0.490956
0.15	4	3	3.00712	0.490956

Table 3: Performance of Algorithm 1 under constraint $c_2 = 0.5$. Across all five seeds the algorithm converges to the theoretically predicted $\bar{V}_2^{\pi^*}(\rho) \approx 3.00712$ and $\bar{\Lambda}_2^{\pi^*}(\rho) \approx 0.490956$.

Figure 4 (left) shows how Agent 2's robust cost value evolves over successive iterations. The dashed horizontal line marks the cost constraint of 0.5, and the empirical cost converges near this boundary, indicating that the gradient-based updates to λ effectively enforce the constraint. Meanwhile, Figure 4 (right) depicts how Agent 2's robust value converges to approximately the theoretical optimum of $\frac{85730}{28509} \approx 3.007$. Hence, under worst-case transitions, Agent 2 not only meets the cost requirement but also attains the robust optimal reward, demonstrating that the learned equilibrium closely matches our analytical predictions.

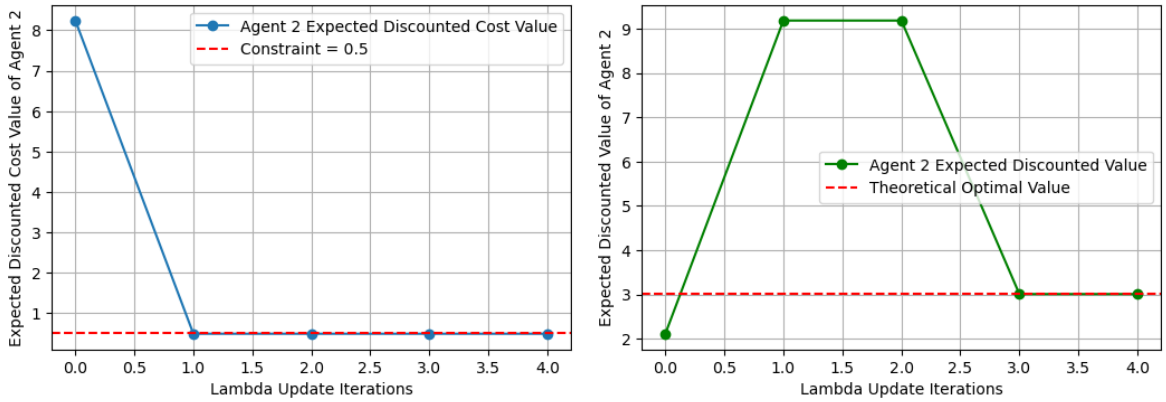

Figure 4: Evolution of Agent 2's expected discounted cumulative cost and reward over iterations. The cost (left) converges near the constraint as λ is updated, while the reward (right) approaches the theoretical optimal value.

E.4 CONSTRAINT-THRESHOLD SWEEP

Having validated the strictest setting $c_2 = 0.5$, we now examine how the constraint level shapes the algorithm's behavior. The threshold plays an inherent role: too small a bound is infeasible for Agent 2, whereas a bound exceeding $\frac{1}{1-\gamma}$ is vacuous and recovers the unconstrained problem. Between these extremes, we gradually increase the threshold while keeping exactly the same set of random seeds. The results in Table 6 show that for moderate constraint levels $c_2 \in [0.6, 1.6]$, both the required number of dual updates T_λ and the overall runtime remain essentially stable. Loosening the constraint therefore does not substantially slow down the convergence of our primal–dual procedure, and the method remains numerically robust across this entire range of thresholds. (Value and cost columns are omitted from the sweep tables since they remain constant at $\bar{V}_2^{\pi^*}(\rho) \approx 3.00712$ and $\bar{\Lambda}_2^{\pi^*}(\rho) \approx 0.490956$ across all feasible levels.)

To illustrate the behavior at larger constraint levels, Figure 5 plots the evolution of $\bar{\Lambda}_2^{\pi^*}(\rho)$ and $\bar{V}_2^{\pi^*}(\rho)$ for $c_2 = 1.5$, among the slowest but still convergent instances in Table 6. Even though the constraint is considerably looser than in the $c_2 = 0.5$

setting, the algorithm still converges reliably, though both curves exhibit pronounced oscillations during the early stages of the dual updates: the cost fluctuates around the feasibility boundary while the reward shows multiple sharp deviations before stabilizing. These transient oscillations reflect the enlarged feasible region under higher constraint levels but do not prevent convergence. In several stages the cost temporarily falls below the threshold while the reward simultaneously rises above the theoretical optimum, because Agent 1 has not yet settled on its best response during these early iterations.

