DISTRIBUTIONALLY ROBUST COOPERATIVE MULTI-AGENT REINFORCEMENT LEARNING VIA ROBUST VALUE FACTORIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Cooperative multi-agent reinforcement learning (MARL) commonly adopts centralized training with decentralized execution, where value-factorization methods enforce the individual-global-maximum (IGM) principle so that decentralized greedy actions recover the team-optimal joint action. However, the reliability of this recipe in real-world settings remains uncertain due to environmental uncertainties arising from the sim-to-real gap, model mismatch, system noise. We address this gap by introducing **D**istributionally **r**obust **IGM** (DrIGM), a principle that requires each agent's robust greedy action to align with the robust team-optimal joint action. We show that DrIGM holds for a novel definition of robust individual action values, which is compatible with decentralized greedy execution and yields a provable robustness guarantee for the whole system. Building on this foundation, we derive DrIGM-compliant robust variants of existing value-factorization architectures (e.g., VDN/QMIX/QTRAN) that (i) train on robust Q-targets, (ii) preserve scalability, and (iii) integrate seamlessly with existing codebases without bespoke per-agent reward shaping. Empirically, on high-fidelity SustainGym simulators and a StarCraft game environment, our methods consistently improve out-of-distribution performance.

1 INTRODUCTION

Multi-agent reinforcement learning (MARL) is a popular framework for studying how multiple agents compete or cooperate in complex environments such as video game playing (Vinyals et al., 2019), economic policy design (Zheng et al., 2022), wireless network communications (Qu et al., 2020), and power grid control (Gao et al., 2021), among others. In this work, we focus on the *cooperative* MARL setting, where each agent can only observe its local history, and agents must collaborate to achieve a joint goal. To address partial observability and reduce real-time communication costs, a widely used paradigm is the centralized training with decentralized execution (CTDE) (Oliehoek et al., 2008). During training, the agents may aggregate global information, coordinate credit assignment, and learn a team structure; at deployment, each agent must act myopically based on its own local history.

The CTDE paradigm is typically realized through value factorization methods (e.g., VDN (Sunehag et al., 2017), QMIX (Rashid et al., 2020), QTRAN (Son et al., 2019)). A key concept that underpins the success of these methods is the individual-global-maximum (IGM) principle (Son et al., 2019), which aligns each agent's greedy action with the team-optimal joint action via a suitable value factorization. However, most examples where the success of this principle is demonstrated are in virtual tasks (games (Vinyals et al., 2017) and grid worlds (Leibo et al., 2017)). It remains unclear whether this principle maintains its reliability in real-world domains, where modeling is imperfect and execution is noisy.

In practice, a major obstacle facing cooperative MARL is environmental uncertainty (Shi et al., 2024): team performance can drop sharply when the deployed environment deviates from the training environment due to model mismatch, system noise, and sim-to-real gap (Zhang et al., 2020b; Balaji et al., 2019). While environmental uncertainty presents challenges in single-agent RL settings, it is a more significant hurdle in cooperative MARL, where partial observability and inter-agent coupling can cause small mismatches to cascade into coordination failures (Capitan et al., 2012; He et al., 2022).

In single-agent RL, uncertainty in the environment is commonly addressed by distributionally robust RL (DR-RL) techniques (Wiesemann et al., 2013; Taori et al., 2020; Nilim & El Ghaoui, 2005; Panaganti & Kalathil, 2021a; Shi et al., 2023) which seek policies that perform well under adversarial perturbations of a nominal environment model. Single-agent DR-RL is well-explored, however extending DR-RL to the cooperative MARL setting is fundamentally more challenging. In particular, each agent acts on a local history yet shares a team reward coupled with teammates' actions, making it nontrivial to define individual robust Q-functions that both evaluate worst-case outcomes and remain compatible with IGM for decentralized greedy execution. Reward engineering can help empirically, but only a narrow class of shaping functions can provably preserve optimality (Foerster et al., 2016), even in the single-agent setting. Thus, *we seek a principled route to distributional robustness for cooperative MARL that remains compatible with decentralized greedy execution.*

Contributions. In this paper, we introduce a family of distributionally robust cooperative MARL algorithms for the CTDE setting. Our central technique is **D**istributionally **r**obust **IGM** (DrIGM), a robustness principle that requires each agent's robust greedy action to coincide with the robust team-optimal joint action, thereby preserving decentralized greedy execution.

We first show, via a concrete counterexample, that naïvely adopting individual robust action-value functions from single-agent DR-RL, where each agent considers its own worst case, does *not* guarantee decentralized alignment (IGM) in the cooperative multi-agent setting. We then provide sufficient conditions under which DrIGM holds: when individual robust action-value functions are defined with respect to the worst-case joint action-value function, DrIGM is guaranteed.

Next, we derive DrIGM-compliant robust variants of existing value factorization architectures (VDN, QMIX, and QTRAN), by training on robust Q-targets while retaining the CTDE information structure. The resulting methods are scalable, easy to implement on top of existing codebases, and maintain robustness at execution without requiring bespoke individual robust value design.

Finally, we evaluate our DrIGM-based algorithms on a realistic simulation an HVAC control task in SustainGym (Yeh et al., 2023), as well as on SMAC, a StarCraft II-based multi-agent game-playing environment (Samvelyan et al., 2019). Across out-of-distribution settings, our methods outperform non-robust value factorization baselines and a recent robust cooperative MARL baseline, consistently mitigating sim-to-real degradation on operational metrics.

Brief discussion of related work. Robustness in cooperative MARL has been studied along several axes: adversarial or heterogeneous teammates (Li et al., 2019; Kannan et al., 2023; Li et al., 2024), state/observation and communication perturbations (Guo et al., 2024; Yu et al., 2024), risk-sensitive (tail-aware) objectives under a fixed model (Shen et al., 2023), and explicit model uncertainty (Kwak et al., 2010; Zhang et al., 2020b; 2021; Liu et al., 2025). Most of the works on model uncertainty adopt a distributionally robust optimization viewpoint and targets Nash solutions with provable algorithms, often assuming full observability or individual rewards (Zhang et al., 2020a; Kardeş et al., 2011; Ma et al., 2023; Blanchet et al., 2023; Shi et al., 2024; Liu et al., 2025). In this work, we focus on the cooperative CTDE regime with partial observability and a single team reward, providing a systematic framework for robustness to model uncertainty without real-time communication. Due to space constraints, we provide an extended discussion of related works in Appendix A.

2 Background and Problem Formulation

Notation. For a set $\mathcal{X}$, $|\mathcal{X}|$ denotes its *cardinality* and $\Delta(\mathcal{X})$ the probability simplex over $\mathcal{X}$. We write $\prod_i \mathcal{X}_i$ for the Cartesian product. For $N \in \mathbb{N}$, we let $[N] := \{1, \dots, N\}$. Let $[x]_+ := \max\{0, x\}$.

Cooperative Dec-POMDPs. A cooperative multi-agent task with N agents is modeled as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP):

$$G = (\mathcal{S}, \{\mathcal{A}_i\}_{i=1}^N, P, r, \{\mathcal{O}_i\}_{i=1}^N, \{\sigma_i\}_{i=1}^N, \gamma),$$

with joint action space $\mathcal{A} := \prod_{i\in[N]} \mathcal{A}_i$. At time t, each agent i obtains an individual observation $o_i^t := \sigma_i(s^t)$ from its observation space $\mathcal{O}_i$, chooses an action $a_i^t \in \mathcal{A}_i$, a joint reward $r(s^t, \mathbf{a}^t) \in [0, 1]$ is received, where $\mathbf{a}^t := (a_1^t, \dots, a_N^t) \in \mathcal{A}$ is the joint action, and then the state evolves via

$s^{t+1} \sim P(\cdot \mid s^t, \mathbf{a}^t)$. Here we assume the joint observation $(o_1, \ldots, o_N)$ can recover the full state.[1] Each agent i acts using a history-based policy $\pi_i(\cdot \mid h_i^t)$ with $h_i^t := (o_i^0, a_i^0, \ldots, o_i^{t-1}, a_i^{t-1}, o_i^t)$; the joint policy is $\pi = \langle \pi_1, \ldots, \pi_N \rangle$. We use $\mathcal{H}_i^t$ to denote the space of possible histories for agent i up to time t. In the following sections, we will omit the superscript t to avoid notational clutter. The joint action-observation history is denoted $\mathbf{h} \in \mathcal{H} := \prod_{i \in [N]} \mathcal{H}_i$. Given the current state s, the joint history $\mathbf{h}$ and the joint action $\mathbf{a}$, we denote the joint action-value function under policy π by $Q_{\text{tot}}^{P,\pi}(\mathbf{h}, \mathbf{a})$, which can be reduced to $|\mathcal{S} \times \mathcal{A}|$ dimension as we assume the joint observation can recover the full state. We use $Q_{\text{tot}}^{P}(\mathbf{h}, \mathbf{a})$ to denote the optimal joint action value $\max_\pi Q_{\text{tot}}^{P,\pi}(\mathbf{h}, \mathbf{a})$.

CTDE. Centralized training with decentralized execution (CTDE) leverages global information during learning while executing individual policies from individual histories. A common CTDE mechanism is value factorization: learn an optimal joint action-value $Q_{\text{tot}}^{P}(\mathbf{h}, \mathbf{a})$ and individual action-value functions $[Q_i^P(h_i, a_i)]_{i \in [N]}$ that satisfy the following *individual–global–max* (IGM) principle (Son et al., 2019).

Definition 1 (IGM). *We say that individual action-value functions* $[Q_i^P : \mathcal{H}_i \times \mathcal{A}_i \to \mathbb{R}]_{i \in [N]}$ *satisfy the* individual-global-max (IGM) principle *for an optimal joint action-value function* $Q_{\text{tot}}^P : \mathcal{H} \times \mathcal{A} \to \mathbb{R}$ *under joint history* $\mathbf{h} = (h_1, \ldots, h_N) \in \mathcal{H}$ *if*

$$\left(\arg\max_{a_1} Q_1^P(h_1, a_1), \ldots, \arg\max_{a_N} Q_N^P(h_N, a_N) \right) \subseteq \arg\max_{\mathbf{a}} Q_{\text{tot}}^P(\mathbf{h}, \mathbf{a}).$$

IGM ensures that greedy individual actions are jointly optimal w.r.t. Q_{tot}^P, enabling decentralized execution without test-time communication.

Robust Dec-POMDPs. To capture model error and deployment shift, we consider an *uncertainty set* $\mathcal{P}$ of environment models around a nominal P^0. In Dec-POMDPs, we work with a history-based view: let $P_{\mathbf{h},\mathbf{a}}(\cdot)$ denote the transition kernel over next joint histories $\mathbf{h}'$, given the current joint history $\mathbf{h}$ and the joint action $\mathbf{a}$. We assume a *history-action rectangular* uncertainty set.

$$\mathcal{P} = \prod_{(\mathbf{h},\mathbf{a}) \in \mathcal{H} \times \mathcal{A}} \mathcal{P}_{\mathbf{h},\mathbf{a}}, \quad \mathcal{P}_{\mathbf{h},\mathbf{a}} \subseteq \Delta(\mathcal{H}), \tag{1}$$

e.g., balls around $P_{\mathbf{h},\mathbf{a}}^0$ under a probability metric with radius $\rho > 0$. This rectangularity assumption is widely adopted in DRRL literature (Blanchet et al., 2023; Shi et al., 2024; Ma et al., 2023), and ensures that a robust policy exists.

Given a function $Q : \mathcal{H} \times \mathcal{A} \to \mathbb{R}$, define the *robust Bellman operator* $\mathcal{T}$ as

$$(\mathcal{T}Q)(\mathbf{h}, \mathbf{a}) := r(s, \mathbf{a}) + \gamma \inf_{P_{\mathbf{h},\mathbf{a}} \in \mathcal{P}_{\mathbf{h},\mathbf{a}}} \mathbb{E}_{\mathbf{h}' \sim P_{\mathbf{h},\mathbf{a}}} \left[\max_{\mathbf{a}' \in \mathcal{A}} Q(\mathbf{h}', \mathbf{a}') \right]. \tag{2}$$

Under standard assumptions (bounded rewards, $\gamma \in (0, 1)$) and rectangularity in Eq. (1), $\mathcal{T}$ is a γ-contraction on the space of bounded Q, so it has a unique fixed point $Q_{\text{tot}}^{\mathcal{P}}$ (Iyengar, 2005) which satisfies

$$Q_{\text{tot}}^{\mathcal{P}}(\mathbf{h}, \mathbf{a}) = \inf_{P \in \mathcal{P}} Q_{\text{tot}}^{P}(\mathbf{h}, \mathbf{a}), \qquad \forall (\mathbf{h}, \mathbf{a}) \in \mathcal{H} \times \mathcal{A}. \tag{3}$$

We call $Q_{\text{tot}}^{\mathcal{P}}$ the *optimal robust joint action-value function* for the Dec-POMDP, and it admits a deterministic robust greedy joint policy $\pi^*(\mathbf{h}) \in \arg\max_{\mathbf{a}} Q_{\text{tot}}^{\mathcal{P}}(\mathbf{h}, \mathbf{a})$.

Robust Cooperative MARL. We study robust cooperative MARL under the CTDE setting. Given a model uncertainty set $\mathcal{P}$, our goal is to learn decentralized policies that maximize $Q_{\text{tot}}^{\mathcal{P}}$. That is, we aim to find $[\pi_i^\star : \mathcal{H}_i \mapsto \mathcal{A}_i]_{i \in [N]}$, such that

$$\langle \pi_1^\star, \ldots, \pi_N^\star \rangle \in \arg\max_{\mathbf{a}} Q_{\text{tot}}^{\mathcal{P}}(\cdot, \mathbf{a}).$$

Specifically, we seek a value factorization method that automatically generate robust individual action values, thereby enabling decentralized policy. This is non-trivial for two reasons. First, no individual reward signals are available, so robust individual action values are ill-defined a priori.

[1] Formally, we assume that $\sigma = (\sigma_1(\cdot), \ldots, \sigma_N(\cdot)) : \mathcal{S} \to \prod_{i \in [N]} \mathcal{O}_i$ is injective. Equivalently, there exists a (deterministic) decoding map $g : \prod_{i \in [N]} \mathcal{O}_i \to \mathcal{S}$ such that $g(\sigma(s)) = s$ for all $s \in \mathcal{S}$.

Second, directly defining robust individual action values from the single-agent DR-RL literature can break standard value factorization: robust individual action may not align with the robust joint action, as demonstrated in Example 1. These challenges motivate the central question of our work: *Can we construct robust individual utilities and a mixing scheme such that decentralized greedy actions recover the joint maximizers of $Q_{\text{tot}}^{\mathcal{P}}$, thereby enabling a robust CTDE framework?*

3 Distributionally Robust IGM (DrIGM)

To address the question above, we propose a novel principle for robust value factorization that builds upon the IGM principle while explicitly incorporating robustness.

