

Beyond Monotonicity: Revisiting Factorization Principles in Multi-Agent Q-Learning *

Tianmeng Hu^{†1}, Yongzheng Cui^{†2}, Rui Tang^{†2}, Biao Luo^{‡2} and Ke Li^{†1}

¹Department of Computer Science, University of Exeter, U.K.

²School of Automation, Central South University, China.

th743@exeter.ac.uk, {yongzhengcui, ruitang02}@csu.edu.cn, biao.luo@hotmail.com,
k.li@exeter.ac.uk

Abstract: Value decomposition is a central approach in multi-agent reinforcement learning (MARL), enabling centralized training with decentralized execution by factorizing the global value function into local values. To ensure individual-global-max (IGM) consistency, existing methods either enforce monotonicity constraints, which limit expressive power, or adopt softer surrogates at the cost of algorithmic complexity. In this work, we present a dynamical systems analysis of non-monotonic value decomposition, modeling learning dynamics as continuous-time gradient flow. We prove that, under approximately greedy exploration, all zero-loss equilibria violating IGM consistency are unstable saddle points, while only IGM-consistent solutions are stable attractors of the learning dynamics. Extensive experiments on both synthetic matrix games and challenging MARL benchmarks demonstrate that unconstrained, non-monotonic factorization reliably recovers IGM-optimal solutions and consistently outperforms monotonic baselines. Additionally, we investigate the influence of temporal-difference targets and exploration strategies, providing actionable insights for the design of future value-based MARL algorithms.

Keywords: Reinforcement learning, multi-agent system.

1 Introduction

Cooperative learning is fundamental to enabling complex collective intelligence in multi-agent systems (MAS), with broad applicability across domains such as multi-robot coordination [1, 2], autonomous driving [3, 4, 5, 6], and smart grid control [7, 8]. Multi-agent reinforcement learning (MARL) [9, 10, 11, 12] offers a unified framework for learning both cooperative and competitive behaviors in complex environments. Within this framework, the paradigm of centralized training with decentralized execution (CTDE) has emerged as a standard approach for addressing cooperative tasks [11, 10]. A central component of CTDE is Value Function Factorization (VFF), which approximates the global joint action-value function Q_{tot} by aggregating individual agent value functions Q_i . This decomposition not only enables effective credit assignment but also enhances scalability in multi-agent learning [13].

The effectiveness of VFF methods hinges on satisfying the Individual-Global-Max (IGM) principle, which ensures that decentralized greedy action selections by individual agents are aligned with the globally optimal joint action. One of the earliest approaches, Value-Decomposition Networks (VDN) [10], achieves IGM by assuming a simple additive decomposition of the global value function. However, this assumption significantly limits its representational capacity. To overcome this limitation, QMIX [13] introduces a more flexible monotonicity constraint, requiring that $\frac{\partial Q_{\text{tot}}}{\partial Q_i} \geq 0$, which permits complex nonlinear relationships between Q_{tot} and the individual Q_i values. This constraint is enforced via a structured mixing network with non-negative weights. The success of QMIX marked

*Accepted as an oral paper at AAAI 2026.

[†]These authors contributed equally.

[‡]Corresponding authors.

a milestone in MARL research and established a dominant design paradigm: ensuring IGM through structured architectural constraints.

While effective in ensuring IGM, the monotonicity constraint limits the model’s expressive power [14, 15]. To overcome this bottleneck, subsequent research has explored more expressive architectures capable of representing the full class of IGM-consistent functions. For instance, QTRAN [14] introduces an auxiliary value function to reformulate the optimization objective, while QPLEX [15] proposes a sophisticated duplex dueling architecture to capture richer value structures. Although these methods expand representational power in theory, they often suffer from instability or excessive architectural complexity in practice. Meanwhile, an alternative line of work attributes the failure in non-monotonic tasks to the phenomenon of relative overgeneralization, arguing that simplistic exploration strategies such as ϵ -greedy are insufficient to escape suboptimal equilibria. This has motivated the development of more elaborate coordination-aware exploration mechanisms, exemplified by methods like MAVEN [16] and UneVEn [17].

However, we observe that prior work typically analyzes IGM-relevant algorithmic behavior in matrix games under a uniformly random exploration policy [15]. Under such settings, both monotonic and non-monotonic variants of QMIX fail to learn IGM-optimal solutions. In contrast, practical Q-learning commonly employs approximately greedy exploration strategies conditioned on the evolving value function. Motivated by this, we conducted preliminary experiments using a non-monotonic variant of QMIX combined with an ϵ -greedy exploration policy on a challenging matrix game. Surprisingly, the algorithm consistently converged to IGM-consistent optimal solutions.

This empirical finding leads us to hypothesize that the underlying learning dynamics are fundamentally altered under approximately greedy policies. Specifically, we conjecture that the feedback loop between the learned value function and the policy induces an implicit self-correcting mechanism, which naturally drives the system away from suboptimal, IGM-inconsistent solutions and toward globally optimal ones—without requiring explicit structural constraints. To validate our theoretical findings, we perform a novel dynamical-systems analysis of non-monotonic value-decomposition Q-learning. Our main contributions are:

- We formulate the non-monotonic value-decomposition Q-learning update as a continuous-time gradient flow and derive its learning dynamics.
- We prove that under approximately greedy exploration policies, every zero-loss fixed point that violates the IGM principle becomes an unstable saddle point, from which the learning trajectory naturally escapes. We further show that only IGM-consistent solutions are stable attractors of the flow.
- Extensive experiments on matrix games and challenging multi-agent benchmarks confirm our theory. Results show that unconstrained QMIX reliably recovers IGM-consistent optimal solutions in matrix games and outperforms monotonic baselines.
- We explore the impact of SARSA-style TD targets [18] and intrinsic-reward-driven exploration mechanisms on the stability and performance of non-monotonic value decomposition.

2 Preliminaries

2.1 Dec-POMDPs

Cooperative multi-agent tasks are commonly modeled as decentralized partially observable Markov decision processes (Dec-POMDPs) [19]. A Dec-POMDP is formally defined as a tuple

$$G = \langle N, S, \{\mathcal{A}_i\}, P, r, \{\Omega_i\}, O, \gamma \rangle, \quad (1)$$

where:

- N is the number of agents.

- S is the global state space of the environment.
- $\mathbf{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$ denotes the joint action space, where $\mathcal{A}_i$ denotes the discrete action space of agent i .
- $P(s' | s, \mathbf{a}) : S \times \mathcal{A} \times S \rightarrow [0, 1]$ is the state transition function.
- $r(s, \mathbf{a}) : S \times \mathcal{A} \rightarrow \mathbb{R}$ is the shared reward function, assigning the same team reward to all agents at each time step.
- $\mathbf{O} = O_1 \times \cdots \times O_n$ is the joint observation space. $Z : S \rightarrow \mathbf{O}$ is the observation function, where each agent i receives a local observation $o_t^i = Z(s_t)_i$.
- $\gamma \in [0, 1)$ is the discount factor.

All agents act according to their individual policies $\mu = (\pi_1, \dots, \pi_N)$, with the collective objective of maximizing the expected cumulative discounted team return:

$$J(\pi) = \mathbb{E}_{s_0, \mathbf{a}_t, s_{t+1}} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, \mathbf{a}_t) \mid \mu \right]. \quad (2)$$

2.2 Value Function Factorization

Value function factorization (VFF) aims to learn a joint action-value function $Q_{\text{tot}}(s, \mathbf{a})$ for cooperative multi-agent settings. This factorization is typically implemented via a mixing function that takes as input the individual agent values Q_i and, optionally, the global state s , and outputs the global value Q_{tot} . For example, VDN adopts a simple additive form:

$$Q_{\text{tot}}(s, \mathbf{a}) = \sum_{i=1}^n Q_i(\eta_i, a_i), \quad (3)$$

where η_i denotes agent i 's action-observation history. QMIX employs a nonlinear mixing network g_{mix} :

$$Q_{\text{tot}}(s, \mathbf{a}) = g_{\text{mix}}(\{Q_i(\eta_i, a_i)\}_{i=1}^n, s). \quad (4)$$

These methods are trained end-to-end by minimizing the joint Bellman error. The typical loss function is defined as:

$$L(\theta) = \mathbb{E}_{s, \mathbf{a}, r, s'} \left[(y^{\text{tot}} - Q_{\text{tot}}(s, \mathbf{a}; \theta))^2 \right], \quad (5)$$

where the target value is given by

$$y^{\text{tot}} = r + \gamma \max_{\mathbf{a}'} Q_{\text{tot}}(s', \mathbf{a}'; \theta^-),$$

and θ and θ^- denote the parameters of the current and target networks, respectively.

2.3 IGM Consistency

A value decomposition is said to satisfy the Individual-Global-Max (IGM) principle if, for any state s , the optimal joint action with respect to the global value function Q_{tot} can be obtained by independently maximizing each agent's local utility function. Formally, this condition is defined as:

$$\begin{aligned} & \arg \max_{\mathbf{a} \in \mathcal{A}} Q_{\text{tot}}(s, \mathbf{a}) \\ &= \left(\arg \max_{a_1 \in \mathcal{A}_1} Q_1(\eta_1, a_1), \dots, \arg \max_{a_n \in \mathcal{A}_n} Q_n(\eta_n, a_n) \right). \end{aligned} \quad (6)$$

The IGM principle is crucial because it enables decentralized execution: when the condition holds, each agent can independently select its action by greedily maximizing its own local utility Q_i , and the resulting joint action will be globally optimal with respect to the learned Q_{tot} .