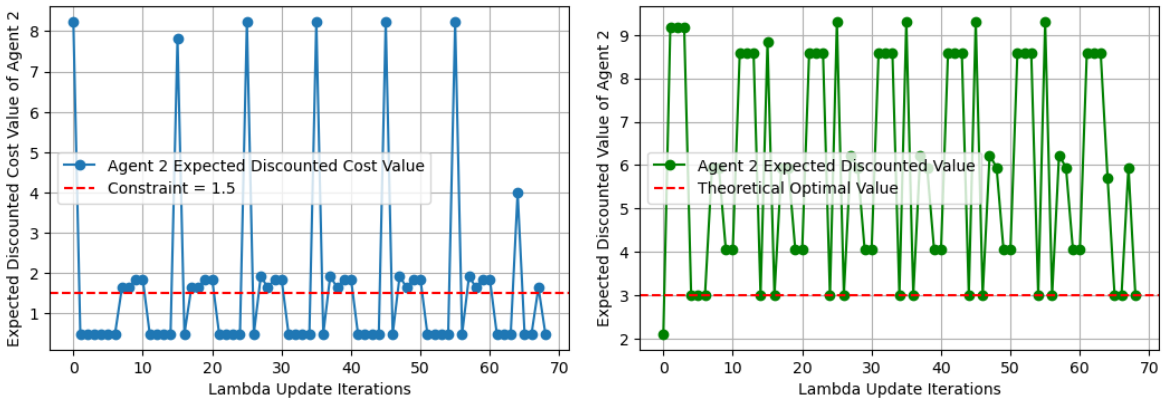

Figure 5: Evolution of Agent 2's cost value and reward value over iterations under $c_2 = 1.5$.

E.5 LIMITATIONS AND STABILIZATION

If we further increase the constraint level to $c_2 = 1.7$, the behavior of the algorithm changes markedly compared to the trend in Table 6. Among the five random seeds, two runs fail to converge within the prescribed iteration limits. To understand this instability, we record the pair $(\bar{V}_1^{\pi^*}(\rho), \bar{V}_2^{\pi^*}(\rho))$ at every outer iteration for the non-divergent runs. The most frequently visited value pairs over all 1000 policy-improvement iterations are summarized in Table 4.

Value pair $(\bar{V}_1^{\pi^*}(\rho), \bar{V}_2^{\pi^*}(\rho))$	**Frequency**
$(6.993, 3.007)$	250
$(5.950, 8.570)$	249
$(5.950, 4.050)$	192
$(6.993, 5.946)$	192
$(5.777, 4.050)$	58
$(6.993, 6.228)$	58
$(8.501, 9.182)$	1

Table 4: Value-pair visit counts at $c_2 = 1.7$ (non-divergent runs). Here $\bar{V}_1^{\pi^*}(\rho)$ is recorded as Agent 1's robust best-response value and $\bar{V}_2^{\pi^*}(\rho)$ as the robust value of the current joint policy; during these transient, non-convergent iterates the two are evaluated at different policies, so the constant-sum identity $\bar{V}_1^{\pi^*}(\rho) + \bar{V}_2^{\pi^*}(\rho) = 1/(1-\gamma) = 10$, which holds only for a single joint policy under one shared kernel, need not hold.

The table reveals that the algorithm does not move steadily toward a single fixed point. Instead, under the heavily relaxed constraint $c_2 = 1.7$, the iterates repeatedly cycle among several distinct value regions. This cyclic behavior arises because, with such a loose constraint, the Lagrangian penalty on cost is too weak relative to the reward scale: the dual variable λ does not sufficiently suppress mildly infeasible policies in early iterations, allowing the primal updates to drift toward high-reward yet high-cost regions. Consequently, the algorithm oscillates among these metastable states until the dual updates eventually become strong enough to counteract the reward-driven drift. This cycling is a transient effect of the over-weak penalty rather than a failure of the procedure, and we remove it next by rescaling the penalty.

A simple and effective remedy is to adjust the relative weighting between reward and cost in the surrogate objective. To strengthen the influence of the cost penalty, we introduce an additional hyperparameter $\kappa \geq 1$ that scales the dual variable, leading to the modified surrogate reward

$$\tilde{R}_i^t(\lambda_i, \kappa) = R_i^t - \kappa \cdot \lambda_i^\top C_i^t. \tag{57}$$

Choosing $\kappa > 1$ amplifies the effective penalty on cost, discouraging overly risky behavior and reducing the tendency of the iterates to fall into cyclic or divergent patterns. In the experiment below we set $\kappa = 5$, which substantially increases the strength of the Lagrangian penalty and stabilizes the learning dynamics under the challenging constraint level $c_2 = 1.7$.