3.1 Distributionally Robust IGM (DrIGM) Principle

Definition 2 (DrIGM)**.** *Given an uncertainty set $\mathcal{P}$, we say that **robust** individual action-value functions $[Q_i^{\text{rob}} : \mathcal{H}_i \times \mathcal{A}_i \to \mathbb{R}]_{i\in[N]}$ satisfy the* Distributionally robust IGM (DrIGM) principle *for the optimal **robust** joint action-value function $Q_{\text{tot}}^{\mathcal{P}} : \mathcal{H} \times \mathcal{A} \to \mathbb{R}$ under joint history $\mathbf{h} = (h_1, \ldots, h_N) \in \mathcal{H}$ if*

$$\left(\arg\max_{a_1} Q_1^{\text{rob}}(h_1, a_1), \ldots, \arg\max_{a_N} Q_N^{\text{rob}}(h_N, a_N)\right) \subseteq \arg\max_{\mathbf{a}} Q_{\text{tot}}^{\mathcal{P}}(\mathbf{h}, \mathbf{a}).$$

DrIGM extends classical IGM to the robust setting by requiring that the *robust* joint greedy action induced by $Q_{\text{tot}}^{\mathcal{P}}$ factorizes into robust individual greedy actions from $[Q_i^{\text{rob}}]_{i\in[N]}$. Note that when $\mathcal{P} = \{P\}$ is a singleton (i.e., there is no uncertainty), then DrIGM is equivalent to IGM.

Satisfying DrIGM is nontrivial. In particular, the single-agent definition $Q_i^{\text{rob}}(s, a) = \inf_{P\in\mathcal{P}} Q_i^P(s, a)$, commonly adopted in the DR-RL literature, does not ensure DrIGM when applied with a global uncertainty set. As shown in Example 1 in Appendix B, an adversarial model $P \in \mathcal{P}$ that minimizes one agent's value need not coincide with the adversarial model $P' \in \mathcal{P}$ that minimizes the joint value. As a result, robust individual greedy actions may fail to align with the robust joint greedy action. Similar inconsistencies arise even under agent-wise uncertainty sets defined in Shi et al. (2024) for essentially the same reason. This highlights the need for a new formulation of robust individual action values to support a consistent robust CTDE framework.

In robust cooperative MARL, the primary concern is the robustness of the *entire system*, as opposed to robustness of individual agents. Thus, it is sufficient to consider the worst case for the joint action value, rather than independently for each agent. Motivated by this idea, we show that the robust individual action value defined under a global worst-case model can guarantee DrIGM.

Theorem 1. *Given a global uncertainty set $\mathcal{P}$ defined in Eq. (1), suppose for all $P \in \mathcal{P}$, there exist $[Q_i^P]_{i\in[N]}$ satisfying IGM for Q_{tot}^P under joint history $\mathbf{h} = (h_1, \ldots, h_N) \in \mathcal{H}$. Let*

$$P^{\text{worst}}(\mathbf{h}, \mathbf{a}) \in \arg\inf_{P\in\mathcal{P}} Q_{\text{tot}}^P(\mathbf{h}, \mathbf{a}), \tag{4}$$

$$\bar{\mathbf{a}} \in \arg\max_{\mathbf{a}} Q_{\text{tot}}^{\mathcal{P}}(\mathbf{h}, \mathbf{a}), \tag{5}$$

denote the global worst-case model and the robust joint greedy action respectively. For each agent $i \in [N]$, define the robust individual action-value functions Q_i^{rob} as

$$Q_i^{\text{rob}}(h_i, a_i) := Q_i^{P^{\text{worst}}(\mathbf{h},\bar{\mathbf{a}})}(h_i, a_i). \tag{6}$$

Then, $[Q_i^{\text{rob}}]_{i\in[N]}$ satisfy DrIGM for $Q_{\text{tot}}^{\mathcal{P}}$ under joint history $\mathbf{h}$.

The proof of Theorem 1 can be found in Appendix C.1. Theorem 1 demonstrates that by anchoring individual robust action values to the global worst-case model evaluated at the robust joint greedy action, individual robust greedy actions become aligned with the robust joint greedy action. This construction resolves the misalignment problem that occurs when individual adversaries differ from the global adversary. More broadly, the result highlights that robust CTDE is achieved not by independently robustifying each agent, but by coordinating all agents against a shared adversarial model tied to the team's worst-case joint outcome. This perspective offers a principled foundation for designing robust value factorization methods that maintain decentralized execution while ensuring robustness guarantees.

Common factorization methods satisfy DrIGM. Having established that robust individual action values can be consistently defined via Theorem 1, we now examine whether these values are compatible with standard factorization methods used in cooperative MARL.

Theorem 2. *Given $\mathcal{P}$ defined in Eq. (1), for a joint history $\mathbf{h} \in \mathcal{H}$, suppose for all $P \in \mathcal{P}$, there exist individual action-value functions $[Q_i^P]_{i\in[N]}$ satisfying one of the following conditions for all $\mathbf{a} = (a_1, \ldots, a_N) \in \mathcal{A}$:*

$$Q_{\text{tot}}^P(\mathbf{h}, \mathbf{a}) = \sum_{i\in[N]} Q_i^P(h_i, a_i), \tag{VDN}$$

$$\frac{\partial Q_{\text{tot}}^P(\mathbf{h}, \mathbf{a})}{\partial Q_i^P(h_i, a_i)} \geq 0, \quad \forall i \in [N], \tag{QMIX}$$

$$\sum_{i\in[N]} Q_i^P(h_i, a_i) - Q_{\text{tot}}^P(\mathbf{h}, \mathbf{a}) + V_{\text{tot}}(\mathbf{h}) = \begin{cases} 0, & \mathbf{a} = \bar{\mathbf{a}}, \\ \geq 0, & \mathbf{a} \neq \bar{\mathbf{a}}, \end{cases} \tag{QTRAN}$$

where $\bar{\mathbf{a}} := [\bar{a}_i]_{i\in[N]}$ with $\bar{a}_i := \arg\max_{a_i} Q_i^P(h_i, a_i)$ and $V_{\text{tot}}(\mathbf{h}) := \max_{\mathbf{a}} Q_{\text{tot}}^P(\mathbf{h}, \mathbf{a}) - \sum_{i\in[N]} Q_i^P(h_i, a_i)$. Then $[Q_i^{\text{rob}}]_{i\in[N]}$ as defined in Eq. (6) satisfy DrIGM for $Q_{\text{tot}}^{\mathcal{P}}$ under joint history $\mathbf{h}$.

The proof of Theorem 2 can be found in Appendix C.2. Theorem 2 shows that when the underlying individual Q-functions satisfy the structural conditions of VDN (Sunehag et al., 2017), QMIX (Rashid et al., 2020), or QTRAN (Son et al., 2019), the robust individual action values $[Q_i^{\text{rob}}]_{i\in[N]}$ constructed from Equation (6) automatically satisfy DrIGM. This result ensures that robust CTDE can be realized directly within widely used value factorization frameworks, enabling principled distributionally robust extensions of existing algorithms. Moreover, as long as the test environment lies within the prescribed uncertainty set, this approach yields a provable robustness guarantee, as formalized in the next theorem.

Theorem 3. *Given $\mathcal{P}$ defined in Eq. (1), suppose the robust individual action values Q_i^{rob} satisfy Definition 2. If the test environment model P_{test} is included in the uncertainty set (i.e., $P_{\text{test}} \in \mathcal{P}$), then the robust joint action values provably lower bound the real joint action values in P_{test}:*

$$Q_{\text{tot}}^{\mathcal{P}}(\mathbf{h}, \mathbf{a}) \leq Q_{\text{tot}}^{P_{\text{test}}}(\mathbf{h}, \mathbf{a}), \ \forall \mathbf{h} \in \mathcal{H}, \ \mathbf{a} \in \mathcal{A}.$$

The proof of Theorem 3 can be found in Appendix C.3.

3.2 Robust Bellman Operators under Specific Uncertainty Sets

To design training loss functions, we next present the DrIGM-based robust Bellman operators for two common uncertainty designs: ρ-contamination and total variation (TV), which are well-studied in the single-agent distributionally robust RL literature (Yang et al., 2022; Panaganti & Kalathil, 2021b; Xu et al., 2023; Dong et al., 2022; Liu & Xu, 2024; Panaganti et al., 2022; Wang & Zou, 2022; Zhang et al., 2024). Both types of uncertainty sets consider perturbations of size $\rho \in (0, 1]$ around a nominal model P^0.

The ρ-contamination uncertainty set is defined as (for all $\mathbf{h} \in \mathcal{H}$ and $\mathbf{a} \in \mathcal{A}$)

$$\mathcal{P}_{\mathbf{h},\mathbf{a}} = \left\{ P \in \Delta(\mathcal{H}) \,\middle|\, P_{\mathbf{h},\mathbf{a}} = (1-\rho)P^0_{\mathbf{h},\mathbf{a}} + \rho H_{\mathbf{h},\mathbf{a}},\ H \in \Delta(\mathcal{H}) \text{ is arbitrary} \right\}, \tag{7}$$

with corresponding robust Bellman operator

$$\begin{aligned}(\mathcal{T}Q_{\text{tot}}^{\mathcal{P}})(\mathbf{h}, \mathbf{a}) &\overset{(a)}{=} r(s, \mathbf{a}) + \gamma(1-\rho)\, \mathbb{E}_{\mathbf{h}'\sim P^0_{\mathbf{h},\mathbf{a}}} \left[\max_{\mathbf{a}'\in\mathcal{A}} Q_{\text{tot}}^{\mathcal{P}}(h'_1, \ldots, h'_N, \mathbf{a}') \right] \\ &\overset{(b)}{=} r(s, \mathbf{a}) + \gamma(1-\rho)\, \mathbb{E}_{\mathbf{h}'\sim P^0_{\mathbf{h},\mathbf{a}}} \left[Q_{\text{tot}}^{\mathcal{P}}(h'_1, \ldots, h'_N, \bar{a}'_1, \ldots, \bar{a}'_N) \right], \end{aligned} \tag{8}$$

where $\bar{a}'_i = \arg\max_{a'_i} Q_i^{\text{rob}}(h'_i, a'_i)$. Here, (a) follows from robust Bellman operator as in the single-agent setting due to the $\mathbf{h} \times \mathbf{a}$-rectangularity from Eq. (1). (b) follows from the DrIGM principle where robust individual greedy actions are aligned with the robust joint greedy action.

Similarly, the TV-uncertainty set is defined as (for all $\mathbf{h} \in \mathcal{H}$ and $\mathbf{a} \in \mathcal{A}$)

$$\mathcal{P}_{\mathbf{h},\mathbf{a}} = \left\{ P \in \Delta(\mathcal{H}) \,\middle|\, \text{TV}(P, P^0_{\mathbf{h},\mathbf{a}}) \leq \rho \right\}, \tag{9}$$

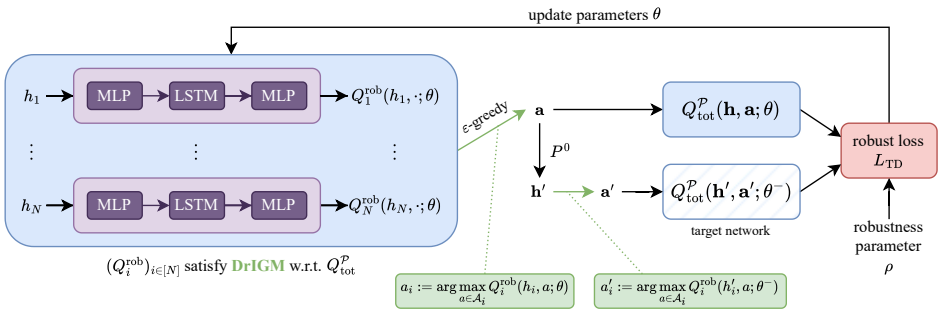


Figure 1: Overview of our robust value factorization algorithms. Because the robust individual action-value functions satisfy DrIGM, greedy actions can be computed efficiently in a decentralized manner while the function parameters are trained with a robust TD loss based on global reward.

with corresponding robust Bellman operator,

$$(\mathcal{T}Q_{\text{tot}}^{\mathcal{P}})(\mathbf{h}, \mathbf{a}) = r(s, \mathbf{a}) - \inf_{\eta \in [0, \frac{2}{\rho(1-\gamma)}]} \gamma\, \mathbb{E}_{\mathbf{h}' \sim P^0_{\mathbf{h},\mathbf{a}}} \Bigg(-(1-\rho)\eta(s, \mathbf{a}) + \Big[\eta(s, \mathbf{a}) - Q_{\text{tot}}^{\mathcal{P}}(h_1', \ldots, h_N', \bar{a}_1', \ldots, \bar{a}_N')\Big]_+ \Bigg), \tag{10}$$

where $\eta(s, a)$ is the dual variable.

Additional details of designing the robust Bellman operators can be found in Appendix D. In the next section, we show how DrIGM leads to practical robust value factorization algorithms.

4 Algorithms: Robust Value Factorization

Overall framework. Guided by DrIGM, we develop six robust value factorization algorithms by combining two types of uncertainty sets (ρ-contamination, TV-uncertainty) with three different value factorization architectures (VDN, QMIX, QTRAN). The overall framework is illustrated in Algorithm 1 and Fig. 1, while detailed pseudocode for each variant can be found in Appendix E. Concretely, we collect trajectories using ε-greedy exploration and train the robust individual action-value network using TD-learning (Sutton & Barto, 2018). For stability, robust one-step targets are evaluated using *target* networks, which are updated periodically.

Robust individual action-value networks. Each agent i uses a DRQN (Deep Recurrent Q Network)-style network that maps its local history h_i (observations and past actions) to action-values $Q_i(h_i, a_i)$, following an *MLP* encoder $\rightarrow$ *LSTM* core $\rightarrow$ *MLP* output architecture. For our training procedure, we follow the approach from Hausknecht & Stone (2015). We sample mini-batches of *sub-trajectories* from the replay buffer $\mathcal{D}$ and use *bootstrapped random updates*. We use 8 burn-in steps to warm-start the LSTM state and only take the last step output to calculate the loss and update the networks. This procedure is computationally and memory efficient while achieving performance comparable to sequential updates from the start of each episode.