3 Analysis

We begin our analysis by simplifying the original Dec-POMDP problem into a single-state matrix game. Conceptually, a single-state game can be viewed as a localized coordination subproblem embedded within any Dec-POMDP state. Instead of using a temporal-difference target, we assume a known, fixed ground-truth payoff function $y(\mathbf{a})$. The formal definition is as follows:

Definition 1 (Single-State Game). *Consider a single-state matrix game involving N agents. Each agent $i \in \{1, \dots, N\}$ has a discrete action set $\mathcal{A}_i$. The joint action is denoted as $\mathbf{a} = (a_1, \dots, a_N)$, and the corresponding ground-truth reward is $y(\mathbf{a}) \in \mathbb{R}$. We assume the existence of a unique globally optimal joint action:*

$$\mathbf{a}^* = \arg \max_{\mathbf{a}} y(\mathbf{a}). \quad (7)$$

The value function is represented by a parameterized value decomposition network. For a given parameter vector $\boldsymbol{\theta}$, the network outputs local values $Q_i(a_i; \boldsymbol{\theta})$ for each agent. A mixing network f_{mix} then aggregates these local values into a joint value:

$$Q_{\text{tot}}(\mathbf{a}; \boldsymbol{\theta}) = f_{\text{mix}}(Q_1(a_1; \boldsymbol{\theta}), \dots, Q_N(a_N; \boldsymbol{\theta}); \boldsymbol{\theta}). \quad (8)$$

To facilitate theoretical analysis, we define a global state vector $\mathbf{q} \in \mathbb{R}^{\sum_i |\mathcal{A}_i|}$ that concatenates all local Q -values:

$$\mathbf{q} := \begin{pmatrix} Q_1(a_{1,1}), \dots, Q_1(a_{1,|\mathcal{A}_1|}), \\ \dots, \\ Q_N(a_{N,1}), \dots, Q_N(a_{N,|\mathcal{A}_N|}) \end{pmatrix}^\top. \quad (9)$$

From this perspective, each $Q_i(a_i)$ is a component of the vector $\mathbf{q}$, and the joint value function $Q_{\text{tot}}(\mathbf{a}; \mathbf{q})$ is treated as a function that directly takes $\mathbf{q}$ as input, with its specific form determined by the structure of f_{mix} . We assume that f_{mix} satisfies a local non-degeneracy condition: it is locally Lipschitz continuous in each coordinate direction, and its Jacobian with respect to $\mathbf{q}$ has full rank on the normal subspace at the points considered in our analysis.

The learning objective is to minimize the mean squared error loss weighted by a behavior policy $\mu(\mathbf{a}; \mathbf{q})$:

$$\mathcal{L}(\mathbf{q}) = \sum_{\mathbf{a}} \mu(\mathbf{a}; \mathbf{q}) (y(\mathbf{a}) - Q_{\text{tot}}(\mathbf{a}; \mathbf{q}))^2. \quad (10)$$

3.1 Fixed Uniform Policy

We begin by considering a fixed, uniform behavior policy μ_0 that is independent of $\mathbf{q}$, i.e., $\mu_0(\mathbf{a}) = 1/|\mathcal{A}|^N$. Under this setting, the loss simplifies to:

$$\mathcal{L}_0(\mathbf{q}) = \frac{1}{|\mathcal{A}|^N} \sum_{\mathbf{a}} (y(\mathbf{a}) - Q_{\text{tot}}(\mathbf{a}; \mathbf{q}))^2. \quad (11)$$

This learning problem reduces to a standard supervised regression task, where the goal is to fit Q_{tot} to a fixed target function $y(\mathbf{a})$. The gradient of the loss with respect to $\mathbf{q}$ is given by:

$$\begin{aligned} \nabla_{\mathbf{q}} \mathcal{L}_0(\mathbf{q}) \\ = -\frac{2}{|\mathcal{A}|^N} \sum_{\mathbf{a}} (y(\mathbf{a}) - Q_{\text{tot}}(\mathbf{a}; \mathbf{q})) \nabla_{\mathbf{q}} Q_{\text{tot}}(\mathbf{a}; \mathbf{q}). \end{aligned} \quad (12)$$

Theorem 1 (Zero-Loss Points Without IGM Consistency). *Under the fixed uniform policy, there exists a set of zero-loss points $\mathcal{M}_0 = \{\mathbf{q} \mid \mathcal{L}_0(\mathbf{q}) = 0\}$ containing infinitely many elements, and this set includes points that do not satisfy IGM consistency.*

Proof. Due to the universal approximation capability of f_{mix} , there may exist many local Q -value configurations $\mathbf{q}$ that violate the IGM condition but can still perfectly fit the target payoff function $y(\mathbf{a})$ via appropriate mixing function parameters. At such points, the loss is exactly zero, and the gradient vanishes accordingly. The inherent ambiguity in credit assignment leads to a multiplicity of solutions. Consequently, the learning process may converge to any of these zero-loss solutions without any inherent bias toward IGM-consistent ones. ■

3.2 Approximately Greedy Policy

When using an approximately greedy policy, the behavior policy depends on the current value estimates, resulting in a value-coupled dynamical system. Directly analyzing ϵ -greedy policies is challenging due to the non-differentiability of the arg max operator. To circumvent this issue, we introduce a smooth and differentiable surrogate based on the softmax policy with temperature parameter $\tau > 0$:

$$\pi_i^\tau(a_i | \mathbf{q}) := \frac{e^{Q_i(a_i)/\tau}}{\sum_{b \in \mathcal{A}_i} e^{Q_i(b)/\tau}}, \quad (13)$$

$$\mu_\tau(\mathbf{a} | \mathbf{q}) = \prod_{i=1}^N \pi_i^\tau(a_i | \mathbf{q}). \quad (14)$$

The corresponding smooth loss function is defined as:

$$\mathcal{L}_\tau(\mathbf{q}) = \sum_{\mathbf{a}} \mu_\tau(\mathbf{a} | \mathbf{q}) (y(\mathbf{a}) - Q_{\text{tot}}(\mathbf{a}; \mathbf{q}))^2. \quad (15)$$

Lemma 1. Let $\{\tau_k\}_{k=1}^\infty$ be a sequence of temperatures such that $\tau_k \downarrow 0$. Suppose that at parameter point $\mathbf{q}$, each agent has a unique greedy action and the mixing function f_{mix} satisfies the local non-degeneracy condition. Then,

$$\lim_{k \rightarrow \infty} \nabla_{\mathbf{q}} \mathcal{L}_{\tau_k}(\mathbf{q}) \in \partial_{\mathbf{q}}^{\text{C}} \mathcal{L}_0(\mathbf{q}), \quad (16)$$

where ∂^{C} denotes the Clarke generalized gradient, and $\mathcal{L}_0$ corresponds to the limiting loss as $\tau \rightarrow 0$ in the softmax policy μ_τ .

Lemma 2 provides a theoretical justification for using the smooth surrogate system to analyze the stability properties of the original non-smooth objective. The complete proof is deferred to Appendix A.

In contrast to the fixed uniform policy case, the gradient of $\mathcal{L}_\tau(\mathbf{q})$ now consists of two distinct components:

$$\begin{aligned} \nabla_{\mathbf{q}} \mathcal{L}_\tau(\mathbf{q}) = & \underbrace{\sum_{\mathbf{a}} (\nabla_{\mathbf{q}} \mu_\tau(\mathbf{a} | \mathbf{q})) (y(\mathbf{a}) - Q_{\text{tot}})^2}_{\text{policy gradient term}} \\ & - 2 \underbrace{\sum_{\mathbf{a}} \mu_\tau(\mathbf{a} | \mathbf{q}) (y(\mathbf{a}) - Q_{\text{tot}}) \nabla_{\mathbf{q}} Q_{\text{tot}}}_{\text{value gradient term}}. \end{aligned} \quad (17)$$

The first term (the policy gradient term) alters the learning dynamics compared to the uniform case. We model the learning process as a continuous-time gradient flow, represented by the following ordinary differential equation (ODE):

$$\dot{\mathbf{q}} = -\nabla_{\mathbf{q}} \mathcal{L}_\tau(\mathbf{q}). \quad (18)$$

The local stability of a fixed point $\mathbf{q}^*$ is determined by the spectrum of its Jacobian matrix, given by $J_\tau(\mathbf{q}^*) = -H_\tau(\mathbf{q}^*)$, where $H_\tau = \nabla_{\mathbf{q}}^2 \mathcal{L}_\tau$ is the Hessian matrix of the loss. If $H_\tau(\mathbf{q}^*)$ is positive definite, then $\mathbf{q}^*$ is asymptotically stable. If it has at least one negative eigenvalue, then $\mathbf{q}^*$ is unstable.

We now focus on the set of all zero-loss points and analyze their stability. These points form a *zero-loss manifold* $\mathcal{M}$, defined as the set of parameter configurations that perfectly fit the ground-truth payoff:

$$\mathcal{M} := \{\mathbf{q} \mid Q_{\text{tot}}(\mathbf{a}; \mathbf{q}) = y(\mathbf{a}), \forall \mathbf{a}\}. \quad (19)$$

Every point within $\mathcal{M}$ is a fixed point of the gradient flow. The local stability of such points is governed by the curvature of the loss function, as characterized by the Hessian matrix. Theorem 2 establishes that for any perturbation direction that alters a local greedy action to a suboptimal one, the corresponding quadratic form is strictly positive. As a result, the fixed point $\mathbf{q}^*$ is locally asymptotically stable along the center-stable manifold.

Theorem 2 (Stability of IGM-Consistent Fixed Points). *Suppose the following conditions hold:*

(H1) *For every $\mathbf{q} \in \mathcal{M}$, each agent's greedy action $\arg \max_{a_i} Q_i(a_i)$ is unique.*

(H2) *The Jacobian of the mixing function f_{mix} has full rank on the normal subspace of $\mathcal{M}$ (local non-degeneracy).*

(H3) *The softmax temperature $\tau > 0$ is sufficiently small such that the induced policy is dominated by greedy actions.*

Then for any zero-loss fixed point $\mathbf{q}^ \in \mathcal{M}$ that satisfies IGM consistency (i.e., $\mathbf{g}(\mathbf{q}^*) = \mathbf{u}^*$), the Hessian $H_\tau(\mathbf{q}^*)$ is positive definite on the normal subspace $(T_{\mathbf{q}^*}\mathcal{M})^\perp$.*

Proof. We determine the spectral properties of the Hessian $H_\tau(\mathbf{q}^*)$ by analyzing the sign of its quadratic form $\mathbf{v}^\top H_\tau(\mathbf{q}^*)\mathbf{v}$ along critical directions. The full proof is provided in Appendix A.

We aim to show that $H_\tau(\mathbf{q}^*)$ is positive definite on the normal subspace $(T_{\mathbf{q}^*}\mathcal{M})^\perp$. Consider an arbitrary perturbation direction $\mathbf{v}$ in this subspace, such as $\mathbf{v} = \mathbf{e}_{iu'_i} - \mathbf{e}_{iu_i^*}$, where $u'_i \neq u_i^*$. This perturbation attempts to flip agent i 's greedy action from the optimal u_i^* to a suboptimal alternative u'_i .