Runtime (s)	T_λ	T_π	$\bar{V}_2^{\pi^*}(\rho)$	$\bar{\Lambda}_2^{\pi^*}(\rho)$
0.20	7	3	3.00712	0.490956
0.64	18	7	3.00712	0.490956
0.20	7	3	3.00712	0.490956
0.28	8	3	3.00712	0.490956
0.24	7	3	3.00712	0.490956

Table 5: Performance of Algorithm 1 with the modified surrogate reward (57) under constraint $c_2 = 1.7$ and $\kappa = 5$.

Using the same $c_2 = 1.7$ instance as in Table 4, Figure 6 plots the evolution of $\bar{\Lambda}_2^{\pi^*}(\rho)$ and $\bar{V}_2^{\pi^*}(\rho)$ under the modified surrogate reward with $\kappa = 5$. The scaled penalty immediately pushes the cost below the constraint threshold and yields a smooth convergence of the reward without the oscillations observed in the unscaled case. Importantly, κ acts purely as a stabilizer: it still converges to the equilibrium value $(\bar{V}_2^{\pi^*}(\rho), \bar{\Lambda}_2^{\pi^*}(\rho)) = (3.00712, 0.490956)$ that the non-cycling runs already reach without it (Table 5). The larger penalty thus only damps the rotational best-response dynamics on the loosely constrained instance, recovering the same fixed point rather than altering it.

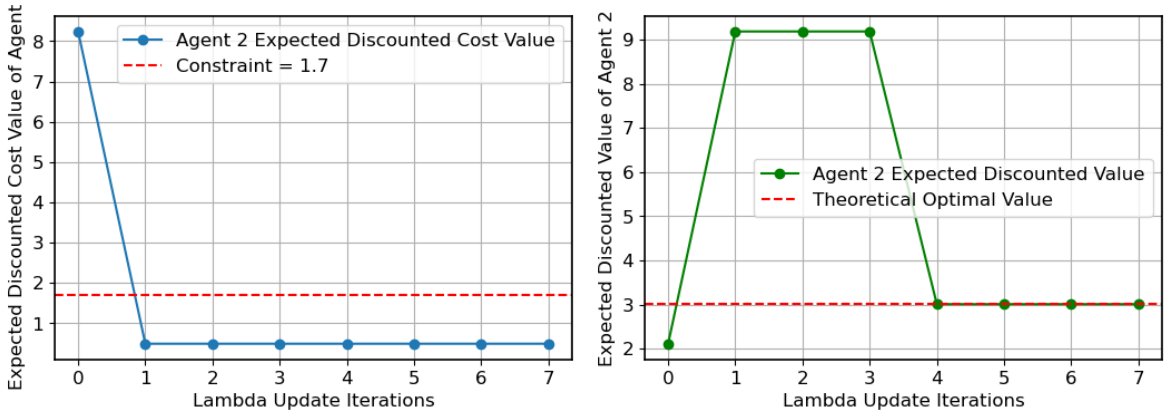

Figure 6: Evolution of Agent 2's value over iterations under the modified surrogate reward ($\kappa = 5$) and $c_2 = 1.7$.

Beyond the case $c_2 = 1.7$, we further evaluate the stabilized algorithm (with $\kappa = 5$) across a wide range of larger constraint levels, from $c_2 = 1.8$ up to $c_2 = 4.0$. The complete results are reported in Table 7. Across all these settings, the algorithm exhibits consistently stable behavior: the number of dual updates T_λ remains moderate, the policy-improvement iterations T_π do not grow uncontrollably, and no divergence or cyclic patterns are observed. These findings indicate that scaling the Lagrangian penalty with $\kappa > 1$ stabilizes the outer iteration across these instances, consistently recovering the same Upper-RFNE without divergence or cycling. By Corollary 5.8, the policy at which the rescaled iteration settles is an Upper-RFNE, so the penalty rescaling reliably drives the outer iteration to the correct equilibrium across all tested instances.

