Towards Understanding Cooperative Multi-Agent Q-Learning with Value Factorization

Anonymous Author(s)
Affiliation
Address
email

Abstract

Value factorization is a popular and promising approach to scaling up multi-agent reinforcement learning in cooperative settings. However, the theoretical understanding of such methods is limited. In this paper, we formalize a multi-agent fitted Q-iteration framework for analyzing factorized multi-agent Q-learning. Based on this framework, we investigate linear value factorization and reveal that multi-agent Q-learning with this simple decomposition implicitly realizes a powerful counterfactual credit assignment, but may not converge in some settings. Through further analysis, we find that on-policy training or richer joint value function classes can improve its local or global convergence properties, respectively. Finally, to support and extend our theoretical implications to practical realization, we conduct an empirical analysis of state-of-the-art deep multi-agent Q-learning algorithms on didactic examples and a broad set of StarCraft II unit micromanagement tasks.

1 Introduction

Cooperative multi-agent reinforcement learning (MARL) has great promise for addressing coordination problems in a variety of applications, such as robotic systems [1], autonomous cars [2], and sensor networks [3]. Such complex tasks often require MARL to learn decentralized policies for agents to jointly optimize a global cumulative reward signal, which posts a number of challenges, including multi-agent credit assignment [4, 5], non-stationarity [6, 7], and scalability [3, 8]. Recently, by leveraging the strength of deep learning techniques, cooperative MARL has made great progress [9–19], particularly in value-based methods that demonstrate state-of-the-art performance on challenging tasks such as StarCraft unit micromanagement [20].

Value factorization is a popular approach to effectively scaling up cooperative multi-agent Q-learning in complex domains. As a representative method, value-decomposition network (VDN) [10] is proposed based on the paradigm of *centralized training with decentralized execution* (CTDE) [21], which learns a centralized but factorizable joint value function Q_{tot} represented as the summation of individual value functions Q_i. During the execution, decentralized policies can be easily derived for each agent i by greedily selecting actions with respect to its local value function Q_i. By utilizing this factorization structure, an implicit multi-agent credit assignment is realized because Q_i is learned by neural network backpropagation from the total temporal-difference error on the single global reward signal, rather than on a local reward signal specific to agent i. This value factorization technique significantly improves the scalability of multi-agent joint policy training and individual policy execution.

Moreover, this linear value factorization adopted by VDN also realizes a sufficient condition for an important principle of CTDE, the IGM (*Individual-Global-Max*) principle [13], which asserts the consistency between joint and individual greedy action selection and ensures the consistent policy learning for CTDE [19]. Inspired by this principle, most recent advances focus on enriching the

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

function expressiveness of the factorization from Q_{tot} to Q_i. QMIX [12] uses a monotonic function to represent the joint value function Q_{tot} through local values Q_i, whose function expressiveness is further improved by QTRAN [13] and QPLEX [19] that aim to achieve full function expressiveness induced by the IGM principle. These approaches show promise and achieve state-of-the-art performance in complicated cooperative multi-agent domains.

Despite the empirical success, theoretical understandings for these MARL approaches are limited. To bridge this gap, this paper considers a general theoretical framework, called *Factorzed Multi-Agent Fitted Q-Iteration* (FMA-FQI), for formally analyzing cooperative MARL with value factorization. This framework generalizes single-agent Fitted Q-Iteration, a popular model for studying Q-learning algorithms with function approximation [22–25]. FMA-FQI models the iterative training procedure of multi-agent Q-learning using empirical Bellman error minimization. By using this general framework, we investigate two popular value factorization methods: the linear factorization used by VDN [10], and the IGM factorization adopted by QTRAN [13] and QPLEX [19]. Our analyses reveal algorithmic properties of these approaches and provide insights on fundamental questions: Why do they work? Do they converge? When they may not perform well?

The main contribution of this paper can be summarized as follows:

1. We formalize *Factorized Multi-Agent Fitted Q-Iteration* (FMA-FQI) as a general theoretical framework for analyzing cooperative multi-agent Q-learning with value factorization.
2. Based on FMA-FQI, we study linear value factorization and derive a closed-form solution to its Bellman error minimization. We then reveal two novel insights that: 1) linear value factorization implicitly realizes a powerful counterfactual credit assignment and 2) it provides local convergence guarantee under on-policy training.
3. Using FMA-FQI, we also study IGM value factorization and prove its global convergence and optimality.
4. Empirical analysis is conducted to connect our theoretical implications to practical scenarios by evaluating state-of-the-art deep MARL approaches.