Factorization networks. We instantiate three networks for robust value factorization:

1. **VDN** factorizes the robust joint action-value as the sum of robust per-agent values,

$$Q_{\text{tot}}^{\mathcal{P},\text{VDN}}(\mathbf{h}, \mathbf{a}) = \sum_{i=1}^{N} Q_i^{\text{rob}}(h_i, a_i).$$

2. Beyond direct summation, **QMIX** uses a *monotone* mixing network

$$Q_{\text{tot}}^{\mathcal{P},\text{QMIX}}(\mathbf{h}, \mathbf{a}) = f_\theta\big(Q_1^{\text{rob}}(h_1, a_1), \ldots, Q_N^{\text{rob}}(h_N, a_N),\, s\big), \tag{11}$$

where s is the global state. A lightweight *hypernetwork* takes s as input and outputs the layer weights of f_θ; to ensure $\partial Q_{\text{tot}}^{\mathcal{P},\text{QMIX}} / \partial Q_i^{\text{rob}} \geq 0$ (the QMIX monotonicity constraint), we enforce elementwise nonnegativity on these weights via an absolute-value (or softplus) transform. Biases remain unconstrained.

Algorithm 1 Robust value factorization

Input robustness parameter ρ, target network update frequency f, and ε
Initialize replay buffer $\mathcal{D}$
Initialize **robust individual action-value networks** $[Q_i^{\text{rob}}]_{i\in[N]}$ with random parameters θ
Initialize **factorization networks** that produce $Q_{\text{tot}}^{\mathcal{P}}$ with random parameters θ
Initialize target parameters $\theta^- = \theta$
for episode $h = 1, \ldots, H$ **do**
  Observe initial state s^0 and observation $o_i^0 = \sigma_i(s^0)$ for each agent i.
  for $t = 1, \ldots, T$ **do**
    Each agent i choose its action a_i^t using ε-greedy policy.
    Take joint action $\mathbf{a}^t$, observe the next state s^{t+1}, reward r^t and observation $o_i^{t+1} = \sigma_i(s^{t+1})$ for each agent i
    Store transition $(\mathbf{h}^t, \mathbf{a}^t, r^t, \mathbf{h}^{t+1})$ in replay buffer $\mathcal{D}$
    Sample a mini-batch of transitions $(\mathbf{h}, \mathbf{a}, r, \mathbf{h}')$ from $\mathcal{D}$
    Calculate **TD loss** L_{TD} using Eq. (14)
    // joint robust action $\bar{\mathbf{a}}$ is obtained from individual robust action $\{\bar{a}_i\}_{i\in[N]}$ by DrIGM
    Update θ by minimizing L_{TD}
    Update $\theta^- = \theta$ with frequency f
  end for
end for

3. **QTRAN** learns a separate joint action-value function $Q_{\text{tot}}^{\mathcal{P},\text{QTRAN}}(\mathbf{h}, \mathbf{a})$ and a baseline $V_{\text{tot}}(\mathbf{h})$. For efficiency and scalability, the joint network shares the encoder/head with the individual DRQN modules. In addition to the robust TD loss, QTRAN imposes two *consistency* terms to align the factorized and joint values:

$$L_{\text{opt}} = \left(Q_{\text{tot}}^{\mathcal{P},\text{VDN}}(\mathbf{h}, \bar{\mathbf{a}}) - \widehat{Q}_{\text{tot}}^{\mathcal{P},\text{QTRAN}}(\mathbf{h}, \bar{\mathbf{a}}) + V_{\text{tot}}(\mathbf{h})\right)^2, \tag{12}$$

$$L_{\text{nopt}} = \left(\min\left[Q_{\text{tot}}^{\mathcal{P},\text{VDN}}(\mathbf{h}, \mathbf{a}) - \widehat{Q}_{\text{tot}}^{\mathcal{P},\text{QTRAN}}(\mathbf{h}, \mathbf{a}) + V_{\text{tot}}(\mathbf{h}),\, 0\right]\right)^2, \tag{13}$$

where the $\hat{Q}$ is the detached Q value for training stability. Intuitively, L_{opt} enforces equality at the (robust) greedy joint action, while L_{nopt} penalizes positive slack elsewhere, recovering the QTRAN constraints in our robust setting.

TD Loss. Given the robust Bellman operator $\mathcal{T}$ defined in Eq. (2) for a Dec-POMDP setting, the generic form of the TD-loss is

$$L_{\text{TD}} = \left(Q_{\text{tot}}^{\mathcal{P}}(\mathbf{h}, \mathbf{a}; \theta) - (\mathcal{T}Q_{\text{tot}}^{\mathcal{P}}(\cdot, \cdot; \theta^-))(\mathbf{h}, \mathbf{a})\right)^2, \tag{14}$$

where θ is the network parameters, and θ^- is the target network parameters for training stability. Specifically, for ρ-contamination uncertainty sets, by the robust Bellman operator in Eq. (8), we have

$$L_{\text{TD}} = \left(Q_{\text{tot}}^{\mathcal{P}}(\mathbf{h}, \mathbf{a}; \theta) - (r(s, \mathbf{a}) + \gamma(1-\rho)\,\mathbb{E}_{\mathbf{h}'\sim P^0_{\mathbf{h},\mathbf{a}}}\, Q_{\text{tot}}^{\mathcal{P}}(\mathbf{h}', \bar{\mathbf{a}}'; \theta^-))\right)^2, \tag{15}$$

For TV uncertainty sets, by the robust Bellman operator in Eq. (10), we have

$$L_{\text{TD}} = \Bigg(Q_{\text{tot}}^{\mathcal{P}}(\mathbf{h}, \mathbf{a}; \theta) - r(s, \mathbf{a}) + \gamma\,\mathbb{E}_{\mathbf{h}'\sim P^0_{\mathbf{h},\mathbf{a}}}\Big[-(1-\rho)\eta(s, \mathbf{a}) + [\eta(s, \mathbf{a}) - Q_{\text{tot}}^{\mathcal{P}}(\mathbf{h}', \bar{\mathbf{a}}'; \theta^-)]_+\Big]\Bigg)^2, \tag{16}$$

where $\eta : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is calculated by minimizing the following empirical loss:

$$L_{\text{dual}}(\eta, Q_{\text{tot}}^{\mathcal{P}}) = \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{h},\mathbf{a},\mathbf{h}')\in\mathcal{D}} \left(\left[\eta(s, \mathbf{a}) - \max_{\mathbf{a}'} Q_{\text{tot}}^{\mathcal{P}}(\mathbf{h}', \mathbf{a}')\right]_+ - (1-\rho)\,\eta(s, \mathbf{a})\right). \tag{17}$$

5 EXPERIMENTS

5.1 SUSTAINGYM

We evaluate our proposed robust value factorization methods on SustainGym (Yeh et al., 2023), a recent benchmark suite designed to simulate real-world control tasks under distribution shift. We focus on multi-agent environments for smart building HVAC control, which inherently involve stochastic dynamics, distribution shifts, partial observability, and inter-agent coupling. These environments are particularly well-suited to test robustness, as the environmental models can vary across different days, building locations (thus climate conditions). More details about the environment and the experiment setup are provided in Appendix F. Our code can be found in `https://github.com/iclr2026-anonymous/robust-coMARL`.

Evaluation protocol. To assess generalization under distribution shift (i.e., model uncertainty), we adopt the following protocol. In the **training phase**, each algorithm is trained on a single environment. For robust MARL baselines that require multiple environments, we follow their standard protocol and train them on a fixed set of environments. In the **evaluation phase**, trained policies are deployed on unseen configurations that differ from those used in training, simulating realistic deployment where distribution shifts inevitably arise. This design allows us to explicitly measure robustness to changes in environment dynamics rather than simple memorization of training conditions.

Baselines. We compare our robust value factorization methods against:

- Non-robust factorization methods: VDN, QMIX, and QTRAN trained without robustness considerations, representing the standard CTDE paradigm.
- Existing robust CTDE baseline: the multi-agent group distributionally robust algorithm from Liu et al. (2025), which we refer to as "GroupDR". While the original work used only the VDN architecture, we extend the algorithm to QMIX and QTRAN for completeness.

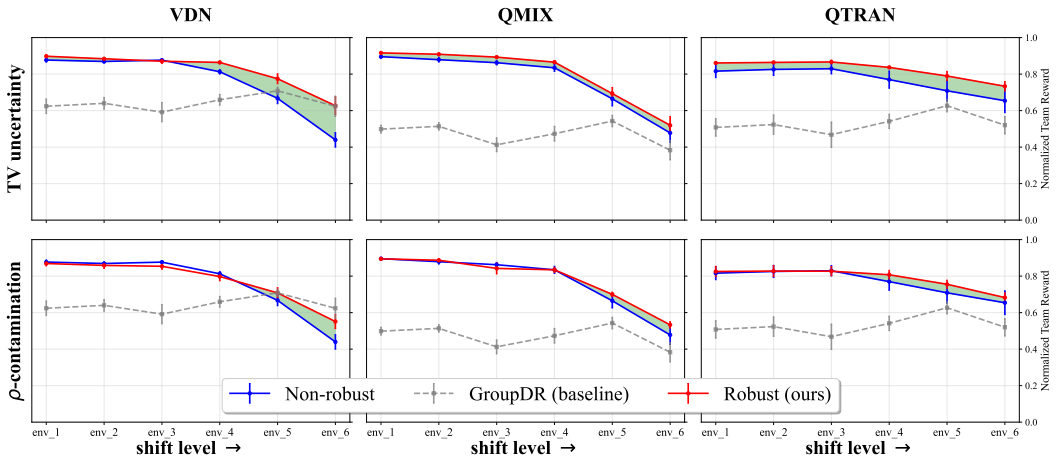


Figure 2: Normalized performance (averaged over 5 independent training runs, with error bars showing standard error) across different environment configurations for our robust MARL algorithms and other baselines. Each panel corresponds to one value factorization method. Robustness gain is the difference in reward (shaded area) between Robust (ours) and Non-robust, which shows the out-of-distribution performance improvement from the robust training.

Experiment 1: climatic shifts. We first test robustness under shifts induced by changes in climate conditions. Results (averaged over five seeds) are shown in Fig. 2. Our robust MARL algorithms consistently outperform both non-robust counterparts and the group DR baseline. Notably, performance degradation scales with the severity of the shift (e.g., `env_6` deviates most from the training environment, `env_1`), but our methods maintain relatively high returns. In contrast, the GroupDR baseline exhibits little sensitivity to shift severity, reflecting its reliance on worst-case rewards computed only from configurations encountered during training.

Experiment 2: seasonal shifts. We next evaluate robustness to seasonal shifts, training algorithms on `season_1` data and evaluating on `season_2`. Results are reported in Table 1, showing mean

and standard error of normalized episodic returns. The results show that robust value factorization algorithms with TV uncertainty set achieve consistent robustness gain against seasonal shifts.

Factorization Methods	VDN	QMIX	QTRAN
Non-robust	0.877 ± 0.012	0.895 ± 0.008	0.816 ± 0.036
baseline (GroupDR)	0.624 ± 0.040	0.499 ± 0.022	0.508 ± 0.048
Robust (TV-uncertainty)	$\mathbf{0.898 \pm 0.008}$	$\mathbf{0.916 \pm 0.006}$	$\mathbf{0.861 \pm 0.006}$
Robust (ρ-contamination)	0.869 ± 0.013	$\mathbf{0.911 \pm 0.005}$	$\mathbf{0.825 \pm 0.028}$

Table 1: Final Performances under seasonal shifts for our robust MARL algorithms and other baselines (mean $\pm$ standard error over 5 independent training runs). Values outperforming both the non-robust and group DR baselines are highlighted in bold.

Experiment 3: climatic and seasonal shifts. Finally, we test on the most extreme case, where we have distribution shifts arising from climatic and seasonal shifts. The results are presented in Table 2, with our robust MARL algorithms achieving 10-40% higher average reward than the non-robust baseline. Notably, QTRAN-based robust MARL algorithms demonstrate strong out-of-distribution performance and stability.

Factorization Methods	VDN	QMIX	QTRAN
Non-robust	0.440 ± 0.040	0.478 ± 0.052	0.654 ± 0.066
baseline (GroupDR)	0.624 ± 0.056	0.383 ± 0.053	0.520 ± 0.049
Robust (TV-uncertainty)	$\mathbf{0.627} \pm 0.049$	$\mathbf{0.520 \pm 0.048}$	$\mathbf{0.733 \pm 0.026}$
Robust (ρ-contamination)	0.551 ± 0.039	$\mathbf{0.500} \pm 0.075$	$\mathbf{0.682 \pm 0.026}$

Table 2: Final Performances under climatic and seasonal shifts for our robust MARL algorithms and other baselines (mean $\pm$ standard error over 5 independent training runs). Values outperforming both the non-robust and group DR baselines are highlighted in bold.

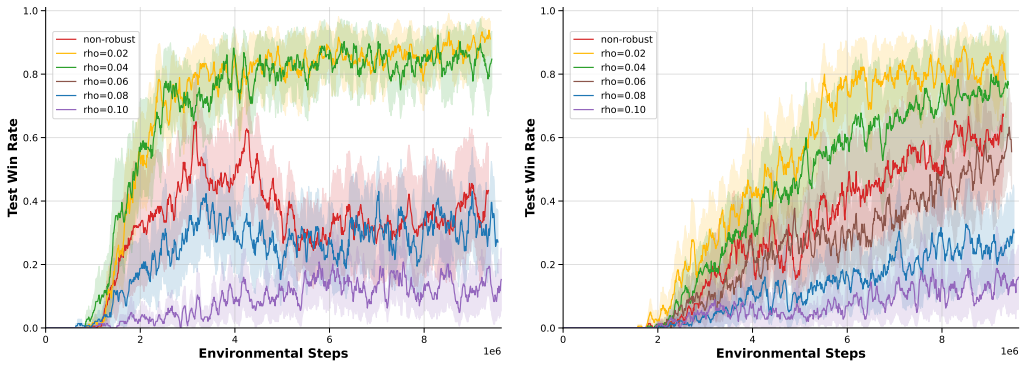
Choice of ρ. Theoretically, ρ should be chosen based on prior estimation of the model uncertainty level. Practically, we select ρ by training on `env_1` and validating on `env_2` and `env_3`, which yields stable performance without overfitting to a single shift.

Robustness in cooperative MARL. A noteworthy finding is that robustness in cooperative MARL does *not necessarily* entail reduced performance in the training environment. Unlike in single-agent robust RL, where conservatism often penalizes in-distribution returns, explicitly modeling robustness here mitigates errors from partial observability and decentralized execution. In several cases, robust training even improves in-distribution performance relative to non-robust baselines, suggesting that robustness can simultaneously enhance stability and adaptability in multi-agent systems.