E.6 COMPLETE SWEEP TABLES

For completeness, we report the full per-seed convergence statistics underlying the constraint-threshold sweep (Appendix E.4) and the stabilization study (Appendix E.5). Table 6 covers the unmodified surrogate across $c_2 \in [0.6, 1.7]$, and Table 7 the modified algorithm ($\kappa = 5$) across $c_2 \in [1.7, 4.0]$; in each table the five rows per constraint level correspond to the five fixed random seeds.

c_2	Runtime (s)	T_λ	T_π
	0.19	5	3
	0.45	12	6
0.6	0.21	5	3
	0.22	6	3
	0.17	5	3
	0.16	5	3
	0.44	12	6
0.7	0.17	5	3
	0.52	14	6
	0.17	5	3
	0.16	5	3
	0.45	12	6
0.8	0.17	5	3
	0.45	12	6
	0.17	5	3

c_2	Runtime (s)	T_λ	T_π
	0.20	5	3
	0.49	13	6
0.9	0.17	5	3
	0.50	13	6
	0.18	5	3
	0.20	7	3
	0.34	13	5
1.0	0.21	7	3
	1.01	29	10
	0.20	7	3
	0.20	7	3
	0.27	8	3
1.1	0.22	7	3
	0.27	8	3
	0.20	7	3

c_2	Runtime (s)	T_λ	T_π
	0.24	7	3
	0.36	13	3
1.2	0.20	7	3
	0.63	19	3
	0.19	7	3
	1.00	30	11
	0.36	13	3
1.3	0.22	7	3
	0.34	13	3
	0.20	7	3
	0.60	19	7
	0.37	13	5
1.4	0.19	7	3
	0.34	13	5
	0.20	7	3

c_2	Runtime (s)	T_λ	T_π
	2.45	68	27
	1.31	38	15
1.5	0.22	9	3
	0.38	15	5
	0.19	7	3
	1.46	38	15
	1.42	39	15
1.6	0.21	7	3
	0.36	13	5
	0.19	7	3
	108.14*	≥ 2500	≥ 1000
	104.78*	≥ 2500	≥ 1000
1.7	0.19	7	3
	0.35	13	5
	0.23	7	3

Table 6: Convergence statistics of Algorithm 1 (unmodified surrogate) across constraint levels $c_2 \in [0.6, 1.7]$. Each constraint level lists the five fixed random seeds; value and cost columns are omitted since they remain constant. The T_λ and T_π columns report the *total* dual updates and outer policy-improvement iterations performed until convergence, as opposed to the per-run caps (T_λ=10000, T_π=1000) of Section 5. Entries marked "*" did not converge within these caps; they were halted at the outer cap T_π=1000, by which point at least 2500 dual updates had accumulated.

c_2	Runtime (s)	T_λ	T_π
	0.20	7	3
	0.64	18	7
1.7	0.20	7	3
	0.28	8	3
	0.24	7	3
	0.22	7	3
	0.61	17	7
1.8	0.20	7	3
	0.25	8	3
	0.20	7	3
	0.20	7	3
	0.54	15	7
1.9	0.20	7	3
	0.28	8	3
	0.21	7	3
	0.32	9	3
	0.29	9	4
2.0	0.20	7	3
	0.29	7	3
	0.20	7	3
	0.30	9	3
	0.37	12	5
2.1	0.23	7	3
	0.39	13	5
	0.24	7	3
	0.36	13	5
	0.34	12	5
2.2	0.21	7	3
	0.37	13	5
	0.21	7	3

c_2	Runtime (s)	T_λ	T_π
	0.36	13	5
	0.38	12	5
2.3	0.20	7	3
	0.39	13	5
	0.23	7	3
	0.39	12	5
	0.31	10	5
2.4	0.20	7	3
	0.32	10	5
	0.21	7	3
	0.34	12	5
	0.34	10	5
2.5	0.20	7	3
	0.32	10	5
	0.23	7	3
	0.34	12	5
	0.34	10	5
2.6	0.21	7	3
	0.32	10	5
	0.21	7	3
	0.35	12	5
	0.32	10	5
2.7	0.21	7	3
	0.32	10	5
	0.24	7	3
	0.37	12	5
	0.32	10	5
2.8	0.20	7	3
	0.32	10	5
	0.23	7	3

c_2	Runtime (s)	T_λ	T_π
	0.35	12	5
	0.34	10	5
2.9	0.20	7	3
	0.32	10	5
	0.21	7	3
	0.39	15	5
	0.35	12	5
3.0	0.23	9	3
	0.35	12	5
	0.23	9	3
	0.40	15	5
	0.34	12	5
3.1	0.36	11	3
	0.35	12	5
	0.24	9	3
	0.40	15	5
	0.35	12	5
3.2	0.33	11	3
	0.39	12	5
	0.24	9	3
	0.38	15	5
	0.35	12	5
3.3	0.27	9	3
	0.35	12	5
	0.24	9	3
	0.75	23	7
	0.70	20	7
3.4	0.76	24	7
	0.69	20	7
	0.23	7	3