2 Related Work

Deep Q-learning algorithms that use neural networks as function approximators have shown great promise in solving complicated decision-making problems [26]. One of the core component of such methods is iterative Bellman error minimization, which can be modelled by a classical framework called Fitted Q-Iteration (FQI) [27]. FQI utilizes a specific Q-function class to iteratively optimize empirical Bellman error on a dataset D. Great efforts have been made towards theoretically characterizing the behavior of FQI with finite samples and imperfect function classes [22–24]. From an empirical perspective, there is also a growing trend to adopt FQI for empirical analysis of deep offline Q-learning algorithms [25, 28]. In MARL, the joint Q-function class grows exponentially with the number of agents, leading many algorithms [10, 13, 19] to utilize different value factorization structures to balance scalability and optimality. In this paper, we generalize FQI to formalize a factorized multi-agent fitted Q-iteration framework for analyzing cooperative multi-agent Q-learning algorithms with value factorization.

To achieve superior effectiveness and scalability in multi-agent settings, centralized training with decentralized executing (CTDE) has become a popular MARL paradigm [29, 30]. *Individual-Global-Max* (IGM) principle [13] is a critical concept for value-based CTDE [31], that ensures the consistency between joint and local greedy action selections and enables effective performance in both training and execution phases. VDN [10] utilizes linear value factorization to satisfy a sufficient condition of IGM. This simple linear structure of VDN has become very popular in MARL due to its excellent scalability [13, 19, 32] and inspired many follow-up methods. QMIX [12] proposes a monotonic Q-network structure to improve the expressiveness of the factorized function class. QTRAN [13] aims to realize the entire IGM function class, but its proposed objective is computationally intractable and requires two extra soft regularizations to approximate IGM, which actually loses rigorous IGM guarantees. QPLEX [19], the state-of-the-art multi-agent Q-learning algorithm, encodes the IGM principle into the Q-network architecture and realize a complete IGM function class, but may have potential limitations in scalability. This paper focuses on the theoretical and empirical understanding of multi-agent Q-learning with linear and IGM value factorization to explore the underlying implications of their corresponding factorized value structures.

3 Notations and Preliminaries

3.1 Decentralized Rich-Observation Markov Decision Process (Dec-ROMDP)

To support theoretical analysis on cooperative multi-agent Q-learning, we consider the problem formulation of *Decentralized Rich-Observation Markov Decision Process* (Dec-ROMDP). Dec-ROMDP is an interpolation of ROMDP [33, 34] and Dec-POMDP [35], in which the latent state space is finite, and the observation space could be arbitrarily large. A Dec-ROMDP is defined as a tuple $\mathcal{M} = \langle \mathcal{N}, \mathcal{S}, \mathcal{X}, \mathcal{A}, P, \Lambda, r, \gamma \rangle$. $\mathcal{N} \equiv \{1, \ldots, n\}$ is a finite set of agents. $\mathcal{S}$ is a finite set of global latent states. $\mathcal{X}$ denotes the observation space (a.k.a. context space) which is larger than the latent state space. $\mathcal{A}$ denotes the action space for an individual agent. The joint action $\mathbf{a} \in \mathbf{A} \equiv \mathcal{A}^n$ is a collection of individual actions $[a_i]_{i=1}^n$. At each timestep t, a selected joint action $\mathbf{a}_t$ results in a latent-state transition $s_{t+1} \sim P(\cdot|s_t, \mathbf{a}_t)$ and a global reward signal $r(s_t, \mathbf{a}_t)$. The observation of agent i is generated from the emission distribution $x_{t,i} \sim \Lambda(\cdot|i, s_t)$.