5.2 StarCraft II

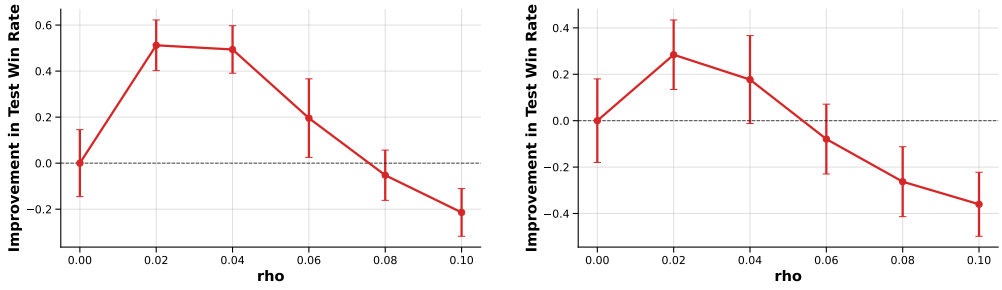
We additionally conduct experiments in SMAC (Samvelyan et al., 2019), a well-known benchmark consisting of two teams of agents engaged in cooperative combat scenarios based on StarCraft II. We focus on the hard `3s_vs_5z` map. In the test environment, we introduce distribution shift by adding noise to each agent's observation of every enemy unit's normalized position, sampled from $\mathcal{N}(0, 0.75^2)$. Results for the ρ-contamination uncertainty set with VDN and QMIX (averaged over five seeds) are shown in Fig. 3. The results demonstrate that for small values of ρ, our robust MARL algorithms significantly improve out-of-distribution performance.

Ablation study. We further compare the final performance of our algorithms against their non-robust baselines, reporting the improvement in the test win rate for different choices of ρ relative to the baseline. The results (averaged over five seeds) are shown in Fig. 4. Interestingly, the test win rate first increases as ρ grows, and then decreases. This observation aligns with our theory: when ρ is small relative to the shift level, explicitly modeling distribution shift during training yields improved out-of-distribution performance. However, when ρ becomes large, the robust MARL algorithms become overly conservative, leading to degraded performance.



(a) Robust VDN (b) Robust QMIX

Figure 3: Performance of our robust MARL algorithms and their non-robust baselines in SMAC (`3s_vs_5z` map). Each algorithm is evaluated every 10,000 environment steps, with each evaluation averaged over 32 episodes. Shaded regions denote the standard error across 5 random seeds. For small ρ, the robust algorithms significantly outperform their non-robust counterparts.



(a) Robust VDN (b) Robust QMIX

Figure 4: Improvement in final test win rate of our robust MARL algorithms over their non-robust baselines in SMAC (`3s_vs_5z` map) for different values of ρ. Error bars denote the standard error across 5 random seeds.

6 Conclusion

In this work, we introduce Distributionally robust IGM (DrIGM), a robustness principle for cooperative MARL that extends the classical IGM property to settings with environmental uncertainties. Whereas naïvely "robustifying" individual agent policies fails to align robust individual policies with the joint robust policy, the DrIGM offers a principled framework for constructing robust individual action values that remain aligned with the joint robust policy, thereby enabling decentralized greedy execution under uncertainty.

Building on this foundation, we derive DrIGM-based robust value factorization algorithms for VDN, QMIX, and QTRAN, trained via robust Bellman operators under standard uncertainty sets (ρ-contamination and total variation). Empirically, on a high-fidelity building HVAC control benchmark, our methods consistently mitigate out-of-distribution performance degradation arising from climatic and seasonal shifts. On a StarCraft II game-playing benchmark, our methods likewise improve out-of-distribution performance under added observation noise. Unlike single-agent robust RL, where conservatism often harms in-distribution returns, we find that robustness in cooperative MARL can simultaneously enhance stability and adaptability.

While we introduced the DrIGM framework for a global uncertainty set, we believe it may be possible to further extend this framework. Future work includes developing DrIGM-compliant algorithms under agent-wise uncertainty sets and exploring additional training paradigms (e.g., decentralized training) to further broaden applicability.

REPRODUCIBILITY STATEMENT

We release an anonymized repository containing all code, configuration files, and scripts needed to reproduce our results, including data generation and figure plotting. All proofs for the main paper are stated in Appendix C. Algorithm psuedocode is also provided in Appendix E.

REFERENCES

Bharathan Balaji, Sunil Mallya, Sahika Genc, Saurabh Gupta, Leo Dirac, Vineet Khare, Gourav Roy, Tao Sun, Yunzhe Tao, Brian Townsend, Eddie Calleja, Sunil Muralidhara, and Dhanasekar Karuppasamy. Deepracer: Educational autonomous racing platform for experimentation with sim2real reinforcement learning. *arXiv preprint arXiv:1911.01562*, 2019. URL https://arxiv.org/abs/1911.01562. 1

Dimitris Bertsimas, Vishal Gupta, and Nathan Kallus. Data-driven robust optimization. *Mathematical Programming*, 167(2):235–292, 2018. 17

Jose Blanchet and Karthyek Murthy. Quantifying distributional model risk via optimal transport. *Mathematics of Operations Research*, 44(2):565–600, 2019. 17

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. *Advances in Neural Information Processing Systems*, 36, 2023. 2, 3, 17

Albert Bou, Matteo Bettini, Sebastian Dittert, Vikash Kumar, Shagun Sodhani, Xiaomeng Yang, Gianni De Fabritiis, and Vincent Moens. Torchrl: A data-driven decision-making library for pytorch. *arXiv preprint arXiv:2306.00577*, 2023. URL https://arxiv.org/abs/2306.00577. 26

Alexander Bukharin, Yan Li, Yue Yu, Qingru Zhang, Zhehui Chen, Simiao Zuo, Chao Zhang, Songan Zhang, and Tuo Zhao. Robust multi-agent reinforcement learning via adversarial regularization: Theoretical foundation and stable algorithms. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=FmZVRe0gn8. 17

Jesus Capitan, Matthijs T.J. Spaan, Luis Merino, and Anibal Ollero. Decentralized multi-robot cooperation with auctioned pomdps. In *2012 IEEE International Conference on Robotics and Automation*, pp. 3323–3328, 2012. doi: 10.1109/ICRA.2012.6224917. 1

Esther Derman and Shie Mannor. Distributional robustness and regularization in reinforcement learning. *arXiv preprint arXiv:2003.02894*, 2020. 17

Esther Derman, Daniel J Mankowitz, Timothy A Mann, and Shie Mannor. Soft-robust actor-critic policy-gradient. In *AUAI press for Association for Uncertainty in Artificial Intelligence*, pp. 208–218, 2018. 17

Jing Dong, Jingwei Li, Baoxiang Wang, and Jingzhao Zhang. Online policy optimization for robust MDP. *arXiv preprint arXiv:2209.13841*, 2022. 5, 17

John Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *arXiv preprint arXiv:1810.08750*, 2018. 17

Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pp. 2145–2153, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819. 2

Rui Gao. Finite-sample guarantees for wasserstein distributionally robust optimization: Breaking the curse of dimensionality. *arXiv preprint arXiv:2009.04382*, 2020. 17

Yuanqi Gao, Wei Wang, and Nanpeng Yu. Consensus Multi-Agent Reinforcement Learning for Volt-VAR Control in Power Distribution Networks. *IEEE Transactions on Smart Grid*, 12(4): 3594–3604, July 2021. ISSN 1949-3061. doi: 10.1109/TSG.2021.3058996. URL https://ieeexplore.ieee.org/document/9353702. Conference Name: IEEE Transactions on Smart Grid. 1

Vineet Goyal and Julien Grand-Clement. Robust markov decision processes: Beyond rectangularity. *Mathematics of Operations Research*, 2022. 17

Weiran Guo, Guanjun Liu, Ziyuan Zhou, Ling Wang, and Jiacun Wang. Enhancing the robustness of qmix against state-adversarial attacks. *Neurocomputing*, 572:127191, March 2024. ISSN 0925-2312. doi: 10.1016/j.neucom.2023.127191. URL http://dx.doi.org/10.1016/j.neucom.2023.127191. 2, 17

Songyang Han, Sanbao Su, Sihong He, Shuo Han, Haizhao Yang, and Fei Miao. What is the solution for state-adversarial multi-agent reinforcement learning? *arXiv preprint arXiv:2212.02705*, 2022. 17

Matthew J. Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. *CoRR*, abs/1507.06527, 2015. URL http://arxiv.org/abs/1507.06527. 6, 26

Keyang He, Prashant Doshi, and Bikramjit Banerjee. Reinforcement learning in many-agent settings under partial observability. In James Cussens and Kun Zhang (eds.), *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pp. 780–789. PMLR, 01–05 Aug 2022. URL https://proceedings.mlr.press/v180/he22a.html. 1

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. 17

Chin Pang Ho, Marek Petrik, and Wolfram Wiesemann. Fast bellman updates for robust mdps. In *International Conference on Machine Learning*, pp. 1979–1988. PMLR, 2018. 17

Chin Pang Ho, Marek Petrik, and Wolfram Wiesemann. Partial policy iteration for l1-robust markov decision processes. *Journal of Machine Learning Research*, 22(275):1–46, 2021. 17

Shariq Iqbal, Christian A. Schröder de Witt, Bei Peng, Wendelin Boehmer, Shimon Whiteson, and Fei Sha. Randomized entity-wise factorization for multi-agent reinforcement learning. In *ICML*, 2021. URL http://proceedings.mlr.press/v139/iqbal21a.html. 17

Garud N Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2): 257–280, 2005. 3, 17

Shyam Sundar Kannan, Vishnunandan LN Venkatesh, and Byung-Cheol Min. Smart-llm: Smart multi-agent robot task planning using large language models. *arXiv preprint arXiv:2309.10062*, 2023. 2, 17

Erim Kardeş, Fernando Ordóñez, and Randolph W Hall. Discounted robust stochastic games and an application to queueing control. *Operations research*, 59(2):365–382, 2011. 2, 17

David L Kaufman and Andrew J Schaefer. Robust modified policy iteration. *INFORMS Journal on Computing*, 25(3):396–410, 2013. 17

Jun-Young Kwak, Rong Yang, Zhengyu Yin, Matthew E. Taylor, and Milind Tambe. Teamwork and coordination under model uncertainty in dec-pomdps. In *Proceedings of the 3rd AAAI Conference on Interactive Decision Theory and Game Theory*, AAAIWS'10-03, pp. 30–36. AAAI Press, 2010. 2, 17

Joel Z. Leibo, Vinicius Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. Multi-agent reinforcement learning in sequential social dilemmas. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '17, pp. 464–473, Richland, SC, 2017. International Foundation for Autonomous Agents and Multiagent Systems. 1

Shihui Li, Yi Wu, Xinyue Cui, Honghua Dong, Fei Fang, and Stuart Russell. Robust multi-agent reinforcement learning via minimax deep deterministic policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 4213–4220, 2019. 2, 17

Simin Li, Jun Guo, Jingqiao Xiu, Ruixiao Xu, Xin Yu, Jiakai Wang, Aishan Liu, Yaodong Yang, and Xianglong Liu. Byzantine robust cooperative multi-agent reinforcement learning as a bayesian game. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=z6KS9D1dxt. 2, 17

Guangyi Liu, Suzan Iloglu, Michael Caldara, Joseph W Durham, and Michael M. Zavlanos. Distributionally robust multi-agent reinforcement learning for dynamic chute mapping. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=DqlN1yFyxN. 2, 8, 17, 26

Zhishuai Liu and Pan Xu. Distributionally robust off-dynamics reinforcement learning: Provable efficiency with linear function approximation. *arXiv preprint arXiv:2402.15399*, 2024. 5, 17

Miao Lu, Han Zhong, Tong Zhang, and Jose Blanchet. Distributionally robust reinforcement learning with interactive data collection: Fundamental hardness and near-optimal algorithms. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=aYWtfsf3uP. 21

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. 2, 3, 17

Daniel J Mankowitz, Nir Levine, Rae Jeong, Yuanyuan Shi, Jackie Kay, Abbas Abdolmaleki, Jost Tobias Springenberg, Timothy Mann, Todd Hester, and Martin Riedmiller. Robust reinforcement learning for continuous control with model misspecification. *arXiv preprint arXiv:1906.07516*, 2019. 17

Arnab Nilim and Laurent El Ghaoui. Robust control of Markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005. 2

Frans A. Oliehoek, Matthijs T. J. Spaan, and Nikos Vlassis. Optimal and approximate q-value functions for decentralized pomdps. *J. Artif. Int. Res.*, 32(1):289–353, May 2008. ISSN 1076-9757. 1

Kishan Panaganti and Dileep Kalathil. Robust reinforcement learning using least squares policy iteration with provable performance guarantees. In *International Conference on Machine Learning (ICML)*, pp. 511–520, 2021a. 2, 17

Kishan Panaganti and Dileep Kalathil. Sample complexity of model-based robust reinforcement learning. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 2240–2245, 2021b. 5, 17

Kishan Panaganti, Zaiyan Xu, Dileep Kalathil, and Mohammad Ghavamzadeh. Robust reinforcement learning using offline data. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 5, 17, 21, 22, 26

Guannan Qu, Yiheng Lin, Adam Wierman, and Na Li. Scalable Multi-Agent Reinforcement Learning for Networked Systems with Average Reward. In *Advances in Neural Information Processing Systems*, volume 33, pp. 2074–2086. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/168efc366c449fab9c2843e9b54e2a18-Abstract.html. 1

Hamed Rahimian and Sanjay Mehrotra. Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659*, 2019. 17

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178):1–51, 2020. 1, 5, 17, 26

Aurko Roy, Huan Xu, and Sebastian Pokutta. Reinforcement learning under model mismatch. In *Advances in Neural Information Processing Systems*, pp. 3043–3052, 2017. 17

Mikayel Samvelyan, Tabish Rashid, Christian Schröder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob N. Foerster, and Shimon Whiteson. The starcraft multi-agent challenge. *CoRR*, abs/1902.04043, 2019. URL http://arxiv.org/abs/1902.04043. 2, 9

Siqi Shen, Mengwei Qiu, Jun Liu, Weiquan Liu, Yongquan Fu, Xinwang Liu, and Cheng Wang. Resq: A residual q function-based approach for multi-agent reinforcement learning value factorization. In *NeurIPS*, 2022. 17

Siqi Shen, Chennan Ma, Chao Li, Weiquan Liu, Yongquan Fu, Songzhu Mei, Xinwang Liu, and Cheng Wang. Riskq: risk-sensitive multi-agent reinforcement learning value factorization. *Advances in Neural Information Processing Systems*, 36:34791–34825, 2023. 2, 17

Laixi Shi, Gen Li, Yuting Wei, Yuxin Chen, Matthieu Geist, and Yuejie Chi. The curious price of distributional robustness in reinforcement learning with a generative model. *Advances in Neural Information Processing Systems*, 36, 2023. 2