c_2	Runtime (s)	T_λ	T_π
	0.74	22	7
	0.69	19	7
3.5	0.75	23	7
	0.73	19	7
	0.24	9	3
	0.81	23	7
	0.76	20	7
3.6	0.82	23	7
	0.75	20	7
	0.24	9	3
	0.82	23	7
	0.81	20	7
3.7	0.81	23	7
	0.76	20	7
	0.27	9	3
	0.80	22	7
	0.77	20	7
3.8	0.84	23	7
	0.82	20	7
	0.25	9	3
	0.43	13	5
	0.48	12	5
3.9	0.50	15	5
	0.45	12	5
	0.25	9	3
	0.45	13	5
	0.44	12	5
4.0	0.47	15	5
	0.44	12	5
	0.26	7	3

Table 7: Convergence statistics of the modified algorithm ($\kappa = 5$) across higher constraint levels $c_2 \in [1.7,\ 4.0]$. Each constraint level lists the five fixed random seeds.

F BASELINE COMPARISONS

To isolate the contribution of each ingredient of the RCMG framework, we compare our method against two natural ablations within the same three-cell environment, all evaluated under the worst-case transition kernel and with the same constraint

$c_2 = 0.5$ (fixed seed):

- **RMG (robust, unconstrained).** We remove Agent 2's constraint and solve the resulting unconstrained robust game, whose constant-sum structure ($R_1 + R_2 \equiv 1$) renders it zero-sum, for its minimax equilibrium value. The robust-optimal evader attains a high worst-case reward ($\bar{V}_2^{\pi^*}(\rho) \approx 6.26$) but ignores feasibility entirely, leaving its worst-case cost at $\bar{\Lambda}_2^{\pi^*}(\rho) \approx 3.46$, far above any meaningful threshold.
- **CMG (constrained, non-robust).** We disable the stochastic cells and train under the nominal deterministic kernel with the cost constraint enforced. The learned policy is feasible *under nominal dynamics* ($\bar{\Lambda}_2^{\pi^*}(\rho) = 0$), but when transferred to the worst-case kernel its cost rises to 1.92 (violating the threshold) and Agent 1's strategic value collapses from 9.47 to 3.51: a policy that appears safe and effective under nominal conditions offers no certified guarantee once transitions are adversarial.

Only the full RCMG simultaneously remains feasible ($\bar{\Lambda}_2^{\pi^*}(\rho) \approx 0.491 \leq 0.5$) and robust (Agent 1 retains a worst-case value of 6.99). Table 8 and Figure 7 summarize the comparison.

Method	**Robust?**	**Worst-case cost $\bar{\Lambda}_2$**	$\bar{V}_1$	$\bar{V}_2$
RMG (robust, unconstrained)	Yes	3.463 (infeasible)	3.738	6.262
CMG (constrained, non-robust), nominal	No	0.000	9.467	0.533
CMG (constrained, non-robust), worst-case	No	1.924 (infeasible)	3.509	6.491
RCMG (ours)	Yes	**0.491** (feasible)	**6.993**	**3.007**

Table 8: Baseline comparison under the worst-case transition kernel with constraint $c_2 = 0.5$. The RMG baseline ignores the constraint and is grossly infeasible; the non-robust CMG policy is feasible under nominal dynamics but violates the constraint and loses strategic value once transitions are adversarial. Only RCMG is simultaneously robust and feasible.