The major difference between Dec-ROMDP and Dec-POMDP is the concept of rich observation. Dec-ROMDP ensures that the observation emission distributions are disjoint across latent states, i.e., there exists an inverse function $\Lambda^{-1} : \mathcal{N} \times \mathcal{X} \to \mathcal{S}$ that decodes full information of latent states from observations. This assumption enables us to consider *reactive function classes* [36, 37], which only takes the latest observation as inputs. Formally, the goal for MARL in Dec-ROMDP is to construct a joint policy $\boldsymbol{\pi} = \langle \pi_1, \ldots, \pi_n \rangle$ maximizing expected discounted rewards $V^{\boldsymbol{\pi}}(\boldsymbol{x}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, \boldsymbol{\pi}(\boldsymbol{x}_t))|\boldsymbol{x}_0 = \boldsymbol{x}\right]$, where $\pi_i : \mathcal{X} \mapsto \mathcal{A}$ denotes an individual policy of agent i which depends on its local observation. The corresponding action-value function is denoted as $Q^{\boldsymbol{\pi}}(\boldsymbol{x}_t, \mathbf{a}_t) = \mathbb{E}[r(s_t, \mathbf{a}_t) + \gamma V^{\boldsymbol{\pi}}(\boldsymbol{x}_{t+1})|\boldsymbol{x}_t]$. More discussions are deferred to Appendix A.

Why Dec-ROMDP? The most widely-used environment setting in deep MARL for empirical studies is Dec-POMDP. However, as implied by prior work, finding optimal policy in infinite-horizon Dec-POMDPs is an incomputable problem in the classical computation model [38]. There is no algorithm can achieve strong theoretical guarantees in the Dec-POMDP setting. Following recent advances in reinforcement learning with function approximation [33, 34, 39, 40], we consider the rich-observation setting to derive implications from a theoretical perspective.

3.2 Centralized Training with Decentralized Execution (CTDE)

Most deep multi-agent Q-learning algorithms with value factorization adopt the paradigm of centralized training with decentralized execution [21]. In the training phase, the centralized trainer can access all global information, including joint trajectories, shared global rewards, agents' policies, and value functions. In the decentralized execution phase, every agent makes individual decisions based on its local information. *Individual-Global-Max* (IGM) [13] is a common principle to realize effective decentralized policy execution. It enforces the action selection consistency between the global joint action-value Q_{tot} and individual action-values $[Q_i]_{i=1}^n$, which are specified as follows:

$$\textbf{(IGM)} \quad \arg\max_{\mathbf{a} \in \mathbf{A}} Q_{\text{tot}}(\boldsymbol{x}, \mathbf{a}) = \left\langle \arg\max_{a_1 \in \mathcal{A}} Q_1(x_1, a_1), \cdots, \arg\max_{a_n \in \mathcal{A}} Q_n(x_n, a_n) \right\rangle. \tag{1}$$

QPLEX [19] is the state-of-the-art deep multi-agent Q-learning algorithm that realizes a complete IGM factorization. QTRAN [13] is another method to approximate IGM principle by two extra soft regularizations. Moreover, as stated in Eq. (2), the linear value factorization adopted by VDN [10] is a sufficient condition for the IGM constraint stated in Eq. (1) but is not a necessary condition, which induces a limited joint action-value function class.

$$\textbf{(Additivity)} \quad Q_{\text{tot}}(\boldsymbol{x}, \mathbf{a}) = \sum_{i=1}^{n} Q_i(x_i, a_i). \tag{2}$$

4 Factorized Multi-Agent Fitted Q-Iteration

Constructing a specific factorized value function class satisfying the IGM condition is a critical step to realize the centralized training with decentralized execution (CTDE) paradigm. In the literature of Q-learning with function approximation, *Fitted Q-Iteration* (FQI) [27] is a popular framework for both theoretical and empirical studies [22–25, 28]. In this section, we generalize single-agent FQI to formalize a *Factorzed Multi-Agent Fitted Q-Iteration* (FMA-FQI) framework, which enables us to characterize the algorithmic properties of different factorization structures in the CTDE paradigm.

For multi-agent Q-learning with value factorization, we use Q_{tot} to denote the global but factorized value function, which can be represented as a function of individual value functions $[Q_i]_{i=1}^n$. In other words, we can use $[Q_i]_{i=1}^n$ to represent Q_{tot}. For brevity, we overload $Q = \langle Q_{\text{tot}}, [Q_i]_{i=1}^n \rangle$ to denote their association. Following fitted Q-iteration, FMA-FQI considers an iterative optimization framework based on a given dataset $D = \{(\boldsymbol{x}, \mathbf{a}, r, \boldsymbol{x}')\}$ as presented in Algorithm 1.