Laixi Shi, Eric Mazumdar, Yuejie Chi, and Adam Wierman. Sample-efficient robust multi-agent reinforcement learning in the face of environmental uncertainty. *arXiv preprint arXiv:2404.18909*, 2024. 1, 2, 3, 4, 17

Elena Smirnova, Elvis Dohmatob, and Jérémie Mary. Distributionally robust reinforcement learning. *arXiv preprint arXiv:1902.08708*, 2019. 17

Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International conference on machine learning*, pp. 5887–5896. PMLR, 2019. 1, 3, 5, 17, 26, 27

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017. 1, 5, 17, 26

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018. 6

Aviv Tamar, Shie Mannor, and Huan Xu. Scaling up robust mdps using function approximation. In *International Conference on Machine Learning*, pp. 181–189, 2014. 17

Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. *Advances in Neural Information Processing Systems*, 33:18583–18599, 2020. 2

Daniel Vial, Sanjay Shakkottai, and R Srikant. Robust multi-agent bandits over undirected graphs. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 6(3):1–57, 2022. 17

Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, John Quan, Stephen Gaffney, Stig Petersen, Karen Simonyan, Tom Schaul, Hado van Hasselt, David Silver, Timothy Lillicrap, Kevin Calderone, Paul Keet, Anthony Brunasso, David Lawrence, Anders Ekermo, Jacob Repp, and Rodney Tsing. Starcraft ii: A new challenge for reinforcement learning. *arXiv preprint arXiv:1708.04782*, 2017. URL https://arxiv.org/abs/1708.04782. 1

Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander S. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L. Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782): 350–354, November 2019. ISSN 1476-4687. doi: 10.1038/s41586-019-1724-z. URL https://www.nature.com/articles/s41586-019-1724-z. Publisher: Nature Publishing Group. 1

Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. Qplex: Duplex dueling multi-agent q-learning. In *ICLR*, 2021. 17

Yue Wang and Shaofeng Zou. Policy gradient method for robust reinforcement learning. In *International Conference on Machine Learning*, pp. 23484–23526, 2022. 5, 17

Wolfram Wiesemann, Daniel Kuhn, and Berç Rustem. Robust Markov decision processes. *Mathematics of Operations Research*, 38(1):153–183, 2013. 2

Eric M Wolff, Ufuk Topcu, and Richard M Murray. Robust control of uncertain markov decision processes with temporal logic specifications. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pp. 3372–3379. IEEE, 2012. 17

Huan Xu and Shie Mannor. Distributionally robust Markov decision processes. *Mathematics of Operations Research*, 37(2):288–300, 2012. 17

Zaiyan Xu, Kishan Panaganti, and Dileep Kalathil. Improved sample complexity bounds for distributionally robust reinforcement learning. *arXiv preprint arXiv:2303.02783*, 2023. 5, 17

Wenhao Yang, Liangyu Zhang, and Zhihua Zhang. Toward theoretical understandings of robust Markov decision processes: Sample complexity and asymptotics. *The Annals of Statistics*, 50(6): 3223–3248, 2022. 5, 17

Yaodong Yang, Jianye Hao, Ben Liao, Kun Shao, Guangyong Chen, Wulong Liu, and Hongyao Tang. Qatten: A general framework for cooperative multiagent reinforcement learning. *CoRR*, abs/2002.03939, 2020. URL https://arxiv.org/abs/2002.03939. 17

Christopher Yeh, Victor Li, Rajeev Datta, Julio Arroyo, Nicolas Christianson, Chi Zhang, Yize Chen, Mehdi Hosseini, Azarang Golmohammadi, Yuanyuan Shi, Yisong Yue, and Adam Wierman. SustainGym: Reinforcement learning environments for sustainable energy systems. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, New Orleans, LA, USA, 12 2023. URL https://openreview.net/forum?id=vZ9tA3o3hr. 2, 8, 22

Lebin Yu, Yunbo Qiu, Quanming Yao, Yuan Shen, Xudong Zhang, and Jian Wang. Robust communicative multi-agent reinforcement learning with active defense. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024. ISBN 978-1-57735-887-9. doi: 10.1609/aaai.v38i16.29708. URL https://doi.org/10.1609/aaai.v38i16.29708. 2, 17

Huan Zhang, Hongge Chen, Duane S Boning, and Cho-Jui Hsieh. Robust reinforcement learning on state observations with learned optimal adversary. In *International Conference on Learning Representations*, 2020a. 2, 17

Huan Zhang, Hongge Chen, Duane Boning, and Cho-Jui Hsieh. Robust reinforcement learning on state observations with learned optimal adversary. *arXiv preprint arXiv:2101.08452*, 2021. 2, 17

Kaiqing Zhang, TAO SUN, Yunzhe Tao, Sahika Genc, Sunil Mallya, and Tamer Basar. Robust multi-agent reinforcement learning with model uncertainty. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 10571–10583. Curran Associates, Inc., 2020b. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/774412967f19ea61d448977ad9749078-Paper.pdf. 1, 2, 17

Zhengfei Zhang, Kishan Panaganti, Laixi Shi, Yanan Sui, Adam Wierman, and Yisong Yue. Distributionally robust constrained reinforcement learning under strong duality. *arXiv preprint arXiv:2406.15788*, 2024. 5, 17, 26

Zhili Zhang, Yanchao Sun, Furong Huang, and Fei Miao. Safe and robust multi-agent reinforcement learning for connected autonomous vehicles under state perturbations. *arXiv preprint arXiv:2309.11057*, 2023. 17

Stephan Zheng, Alexander Trott, Sunil Srinivasa, David C. Parkes, and Richard Socher. The AI Economist: Taxation policy design via two-level deep multiagent reinforcement learning. *Science Advances*, 8(18):eabk2607, May 2022. doi: 10.1126/sciadv.abk2607. URL https://www.science.org/doi/10.1126/sciadv.abk2607. 1

Ziyuan Zhou and Guanjun Liu. Robustness testing for multi-agent reinforcement learning: State perturbations on critical agents. *arXiv preprint arXiv:2306.06136*, 2023. 17

A Related Work

Single-agent Distributionally Robust RL (DR-RL). The single-agent setting is typically formalized as a robust Markov decision process (MDP). A substantial literature studies finite-sample guarantees for distributionally robust RL, exploring a variety of ambiguity-set designs (Iyengar, 2005; Xu & Mannor, 2012; Wolff et al., 2012; Kaufman & Schaefer, 2013; Ho et al., 2018; Smirnova et al., 2019; Ho et al., 2021; Goyal & Grand-Clement, 2022; Derman & Mannor, 2020; Tamar et al., 2014; Panaganti & Kalathil, 2021a; Roy et al., 2017; Derman et al., 2018; Mankowitz et al., 2019). Most relevant to our work are tabular robust MDPs with (s, a)-rectangular uncertainty sets defined by total-variation balls (Yang et al., 2022; Panaganti & Kalathil, 2021b; Xu et al., 2023; Dong et al., 2022; Liu & Xu, 2024; Panaganti et al., 2022) or ρ-contamination models (Wang & Zou, 2022; Zhang et al., 2024), for which minimax dynamic programming and learning algorithms admit provable performance bounds.

Value factorization methods for cooperative MARL. Value factorization is the standard mechanism for scalable cooperative MARL under CTDE. Early work adopts simple additivity (VDN (Sunehag et al., 2017)), while QMIX (Rashid et al., 2020) learn a state-conditioned monotone combiner to enlarge the function class without violating the IGM requirement. QTRAN (Son et al., 2019) further relax the monotonicity assumption with consistency constraints. Other approaches include attention-based mixers (e.g., QAtten (Yang et al., 2020), REFIL (Iqbal et al., 2021)), dueling-style decompositions (QPlex (Wang et al., 2021)) and residual designs (ResQ (Shen et al., 2022)). Building on this body of work, we develop robust value-factorization algorithms with provable robustness guarantees under model uncertainty, enabling robust decentralized execution in partially observable cooperative settings.

Robustness in MARL. In general MARL, robustness is typically studied within Markov games, where uncertainty can be modeled in different components, such as the state space (Han et al., 2022; He et al., 2023; Zhou & Liu, 2023; Zhang et al., 2023), other agents (Li et al., 2019; Kannan et al., 2023), and environmental dynamics (Zhang et al., 2021; Liu et al., 2025). We refer readers to Vial et al. (2022) for an overview. This work considers robustness to model uncertainty, primarily studied via distributionally robust optimization (DRO) (Rahimian & Mehrotra, 2019; Gao, 2020; Bertsimas et al., 2018; Duchi & Namkoong, 2018; Blanchet & Murthy, 2019), where most prior efforts target Nash equilibria and provide provable (actor–critic / Q-learning) algorithms (Zhang et al., 2020a; Kardeş et al., 2011; Ma et al., 2023; Blanchet et al., 2023; Shi et al., 2024; Liu et al., 2025), often under full observability or individually rewarded settings. We complement this line by addressing the cooperative, partially observable CTDE regime, where agents receive a single joint reward and act only local observations.

In cooperative MARL, robustness has been modeled along several complementary axes, including adversarial (Byzantine) teammates (Li et al., 2024), state/observation disturbances (Guo et al., 2024), communication errors (Yu et al., 2024), risk-sensitive objectives that guard against tail events under a fixed model (Shen et al., 2023), and explicit model uncertainty Kwak et al. (2010); Zhang et al. (2020b). Focusing on the last category, Kwak et al. (2010) address model uncertainty with sparse, execution-time communication, whereas Zhang et al. (2020b) study settings in which each agent observes the full state and receives individual reward. Similarly, Bukharin et al. (2023) also considers settings where each agent receives individual reward, and achieve robustness by controlling the Lipschitz constant of each agent's policy. In contrast, our work targets robustness to model uncertainty in the cooperative CTDE setting, complementing prior approaches by providing a systematic framework that does *not* require real-time communication and operates under partial observability with a single team reward.

B Examples

Example 1 (Naïve single-agent robust action values cannot guarantee DrIGM)**.** *Consider a robust cooperative two-agent task (illustrated in Fig. 5) with action spaces $\mathcal{A}_1 = \mathcal{A}_2 = \{1, 2\}$, state space $\mathcal{S} = \{s_0, s_1, s_2, s_3, s_4\}$, and uncertainty set $\mathcal{P} = \{P_1, P_2\}$. Let s_0 be the initial state, and let s_1,s_2,s_3 and s_4 all be absorbing states with zero reward. We assume each agent observes the full state. For P_1, all the transitions are fully deterministic. P_2 differs from P_1 only in transitions on joint actions $(1, 2)$ and $(2, 1)$, given by:*

$$\mathbb{P}(S_2 \mid 1, 2) = \frac{1}{3}, \mathbb{P}(S_3 \mid 1, 2) = \frac{2}{3},$$

$$\mathbb{P}(S_2 \mid 2,1) = \frac{2}{3}, \mathbb{P}(S_3 \mid 2,1) = \frac{1}{3},$$

As shown in Fig. 5, the optimal joint action value function at state s_0 is (we omit the $\gamma/(1-\gamma)$ factor for clarity)

$$Q_{\text{tot}}^{P_1}(s_0,1,1) = 0.7, \quad Q_{\text{tot}}^{P_1}(s_0,1,2) = 0.4, \quad Q_{\text{tot}}^{P_1}(s_0,2,1) = 1.0, \quad Q_{\text{tot}}^{P_1}(s_0,2,2) = 0.7;$$
$$Q_{\text{tot}}^{P_2}(s_0,1,1) = 0.7, \quad Q_{\text{tot}}^{P_2}(s_0,1,2) = 0.8, \quad Q_{\text{tot}}^{P_2}(s_0,2,1) = 0.6, \quad Q_{\text{tot}}^{P_2}(s_0,2,2) = 0.7.$$

Therefore, the robust joint action-value function $Q_{\text{tot}}^{\mathcal{P}}(s,\mathbf{a}) = \inf_{P\in\mathcal{P}} Q_{\text{tot}}^{\mathcal{P}}(s,\mathbf{a})$ is given by:

$$Q_{\text{tot}}^{\mathcal{P}}(s_0,1,1) = 0.7, \quad Q_{\text{tot}}^{\mathcal{P}}(s_0,1,2) = 0.4, \quad Q_{\text{tot}}^{\mathcal{P}}(s_0,2,1) = 0.6, \quad Q_{\text{tot}}^{\mathcal{P}}(s_0,2,2) = 0.7,$$

It is straightforward to check that the following individual action-value functions $\{Q_i^{P_j}\}_{i,j\in[2]}$ satisfy $Q_{\text{tot}}^{P}(s,a_1,a_2) = Q_1^P(s,a_1) + Q_2^P(s,a_2)$ for all $s \in \mathcal{S}$ and $P \in \mathcal{P}$, which is a special case of the classical IGM property:

$$Q_1^{P_1}(s_0,1) = 0, \quad Q_1^{P_1}(s_0,2) = 0.3, \quad Q_2^{P_1}(s_0,1) = 0.7, \quad Q_2^{P_1}(s_0,2) = 0.4,$$
$$Q_1^{P_2}(s_0,1) = 0.2, \quad Q_1^{P_2}(s_0,2) = 0.1, \quad Q_2^{P_2}(s_0,2) = 0.5, \quad Q_2^{P_2}(s_0,2) = 0.6.$$

Suppose the robust individual action-value function is defined as $Q_i^{\text{rob}}(s,a) = \inf_{P\in\mathcal{P}} Q_i^P(s,a)$, as in the single-agent DR-RL literature. Therefore, the robust individual action-value functions are given by:

$$Q_1^{\text{rob}}(s_0,1) = 0, \quad Q_1^{\text{rob}}(s_0,2) = 0.1, \quad Q_2^{\text{rob}}(s_0,1) = 0.5, \quad Q_2^{\text{rob}}(s_0,2) = 0.4,$$

At s_0, these robustifications fail to satisfy DrIGM:

$$(2,1) = \left(\arg\max_{a_1} Q_1^{\text{rob}}(s_0,a_1), \arg\max_{a_2} Q_2^{\text{rob}}(s_0,a_2)\right) \notin \arg\max_{\mathbf{a}} Q_{\text{tot}}^{\mathcal{P}}(s_0,\mathbf{a}) = \{(1,1),(2,2)\}.$$

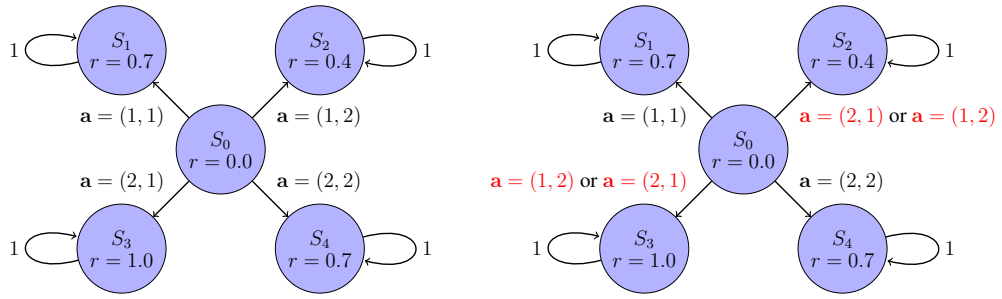


Figure 5: Fig. 5a is the MDP under transition kernel P_1, Fig. 5b is under P_2. The two differ in their transition probabilities to s_2 and s_3.