On the zero-loss manifold $\mathcal{M}$, the quadratic form of the Hessian in this direction can be computed explicitly as:

$$\begin{aligned} \mathbf{v}^\top H_\tau(\mathbf{q}^*)\mathbf{v} &= \frac{1}{\tau} \sum_{\mathbf{a}} \mu_\tau(\mathbf{a} \mid \mathbf{q}^*) \left[\left(\mathbf{1}[a_i = u'_i] - \pi_i^\tau(u'_i) \right) \right. \\ &\quad \left. - \left(\mathbf{1}[a_i = u_i^*] - \pi_i^\tau(u_i^*) \right) \right]^2. \end{aligned} \quad (20)$$

In the low-temperature limit $\tau \rightarrow 0$, the policy $\mu_\tau(\mathbf{a} \mid \mathbf{q}^*)$ concentrates its mass on the greedy joint action $\mathbf{u}^*$. For this dominant term $\mathbf{a} = \mathbf{u}^*$, we have $\pi_i^\tau(u'_i) \rightarrow 0$, $\pi_i^\tau(u_i^*) \rightarrow 1$, and the indicator functions yield $\mathbf{1}[a_i = u'_i] = 0$, $\mathbf{1}[a_i = u_i^*] = 1$. The expression inside the brackets thus converges to a nonzero constant. Since the quadratic form is a weighted sum of squared terms with the dominant contribution strictly positive, we conclude that $\mathbf{v}^\top H_\tau(\mathbf{q}^*)\mathbf{v} > 0$. Therefore, $H_\tau(\mathbf{q}^*)$ is positive definite in all directions that perturb the optimal greedy action, ensuring that $\mathbf{q}^*$ is asymptotically stable. ■

Theorem 3 (Instability of IGM-Inconsistent Fixed Points). *Under the same assumptions (H1) – (H3) as in Theorem 2, consider a zero-loss fixed point $\mathbf{q}^* \in \mathcal{M}$ that violates IGM consistency (i.e., $\mathbf{g}(\mathbf{q}^*) \neq \mathbf{u}^*$). Then there exists an agent i and a perturbation direction*

$$\mathbf{v} = \mathbf{e}_{i,u_i^*} - \mathbf{e}_{i,g_i(\mathbf{q}^*)} \quad (21)$$

such that

$$\mathbf{v}^\top H_\tau(\mathbf{q}^*)\mathbf{v} \approx -\frac{2}{\tau} [y(\mathbf{u}^*) - y(\mathbf{g}(\mathbf{q}^*))] < 0, \quad (22)$$

indicating that $H_\tau(\mathbf{q}^)$ has a negative eigenvalue, and $\mathbf{q}^*$ is a structurally unstable saddle point.*

Proof. If $\mathbf{g}(\mathbf{q}^*) \neq \mathbf{u}^*$, we aim to show that $H_\tau(\mathbf{q}^*)$ admits at least one negative eigenvalue. Let $g_i = g_i(\mathbf{q}^*)$ denote the suboptimal greedy action selected by agent i . We construct a perturbation direction intended to "correct" this suboptimal choice:

$$\mathbf{v} = \mathbf{e}_{iu_i^*} - \mathbf{e}_{ig_i}. \quad (23)$$

This perturbation attempts to redirect the greedy policy toward an action with potentially higher reward.

In the low-temperature regime, the softmax policy induces a gradient amplification effect that makes the Hessian’s curvature primarily governed by reward differences. It can be shown that the quadratic form in this direction approximately satisfies:

$$\mathbf{v}^\top H_\tau(\mathbf{q}^*)\mathbf{v} \approx -\frac{C}{\tau} [y(\mathbf{g}') - y(\mathbf{g}(\mathbf{q}^*))], \quad (24)$$

where $C > 0$ is a constant, and $\mathbf{g}'$ is the new greedy joint action induced by the perturbation. Since $\mathbf{g}(\mathbf{q}^*)$ is not globally optimal, there always exists a local improving direction such that $y(\mathbf{g}') > y(\mathbf{g}(\mathbf{q}^*))$. This implies that the quadratic form is strictly negative in the direction $\mathbf{v}$, i.e., $\mathbf{v}^\top H_\tau(\mathbf{q}^*)\mathbf{v} < 0$, thereby confirming the existence of a negative eigenvalue in the Hessian. Consequently, the fixed point $\mathbf{q}^*$ is a saddle point. ■

Combining Theorems 2 and 3, we conclude that the set of IGM-consistent zero-loss solutions forms the unique center-stable manifold of the learning dynamics, while any IGM-inconsistent zero-loss point is a saddle point and hence unstable. Under continued exploration, the system is guaranteed to escape from these unstable saddle points and ultimately converge to the stable submanifold:

$$\mathcal{M}_{\text{IGM}} = \{\mathbf{q} \in \mathcal{M} \mid \mathbf{g}(\mathbf{q}) = \mathbf{u}^*\}. \quad (25)$$

4 Methodology

In this section, we investigate a non-monotonic variant of the QMIX algorithm and its two extensions designed to address the challenges introduced by removing the monotonicity constraint. Additional implementation details, including hyperparameters and pseudocode, are provided in Appendix B.

4.1 Non-Monotonic Mixing Function

We investigate a non-monotonic variant of the QMIX algorithm. Standard QMIX imposes a monotonicity constraint on the mixing network, requiring that the joint action-value function Q_{tot} be non-decreasing with respect to each individual agent’s utility Q_i . Formally, $\frac{\partial Q_{\text{tot}}}{\partial Q_i} \geq 0$, which is enforced by constraining the mixing network weights to be non-negative. While this constraint guarantees IGM consistency by design, it restricts the expressive capacity of the model.

In contrast, our approach removes this structural constraint, allowing the mixing network to learn an arbitrary aggregation function of the individual values:

$$Q_{\text{tot}}(s, \mathbf{a}) = g_{\text{mix}}(\{Q_i(\eta_i, a_i)\}_{i=1}^n, s), \quad (26)$$

where s denotes the global state, η_i represents the local observation history of agent i , and g_{mix} is implemented by a feed-forward neural network. Apart from the removal of the monotonicity constraint, the architecture remains identical to QMIX.

Our central hypothesis is that under approximately greedy exploration policies, the learning dynamics are sufficient to drive the system towards IGM-consistent solutions, even in the absence of explicit architectural constraints. The network is trained end-to-end by minimizing the joint Bellman error:

$$L(\theta) = \mathbb{E}_{s, \mathbf{a}, r, s'} [(y^{\text{tot}} - Q_{\text{tot}}(s, \mathbf{a}; \theta))^2], \quad (27)$$

where the target is computed using a target network,

$$y^{\text{tot}} = r + \gamma \max_{\mathbf{a}'} Q_{\text{tot}}(s', \mathbf{a}'; \theta^-). \quad (28)$$

4.2 SARSA-Style Updating with TD(λ)

Removing the monotonicity constraint introduces a potential issue: during the learning process, IGM consistency can no longer be guaranteed prior to convergence. In standard Q-learning, this may lead to undesirable effects because the target

$$y_Q^{\text{tot}} = r + \gamma \max_{\mathbf{a}'} Q_{\text{tot}}(s', \mathbf{a}'; \theta^-), \quad (29)$$

which relies on the $\max_{\mathbf{a}'}$ operator, is no longer a reliable training signal in the absence of IGM.

To address this issue, we adopt a SARSA-style update rule that removes the problematic max operator. Instead of computing the target based on the maximum over all possible next actions, we use the joint action $\mathbf{a}'$ that was actually sampled from the replay buffer:

$$y_{\text{SARSA}}^{\text{tot}} = r + \gamma Q_{\text{tot}}(s', \mathbf{a}'; \theta^-), \quad (30)$$

where $\mathbf{a}'$ is the next joint action executed by the behavior policy. This formulation aligns the training signal with the policy’s actual behavior and mitigates the bias caused by the max operator. We then incorporate the TD(λ) algorithm:

$$y_\lambda^{\text{tot}} = (1 - \lambda) \sum_{n=1}^{\infty} \lambda^{n-1} y_{(n)}^{\text{tot}}, \quad (31)$$

where $y_{(n)}^{\text{tot}}$ denotes the n -step return. By averaging over multiple step returns, TD(λ) smooths the learning signal, mitigates variance, and improves credit assignment over longer horizons, which is particularly beneficial for non-monotonic value decomposition.

It is worth noting that SARSA-type updates are theoretically on-policy and would typically require off-policy corrections, such as importance sampling. However, prior work by (author?) [20] has shown that, in the context of deep RL, applying such corrections to SARSA often degrades performance. Therefore, we omit off-policy corrections in our implementation.

4.3 Intrinsic Reward Driven Exploration-Exploitation

In our preliminary experiments, we observed that simply increasing the exploration rate in the original QMIX algorithm provided no benefit. In contrast, for our non-monotonic variant of QMIX, a higher degree of exploration consistently improved performance. Motivated by this observation, we integrate a curiosity-driven exploration mechanism based on Random Network Distillation (RND)[21]. RND encourages exploration by assigning intrinsic rewards for visiting novel states. It employs two neural networks: (i) a fixed, randomly initialized target network that maps states to feature embeddings, and (ii) a predictor network trained to approximate the target network’s output for visited states. The prediction error of the predictor network then serves as the intrinsic reward signal. The total reward used to train the value function is given by:

$$r = r_{\text{ext}} + \beta \cdot r_{\text{int}}, \quad (32)$$

where r_{ext} is the extrinsic team reward from the environment, r_{int} is the intrinsic curiosity reward generated by RND. This curiosity-driven mechanism serves as a more sophisticated alternative to ϵ -greedy exploration and allows us to investigate the role of exploration in non-monotonic value factorization.

5 Experiments

We evaluate our approach on three representative benchmarks: one-step matrix games, the StarCraft Multi-Agent Challenge (SMAC) [22], and Google Research Football (GRF) [23]. We compare our method against three strong baselines: QMIX, QPLEX, and QTRAN. These baselines were selected

to cover the major paradigms in value factorization: a strictly monotonic approach (QMIX), a refined architecture that preserves a relaxed form of monotonicity (QPLEX), and a non-monotonic method based on an alternative factorization principle (QTRAN). Complete experimental results, along with additional ablation studies, are provided in Appendix C.

$u_1 \backslash u_2$	A	B	C
A	12	-12	-12
B	-12	0	0
C	-12	0	0

(a) Payoff of Game A

$Q_1 \backslash Q_2$	A(7.4)	B(-4.0)	C(-4.1)
A(8.1)	12.0	-12.0	-12.0
B(-4.3)	-12.0	0.0	0.0
C(-4.3)	-12.0	0.0	0.0

(b) Our method on Game A

$Q_1 \backslash Q_2$	A(-5.2)	B(0.1)	C(0.1)
A(-5.3)	-12.0	-12.0	-12.0
B(0.1)	-12.0	0.0	-0.1
C(0.1)	-12.0	-0.0	-0.0

(c) QMIX on Game A

$u_1 \backslash u_2$	A	B	C
A	12	0	10
B	0	0	10
C	10	10	10

(d) Payoff of Game B

$Q_1 \backslash Q_2$	A(6.0)	B(-3.7)	C(2.5)
A(6.2)	12.0	0.0	10.0
B(-4.4)	0.0	0.0	10.0
C(1.7)	10.0	10.0	10.0

(e) Our method on Game B

$Q_1 \backslash Q_2$	A(-0.6)	B(-0.6)	C(0.5)
A(-0.6)	0.1	0.1	4.1
B(-0.6)	0.1	0.1	4.1
C(0.5)	5.6	5.6	10.0

(f) QMIX on Game B

Table 1: True payoff matrices and estimated value functions for two matrix games. Each row corresponds to one game: the left column shows the ground-truth payoff matrix, the middle column shows value estimates from our non-monotonic QMIX variant, and the right column shows estimates from standard QMIX. Top row: Game A. Bottom row: Game B.