Algorithm 1 Factorized Multi-Agent Fitted Q-Iteration (FMA-FQI)

1: **Input:** dataset D, the number of iterations T, factorized multi-agent value function class $\mathcal{Q}^{\text{FMA}}$
2: Randomly initialize $Q^{(0)}$ from $\mathcal{Q}^{\text{FMA}}$
3: **for** $t = 0 \dots T-1$ **do**
4: Iteratively update

$$Q^{(t+1)} \leftarrow \mathcal{T}_D^{\text{FMA}} Q^{(t)} \equiv \underset{Q \in \mathcal{Q}^{\text{FMA}}}{\arg\min} \underset{(\boldsymbol{x}, \mathbf{a}, r, \boldsymbol{x}') \sim D}{\mathbb{E}} \left(\hat{y}^{(t)}(\boldsymbol{x}, \mathbf{a}, \boldsymbol{x}') - Q_{\text{tot}}(\boldsymbol{x}, \mathbf{a}) \right)^2 \tag{3}$$

where $\hat{y}^{(t)}(\boldsymbol{x}, \mathbf{a}, \boldsymbol{x}') = r + \gamma \max_{\mathbf{a}' \in \mathbf{A}} Q_{\text{tot}}^{(t)}(\boldsymbol{x}', \mathbf{a}')$ denotes the one-step TD target.

5: Construct policies by individual value functions

$$\forall i \in \mathcal{N},\ \pi_i^{(t+1)}(x_i) = \underset{a_i \in \mathcal{A}}{\arg\max}\, Q_i^{(t+1)}(x_i, a_i) \tag{4}$$

6: **Return:** $[\pi_i^{(T)}]_{i=1}^n$

Formally, FMA-FQI specifies the Q-function class through a multi-agent value factorization structure:

$$\mathcal{Q}^{\text{FMA}} = \left\{ Q \;\middle|\; \exists f \in \mathcal{F}^{\text{FMA}},\ Q_{\text{tot}}(\boldsymbol{x}, \mathbf{a}) = f\left(\boldsymbol{x}, \mathbf{a}, [Q_i]_{i=1}^n\right),\ [Q_i]_{i=1}^n \in \mathbb{R}^{|\mathcal{X} \times \mathcal{A}|^n} \right\}, \tag{5}$$

where $\mathcal{F}^{\text{FMA}}$ denotes the function class characterizing the factorization structure. For example, the IGM constraint stated as Eq. (1) and the additivity constraint stated as Eq. (2) correspond to the value factorization structures used by QPLEX [19] and VDN [10], respectively.

The empirical Bellman error minimization in FMA-FQI is established using the global reward signals. To facilitate further discussions, we rewrite the regression objective as follows:

$$Q^{(t+1)} \leftarrow \mathcal{T}_D^{\text{FMA}} Q^{(t)} \equiv \underset{Q \in \mathcal{Q}^{\text{FMA}}}{\arg\min} \underset{(\boldsymbol{x}, \mathbf{a}) \sim D}{\mathbb{E}} \left(y^{(t)}(\boldsymbol{x}, \mathbf{a}) - Q_{\text{tot}}(\boldsymbol{x}, \mathbf{a}) \right)^2 \tag{6}$$

where $y^{(t)}(\boldsymbol{x}, \mathbf{a}) = r + \gamma \mathbb{E}_{\boldsymbol{x}'} \left[\max_{\mathbf{a}'} Q_{\text{tot}}^{(t)}(\boldsymbol{x}', \mathbf{a}') \right]$ denotes the expected one-step TD target. Assume the dataset is adequate, the equivalence between Eq. (3) and Eq. (6) is proved in Appendix A.3.

Modeling CTDE by FMA-FQI. Different from the single-agent case, FMA-FQI considers the multi-agent value factorization structure, Q_{tot} and $[Q_i]_{i=1}^n$, to support effective training and scalable execution. In the global reward setting, the shared reward signal can only supervise the training of the joint value function Q_{tot} (see Eq. (6)). The learned joint value function Q_{tot} would be automatically decomposed to individual value functions $[Q_i]_{i=1}^n$ through the factorized function structure, which can be further used to perform decentralized execution (see Eq. (4)). In practice, FMA-FQI illustrated in Eq. (5) provides the end-to-end learning of $[Q_i]_{i=1}^n$ as the iterative optimization of joint value function Q_{tot}. The greedy action selection can be searched in the individual action space $\mathcal{A}$ rather than the joint action space $\mathcal{A}^n$, which significantly reduces the computation costs.