Example 2 (DrIGM can address cases where IGM fails.)**.** *Consider a similar robust cooperative two-agent task (illustrated in Fig. 6) with action spaces $\mathcal{A}_1 = \mathcal{A}_2 = \{1,2\}$, state space $\mathcal{S} = \{s_0, s_1, s_2, s_3, s_4\}$, and uncertainty set $\mathcal{P} = \{P_1, P_2\}$. Let P_1 be the training and testing environment. For P_1, all the transitions are fully deterministic, the optimal joint action value function at state s_0 is (we omit the $\gamma/(1-\gamma)$ factor for clarity):*

$$Q_{\text{tot}}^{P_1}(s_0,1,1) = 0.7, \; Q_{\text{tot}}^{P_1}(s_0,1,2) = 0.4, \quad Q_{\text{tot}}^{P_1}(s_0,2,1) = 1.0, \; Q_{\text{tot}}^{P_1}(s_0,2,2) = 0.5.$$

P_2 differs from P_1 in that all actions leads to S_4. Therefore, the optimal joint action value function at state s_0 is (we omit the $\gamma/(1-\gamma)$ factor for clarity):

$$Q_{\text{tot}}^{P_2}(s_0,1,1) = 0.4, \; Q_{\text{tot}}^{P_2}(s_0,1,2) = 0.4, \quad Q_{\text{tot}}^{P_2}(s_0,2,1) = 0.4, \; Q_{\text{tot}}^{P_2}(s_0,2,2) = 0.4.$$

It can be verified that P_1 does not admit a VDN-style value decomposition, but the worst case, P_2, admits a feasible VDN-style value decomposition.

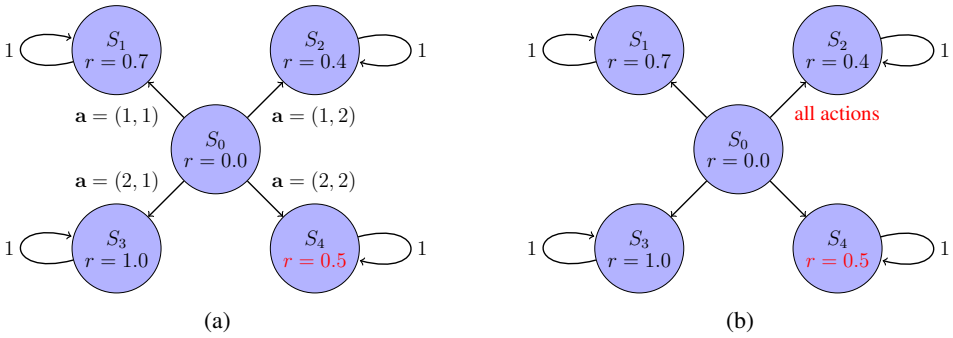


Figure 6: Fig. 6a is the MDP under transition kernel P_1, Fig. 6b is under P_2.

C Proofs

C.1 Proof of Theorem 1

Proof. Recall that for all $\mathbf{a} \in \mathcal{A}$, we have

$$Q^{\mathcal{P}}_{\text{tot}}(\mathbf{h}, \mathbf{a}) = \inf_{P \in \mathcal{P}} Q^{P}_{\text{tot}}(\mathbf{h}, \mathbf{a}) \qquad \text{(Eq. (3))}$$

$$P^{\text{worst}}(\mathbf{h}, \mathbf{a}) \in \arg\inf_{P \in \mathcal{P}} Q^{P}_{\text{tot}}(\mathbf{h}, \mathbf{a}). \qquad \text{(Equation (4))}$$

Thus, $Q^{P^{\text{worst}}(\mathbf{h},\mathbf{a})}_{\text{tot}}(\mathbf{h}, \mathbf{a}) = Q^{\mathcal{P}}_{\text{tot}}(\mathbf{h}, \mathbf{a})$. In Equation (4), we also defined $\bar{\mathbf{a}} \in \arg\max_{\mathbf{a} \in \mathcal{A}} Q^{\mathcal{P}}_{\text{tot}}(\mathbf{h}, \mathbf{a})$. Since $P^{\text{worst}}(\mathbf{h}, \bar{\mathbf{a}}) \in \mathcal{P}$, by assumption there exist $[Q_i^{P^{\text{worst}}(\mathbf{h},\bar{\mathbf{a}})}]_{i \in [N]}$ that satisfy IGM for $Q^{P^{\text{worst}}(\mathbf{h},\bar{\mathbf{a}})}_{\text{tot}}$ under $\mathbf{h}$. Therefore,

$$\begin{aligned}
&\left(\arg\max_{a_1} Q^{\text{rob}}_1(h_1, a_1), \ldots, \arg\max_{a_N} Q^{\text{rob}}_N(h_N, a_N)\right) \\
&= \left(\arg\max_{a_1} Q_1^{P^{\text{worst}}(\mathbf{h},\bar{\mathbf{a}})}(h_1, a_1), \ldots, \arg\max_{a_N} Q_N^{P^{\text{worst}}(\mathbf{h},\bar{\mathbf{a}})}(h_N, a_N)\right) && \text{(Eq. (4))} \\
&\subseteq \arg\max_{\mathbf{a}} Q^{P^{\text{worst}}(\mathbf{h},\bar{\mathbf{a}})}_{\text{tot}}(\mathbf{h}, \mathbf{a}) && \text{(IGM)} \\
&\subseteq \arg\max_{\mathbf{a}} Q^{P^{\text{worst}}(\mathbf{h},\mathbf{a})}_{\text{tot}}(\mathbf{h}, \mathbf{a}) && \text{(Eq. (4))} \\
&= \arg\max_{\mathbf{a}} Q^{\mathcal{P}}_{\text{tot}}(\mathbf{h}, \mathbf{a}),
\end{aligned}$$

which shows that $[Q^{\text{rob}}_i]_{i \in [N]}$ satisfy DrIGM under $\mathbf{h}$. □

Remark 1. *The statement of Theorem 1 assumes that for all $P \in \mathcal{P}$ there exist $[Q^P_i]_{i \in [N]}$ satisfying IGM for Q^P_{tot} under joint history $\mathbf{h} \in \mathcal{H}$. However, we note here that this assumption can be relaxed to only requiring that there exist $[Q_i]_{i \in [N]}$ satisfying IGM for $Q^{P^{\text{worst}}}_{\text{tot}}$ under $\mathbf{h}$.*

Remark 2. *As shown in Example 1, an adversarial model $P \in \mathcal{P}$ that minimizes one agent's value need not coincide with the adversarial model $P' \in \mathcal{P}$ that minimizes the joint value. Theorem 1 circumvents this problem by directly considering the global worst case, i.e.,*

$$Q^{\text{rob}}_i(h_i, a_i) := Q_i^{P^{\text{worst}}(\mathbf{h},\bar{\mathbf{a}})}(h_i, a_i). \qquad (18)$$

Given this definition, a robust joint action is given by $\bar{\mathbf{a}} = (1,1)$ or $(2,2)$, and there exist robust individual action-value functions are given by:

$$Q^{\text{rob}}_1(s_0, 1) = 0.2, \quad Q^{\text{rob}}_1(s_0, 2) = 0.1, \quad Q^{\text{rob}}_2(s_0, 1) = 0.7, \quad Q^{\text{rob}}_2(s_0, 2) = 0.4,$$

where the robust individual action is given by $a_1 = 1, a_2 = 1$. Alternatively, another set of individual action-value functions are given by:

$$Q_1^{\text{rob}}(s_0, 1) = 0, \qquad Q_1^{\text{rob}}(s_0, 2) = 0.3, \quad Q_2^{\text{rob}}(s_0, 1) = 0.5, \qquad Q_2^{\text{rob}}(s_0, 2) = 0.6,$$

where the robust individual action is given by $a_1 = 2, a_2 = 2$. Either way, the robust individual actions are aligned with the joint actions.

C.2 Proof of Theorem 2

Proof. We proceed by proving that IGM holds for under each of the three conditions given in Theorem 2. By Theorem 1, this suffices to show that DrIGM holds for under each of the three conditions given in Theorem 2.

VDN condition. Given a joint history $\mathbf{h} \in \mathcal{H}$, for any $P \in \mathcal{P}$, we have

$$Q_{\text{tot}}^P(\mathbf{h}, \mathbf{a}) = \sum_{i \in [N]} Q_i^P(h_i, a_i), \quad \forall \mathbf{a} = (a_1, \ldots, a_N) \in \mathcal{A}.$$

Let $\bar{a}_i = \arg\max_{a_i} Q_i^P(h_i, a_i)$ for $i \in [N]$, and let $\bar{\mathbf{a}} = [\bar{a}_i]_{i \in [N]}$. Then, for any $\mathbf{a} \in \mathcal{A}$,

$$\begin{aligned} Q_{\text{tot}}^P(\mathbf{h}, \mathbf{a}) &= \sum_{i \in [N]} Q_i^P(h_i, a_i), \\ &\leq \sum_{i \in [N]} Q_i^P(h_i, \bar{a}_i) && \text{(Definition of } \bar{a}_i\text{)} \\ &= Q_{\text{tot}}^P(\mathbf{h}, \bar{\mathbf{a}}). \end{aligned}$$

This implies that

$$\left(\arg\max_{a_1} Q_1^P(h_1, a_1), \ldots, \arg\max_{a_N} Q_N^P(h_N, a_N) \right) \subseteq \arg\max_{\mathbf{a}} Q_{\text{tot}}^P(\mathbf{h}, \mathbf{a}),$$

so $[Q_i^P]_{i \in [N]}$ satisfy IGM for Q_{tot}^P under $\mathbf{h}$. Therefore, by Theorem 1, $[Q_i^{\text{rob}}]_{i \in [N]}$ satisfy DrIGM for $Q_{\text{tot}}^{\mathcal{P}}$ under $\mathbf{h}$.

QMIX condition. Given a joint history $\mathbf{h} \in \mathcal{H}$, for any $P \in \mathcal{P}$, suppose the following monotonicity property holds:

$$\frac{\partial Q_{\text{tot}}^P(\mathbf{h}, \mathbf{a})}{\partial Q_i^P(h_i, a_i)} \geq 0, \quad \forall i \in [N],\ \mathbf{a} = (a_1, \ldots, a_N) \in \mathcal{A}.$$

Let $\bar{a}_i = \arg\max_{a_i} Q_i^P(h_i, a_i)$ for $i \in [N]$, and let $\bar{\mathbf{a}} = [\bar{a}_i]_{i \in [N]}$. Given that $\partial Q_{\text{tot}}^P(\mathbf{h}, \mathbf{a}) / \partial Q_1^P(h_1, a_1) \geq 0$ and $\bar{a}_1 = \arg\max_{a_1} Q_1^P(h_1, a_1)$, we have (for any $\mathbf{a} \in \mathcal{A}$)

$$Q_{\text{tot}}^P(\mathbf{h}, \mathbf{a}) \leq Q_{\text{tot}}^P(\mathbf{h}, \bar{a}_1, a_2, \ldots, a_N).$$

Applying the same logic to all $i \in [N]$ yields that for any $\mathbf{a} \in \mathcal{A}$,

$$\begin{aligned} Q_{\text{tot}}^P(\mathbf{h}, \mathbf{a}) &\leq Q_{\text{tot}}^P(\mathbf{h}, \bar{a}_1, \bar{a}_2, \ldots, \bar{a}_N) && (19) \\ &= Q_{\text{tot}}^P(\mathbf{h}, \bar{\mathbf{a}}). && (20) \end{aligned}$$

This implies that

$$\left(\arg\max_{a_1} Q_1^P(h_1, a_1), \ldots, \arg\max_{a_N} Q_N^P(h_N, a_N) \right) \subseteq \arg\max_{\mathbf{a}} Q_{\text{tot}}^P(\mathbf{h}, \mathbf{a}),$$

so $[Q_i^P]_{i \in [N]}$ satisfy IGM for Q_{tot}^P under $\mathbf{h}$. Therefore, by Theorem 1, $[Q_i^{\text{rob}}]_{i \in [N]}$ satisfy DrIGM for $Q_{\text{tot}}^{\mathcal{P}}$ under $\mathbf{h}$.

QTRAN condition. Given a joint history $\mathbf{h} \in \mathcal{H}$, for any $P \in \mathcal{P}$, we have (for all $\mathbf{a} = (a_1, \ldots, a_N) \in \mathcal{A}$):

$$\sum_{i=1}^{N} Q_i^P(h_i, a_i) - Q_{\text{tot}}^P(\mathbf{h}, \mathbf{a}) + V_{\text{tot}}(\mathbf{h}) = \begin{cases} 0, & \mathbf{a} = \bar{\mathbf{a}}, \\ \geq 0, & \mathbf{a} \neq \bar{\mathbf{a}}, \end{cases} \tag{21}$$

where $\bar{\mathbf{a}} = [\bar{a}_i]_{i \in [N]}$ with $\bar{a}_i = \arg\max_{a_i} Q_i^P(h_i, a_i)$ and $V_{\text{tot}}(\mathbf{h}) = \max_{\mathbf{a}} Q_{\text{tot}}^P(\mathbf{h}, \mathbf{a}) - \sum_{i=1}^{N} Q_i^P(h_i, a_i)$. Therefore,

$$\begin{aligned} Q_{\text{tot}}^P(\mathbf{h}, \bar{\mathbf{a}}) &= \sum_{i=1}^{N} Q_i^P(h_i, \bar{a}_i) + V_{\text{tot}}(\mathbf{h}) && \text{(Eq. (21))} \\ &= \max_{\mathbf{a}} Q_{\text{tot}}^P(\mathbf{h}, \mathbf{a}), && \text{(Definition of } V(\mathbf{h})\text{)} \end{aligned}$$

This implies that

$$\left(\arg\max_{a_1} Q_1^P(h_1, a_1), \ldots, \arg\max_{a_N} Q_N^P(h_N, a_N) \right) \subseteq \arg\max_{\mathbf{a}} Q_{\text{tot}}^P(\mathbf{h}, \mathbf{a}),$$

so $[Q_i^P]_{i \in [N]}$ satisfy IGM for Q_{tot}^P under $\mathbf{h}$. Therefore, by Theorem 1, $[Q_i^{\text{rob}}]_{i \in [N]}$ satisfy DrIGM for $Q_{\text{tot}}^{\mathcal{P}}$ under $\mathbf{h}$.