5.1 Matrix Game

We begin by evaluating non-monotonic QMIX with SARSA-style updates on two one-step matrix games, which are adapted from [14]. Among them, Game B is more challenging than Game A, as it contains stronger local optima that are more difficult to escape. In addition, unlike previous studies on matrix games that employ uniformly random policies, we adopt an ϵ -greedy strategy, which is consistent with our theoretical analysis.

The results, summarized in Table 3, show that our method successfully learns the correct global payoff matrix. In contrast, QMIX fails to recover the true payoffs. These findings provide initial empirical evidence that, under appropriate exploration-exploitation mechanisms, removing the monotonicity constraint enables the algorithm to accurately recover IGM-optimal solutions in non-monotonic settings.

5.2 SMAC and GRF

In order to validate our propositions, we implemented RND to generate intrinsic exploration rewards for non-monotonic QMIX with SARSA-style updates, and conducted experiments in complex, high-dimensional SMAC and GRF environments. As shown in Figures 1 and 2, our method demonstrates clear advantages over the baselines across a variety of maps. QTRAN, despite its success in the matrix game, performs poorly in these complex environments. Similarly, QMIX and QPLEX struggle to find optimal solutions in these non-monotonic sequential decision-making problems due to their inherent structural constraints. Further ablation studies on additional variants of our method are provided in Appendix C.

5.3 Discussion

The empirical results provide strong evidence supporting our theoretical analysis. The observed learning dynamics closely align with our model of stable and unstable equilibria. Unlike QTRAN, whose projection mechanism can restrict the use of non-optimal global actions and thus impede learning, our unconstrained approach leverages broad exploration. This enables the value function to be estimated across a wider range of joint actions, mitigating the risk of premature convergence to suboptimal solutions caused by relative overgeneralization.

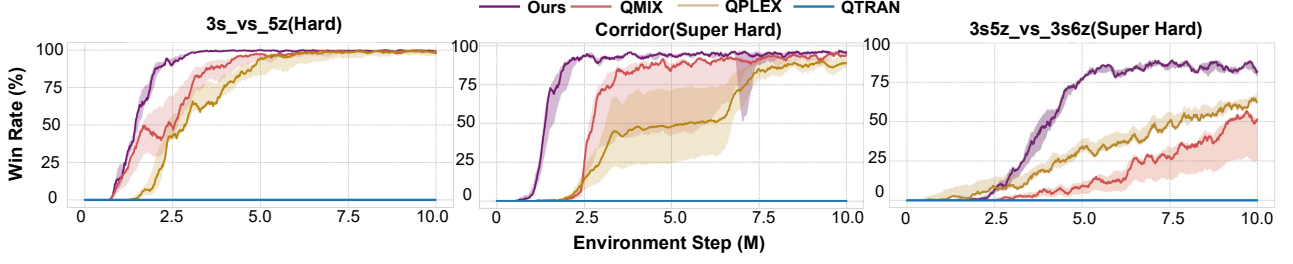


Figure 1: Comparisons of test win rate on SMAC maps: `3s_vs_5z`, `corridor`, and `3s5z_vs_3s6z`. The results are averaged over five independent runs, with the 25%–75% interquartile range shown as a shaded region.

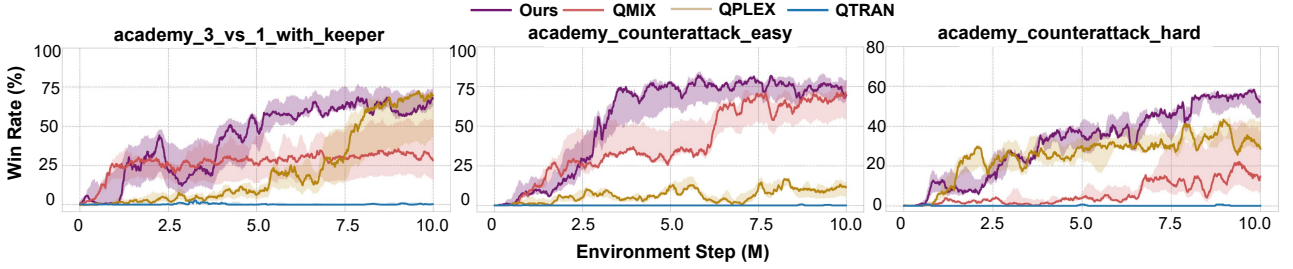


Figure 2: Comparisons of test win rate on GRF tasks: `academy_3_vs_1_keeper`, `academy_counterattack_easy`, and `academy_counterattack_hard`. The results are averaged over five independent runs.

We initially expected non-monotonic QMIX to perform comparably to the original QMIX. Surprisingly, however, on several challenging SMAC tasks, our method not only consistently outperforms the original QMIX but also converges significantly faster. We hypothesize that this advantage arises from the removal of the monotonicity constraint, which substantially enhances the expressiveness of the mixing network. Under an appropriate exploration mechanism, this increased flexibility allows non-monotonic QMIX to more effectively discover optimal policies. In addition, both QPLEX and QTRAN enhance the representational capacity of the value function through specialized network architectures and additional soft constraints. However, on the challenging tasks we evaluated, neither method demonstrated superior performance. We hypothesize that the increased algorithmic complexity of these approaches may reduce their robustness and heighten sensitivity to hyperparameter choices, thereby limiting their practical effectiveness compared to our simpler unconstrained formulation.

It is worth noting that in the GRF tasks, our method exhibits a slower reward increase during the early stage of training, followed by a sharp improvement in the later stage. We attribute this behavior to the existence of multiple zero-loss solutions in the initial learning dynamics, including suboptimal ones. During this exploratory phase, the algorithm navigates the policy space and transiently approaches unstable saddle points. The system eventually escapes these unstable regions and converges to the stable, IGM-consistent submanifold, resulting in a rapid performance surge and superior final results.

6 Related Work

Value Function Factorization (VFF) is a central branch of the CTDE paradigm in multi-agent reinforcement learning. VFF methods seek to learn a global joint action-value function Q_{tot} and decompose it into per-agent value functions Q_i . This factorized structure is designed to ensure that optimizing each local utility Q_i leads to the optimization of the global objective Q_{tot} , while enabling decentralized execution based solely on local observations.

One of the earliest VFF methods, Value-Decomposition Networks (VDN)[10], assumes that the

global joint action-value function is the sum of individual agent values. This additive assumption simplifies credit assignment but severely limits representational capacity, failing to capture nonlinear agent interactions. QMIX [13] extends VDN by introducing a nonlinear mixing network, thereby enhancing expressiveness. QMIX imposes a constraint that requires Q_{tot} to be a monotonically non-decreasing function with respect to each individual Q_i . Although more expressive than VDN, this constraint still restricts the representation of tasks with inherently non-monotonic value functions.

To overcome the expressiveness limitations imposed by the monotonicity constraint, subsequent research has explored two main directions. The first focuses on enhancing the mixing network. WQMIX [24] introduces a weighting mechanism in the projection step of monotonic value decomposition, emphasizing more promising joint actions and biasing learning toward better solutions. Qatten [25] applies multi-head attention to adaptively learn interaction weights among agents, enabling a more flexible nonlinear combination.

Another line aims to develop more general frameworks capable of representing the full class of IGM functions. QTRAN [14] introduces a transformed surrogate value function, reformulating the IGM condition as a set of linear constraints and incorporating them via soft regularization. While QTRAN is theoretically more expressive, it often suffers from instability during training and performs suboptimally in practice. QPLEX [15] introduces a duplex dueling architecture, which factorizes Q_{tot} into a state-value term $V(s)$ and a joint advantage function $A(s, \mathbf{u})$. The IGM constraint is enforced by imposing monotonicity only on the advantage component. QPLEX theoretically covers the complete IGM function class and achieves strong empirical performance, though at the cost of increased architectural complexity.

Current research in VFF largely focuses on designing increasingly complex network architectures and decomposition schemes to enhance representational capacity, aiming to capture a broader range of tasks while preserving IGM guarantees. In contrast, our work offers a different perspective: we argue that under standard approximately greedy exploration strategies, the learning dynamics may exhibit an implicit stabilizing mechanism. This mechanism can naturally guide unconstrained learning process toward IGM-consistent optimal solutions.

7 Conclusion

This paper challenges the prevailing assumption that structural monotonicity is necessary for multi-agent value decomposition to ensure IGM consistency. We model the non-monotonic learning process as a continuous-time gradient flow and theoretically demonstrate that, under approximately greedy exploration, the dynamics themselves provide an implicit self-correction mechanism. This mechanism drives IGM-inconsistent solutions to unstable saddle points, while establishing IGM-consistent solutions as stable attractors. Experiments on synthetic matrix games, SMAC, and GRF empirically validate our theory, showing that removing monotonicity constraints not only reliably recovers optimal solutions but also consistently outperforms monotonic baselines. Although our formal analysis is developed in the single-state setting, these findings suggest that leveraging the natural dynamics of learning rather than imposing rigid architectural constraints offers a promising direction for designing more expressive, and more effective value-based MARL algorithms.

Acknowledgment

This work was supported in part by the UKRI Future Leaders Fellowship under Grant MR/S017062/1 and MR/X011135/1; in part by NSFC under Grant 62376056 and 62076056; in part by the Royal Society under Grant IES/R2/212077; in part by the EPSRC under Grant 2404317; in part by the Kan Tong Po Fellowship (KTP\R1\231017); and in part by the Amazon Research Award and Alan Turing Fellowship.