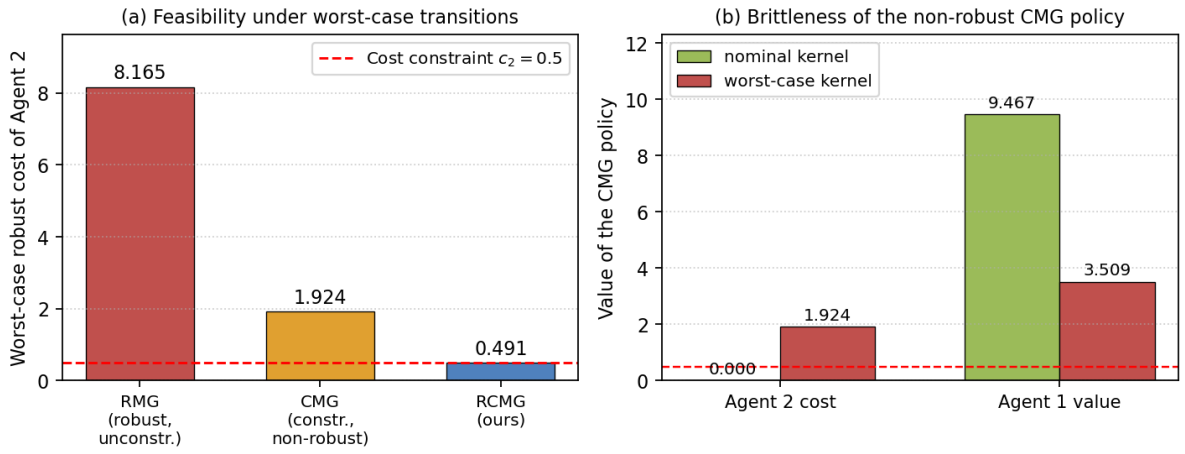

Figure 7: (a) Worst-case robust cost of Agent 2 for the three methods; only RCMG stays below the constraint. (b) The non-robust CMG policy is brittle: it is feasible and high-value under the nominal kernel but, under the worst-case kernel, its cost rises above the threshold and Agent 1's value collapses.

G THE NON-TIGHT REGIME: DIVERGENT EQUILIBRIA

Here we specify the coupled two-agent instance of Example 4.4 in full and derive its RFNE and Upper-RFNE in closed form, showing that the two solution concepts are *distinct equilibria* when the surrogate is not tight and coincide when it is. The non-tightness of Remark 4.3 is what drives the equilibria apart, now expressed through the agents' strategic interaction rather than at a single fixed policy.

The instance keeps the states $\{s_1, s_2\}$ with s_2 absorbing, the discount $\gamma = 0.9$, and the binary actions $a_i \in \{0, 1\}$ ($1 =$ advance), together with the uncertain transition of Example 4.4 that is triggered by the joint action,

$$P\big(s_2 \mid s_1, (1,1)\big) = p \in [0.6, 0.95], \qquad P\big(s_1 \mid s_1, (a_1, a_2)\big) = 1 \ \text{ for } (a_1, a_2) \neq (1,1),$$

so the chain advances toward absorption only when *both* agents advance, which is the source of the coupling. Rewards depend on the acting agent's own action, $R_i\big(s_1, (a_1, a_2), \cdot\big) = R_i^{\mathrm{hi}}$ if $a_i = 1$ and R_i^{lo} if $a_i = 0$, with $R_i(s_2, \cdot, \cdot) = 0$; we take $(R_1^{\mathrm{hi}}, R_1^{\mathrm{lo}}) = (3.85, 1)$ and $(R_2^{\mathrm{hi}}, R_2^{\mathrm{lo}}) = (2, 0)$. Agent 2 alone is constrained, paying $C_2\big(s_1, (\cdot, 1), \cdot\big) = 1$ whenever it advances and $C_2\big(s_1, (\cdot, 0), \cdot\big) = C_2(s_2, \cdot, \cdot) = 0$ otherwise, with threshold $c_2 = 1.5$; Agent 1 is unconstrained. Writing $x_i := \pi_i(a_i{=}1 \mid s_1)$ for each agent's advance probability, advancing is at once the only way to earn the higher reward, the sole source of Agent 2's cost, and, if matched by the opponent, the only route to absorption, which truncates the shared reward stream. The two agents therefore face a genuine tension between robustness and feasibility against each other.

Under the stationary profile (x_1, x_2) and kernel p the chain stays in s_1 with per-step probability $1 - x_1 x_2 p$, so the discounted occupancy is $d(p) = \big[1 - \gamma(1 - x_1 x_2 p)\big]^{-1}$ and

$$\bar{V}_i(p) = \big(R_i^{\mathrm{lo}} + (R_i^{\mathrm{hi}} - R_i^{\mathrm{lo}})\, x_i\big)\, d(p), \qquad \bar{\Lambda}_2(p) = x_2\, d(p).$$

Each value is worst (smallest) at $p = 0.95$ and the cost is worst (largest) at $p = 0.6$, so reward and cost again call for separate adversarial kernels.