Combining the three cases concludes the proof of Theorem 2. □

C.3 Proof of Theorem 3

Proof. Recall that given an uncertainty set $\mathcal{P}$, the robust joint action value is defined as,

$$Q_{\text{tot}}^{\mathcal{P}}(\mathbf{h}, \mathbf{a}) := \inf_{P \in \mathcal{P}} Q_{\text{tot}}^P(\mathbf{h}, \mathbf{a}), \ \forall (\mathbf{h}, \mathbf{a}) \in \mathcal{H} \times \mathcal{A}. \tag{22}$$

Given that $P_{\text{test}} \in \mathcal{P}$, we directly have:

$$Q_{\text{tot}}^{\mathcal{P}}(\mathbf{h}, \mathbf{a}) \leq Q_{\text{tot}}^{P_{\text{test}}}(\mathbf{h}, \mathbf{a}), \ \forall \mathbf{h} \in \mathcal{H}, \ \mathbf{a} \in \mathcal{A}. \tag{23}$$

This concludes the proof. □

D Robust Bellman Operators

We start by introducing the assumptions needed to derive the robust bellman operators.

Assumption 1 (Fail-state (Panaganti et al., 2022)). *The robust Dec-POMDP has a* fail state s_f *such that*

$$r(s_f, \mathbf{a}) = 0 \quad \textit{and} \quad P_{s_f, \mathbf{a}}(s_f) = 1, \qquad \forall \mathbf{a} \in \mathcal{A}, \ \forall P \in \mathcal{P}. \tag{24}$$

This requirement is mild, as fail states naturally arise in both simulated and physical systems. For example, in robotics, a configuration where the robot falls and cannot recover, whether in simulators such as MuJoCo or in real hardware, serves as a natural fail state. We can further relax it to the following assumption.

Assumption 2 (Vanishing minimal value (Lu et al., 2024)). *The underlying RMDP satisfies*

$$\min_{s \in \mathcal{S}} V_{\text{tot}}^{\mathcal{P}}(s) = 0. \tag{25}$$

Without loss of generality, we also assume that any initial state $s_1 \notin \arg\min_{s \in \mathcal{S}} V_{\text{tot}}^{\mathcal{P}}(s)$.

This assumption states that the lowest achievable robust value across all states is normalized to zero. The exclusion of the minimizing state as the starting point rules out the degenerate case where the agent begins with zero guaranteed return.

ρ-contamination uncertainty set. Given a ρ-contamination uncertainty set defined in Eq. (7), the robust bellman operator can be expanded as:

$$(\mathcal{T}^{\mathcal{P}}Q)(\mathbf{h},\mathbf{a}) = r(s,\mathbf{a}) + \gamma \inf_{P_{\mathbf{h},\mathbf{a}}\in\mathcal{P}_{\mathbf{h},\mathbf{a}}} \mathbb{E}_{\mathbf{h}'\sim P_{\mathbf{h},\mathbf{a}}}\left[\max_{\mathbf{a}'\in\mathcal{A}} Q(\mathbf{h}',\mathbf{a}')\right]. \tag{26}$$

$$= r(s,\mathbf{a}) + \gamma(1-\rho)\,\mathbb{E}_{\mathbf{h}'\sim P^0_{\mathbf{h},\mathbf{a}}}\left[\max_{\mathbf{a}'\in\mathcal{A}} Q(\mathbf{h}',\mathbf{a}')\right] \tag{27}$$

$$+ \rho \min_{s'\in\mathcal{S}} V^{\mathcal{P}}_{\text{tot}}(s'). \tag{28}$$

Under Assumption 1 (Assumption 2), we obtain that:

$$(\mathcal{T}^{\mathcal{P}}Q)(\mathbf{h},\mathbf{a}) = r(s,\mathbf{a}) + \gamma(1-\rho)\,\mathbb{E}_{\mathbf{h}'\sim P^0_{\mathbf{h},\mathbf{a}}}\left[\max_{\mathbf{a}'\in\mathcal{A}} Q(\mathbf{h}',\mathbf{a}')\right] \quad \text{(Assumption 1 (Assumption 2))}$$

$$= r(s,\mathbf{a}) + \gamma(1-\rho)\,\mathbb{E}_{\mathbf{h}'\sim P^0_{\mathbf{h},\mathbf{a}}}\left[Q(\mathbf{h}',\bar{\mathbf{a}}')\right], \quad \text{(Definition 2)}$$

where $\bar{a}'_i = \arg\max_{a'_i} Q^{\text{rob}}_i(h'_i, a'_i)$.

TV-uncertainty set. Leveraging Panaganti et al. (2022)[Proposition 1], given a TV-uncertainty set defined in Eq. (9), the robust bellman operator can be expanded as:

$$(\mathcal{T}Q^{\mathcal{P}}_{\text{tot}})(\mathbf{h},\mathbf{a}) = r(s,\mathbf{a}) - \inf_{\eta\in[0,\frac{2}{\rho(1-\gamma)}]} \gamma\,\mathbb{E}_{\mathbf{h}'\sim P^0_{\mathbf{h},\mathbf{a}}}\Bigg(\rho\left[\eta(s,\mathbf{a}) - \inf_{s''\in\mathcal{S}} V^{\mathcal{P}}_{\text{tot}}(s'')\right]_+ - \eta$$
$$+ \left[\eta(s,\mathbf{a}) - \max_{\mathbf{a}'\in\mathcal{A}} Q^{\mathcal{P}}_{\text{tot}}(\mathbf{h}',\mathbf{a}')\right]_+\Bigg). \tag{29}$$

Under Assumption 1 (Assumption 2), we obtain that:

$$(\mathcal{T}Q^{\mathcal{P}}_{\text{tot}})(\mathbf{h},\mathbf{a}) = r(s,\mathbf{a}) - \inf_{\eta\in[0,\frac{2}{\rho(1-\gamma)}]} \gamma\,\mathbb{E}_{\mathbf{h}'\sim P^0_{\mathbf{h},\mathbf{a}}}\Bigg(-(1-\rho)\eta(s,\mathbf{a})$$
$$+ \left[\eta(s,\mathbf{a}) - \max_{\mathbf{a}'\in\mathcal{A}} Q^{\mathcal{P}}_{\text{tot}}(\mathbf{h}',\mathbf{a}')\right]_+\Bigg). \quad \text{(Assumption 1 (Assumption 2))}$$

$$= r(s,\mathbf{a}) - \inf_{\eta\in[0,\frac{2}{\rho(1-\gamma)}]} \gamma\,\mathbb{E}_{\mathbf{h}'\sim P^0_{\mathbf{h},\mathbf{a}}}\Bigg(-(1-\rho)\eta(s,\mathbf{a})$$
$$+ \left[\eta(s,\mathbf{a}) - \max_{\mathbf{a}'\in\mathcal{A}} Q^{\mathcal{P}}_{\text{tot}}(h'_1,\ldots,h'_N,\bar{a}'_1,\ldots,\bar{a}'_N)\right]_+\Bigg). \quad \text{(Definition 2)}$$

E Algorithms

We offer a full description of our algorithms in this section, presented in Algorithms 2 to 7.

F Experiment Details

F.1 Task Description

We test our algorithms and baseline algorithms in `BuildingEnv` in Yeh et al. (2023). This environment considers the control of the heat flow in a multi-zone building so as to maintain a desired temperature setpoint. Building temperature simulation uses f*irst-principled physics models*, to capture the real-world dynamics. The environmental model and reward functions can differ from three climate types and locations (San Diego, Tucson, New York), which jointly decide the climate.

Episode. In `BuildingEnv`, each episode runs for 1 day, with 5-minute time intervals. That is, the horizon length $H = 288$, and the time interval length $\tau = 5/60$ hours. We set the discount factor $\gamma \simeq 0.997$ by using H as the effective horizon length $H = \frac{1}{1-\gamma}$.

State Space. For a building with N indoor zones, the state contains observable properties of the building environment at timestep t:

$$s(t) = (T_1(t),\ldots,T_N(t),T_E(t),T_G(t),Q^{GHI}(t),\bar{Q}^p(t)), \tag{30}$$

where $T_i(t)$ denotes zone i's temperature at time step t, $\bar{Q}^p(t)$ is the heat acquisition from occupants' activities, $Q^{GHI}(t)$ is the heat gain from the solar irradiance, and $T_G(t)$ and $T_E(t)$ denote the ground and outdoor environment temperature.

Algorithm 2 Robust VDN with ρ-contamination uncertainty set

Input robustness parameter ρ, target network update frequency f and ε
Initialize replay buffer $\mathcal{D}$
Initialize $[Q_i^{\text{rob}}]_{i\in[N]}$ with random parameters θ
Initialize target parameters $\theta^- = \theta$
for episode $h = 1, \ldots, H$ **do**
  Observe initial state s^0 and observation $o_i^0 = \sigma_i(s^0)$ for each agent i.
  for $t = 1, \ldots, T$ **do**
    Each agent i choose its action a_i^t using ε-greedy policy.
    Take joint action $\mathbf{a}^t$, observe the next state s^{t+1}, reward r^t and observation $o_i^{t+1} = \sigma_i(s^{t+1})$ for each agent i
    Store transition $(\mathbf{h}^t, \mathbf{a}^t, r^t, \mathbf{h}^{t+1})$ in replay buffer $\mathcal{D}$
    Sample a mini-batch of transitions $(\mathbf{h}, \mathbf{a}, r, \mathbf{h}')$ from $\mathcal{D}$
    Set $\bar{\mathbf{a}}' = [\arg\max_{a_i'} Q_i^{\text{rob}}(h_i', a_i'; \theta^-)]_{i\in[N]}$
    Set $Q_{\text{tot}}^{\mathcal{P},\text{VDN}}(\mathbf{h}, \mathbf{a}; \theta) = \sum_{i\in[N]} Q_i^{\text{rob}}(h_i, a_i; \theta)$
    Set $Q_{\text{tot}}^{\mathcal{P},\text{VDN}}(\mathbf{h}', \bar{\mathbf{a}}'; \theta^-) = \sum_{i\in[N]} Q_i^{\text{rob}}(h_i', \bar{a}_i'; \theta^-)$
    Set $y^{\text{target}} = r + \gamma(1-\rho) Q_{\text{tot}}^{\mathcal{P},\text{VDN}}(\mathbf{h}', \bar{\mathbf{a}}'; \theta^-)$
    Calculate TD loss $L_{\text{TD}} = (Q_{\text{tot}}^{\mathcal{P},\text{VDN}}(\mathbf{h}, \mathbf{a}; \theta) - y^{\text{target}})^2$
    Update θ by minimizing L_{TD}
    Update $\theta^- = \theta$ with frequency f
  end for
end for

Algorithm 3 Robust QMIX with ρ-contamination uncertainty set

Input robustness parameter ρ, target network update frequency f and ε
Initialize replay buffer $\mathcal{D}$
Initialize $[Q_i^{\text{rob}}]_{i\in[N]}$ and mixing network f_θ with random parameters θ
Initialize target parameters $\theta^- = \theta$
for episode $h = 1, \ldots, H$ **do**
  Observe initial state s^0 and observation $o_i^0 = \sigma_i(s^0)$ for each agent i.
  for $t = 1, \ldots, T$ **do**
    Each agent i choose its action a_i^t using ε-greedy policy.
    Take joint action $\mathbf{a}^t$, observe the next state s^{t+1}, reward r^t and observation $o_i^{t+1} = \sigma_i(s^{t+1})$ for each agent i
    Store transition $(\mathbf{h}^t, \mathbf{a}^t, s^t, r^t, \mathbf{h}^{t+1}, s^{t+1})$ in replay buffer $\mathcal{D}$
    Sample a mini-batch of transitions $(\mathbf{h}, \mathbf{a}, s, r, \mathbf{h}', s')$ from $\mathcal{D}$
    Set $\bar{\mathbf{a}}' = [\arg\max_{a_i'} Q_i^{\text{rob}}(h_i', a_i'; \theta^-)]_{i\in[N]}$
    Set $Q_{\text{tot}}^{\mathcal{P},\text{QMIX}}(\mathbf{h}, \mathbf{a}; \theta) = f_\theta((Q_i^{\text{rob}}(h_i, a_i; \theta))_{i\in[N]}, s)$
    Set $Q_{\text{tot}}^{\mathcal{P},\text{QMIX}}(\mathbf{h}', \bar{\mathbf{a}}'; \theta^-) = f_{\theta^-}((Q_i^{\text{rob}}(h_i', \bar{a}_i'; \theta^-))_{i\in[N]}, s')$
    Set $y^{\text{target}} = r + \gamma(1-\rho) Q_{\text{tot}}^{\mathcal{P},\text{QMIX}}(\mathbf{h}', \bar{\mathbf{a}}'; \theta^-)$
    Calculate TD loss $L_{\text{TD}} = (Q_{\text{tot}}^{\mathcal{P},\text{QMIX}}(\mathbf{h}, \mathbf{a}; \theta) - y^{\text{target}})^2$
    Update θ by minimizing L_{TD}
    Update $\theta^- = \theta$ with frequency f
  end for
end for

Algorithm 4 Robust QTRAN with ρ-contamination uncertainty set

Input robustness parameter ρ, target network update frequency f and ε
Initialize replay buffer $\mathcal{D}$
Initialize $[Q_i^{\text{rob}}]_{i\in[N]}$, $Q_{\text{tot}}^{\mathcal{P},\text{QTRAN}}$ and $V_{\text{tot}}^{\mathcal{P},\text{QTRAN}}$ with random parameters θ
Initialize target parameters $\theta^- = \theta$
for episode $h = 1, \ldots, H$ **do**
  Observe initial state s^0 and observation $o_i^0 = \sigma_i(s^0)$ for each agent i.
  for $t = 1, \ldots, T$ **do**
    Each agent i choose its action a_i^t using ε-greedy policy.
    Take joint action $\mathbf{a}^t$, observe the next state s^{t+1}, reward r^t and observation $o_i^{t+1} = \sigma_i(s^{t+1})$ for each agent i
    Store transition $(\mathbf{h}^t, \mathbf{a}^t, r^t, \mathbf{h}^{t+1})$ in replay buffer $\mathcal{D}$
    Sample a mini-batch of transitions $(\mathbf{h}, \mathbf{a}, r, \mathbf{h}')$ from $\mathcal{D}$
    Set $\bar{\mathbf{a}}' = [\arg\max_{a_i'} Q_i^{\text{rob}}(h_i', a_i'; \theta^-)]_{i\in[N]}$
    Set $y^{\text{target}} = r + \gamma(1-\rho) Q_{\text{tot}}^{\mathcal{P},\text{QTRAN}}(\mathbf{h}', \bar{\mathbf{a}}'; \theta^-)$
    Calculate TD loss $L_{\text{TD}} = (Q_{\text{tot}}^{\mathcal{P},\text{QTRAN}}(\mathbf{h}, \mathbf{a}; \theta) - y^{\text{target}})^2$
    Calculate L_{opt} using Eq. (12)
    Calculate L_{nopt} using Eq. (13)
    Update θ by minimizing $L = L_{\text{TD}} + L_{\text{opt}} + L_{\text{nopt}}$
    Update $\theta^- = \theta$ with frequency f
  end for
end for