References

- [1] P. Long, T. Fan, X. Liao, W. Liu, H. Zhang, and J. Pan, “Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning,” in *Proc. IEEE International Conference on Robotics and Automation*, 2018, pp. 6252–6259.
- [2] J. Duan, W. Wang, L. Xiao, J. Gao, S. E. Li, C. Liu, Y.-Q. Zhang, B. Cheng, and K. Li, “Distributional soft actor-critic with three refinements,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 47, no. 5, pp. 3935–3946, 2025.
- [3] T. Chu, J. Wang, L. Codecà, and Z. Li, “Multi-agent deep reinforcement learning for large-scale traffic signal control,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1086–1095, 2019.
- [4] Q. Li, Z. Peng, L. Feng, Q. Zhang, Z. Xue, and B. Zhou, “Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3461–3475, 2022.
- [5] R. Zhang, J. Hou, F. Walter, S. Gu, J. Guan, F. Röhrbein, Y. Du, P. Cai, G. Chen, and A. Knoll, “Multi-agent reinforcement learning for autonomous driving: A survey,” *arXiv preprint arXiv:2408.09675*, 2024.
- [6] L. Chen, P. Wu, K. Chitta, B. Jaeger, A. Geiger, and H. Li, “End-to-end autonomous driving: Challenges and frontiers,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 10 164–10 183, 2024.
- [7] M. Roesch, C. Linder, R. Zimmermann, A. Rudolf, A. Hohmann, and G. Reinhart, “Smart grid for industry using multi-agent reinforcement learning,” *Applied Sciences*, vol. 10, no. 19, p. 6900, 2020.
- [8] H.-M. Chung, S. Maharjan, Y. Zhang, and F. Eliassen, “Distributed deep reinforcement learning for intelligent load scheduling in residential smart grids,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2752–2763, 2020.
- [9] M. Tan, “Multi-agent reinforcement learning: Independent vs. cooperative agents,” in *Proc. International Conference on Machine Learning*, 1993, pp. 330–337.
- [10] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls *et al.*, “Value-decomposition networks for cooperative multi-agent learning based on team reward,” in *Proc. International Conference on Autonomous Agents and MultiAgent Systems*, 2018, pp. 2085–2087.
- [11] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, “Multi-agent actor-critic for mixed cooperative-competitive environments,” in *Proc. Advances in Neural Information Processing Systems*, 2017, pp. 6382–6393.
- [12] T. Hu, B. Luo, C. Yang, and T. Huang, “Mo-mix: Multi-objective multi-agent cooperative decision-making with deep reinforcement learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 10, pp. 12 098–12 112, 2023.
- [13] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson, “Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning,” in *Proc. International Conference on Machine Learning*, 2018, pp. 4295–4304.
- [14] K. Son, D. Kim, W. J. Kang, D. E. Hostallero, and Y. Yi, “Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning,” in *Proc. International Conference on Machine Learning*, 2019, pp. 5887–5896.

- [15] J. Wang, Z. Ren, T. Liu, Y. Yu, and C. Zhang, “Qplex: Duplex dueling multi-agent q-learning,” in *Proc. International Conference on Learning Representations*, 2021, pp. 1–11.
- [16] A. Mahajan, T. Rashid, M. Samvelyan, and S. Whiteson, “Maven: Multi-agent variational exploration,” *Advances in Neural Information Processing Systems*, vol. 32, no. 1, pp. 7613–7624, 2019.
- [17] T. Gupta, A. Mahajan, B. Peng, W. Böhmer, and S. Whiteson, “Uneven: Universal value exploration for multi-agent reinforcement learning,” in *Proc. International Conference on Machine Learning*. PMLR, 2021, pp. 3930–3941.
- [18] G. A. Rummery and M. Niranjan, *On-line Q-learning using connectionist systems*. University of Cambridge, Department of Engineering Cambridge, UK, 1994, vol. 37.
- [19] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein, “The complexity of decentralized control of markov decision processes,” *Mathematics of operations research*, vol. 27, no. 4, pp. 819–840, 2002.
- [20] J. F. Hernandez-Garcia and R. S. Sutton, “Understanding multi-step deep reinforcement learning: A systematic study of the dqn target,” *arXiv preprint arXiv:1901.07510*, 2019.
- [21] Y. Burda, H. Edwards, A. Storkey, and O. Klimov, “Exploration by random network distillation,” *arXiv preprint arXiv:1810.12894*, 2018.
- [22] S. Whiteson, M. Samvelyan, T. Rashid, C. De Witt, G. Farquhar, N. Nardelli, T. Rudner, C. Hung, P. Torr, and J. Foerster, “The StarCraft multi-agent challenge,” in *Proc. International Conference on Autonomous Agents and MultiAgent Systems*, 2019, pp. 2186–2188.
- [23] K. Kurach, A. Raichuk, P. Stańczyk *et al.*, “Google research football: A novel reinforcement learning environment,” in *Proc. AAAI conference on artificial intelligence*, vol. 34, no. 04, 2020, pp. 4501–4510.
- [24] T. Rashid, G. Farquhar, B. Peng, and S. Whiteson, “Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 10 199–10 210, 2020.
- [25] Y. Yang, J. Hao, B. Liao, K. Shao, G. Chen, W. Liu, and H. Tang, “Qatten: A general framework for cooperative multiagent reinforcement learning,” *arXiv preprint arXiv:2002.03939*, 2020.

A Proofs

A.1 Proof of Lemma 2

Lemma 2. Let $\{\tau_k\}_{k=1}^\infty$ be a sequence of temperatures such that $\tau_k \downarrow 0$. Suppose that at parameter point $\mathbf{q}$, each agent has a unique greedy action and the mixing function f_{mix} satisfies the local non-degeneracy condition. Then,

$$\lim_{k \rightarrow \infty} \nabla_{\mathbf{q}} \mathcal{L}_{\tau_k}(\mathbf{q}) \in \partial_{\mathbf{q}}^{\mathcal{C}} \mathcal{L}_0(\mathbf{q}), \quad (33)$$

where $\partial^{\mathcal{C}}$ denotes the Clarke generalized gradient, and $\mathcal{L}_0$ corresponds to the limiting loss as $\tau \rightarrow 0$ in the softmax policy μ_τ .

Proof. For any fixed $\tau > 0$, define the policy of agent i as

$$\pi_i^\tau(a_i; \mathbf{q}) = \frac{\exp(Q_i(a_i)/\tau)}{\sum_b \exp(Q_i(b)/\tau)}. \quad (34)$$

π_i^τ is infinitely differentiable with respect to $\mathbf{q}$. Consequently, the joint policy

$$\mu_\tau(\mathbf{a}; \mathbf{q}) = \prod_i \pi_i^\tau(a_i; \mathbf{q}), \quad (35)$$

and the corresponding loss function

$$\mathcal{L}_\tau(\mathbf{q}) = \sum_{\mathbf{a}} \mu_\tau(\mathbf{a}; \mathbf{q}) (y - Q_{\text{tot}})^2 \quad (36)$$

are both smooth and differentiable.

In the limit $\tau \rightarrow 0$, since each agent's greedy action $g_i(\mathbf{q})$ is unique, we have

$$\pi_i^\tau(g_i(\mathbf{q}); \mathbf{q}) \rightarrow 1, \quad \pi_i^\tau(a_i; \mathbf{q}) \rightarrow 0 \quad (a_i \neq g_i(\mathbf{q})). \quad (37)$$

Hence, $\mu_\tau(\mathbf{a}; \mathbf{q})$ converges pointwise to the deterministic greedy policy.

The Clarke generalized gradient $\partial^{\mathcal{C}} f(\mathbf{q})$ of a non-smooth function f is defined as the convex hull of all limit points of gradients of smooth approximations converging to f :

$$\partial^{\mathcal{C}} f(\mathbf{q}) = \text{conv} \left\{ \lim_{k \rightarrow \infty} \nabla f_k(\mathbf{q}) \mid f_k \rightarrow f, f_k \text{ smooth} \right\}. \quad (38)$$

If a sequence of smooth functions $\{f_k\}$ converges pointwise to f and has uniformly bounded Lipschitz constants, then

$$\lim_{k \rightarrow \infty} \nabla f_k(\mathbf{q}) \in \partial^{\mathcal{C}} f(\mathbf{q}). \quad (39)$$

This property directly connects the gradients of smooth approximations to the Clarke generalized gradient of the limiting non-smooth function.

We now apply this result to our problem. Define $f_k(\mathbf{q}) = \mathcal{L}_{\tau_k}(\mathbf{q})$, where $\tau_k \downarrow 0$ and the Lipschitz constants of $\{\mathcal{L}_{\tau_k}\}$ remain uniformly bounded. Clearly, f_k converges pointwise to

$$f_0(\mathbf{q}) = \sum_{\mathbf{a}} \mu_{\text{greedy}}(\mathbf{a}; \mathbf{q}) (y - Q_{\text{tot}})^2, \quad (40)$$

which corresponds precisely to the greedy policy loss. By the limit transfer theorem (Rockafellar and Wets, *Variational Analysis*, Theorem 9.13), we obtain

$$\lim_{k \rightarrow \infty} \nabla f_k(\mathbf{q}) = \lim_{k \rightarrow \infty} \nabla_{\mathbf{q}} \mathcal{L}_{\tau_k}(\mathbf{q}) \in \partial^{\mathcal{C}} f_0(\mathbf{q}). \quad (41)$$

This completes the proof. ■

A.2 Proof of Theorem 2

Theorem 2 (Stability of IGM-Consistent Fixed Points). *Suppose the following conditions hold:*

(H1) *For every $\mathbf{q} \in \mathcal{M}$, each agent's greedy action $\arg \max_{a_i} Q_i(a_i)$ is unique.*

(H2) *The Jacobian of the mixing function f_{mix} has full rank on the normal subspace of $\mathcal{M}$ (local non-degeneracy).*

(H3) *The softmax temperature $\tau > 0$ is sufficiently small such that the induced policy is dominated by greedy actions.*

Then for any zero-loss fixed point $\mathbf{q}^ \in \mathcal{M}$ that satisfies IGM consistency (i.e., $\mathbf{g}(\mathbf{q}^*) = \mathbf{u}^*$), the Hessian $H_\tau(\mathbf{q}^*)$ is positive definite on the normal subspace $(T_{\mathbf{q}^*}\mathcal{M})^\perp$.*

Proof. To establish the local stability of IGM-consistent zero-loss points, we demonstrate that for any perturbation direction $\mathbf{v}$ attempting to change the greedy action from $\mathbf{u}^*$ to some suboptimal action $\mathbf{g}'$, the quadratic form

$$\mathbf{v}^\top H_\tau(\mathbf{q}^*) \mathbf{v} \quad (42)$$

is strictly positive.

Without loss of generality, consider agent 1 and a suboptimal action $u'_1 \neq u_1^*$. Define the perturbation direction as

$$\mathbf{v} = \mathbf{e}_{1u'_1} - \mathbf{e}_{1u_1^*}, \quad (43)$$

and examine the perturbed parameter vector

$$\mathbf{q}(\epsilon) = \mathbf{q}^* + \epsilon \mathbf{v}, \quad (44)$$

for sufficiently small $\epsilon > 0$.