Relation to Prior Work. Most prior work of FQI focus on finite-sample analyses and exploring minimum requirements for performance guarantees in a general sense [22–24]. By contrast, the analyses in this paper are motivated by recent advances in practical algorithms. We aim to characterize the algorithmic properties of two specific value function classes, linear value factorization and IGM value factorization, which are widely used in deep multi-agent Q-learning algorithms [10, 13, 19].

5 Multi-Agent Q-Learning with Linear Value Factorization

Linear value factorization proposed by VDN [10] is a simple yet effective method to realize a sufficient condition of the IGM principle in CTDE paradigm. In this section, we provide theoretical analysis towards a deeper understanding of this popular factorization structure. Our result is based on

FMA-FQI with linear value factorization, named FQI-LVF. We derive the closed-form update rule of FQI-LVF, and then reveal the underlying credit assignment mechanism realized by linear value factorization learning. We also find that FQI-LVF has the potential risk of unbounded divergence in the most unfavorable environment. To improve its training stability, we theoretically demonstrate that on-policy data collection can provide local convergence guarantee.

5.1 Multi-Agent Fitted Q-Iteration with Linear Value Factorization (FQI-LVF)

We define multi-agent fitted Q-iteration with linear value factorization as follows.

Definition 1 (FQI-LVF)**.** *FQI-LVF is an instance of FMA-FQI stated in Algorithm 1, which specifies the action-value function class with linear value factorization:*

$$\mathcal{Q}^{LVF} = \left\{ Q \;\middle|\; Q_{tot}(\boldsymbol{x}, \mathbf{a}) = \sum_{i=1}^{n} Q_i(x_i, a_i),\ [Q_i]_{i=1}^{n} \in \mathbb{R}^{|\mathcal{X}\times\mathcal{A}|^n} \right\}. \tag{7}$$

FQI-LVF considers a popular linear value factorization structure in MARL [10, 13, 19], which reduces the action-value function class to provide attractive scalability for policy training [32]. Value-decomposition network (VDN) [10] provides a deep-learning-based implementation of FQI-LVF, in which individual value functions $[Q_i]_{i=1}^{n}$ are parameterized by deep neural networks, and the joint value function Q_{tot} can be simply formed by their summation.

5.2 Implicit Counterfactual Credit Assignment in Linear Value Factorization

In the formulation of FQI-LVF, the empirical Bellman error minimization with linear value function class $\mathcal{Q}^{\text{LVF}}$ can be regarded as a weighted linear least-squares problem, which contains $n|\mathcal{X}\times\mathcal{A}|$ variables to form individual value functions $[Q_i]_{i=1}^{n}$ and $|\mathcal{X}\times\mathcal{A}|^n$ data points corresponding to all entries of the regression target $y^{(t)}(\boldsymbol{x}, \mathbf{a})$. Formally, by plugging the definition of $\mathcal{Q}^{\text{LVF}}$ into multi-agent fitted Q-iteration, FQI-LVF iteratively optimizes the following least-squares problem:

$$Q^{(t+1)} \leftarrow \mathcal{T}_D^{\text{LVF}} Q^{(t)} \equiv \operatorname*{arg\,min}_{Q\in\mathcal{Q}^{\text{LVF}}} \sum_{\boldsymbol{x},\mathbf{a}} p_D(\boldsymbol{x}, \mathbf{a}) \left(y^{(t)}(\boldsymbol{x}, \mathbf{a}) - \sum_{i=1}^{n} Q_i^{(t+1)}(x_i, a_i) \right)^2,$$

where p_D denotes the probability measured by the dataset D. The closed-form solution of this least-squares problem is presented in Theorem 1 with the following assumption:

Assumption 1 (Decentralized Data Collection)**.** *The dataset D is collected by a decentrailized and exploratory policy $\boldsymbol{\pi}^D$ satisfying:*

$$\forall(\boldsymbol{x}\times\boldsymbol{a}) \in \boldsymbol{\mathcal{X}}\times\boldsymbol{A},\ \ \boldsymbol{\pi}^D(\mathbf{a}|\boldsymbol{x}) = \prod_{i\in\mathcal{N}} \pi_i^D(a_i|x_i) > 0.$$