We first locate Agent 1's best response. Under its reward worst case $p = 0.95$, its robust value is

$$\bar{V}_1 = \frac{R_1^{\mathrm{lo}} + (R_1^{\mathrm{hi}} - R_1^{\mathrm{lo}})\, x_1}{1 - \gamma\big(1 - 0.95\, x_1 x_2\big)} = \frac{1 + 2.85\, x_1}{0.1 + 0.855\, x_1 x_2},$$

whose derivative with respect to x_1 has the sign of $0.285 - 0.855\, x_2$. This vanishes at

$$x_2^\star = \frac{(R_1^{\mathrm{hi}} - R_1^{\mathrm{lo}})(1 - \gamma)}{R_1^{\mathrm{lo}}\, \gamma\, (0.95)} = \frac{0.285}{0.855} = \frac{1}{3},$$

where the numerator and denominator share the factor $1 + 2.85\, x_1$ and $\bar{V}_1 \equiv 10$ for every x_1. Agent 1 is thus indifferent and willing to randomize, which fixes Agent 2's equilibrium probability at $x_2^\star = \frac{1}{3}$. Agent 2's robust value $2x_2\, d(0.95)$ is in turn strictly increasing in x_2, since its derivative carries the sign of $(R_2^{\mathrm{hi}} - R_2^{\mathrm{lo}})(1 - \gamma) = 0.2 > 0$, so Agent 2 advances until its constraint binds, and the equilibrium value of x_1 is the one making that binding constraint hold at $x_2 = \frac{1}{3}$.

The RFNE charges the cost at its true worst-case kernel $p = 0.6$. Substituting $x_2 = \frac{1}{3}$ and $\gamma(0.6) = 0.54$,

$$\bar{\Lambda}_2(0.6) = \frac{x_2}{1 - \gamma\big(1 - 0.6\, x_1 x_2\big)} = \frac{1/3}{0.1 + 9x_1/50},$$

and setting $\bar{\Lambda}_2(0.6) = c_2 = \frac{3}{2}$ gives $\frac{1}{3} = \frac{3}{20} + \frac{27}{100}\, x_1$, hence $\frac{11}{60} = \frac{27}{100}\, x_1$ and

$$x_1^\star = \frac{55}{81} \approx 0.679.$$

At this profile the reward occupancy is $d(0.95) = \big(0.1 + 57x_1/200\big)^{-1} = \frac{1080}{317}$, so the robust reward is $\bar{V}_2 = 2x_2\, d(0.95) = 2 \cdot \frac{1}{3} \cdot \frac{1080}{317} = \frac{720}{317} \approx 2.271$, and the worst-case cost equals the threshold 1.5 by construction.

The Upper-RFNE instead optimizes the surrogate value (8), which for Agent 2 becomes

$$\bar{V}_2^{SL}(\pi, \lambda) = \min_P \mathbf{E}\Big[\textstyle\sum_t \gamma^t \big(R_2 - \lambda C_2\big)\Big] + \lambda c_2 = \min_p\, (2 - \lambda)\, x_2\, d(p) + \lambda c_2,$$

since the per-step surrogate return earned in s_1 is $2x_2 - \lambda x_2 = (2 - \lambda)x_2$. For $\lambda < R_2^{\mathrm{hi}} = 2$ the coefficient $(2 - \lambda)x_2$ is positive, so the single worst-case kernel is again $p = 0.95$ and

$$\bar{V}_2^{SL}(\lambda) = (2 - \lambda)\, x_2\, d(0.95) + \lambda c_2 = 2x_2\, d(0.95) + \lambda\big(c_2 - x_2\, d(0.95)\big).$$

The outer dual minimizes this over $\lambda \geq 0$, raising λ until the surrogate cost $\bar{\Lambda}_2(0.95) = x_2\, d(0.95)$ reaches the threshold; this pins Agent 1's probability through

$$\bar{\Lambda}_2(0.95) = \frac{1/3}{0.1 + 57x_1/200} = c_2 = \frac{3}{2} \implies \frac{11}{60} = \frac{171}{400}\, x_1 \implies x_1^\star = \frac{220}{513} \approx 0.429.$$