Since $\mathbf{q}^*$ is a stationary point with zero gradient, a second-order Taylor expansion of the loss function $\mathcal{L}_\tau$ around $\mathbf{q}^*$ yields

$$\mathcal{L}_\tau(\mathbf{q}(\epsilon)) = \frac{\epsilon^2}{2} \mathbf{v}^\top H_\tau(\mathbf{q}^*) \mathbf{v} + \mathcal{O}(\epsilon^3). \quad (45)$$

Thus, proving local stability reduces to showing that $\mathbf{v}^\top H_\tau(\mathbf{q}^*) \mathbf{v} > 0$, which would imply the perturbed loss increases quadratically with respect to ϵ along any direction that attempts to replace the greedy action with a suboptimal one.

Under the low-temperature limit ($\tau \rightarrow 0$), the perturbed greedy action of agent 1 switches from u_1^* to u'_1 , leading to a new joint greedy action $\mathbf{g}' = (u'_1, u_2^*, \dots, u_N^*)$. The loss is then dominated by the error associated with $\mathbf{g}'$:

$$\mathcal{L}_\tau(\mathbf{q}(\epsilon)) \approx [y(\mathbf{g}') - Q_{\text{tot}}(\mathbf{g}'; \mathbf{q}(\epsilon))]^2. \quad (46)$$

Since $\mathbf{q}^* \in \mathcal{M}$, we have $Q_{\text{tot}}(\mathbf{g}'; \mathbf{q}^*) = y(\mathbf{g}')$. Using a first-order Taylor expansion around $\mathbf{q}^*$, we obtain

$$Q_{\text{tot}}(\mathbf{g}'; \mathbf{q}(\epsilon)) \approx y(\mathbf{g}') + \epsilon \cdot \nabla_{\mathbf{q}} Q_{\text{tot}}(\mathbf{g}'; \mathbf{q}^*)^\top \mathbf{v}. \quad (47)$$

Substituting this approximation into the loss expression, we get

$$\mathcal{L}_\tau(\mathbf{q}(\epsilon)) \approx \epsilon^2 \left(\nabla_{\mathbf{q}} Q_{\text{tot}}(\mathbf{g}'; \mathbf{q}^*)^\top \mathbf{v} \right)^2. \quad (48)$$

Thus, by definition of the Hessian, we have

$$\mathbf{v}^\top H_\tau(\mathbf{q}^*) \mathbf{v} = 2 \left(\nabla_{\mathbf{q}} Q_{\text{tot}}(\mathbf{g}'; \mathbf{q}^*)^\top \mathbf{v} \right)^2. \quad (49)$$

Finally, by assumption (H2), the local non-degeneracy condition implies

$$\nabla_{\mathbf{q}} Q_{\text{tot}}(\mathbf{g}'; \mathbf{q}^*)^\top \mathbf{v} \neq 0. \quad (50)$$

Hence, we conclude

$$\mathbf{v}^\top H_\tau(\mathbf{q}^*) \mathbf{v} > 0, \quad (51)$$

proving the strict positive definiteness of the Hessian along any action-switching direction, and thereby establishing the local stability of the IGM-consistent fixed point $\mathbf{q}^*$. ■

A.3 Proof of Theorem 3

Theorem 3 (Instability of IGM-Inconsistent Fixed Points). *Under the same assumptions (H1) – (H3) as in Theorem 3, consider a zero-loss fixed point $\mathbf{q}^* \in \mathcal{M}$ that violates IGM consistency (i.e., $\mathbf{g}(\mathbf{q}^*) \neq \mathbf{u}^*$). Then there exists an agent i and a perturbation direction*

$$\mathbf{v} = \mathbf{e}_{i,u_i^*} - \mathbf{e}_{i,g_i(\mathbf{q}^*)} \quad (52)$$

such that

$$\mathbf{v}^\top H_\tau(\mathbf{q}^*) \mathbf{v} \approx -\frac{2}{\tau} [y(\mathbf{u}^*) - y(\mathbf{g}(\mathbf{q}^*))] < 0, \quad (53)$$

indicating that $H_\tau(\mathbf{q}^*)$ has a negative eigenvalue, and $\mathbf{q}^*$ is a structurally unstable saddle point.

Proof. To prove instability, it suffices to identify a perturbation direction $\mathbf{v}$ along which the Hessian exhibits negative curvature. Specifically, we aim to show that there exists at least one agent i for which perturbing the greedy action from the current suboptimal choice $g_i(\mathbf{q}^*)$ to the globally optimal choice u_i^* results in a strictly negative quadratic form:

$$\mathbf{v}^\top H_\tau(\mathbf{q}^*) \mathbf{v} < 0. \quad (54)$$

Since $\mathbf{g}(\mathbf{q}^*) \neq \mathbf{u}^*$, at least one agent’s greedy action must differ from the globally optimal action. Without loss of generality, consider agent i with $g_i(\mathbf{q}^*) \neq u_i^*$, and define the perturbation direction as

$$\mathbf{v} = \mathbf{e}_{i,u_i^*} - \mathbf{e}_{i,g_i(\mathbf{q}^*)}. \quad (55)$$

This perturbation attempts to adjust the greedy action of agent i toward its globally optimal action u_i^* , yielding a corrected joint action

$$\mathbf{g}' = (u_i^*, g_{-i}(\mathbf{q}^*)). \quad (56)$$

In the low-temperature limit ($\tau \rightarrow 0$), the loss curvature along this direction is dominated by the payoff difference induced by switching actions. Specifically, a second-order Taylor expansion around $\mathbf{q}^*$ gives

$$\mathbf{v}^\top H_\tau(\mathbf{q}^*) \mathbf{v} \approx -\frac{C}{\tau} [y(\mathbf{g}') - y(\mathbf{g}(\mathbf{q}^*))], \quad (57)$$

where $C > 0$ is a positive constant depending on the policy gradient terms of the softmax function.

Since $\mathbf{g}(\mathbf{q}^*)$ is not globally optimal, we have

$$y(\mathbf{g}') > y(\mathbf{g}(\mathbf{q}^*)), \quad (58)$$

implying

$$\mathbf{v}^\top H_\tau(\mathbf{q}^*) \mathbf{v} < 0. \quad (59)$$

Thus, there exists a perturbation direction along which the Hessian has negative curvature, demonstrating the existence of a negative eigenvalue. Equivalently, the corresponding Jacobian of the learning dynamics possesses a positive eigenvalue, establishing that the IGM-inconsistent fixed point $\mathbf{q}^*$ is a structurally unstable saddle point. ■

B Implementation Details

B.1 Model Architecture

Our model’s architecture, as illustrated in Figure 3, is based on the QMIX [13] framework and is composed of a set of decentralized agent networks and a centralized mixing network.

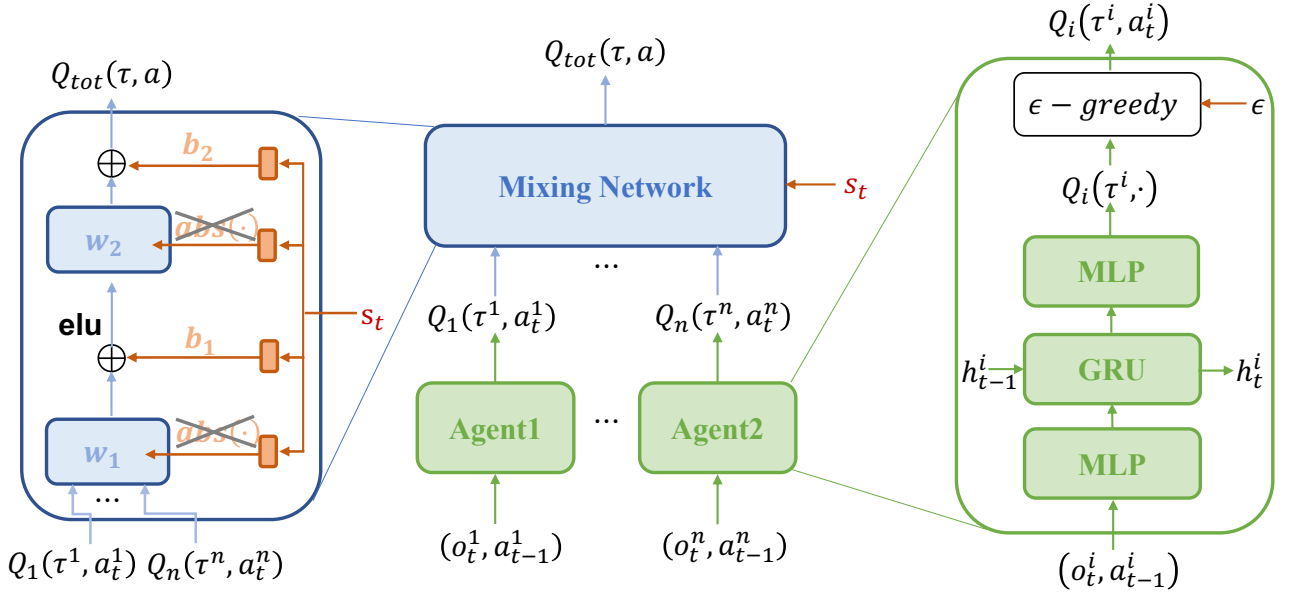


Figure 3: Architecture of our method.

Agent Networks Each agent possesses an individual recurrent network responsible for approximating its local action-value function, $Q_i(\tau^i, a_t^i)$. This network processes the agent’s current local observation (o_t^i) and its previous action (a_{t-1}^i). The input is first passed through a Multi-Layer Perceptron (MLP), and the output is then fed into a Gated Recurrent Unit (GRU) which also receives the previous hidden state h_{t-1}^i . The GRU’s new hidden state, h_t^i , which encodes the agent’s action-observation history τ^i , is then passed to a final MLP to produce the Q-values for all of the agent’s discrete actions. During execution, actions are chosen from these local Q-values using an ϵ -greedy policy.

Mixing Network The mixing network is a centralized module that combines the individual Q-values from all agent networks, $\{Q_1(\tau^1, a_t^1), \dots, Q_n(\tau^n, a_t^n)\}$, into a single global joint action-value, $Q_{tot}(\tau, a)$. This network also takes the global state s_t as an input to enable state-dependent mixing.