Theorem 1. *Let $Q^{(t+1)} = \mathcal{T}_D^{LVF} Q^{(t)}$ denote a single iteration of the empirical Bellman operator. $\forall i \in \mathcal{N}, \forall(\boldsymbol{x}, \mathbf{a}) \in \boldsymbol{\mathcal{X}}\times\mathbf{A}$, the individual action-value function $Q_i^{(t+1)}(x_i, a_i)$ is updated to*

$$\underbrace{\mathop{\mathbb{E}}_{(x'_{-i},a'_{-i})\sim p_D(\cdot|x_i)} \left[y^{(t)}\left(x_i \oplus x'_{-i}, a_i \oplus a'_{-i}\right) \right]}_{\textit{evaluation of the individual action } a_i} - \frac{n-1}{n} \underbrace{\mathop{\mathbb{E}}_{\boldsymbol{x}',\mathbf{a}'\sim p_D(\cdot|\Lambda^{-1}(x_i))} \left[y^{(t)}\left(\boldsymbol{x}', \mathbf{a}'\right) \right]}_{\textit{counterfactual baseline}} + w_i(x_i), \tag{8}$$

where $z_i \oplus z'_{-i}$ denotes $\langle z'_1, \cdots, z'_{i-1}, z_i, z'_{i+1}, \cdots, z'_n\rangle$, and z'_{-i} denotes the elements of all agents except for agent i. $\Lambda^{-1}(x_i)$ denotes the inverse of observation emission, which decodes the current latent state from x_i. The residue term $\mathbf{w} \equiv [w_i]_{i=1}^{n}$ is an arbitrary function satisfying $\forall\boldsymbol{x} \in \boldsymbol{\mathcal{X}}$, $\sum_{i=1}^{n} w_i(x_i) = 0$.

The proof of Theorem 1 is based on *Moore-Penrose inverse* [41] for weighted linear regression analysis. The detailed proofs for all theory statements in this paper are deferred to Appendix.

As shown in Theorem 1, the update rule of local action-value $Q_i^{(t+1)}$ consists of three terms. The first term is the expectation of one-step TD target value over the actions of other agents, which evaluates the expected return of executing an individual action a_i. The second term is the expectation of one-step target TD values over all joint actions, which can be regarded as a baseline function evaluating the average performance. The arbitrary vector $\mathbf{w}$ indicates the entire valid individual action-value function space. We can ignore this term because $\mathbf{w}$ does not affect the local action selection of each agent and will be eliminated in the summation operator of linear value factorization (see Eq. (2)), which indicates that joint action-value $Q_{\text{tot}}^{(t+1)}$ has a unique closed-form solution.

Implicit Counterfactual Credit Assignment. Note that, if we regard the empirical probability $p_D(\mathbf{a}|s)$ within the dataset D as a *default policy*, the first term of Eq. (8) is the expected value of an individual action a_i, and the second term is the expected value of the default policy, which is considered as the *counterfactual baseline*. Their difference corresponds to a credit assignment mechanism called *counterfactual difference rewards*, which has been used by counterfactual multi-agent policy gradient (COMA) [11]. Moreover, the extra importance weight $(n-1)/n$ makes our derived credit assignment of FQI-LVF to be more consistent and meaningful in the sense that all global rewards should be assigned to agents. Consider a simple case where all joint actions generate the same reward signals, Eq. (8) will assign $1/n$ unit of rewards to each agent, but COMA will assign 0. This gap will asymptotically close when n becomes sufficiently large.

Remark on Assumptions. The closed-form solution derived in Theorem 1 relies on two assumptions, decentralized data collection and rich-observation problem formulation. The assumption of decentralized data collection is natural in the CTDE paradigm since we do not consider centralized policies. The assumption of rich observations corresponds to the least requirement we found to make the closed-form solution meaningful and explainable. Note that, in terms of problem complexity, computing Bellman updates in general Dec-POMDPs with linear value factorization (i.e., without the rich-observation assumption) is equivalent to computing arbitrary linear least-squares problems with binary matrices, which does not have existing closed-form solutions. Please refer to Proposition 3 in Appendix B.3 for detailed discussions.