At this binding point $x_2\, d(0.95) = c_2$, so $d(0.95) = c_2/x_2 = \frac{3/2}{1/3} = \frac{9}{2}$, the λ-term vanishes, and the surrogate value is

$$\bar{V}_2^{SL} = 2x_2\, d(0.95) = 2c_2 = 3,$$

which here equals Agent 2's genuine robust reward because the reward itself is never relaxed. Evaluating the same profile under the true cost kernel $p = 0.6$, where $d(0.6) = \big(0.1 + 9x_1/50\big)^{-1} = \frac{570}{101}$, the genuine worst-case cost is

$$\bar{\Lambda}_2(0.6) = x_2\, d(0.6) = \frac{1}{3} \cdot \frac{570}{101} = \frac{190}{101} \approx 1.881,$$

so the Upper-RFNE exceeds the threshold by $\frac{190}{101} - \frac{3}{2} = \frac{77}{202} \approx 0.381$. Both agents genuinely randomize in both equilibria, and Agent 2 advances with the same probability $\frac{1}{3}$; what differs is Agent 1's equilibrium mixing (0.679 versus 0.429), together with the value and feasibility of Agent 2 (Figure 8). The Upper-RFNE reports a strictly higher value for Agent 2 ($\bar{V}_2^{SL} = 3$ versus the RFNE's $\bar{V}_2 \approx 2.271$), but attains it only by tolerating a constraint violation of $\frac{77}{202} \approx 0.381$ under the genuine worst-case kernel.

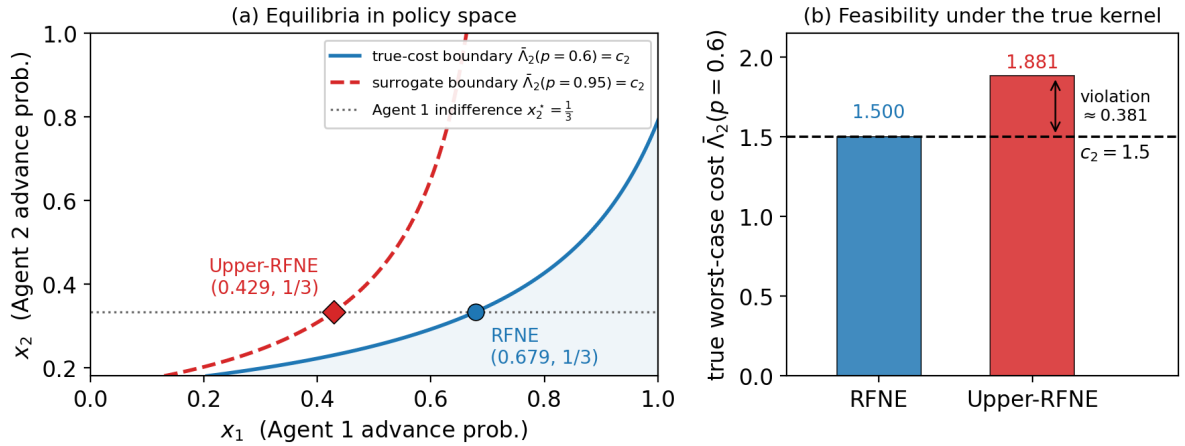

Figure 8: Divergent equilibria in the coupled non-tight game. (left) Policy space (x_1, x_2): the two agents meet on Agent 1's indifference line $x_2^\star = \frac{1}{3}$, where the equilibrium is selected by Agent 2's binding constraint: the RFNE (circle) sits on the true-cost boundary $\bar{\Lambda}_2(p{=}0.6){=}c_2$, while the Upper-RFNE (diamond) sits on the surrogate boundary $\bar{\Lambda}_2(p{=}0.95){=}c_2$. (right) True worst-case cost of each equilibrium: the RFNE is feasible at the threshold, whereas the Upper-RFNE, optimistic about cost, violates it by ≈ 0.381.

The mechanism is again the non-tightness of Remark 4.3, now expressed through the opponents' interaction. Because the surrogate assesses Agent 2's cost under the reward worst case $p = 0.95$, where rapid absorption keeps the horizon short and the cost small, it under-penalizes advancing, and the surrogate equilibrium consequently tolerates a smaller absorption pressure from Agent 1 ($x_1^\star = 0.429$). Under the genuine cost kernel $p = 0.6$, however, Agent 2 then lingers in s_1 and accrues a worst-case cost well above c_2. The exact RFNE anticipates $p = 0.6$ and therefore requires Agent 1 to advance more aggressively ($x_1^\star = 0.679$), shortening Agent 2's effective horizon just enough to hold its true worst-case cost at the threshold. Relocating Agent 2's cost to the absorbing state s_2 would align both worst cases at $p = 0.95$, collapse the surrogate onto the exact value for every profile, and merge the two equilibria, recovering exactly the tightness condition and the vanishing-uncertainty limit of Remark 4.3.