Hypernetwork Structure The mixing network utilizes a hypernetwork architecture. It consists of several linear layers that take the global state s_t as input and generate the weights (e.g., w_1, w_2) and biases (e.g., b_1, b_2) for the main mixing layers. The individual agent Q-values are then fed through these state-generated layers, with an ELU activation function applied between them, to compute the final Q_{tot} . For standard QMIX, the weights of the hypernetwork are constrained to be non-negative to enforce the monotonicity constraint $\frac{\partial Q_{tot}}{\partial Q_i} \geq 0$. Our non-monotonic approach, central to this paper, removes this constraint.

B.2 Intrinsic Reward Generation for Exploration

To facilitate robust exploration, especially in complex environments with sparse extrinsic rewards, we integrate an intrinsic curiosity mechanism based on Random Network Distillation (RND) [21]. As illustrated in Figure 4, the RND module consists of two distinct neural networks:

- A **target network** (g), which is initialized randomly and remains fixed throughout training. It takes the current state s_t and maps it to a feature embedding.
- A **predictor network** ($\hat{g}$), which is trained to predict the output of the target network for the same state s_t .

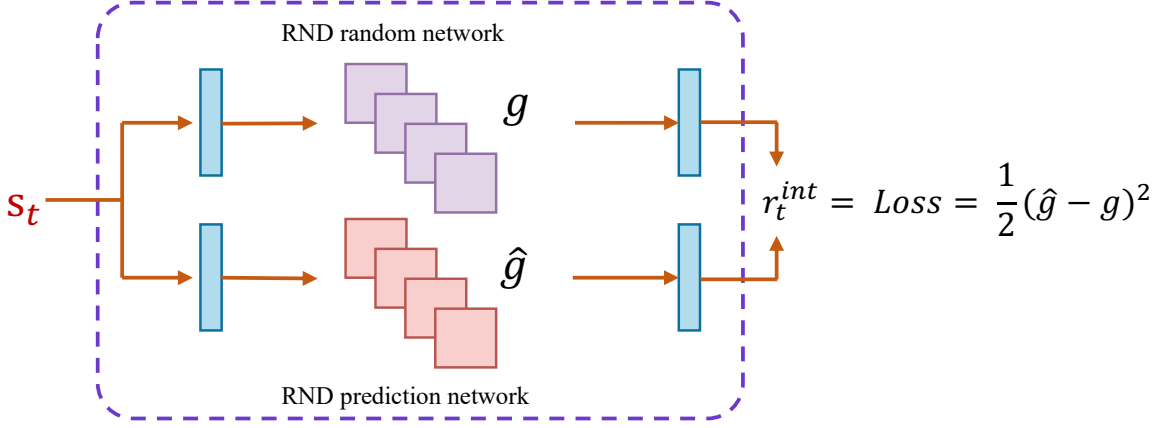


Figure 4: Architecture of RND.

The prediction error between the outputs of these two networks serves as the intrinsic reward signal. Specifically, the intrinsic reward r_t^{int} is calculated as the mean squared error between the feature embeddings:

$$r_t^{int} = \frac{1}{2} \|\hat{g}(s_t; \theta_{\text{predictor}}) - g(s_t)\|_2^2 \quad (60)$$

where $\theta_{\text{predictor}}$ denotes the parameters of the trainable predictor network. This reward is high for novel, unfamiliar states where the predictor network makes large errors, and low for familiar states that have been visited frequently. By adding this curiosity-driven reward to the extrinsic environment reward, we encourage the agents to systematically explore their state space and discover more promising regions of the environment.

B.3 Pseudocode

The complete training procedure of our proposed method is summarized in Algorithm 1. We first initialize all agent networks, the non-monotonic mixing network, and their corresponding target networks, which are cloned from the initial parameters. In parallel, we initialize the RND module, consisting of a fixed random target network and a trainable predictor network, as well as an empty replay buffer for experience storage.

During data collection, agents interact with the environment over full episodes. At each timestep, every agent selects an action according to its local action-value estimates using an ϵ -greedy exploration policy. The resulting joint action, extrinsic reward, and next state are then stored in the replay buffer as part of the trajectory.

The training phase is executed periodically. We sample a mini-batch of complete trajectories from the replay buffer. For each step within these trajectories, we first compute an intrinsic reward using the RND module’s prediction error. This is added to the extrinsic reward to form a total reward. Next, we calculate the target value for the Bellman update. Following our methodology, we use the TD(λ) return, which forms a robust target by calculating a weighted average of all n-step SARSA returns along the trajectory. The final loss is the mean squared error between these TD(λ) targets and the Q-values produced by the online mixing network. This loss is used to update the parameters of both the agent networks and the mixing network via gradient descent. The RND predictor network is separately updated to minimize its prediction error. Finally, the target networks θ^- are periodically updated by copying the parameters from the online networks θ .

B.4 Hyperparameters

The important parameters used in the experiment are presented in Table 12. Mini-batch size refers to the number of episodes sampled from the buffer for training. Annual start weight means that the initial value of β is 0.5 and reduces to 0.05 over 100k steps.

Algorithm 1: Training Procedure

Initialize: Agent networks $Q_i(\cdot; \theta_i)$ for $i = 1..N$, and mixing network $Q_{tot}(\cdot; \theta_{mix})$
Initialize: Target networks $\theta_i^-, \theta_{mix}^-$ with parameters copied from online networks
Initialize: RND target network g (fixed) and predictor network $\hat{g}(\cdot; \theta_{\text{predictor}})$
Initialize: Replay buffer $D \leftarrow \emptyset$

- 1: **for** episode = 1, ..., M **do**
- 2: Initialize environment and get initial state s_0
- 3: **for** $t = 0, \dots, T - 1$ **do**
- 4: **for** agent $i = 1, \dots, N$ **do**
- 5: Based on observation history τ_t^i , select action a_t^i using an ϵ -greedy policy on $Q_i(\tau_t^i, \cdot; \theta_i)$
- 6: **end for**
- 7: Execute joint action $\mathbf{a}_t$, observe extrinsic reward r_t , next state s_{t+1} , and terminated flag d_{t+1}
- 8: Store transition sequence in buffer D
- 9: **if** time to update **then**
- 10: Sample a random mini-batch of trajectories B from D
- 11: Initialize loss $\mathcal{L} \leftarrow 0$
- 12: For each trajectory $\tau_j = \{(s_t, \mathbf{a}_t, r_t, d_t)\}_{t=0..T}$ in B :
- 13: // First, calculate total rewards for the trajectory
- 14: For $t = 0, \dots, T$:
- 15: $r_{int,t} \leftarrow \|\hat{g}(s_{t+1}; \theta_{\text{predictor}}) - g(s_{t+1})\|_2^2$
- 16: $r_{total,t} \leftarrow r_t + \beta \cdot r_{int,t}$
- 17: For $t = 0, \dots, T$:
- 18: // Calculate n-step SARSA returns starting from time t
- 19: For $n = 1, \dots, T - t$:
- 20: $y_t^{(n)} \leftarrow \left(\sum_{k=0}^{n-1} \gamma^k r_{total,t+k} \right) + (1 - d_{t+n}) \gamma^n Q_{tot}(s_{t+n}, \mathbf{a}_{t+n}; \theta_{mix}^-, \theta_i^-)$
- 21: // Combine n-step returns to form the TD(λ) target
- 22: $y_t^\lambda \leftarrow (1 - \lambda) \sum_{n=1}^{T-t-1} \lambda^{n-1} y_t^{(n)} + \lambda^{T-t-1} y_t^{(T-t)}$
- 23: $Q_{tot,t} \leftarrow Q_{tot}(s_t, \mathbf{a}_t; \theta_{mix})$
- 24: $\mathcal{L} \leftarrow \mathcal{L} + (y_t^\lambda - Q_{tot,t})^2$
- 25: Update RND predictor network $\theta_{\text{predictor}}$ to minimize $\sum_{j,t} r_{int,t}$
- 26: Update agent networks θ_i and mixing network θ_{mix} by minimizing $\mathcal{L}$
- 27: Periodically update target networks: $\theta_i^- \leftarrow \theta_i$, $\theta_{mix}^- \leftarrow \theta_{mix}$
- 28: **end if**
- 29: **end for**
- 30: **end for**

Table 2: Hyperparameters

Hyperparameter	Value	Hyperparameter	Value
Optimizer	Adam	Buffer size	5000
Agent learning rate	1e-3	RND learning rate	5e-4
Mini-batch size	128	Anneal start weight	0.5
Anneal step for β	100K	Anneal finish weight	0.05
Use learning rate decay	False		

C Experimental Validation and Ablation Studies

In this section, we present additional experiments to further substantiate the theoretical results established in the main paper.

C.1 Matrix Game Results

We provide the full set of results from the matrix-game experiments summarized in the main text. Table 3 compares the learned value functions produced by our method, QTRAN [14], QMIX[13], and QPLEX[15] against the true payoff matrices across two distinct games.

As the results indicate, both our unconstrained method and QTRAN accurately recovered the global payoff matrices. In contrast, QMIX and QPLEX, which enforce monotonic constraints, failed to represent the true payoffs, especially in the non-monotonic Game A scenario. These findings strongly support our theoretical claim that removing monotonicity constraints enables reliable recovery of IGM-optimal solutions in cases where monotonic approaches inherently fall short.

$u_1 \backslash u_2$	A	B	C
A	12	-12	-12
B	-12	0	0
C	-12	0	0

(a) True payoff of matrix game A

$Q_1 \backslash Q_2$	A(7.4)	B(-4.0)	C(-4.1)
A(8.1)	12.0	-12.0	-12.0
B(-4.3)	-12.0	0.0	0.0
C(-4.3)	-12.0	0.0	0.0

(c) Our method on game A

$Q_1 \backslash Q_2$	A(-5.2)	B(0.1)	C(0.1)
A(-5.3)	-12.0	-12.0	-12.0
B(0.1)	-12.0	0.0	-0.1
C(0.1)	-12.0	-0.0	-0.0

(e) QMIX on game A

$Q_1 \backslash Q_2$	A(2.0)	B(-8.5)	C(-8.9)
A(1.9)	12.0	-17.8	-17.4
B(-8.7)	-14.6	-44.4	-44.0
C(-8.8)	-15.1	-44.6	-44.9

(g) QPLEX on game A

$Q_1 \backslash Q_2$	A(6.3)	B(0.2)	C(0.7)
A(6.0)	12.0	-12.0	-12.0
B(0.1)	-12.0	0.0	0.0
C(0.6)	-12.0	0.0	0.0

(i) QTRAN on game A

$u_1 \backslash u_2$	A	B	C
A	12	0	10
B	0	0	10
C	10	10	10

(b) True payoff of matrix game B

$Q_1 \backslash Q_2$	A(6.0)	B(-3.7)	C(2.5)
A(6.2)	12.0	0.0	10.0
B(-4.4)	0.0	0.0	10.0
C(1.7)	10.0	10.0	10.0

(d) Our method on game B

$Q_1 \backslash Q_2$	A(-0.6)	B(-0.6)	C(0.5)
A(-0.6)	0.1	0.1	4.1
B(-0.6)	0.1	0.1	4.1
C(0.5)	5.6	5.6	10.0

(f) QMIX on game B

$Q_1 \backslash Q_2$	A(0.4)	B(0.0)	C(-27.1)
A(0.4)	12.1	6.9	-347.0
B(0.0)	5.3	0.2	-353.7
C(-26.9)	-430.9	-436.0	-789.9

(h) QPLEX on game B

$Q_1 \backslash Q_2$	A(4.5)	B(3.3)	C(3.6)
A(4.0)	12.0	0.0	10.0
B(2.7)	-0.0	0.0	10.0
C(3.4)	10.0	10.0	10.0

(j) QTRAN on game B

Table 3: True payoff matrices and estimated value functions for two matrix games. Each column corresponds to one game: the first row shows the ground-truth payoff matrix of Game A and Game B, and each of the remaining rows represents a method’s value estimate.