5.3 Data Distribution Matters for Linear Value Factorization

The closed-form update rule of FQI-LVF stated in Theorem 1 enables us to investigate more algorithmic properties of linear value factorization in multi-agent Q-learning. In the empirical literature, the major strength of linear value factorization is its high scalability [32], since the additive constraint between the global and local values is non-parametric which does not require additional learnable parameters. Meanwhile, the limitation of linear value factorization structure is also induced by its simplicity. The function capacity of $\mathcal{Q}^{\text{LVF}}$ cannot express all valid global values in $\mathbb{R}^{|\mathcal{X}\times\mathcal{A}|^n}$ [13, 19, 31]. As suggested by prior work [24, 42], the performance of Q-learning algorithms would become sensitive to the training data distribution when the function approximator is not perfect. In this section, we investigate how linear value factorization structure interacts with different data distributions. Our theoretical results contain two aspects:

1. When the dataset is collected by a fully offline manner, e.g., by a uniform random policy, FQI-LVF would suffer from unbounded divergence from arbitrary initialization in the most unfavorable environment.
2. By iteratively updating the dataset to a nearly on-policy data distribution, FQI-LVF would have local convergence guarantees near the optimal solution.

5.3.1 Divergence Risk in Offline Training

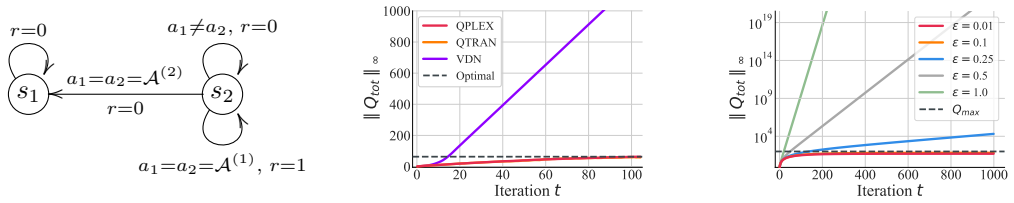
To begin with the discussion on linear value factorization in offline training settings, we first investigate an important property, named γ-contraction, in the FQI-LVF framework (see Proposition 1).

Proposition 1. *The empirical Bellman operator $\mathcal{T}_D^{LVF}$ is not a γ-contraction, i.e., the following important property of the standard Bellman optimality operator $\mathcal{T}$ does not hold for $\mathcal{T}_D^{LVF}$ anymore.*

$$(\gamma\text{-contraction}) \qquad \forall Q_{tot}, Q'_{tot} \in \mathcal{Q},\ \ \|\mathcal{T}Q_{tot} - \mathcal{T}Q'_{tot}\|_\infty \le \gamma \|Q_{tot} - Q'_{tot}\|_\infty$$

For the standard *Bellman optimality operator* $\mathcal{T}$, γ-contraction property is critical to derive the convergence guarantee of Q-learning algorithms [43]. In the context of FQI-LVF, the additivity constraint limits the joint action-value function class that it can express, which deviates the empirical Bellman operator $\mathcal{T}_D^{\text{LVF}}$ from the original *Bellman optimality operator* $\mathcal{T}$ (see Theorem 1). This deviation is also known as *inherent Bellman error* [22] or projection error, which corrupts a broad set of stability properties, including γ-contraction.

To serve a concrete example, we construct a simple MMDP with two agents, two global states, and two actions (see Figure 1a). Multi-agent MDP (MMDP) [44] corresponds to a special case of Dec-ROMDP where all agents can obtain the global state as their observations. The optimal policy of this MMDP is simply executing the action $\mathcal{A}^{(1)}$ at state s_2, which is the only way for two agents to



(a) Two-state task (b) Deep MARL (c) On-policy FQI-LVF

Figure 1: (a) A two-state MMDP where FQI-LVF will diverge to infinity when $\gamma \in \left(\frac{4}{5}, 1\right)$ from arbitrary initialization $Q^{(0)}$. (b) The learning curves of $\|Q_{\text{tot}}\|_\infty$ produced by running several deep multi-agent Q-learning algorithms. (c) The learning curves of $\|Q_{\text{tot}}\|_\infty$ of on-policy FQI-LVF on the given task where the dataset is generated by different choices of hyper-parameters ϵ for ϵ-greedy.

obtain a positive reward. The learning curve of $\epsilon = 1.0$ (green one) in Figure 1c refers to an offline setting with uniform data distribution, in which an unbounded divergence can be observed as depicted by the following proposition.

Proposition 2. *There exist an MMDP such that, when using uniform data distribution, the value function of FQI-LVF diverges to infinity from an arbitrary initialization $Q^{(0)}$.*

In the proof of Proposition 2, we will show that the unbounded divergence would happen to an arbitrary initialization $Q^{(0)}$ in this example. To provide an implication for practical scenarios, we also investigate the performance of several deep multi-agent Q-learning algorithms in this two-state task. As shown in Figure 1b, VDN [10], a deep-learning-based implementation of FQI-LVF, also results in unbounded divergence.