Algorithm 5 Robust VDN with TV uncertainty set

Input robustness parameter ρ, target network update frequency f and ε
Initialize replay buffer $\mathcal{D}$
Initialize $[Q_i^{\text{rob}}]_{i\in[N]}$ with random parameters θ
Initialize dual network η_ξ with random parameters ξ
Initialize target parameters $\theta^- = \theta$
for episode $h = 1, \ldots, H$ **do**
  Observe initial state s^0 and observation $o_i^0 = \sigma_i(s^0)$ for each agent i.
  for $t = 1, \ldots, T$ **do**
    Each agent i choose its action a_i^t using ε-greedy policy.
    Take joint action $\mathbf{a}^t$, observe the next state s^{t+1}, reward r^t and observation $o_i^{t+1} = \sigma_i(s^{t+1})$ for each agent i
    Store transition $(\mathbf{h}^t, \mathbf{a}^t, s^t, r^t, \mathbf{h}^{t+1})$ in replay buffer $\mathcal{D}$
    Sample a mini-batch of transitions $(\mathbf{h}, \mathbf{a}, r, s, \mathbf{h}')$
    Calculate dual loss L_{dual} using Eq. (17)
    Update ξ by minimizing L_{dual}
    Sample another mini-batch of transitions $(\mathbf{h}, \mathbf{a}, s, r, \mathbf{h}')$ from $\mathcal{D}$
    Set $\bar{\mathbf{a}}' = [\arg\max_{a_i'} Q_i^{\text{rob}}(h_i', a_i'; \theta^-)]_{i\in[N]}$
    Set $Q_{\text{tot}}^{\mathcal{P},\text{VDN}}(\mathbf{h}, \mathbf{a}; \theta) = \sum_{i\in[N]} Q_i^{\text{rob}}(h_i, a_i; \theta)$
    Set $Q_{\text{tot}}^{\mathcal{P},\text{VDN}}(\mathbf{h}', \bar{\mathbf{a}}'; \theta^-) = \sum_{i\in[N]} Q_i^{\text{rob}}(h_i', \bar{a}_i'; \theta^-)$
    Set $y^{\text{target}} = r + \gamma(1-\rho)\eta_\xi(s, \mathbf{a}) - \gamma[\eta_\xi(s, \mathbf{a}) - Q_{\text{tot}}^{\mathcal{P},\text{VDN}}(\mathbf{h}', \bar{\mathbf{a}}'; \theta^-)]_+$
    Calculate TD loss $L_{\text{TD}} = (Q_{\text{tot}}^{\mathcal{P},\text{VDN}}(\mathbf{h}, \mathbf{a}; \theta) - y^{\text{target}})^2$
    Update θ by minimizing L_{TD}
    Update $\theta^- = \theta$ with frequency f
  end for
end for

Algorithm 6 Robust QMIX with TV uncertainty set

Input robustness parameter ρ, target network update frequency f and ε
Initialize replay buffer $\mathcal{D}$
Initialize $[Q_i^{\text{rob}}]_{i\in[N]}$ and mixing network f_θ with random parameters θ
Initialize dual network η_ξ with random parameters ξ
Initialize target parameters $\theta^- = \theta$
for episode $h = 1, \ldots, H$ **do**
  Observe initial state s^0 and observation $o_i^0 = \sigma_i(s^0)$ for each agent i.
  for $t = 1, \ldots, T$ **do**
    Each agent i choose its action a_i^t using ε-greedy policy.
    Take joint action $\mathbf{a}^t$, observe the next state s^{t+1}, reward r^t and observation $o_i^{t+1} = \sigma_i(s^{t+1})$ for each agent i
    Store transition $(\mathbf{h}^t, \mathbf{a}^t, s^t, r^t, \mathbf{h}^{t+1})$ in replay buffer $\mathcal{D}$
    Sample a mini-batch of transitions $(\mathbf{h}, \mathbf{a}, r, \mathbf{h}')$
    Calculate dual loss L_{dual} using Eq. (17)
    Update ξ by minimizing L_{dual}
    Sample another mini-batch of transitions $(\mathbf{h}, \mathbf{a}, s, r, \mathbf{h}')$ from $\mathcal{D}$
    Set $\bar{\mathbf{a}}' = [\arg\max_{a_i'} Q_i^{\text{rob}}(h_i', a_i'; \theta^-)]_{i\in[N]}$
    Set $Q_{\text{tot}}^{\mathcal{P},\text{QMIX}}(\mathbf{h}, \mathbf{a}; \theta) = f_\theta((Q_i^{\text{rob}}(h_i, a_i; \theta))_{i\in[N]}, s)$
    Set $Q_{\text{tot}}^{\mathcal{P},\text{QMIX}}(\mathbf{h}', \bar{\mathbf{a}}'; \theta^-) = f_{\theta^-}((Q_i^{\text{rob}}(h_i', \bar{a}_i'; \theta^-))_{i\in[N]}, s')$
    Set $y^{\text{target}} = r + \gamma(1-\rho)\eta_\xi(s, \mathbf{a}) - \gamma[\eta_\xi(s, \mathbf{a}) - Q_{\text{tot}}^{\mathcal{P},\text{QMIX}}(\mathbf{h}', \bar{\mathbf{a}}'; \theta^-)]_+$
    Calculate TD loss $L_{\text{TD}} = (Q_{\text{tot}}^{\mathcal{P},\text{QMIX}}(\mathbf{h}, \mathbf{a}; \theta) - y^{\text{target}})^2$
    Update θ by minimizing L_{TD}
    Update $\theta^- = \theta$ with frequency f
  end for
end for

Algorithm 7 Robust QTRAN with TV uncertainty set

Input robustness parameter ρ, target network update frequency f and ε
Initialize replay buffer $\mathcal{D}$
Initialize $[Q_i^{\text{rob}}]_{i\in[N]}$, $Q_{\text{tot}}^{\mathcal{P},\text{QTRAN}}$ and $V_{\text{tot}}^{\mathcal{P},\text{QTRAN}}$ with random parameters θ
Initialize dual network η_ξ with random parameters ξ
Initialize target parameters $\theta^- = \theta$
for episode $h = 1, \ldots, H$ **do**
  Observe initial state s^0 and observation $o_i^0 = \sigma_i(s^0)$ for each agent i.
  for $t = 1, \ldots, T$ **do**
    Each agent i choose its action a_i^t using ε-greedy policy.
    Take joint action $\mathbf{a}^t$, observe the next state s^{t+1}, reward r^t and observation $o_i^{t+1} = \sigma_i(s^{t+1})$ for each agent i
    Store transition $(\mathbf{h}^t, \mathbf{a}^t, s^t, r^t, \mathbf{h}^{t+1})$ in replay buffer $\mathcal{D}$
    Sample a mini-batch of transitions $(\mathbf{h}, \mathbf{a}, r, s, \mathbf{h}', s')$
    Calculate dual loss L_{dual} using Eq. (17)
    Update ξ by minimizing L_{dual}
    Sample another mini-batch of transitions $(\mathbf{h}, \mathbf{a}, s, r, \mathbf{h}')$ from $\mathcal{D}$
    Set $\bar{\mathbf{a}}' = [\arg\max_{a_i'} Q_i^{\text{rob}}(h_i', a_i'; \theta^-)]_{i\in[N]}$
    Set $y^{\text{target}} = r + \gamma(1-\rho)\eta_\xi(s, \mathbf{a}) - \gamma[\eta_\xi(s, \mathbf{a}) - Q_{\text{tot}}^{\mathcal{P},\text{QTRAN}}(\mathbf{h}', \bar{\mathbf{a}}'; \theta^-)]_+$
    Calculate TD loss $L_{\text{TD}} = (Q_{\text{tot}}^{\mathcal{P},\text{QTRAN}}(\mathbf{h}, \mathbf{a}; \theta) - y^{\text{target}})^2$
    Calculate L_{opt} using Eq. (12)
    Calculate L_{nopt} using Eq. (13)
    Update θ by minimizing $L = L_{\text{TD}} + L_{\text{opt}} + L_{\text{nopt}}$
    Update $\theta^- = \theta$ with frequency f
  end for
end for

Observation Space. For agent i, given the current state $s(t)$, its observation $o_i(t)$ is given by:

$$\begin{aligned} o_i(t) &= \sigma_i(s(t)) \\ &= (T_i(t), T_E(t), T_G(t), Q^{GHI}(t), \bar{Q}^p(t)). \end{aligned} \tag{31}$$

That is, agent i can observe the tempature at zone i, the heat acquisition from occupants' activities, the heat gain from the solar irradiance, and the ground and outdoor environment temperature.

Action Space. At time t, agent i's action is a scalar $a_i(t) \in [-1, 1]$, which sets the controlled heating supplied to zone i. The joint action $\mathbf{a}(t) = (a_1(t), \ldots, a_N(t))$

Reward Function. The objective is to reduce energy consumption while keeping the temperature within a given comfort range. Therefore, the reward function is a weighted average of these two goals:

$$r(t) = -(1-\beta)||\mathbf{a}(t)||_2 - \beta||T^{\text{target}} - T(t)||_2, \tag{32}$$

where $T^{\text{target}} = (T_1^{\text{target}}(t), \ldots, T_N^{\text{target}})$ are the target temperatures and $T(t) = (T_1(t), \ldots, T_N(t))$ are the actual zonal temperature, and $||\cdot||_2$ denote the ℓ_2 norm. We use the same default hyperparameter β across all experiments.

Environmental Uncertainty. The environmental model and reward functions can differ from three climate types and locations (San Diego, Tucson, New York), which jointly decide the climate. Besides, `BuildingEnv` contains distribution shifts in the ambient outdoor temperature profile T_E incurred by seasonal shifts.

F.2 Experiments Setup.

we implement distributionally robust algorithms with two types of uncertainty sets (ρ-contamination Zhang et al. (2024), TV-uncertaintyPanaganti et al. (2022)) and three different value factorization methods (VDN (Sunehag et al., 2017), QMIX (Rashid et al., 2020), QTRAN (Son et al., 2019)). We also implement the groupDR MARL algorithm from (Liu et al., 2025), also with these three different value factorization methods. Each experiment is run independently with 5 different random seeds.

Hyperparameters. Training lasts 600 episodes with parameter updates every 2 steps and target updates every 25k steps. Replay buffer size is 10k; batch size is 64. We use ε-greedy exploration with ε annealed from 1.0 to 0.01 over 120k steps. Hidden layer size is 64.

Environment configurations. The environment configurations we use throughout the three experiments are shown in Table 3 where `Env_1` is the training environment, and the other environments are numbered in order of how much they differ from `Env_1`.

Environment	Weather	Location
`Env_1`	Hot_Dry	Tucson
`Env_2`	Hot_Humid	Tampa
`Env_3`	Very_Hot_Humid	Honolulu
`Env_4`	Warm_Dry	El Paso
`Env_5`	Cool_Marine	Seattle
`Env_6`	Mixed_Humid	New York

Table 3: Environment configurations

Robust individual Q-networks. Each agent employs a recurrent Q-network (Hausknecht & Stone, 2015), encoding its local observation concatenated with the previous action (one-hot). Features pass through a fully connected layer with ReLU, followed by a single-layer LSTM (hidden size 64). The final hidden state is projected into Q-values. We optimize with RMSprop (lr $= 5 \times 10^{-4}$).

Mixing networks (QMIX). Following TorchRL (Bou et al., 2023), we use hypernetworks to generate state-dependent mixing weights while enforcing monotonicity. Per-agent Q-values are embedded, passed through ELU, and projected to a scalar joint Q. A state-conditioned bias is added via a two-layer MLP. Optimizer: RMSprop (lr $= 5 \times 10^{-4}$).

QTRAN Networks. We implement joint action-value and state-value networks as in Son et al. (2019). The joint action is encoded by concatenated one-hots, projected with ReLU, and optionally combined with agent features. The state-value network processes the global state and summed agent features in parallel, concatenates them, and outputs a scalar. Both use Adam ($\text{lr} = 10^{-3}$).

Dual networks (TV uncertainty). The global state and joint action (concatenated one-hots) are separately embedded via ReLU MLPs, concatenated, and passed through another ReLU layer before a final linear projection to a scalar. Optimizer: Adam ($\text{lr} = 10^{-3}$).

GroupDR training. GroupDR first fits a contextual-bandit-based worst-case reward estimator to estimate the worst-case reward, using data from `Env_1` to `Env_5` collected under a VDN-trained behavior policy. Using this estimator, we then train the individual Q-networks by randomly sampling episodes from `Env_1` to `Env_5`. (For fairness, `Env_6` is never used for training by any method.)

Normalized Return. Because the reward in `BuildingEnv` is negative, we normalize the returns to $[0, 1]$ by the following transformation:

$$\text{Normalized\_return} = \frac{\text{Return} + 9000}{9000}. \tag{33}$$

Experiment 1: climatic shifts. During training, we train each algorithm in `Env_1` by sampling episodes from Day 1 to Day 200. During evaluation, we evaluate each algorithm in `Env_1` to `Env_6`, by sampling 50 episodes from Day 1 to Day 200 in each environment. This setup demonstrates distribution shift induced by climate differences.

Experiment 2: seasonal shifts. During training, we train each algorithm in `Env_1` by sampling episodes from Day 1 to Day 200. During evaluation, we evaluate each algorithm in `Env_1`, by sampling 50 episodes from Day 400 to Day 600 in each environment. This setup demonstrates distribution shift induced by seasonality within the same location.

Experiment 3: climatic and seasonal shifts. During training, we train each algorithm in `Env_1` by sampling episodes from Day 1 to Day 200. During evaluation, we evaluate each algorithm in `Env_6`, by sampling 50 episodes from Day 400 to Day 600 in each environment. This setup demonstrates the combined effects of climate and seasonal shifts.

G Disclosure of LLM Usage

To improve clarity and readability, we used a large language model (LLM) to assist in polishing the writing. The LLM was only employed for language refinement (e.g., grammar, style, and conciseness) and was not involved in designing methods, experiments, or drawing conclusions.