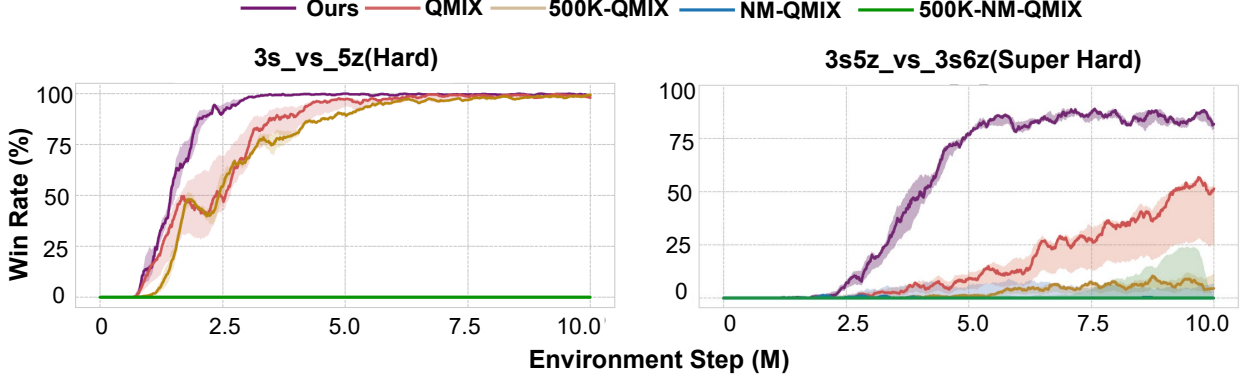


Figure 5: Comparisons of test win rate on SMAC maps: `3s_vs_5z`, and `3s5z_vs_3s6z`. The results are averaged over five independent runs, with the 25%–75% interquartile range shown as a shaded region.

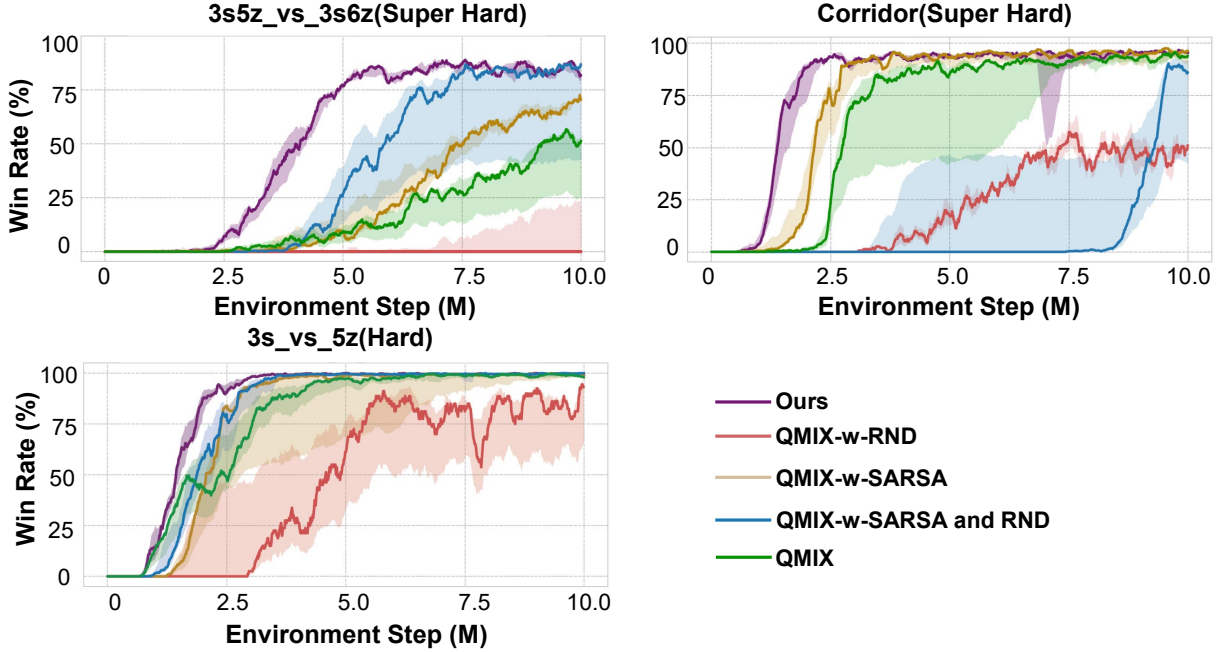


Figure 6: Comparisons of test win rate on SMAC maps: `3s_vs_5z`, `corridor`, and `3s5z_vs_3s6z`. The results are averaged over five independent runs, with the 25%–75% interquartile range shown as a shaded region.

C.2 Ablation Studies

We conduct ablation studies to validate the key design choices of our proposed method.

First, we investigate the effect of directly removing the monotonicity constraint from QMIX, denoted as NM-QMIX. As shown in Figure 5, NM-QMIX performs very poorly on tasks such as `3s5z_vs_3s6z`. This indicates that the monotonicity constraint is indeed beneficial in most cases, despite its limited representational capacity. By default, QMIX uses an exploration annealing schedule of 50k steps; we extend this to 500k but observe no improvement. This suggests that simply increasing ϵ -greedy exploration provides limited benefit. In contrast, our method achieves substantially better performance, which we attribute to the use of a more effective exploration strategy and the SARSA-style update. These findings align with our theoretical analysis: (1) sufficient exploration is crucial for escaping unstable saddle points and ultimately converging to the stable submanifold, and (2) the SARSA-style update, which does not rely on the $\max_{a'}$ operator, provides a more reliable learning

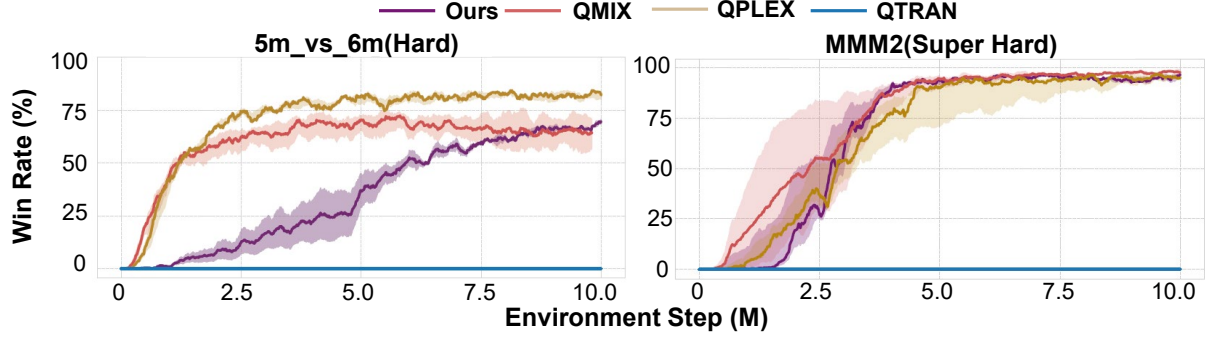


Figure 7: Comparisons of test win rate on SMAC maps: 5m_vs_6m and MMM2. The results are averaged over five independent runs, with the 25%–75% interquartile range shown as a shaded region.

signal.

Second, we examine the effect of adding the RND component and the SARSA-style update to the original QMIX. As shown in the Figure 6, neither component consistently improves performance for standard QMIX. The SARSA-style update is specifically designed for non-monotonic factorization; adding it to monotonic QMIX is unnecessary. As for RND, it enhances exploration, but stronger exploration is not always beneficial: it helps our non-monotonic method recover IGM-optimal solutions, yet it tends to slow convergence in QMIX. These results further indicate that with an effective exploration mechanism and an appropriate algorithmic design, removing the monotonicity constraint can indeed lead to improved performance.

C.3 Failure Cases

We also observe that our method does not consistently outperform the baselines on a small number of environments, as shown in Figure 7. In the 5m_vs_6m environment, our method converges more slowly but eventually reaches performance comparable to QMIX. In MMM2, all methods exhibit very similar learning curves, except for QTRAN, and our method converges slightly more slowly than QMIX. Although we do not achieve absolute superiority on every benchmark, the results still demonstrate that multi-agent Q-learning can perform strongly even without imposing any constraints. We believe that further exploration of this unconstrained setting is a promising future direction.

D Limitations and Future Work

We provide several discussions on the limitations of our work and potential directions for future research.

First, our theoretical analysis of learning dynamics is conducted in a simplified setting. Although the empirical results on challenging sequential benchmarks such as SMAC and GRF indicate that the core insights generalize well, a rigorous extension of the stability analysis to the full multi-state Dec-POMDP setting remains an important avenue for future investigation.

We also discuss a possible direction for extending the theoretical analysis to practical settings. For RL based on multi-state MDPs, the overall training objective can be formulated as a weighted expectation of per-state losses. Consequently, the global Hessian can be regarded as the expectation of per-state Hessians. Therefore, the conclusion of Theorem 2 naturally extends to standard MDPs: if the quadratic form on the normal subspace remains positive for IGM-consistent points at each visited state, the expected Hessian will also be positive definite in the normal directions, implying that IGM-consistent points remain asymptotically stable.

Lastly, our approach adopts a SARSA-style update rule, which is theoretically on-policy. For practical reasons, and following prior work [20], we omit off-policy corrections such as importance sampling. The implications of this omission, particularly the potential distribution shift between the

replay buffer and the current policy, are not explicitly captured in our theoretical analysis and warrant further study.