5.3.2 On-Policy Data Collection Provides Local Convergence Guarantee for FQI-LVF

As shown in Theorem 1, the choice of training data distribution affects the output of the empirical Bellman operator $\mathcal{T}_D^{\text{LVF}}$. In the literature of temporal-difference learning with function approximation, on-policy data distribution is known to be beneficial to the learning stability in some settings [45–47]. In this section, we show that FQI-LVF would have a local convergence guarantee with online data collection. Formally, we consider the dataset D_t is accumulated by running an ϵ-greedy policy [26] at t-th iteration, i.e., each agent performs a random action with probability ϵ. To serve intuitions, we present an informal statement of local stability of FQI-LVF here and defer the detailed version, its proof, and the algorithm box of on-policy FQI-LVF to Appendix C.2.

Theorem 2 (Informal)**.** *On-policy FQI-LVF will locally converge to the optimal policy and have a fixed point value function when the hyper-parameter ϵ is sufficiently small.*

The proof of this statement is based on Brouwer's fixed-point theorem [48], in which we construct a bounding box in the value function space such that $\mathcal{T}_D^{\text{LVF}}$ is a closed operator. In addition, within this bounding box, the induced policies of these value functions are equal to the optimal policy π^* in the given MDP. It indicates that multi-agent Q-learning with linear value factorization has a convergent region, where the value function induces optimal actions.

Figure 1c visualizes the performance of on-policy FQI-LVF with different values of the hyper-parameter ϵ. With a smaller ϵ (such as 0.1 or 0.01), on-policy FQI-LVF demonstrates excellent training stability, and their corresponding collected datasets are closer to on-policy data distribution.

6 Multi-Agent Q-Learning with IGM Value Factorization

Recently, advanced deep multi-agent Q-learning algorithms, QPLEX [19] and QTRAN [13], aim to utilize IGM factorization method to formulate their value function class for CTDE paradigm and achieve state-of-the-art performance in StarCraft II benchmark [20] with online data collection. In this section, we introduce FMA-FQI with IGM factorization, named FQI-IGM. Compared with the unbounded divergence risk of linear value factorization analyzed in section 5, we find that FQI-IGM provides a global optimality convergence guarantee in Dec-ROMDPs, which implies IGM value factorization is a stable choice for cooperative multi-agent Q-learning.

6.1 Multi-Agent Fitted Q-Iteration with IGM Value Factorization (FQI-IGM)

We define multi-agent fitted Q-iteration with with IGM value factorization as follows.

Definition 2 (FQI-IGM)**.** *FQI-IGM is an instance of FMA-FQI stated in Algorithm 1, which specifies the action-value function class with a complete IGM principle realization*

$$\mathcal{Q}^{IGM} = \left\{ Q \;\middle|\; \arg\max_{\mathbf{a}\in\mathbf{A}} Q_{tot}(\boldsymbol{x}, \mathbf{a}) = \left\langle \arg\max_{a_1\in\mathcal{A}} Q_i(x_i, a_i) \right\rangle_{i=1}^{n},\ [Q_i]_{i=1}^{n} \in \mathbb{R}^{|\mathcal{X}\times\mathcal{A}|^n} \right\}. \tag{9}$$

Compared with FQI-LVF stated in Definition 1, the differences are the factorized value function classes, i.e, $\mathcal{Q}^{\text{LVF}}$ vs. $\mathcal{Q}^{\text{IGM}}$. Note that $\mathcal{Q}^{\text{LVF}} \subset \mathcal{Q}^{\text{IGM}}$ indicates that the linear factorization structure realizes a subspace of IGM value functions. In empirical studies [19], the enriched value function class $\mathcal{Q}^{\text{IGM}}$ is observed to have better training stability and overall performance than that of linear value factorization $\mathcal{Q}^{\text{LVF}}$. In the following section, we will provide theoretical supports regarding the differences between the algorithmic properties of $\mathcal{Q}^{\text{LVF}}$ and $\mathcal{Q}^{\text{IGM}}$.

6.2 Global Convergence Guarantee of IGM Value Factorization

The major difference between $\mathcal{Q}^{\text{LVF}}$ and $\mathcal{Q}^{\text{IGM}}$ is their representation power of function expressiveness. Recall that the unbounded divergence of linear value factorization is caused by the projection error induced from the limited function expressiveness of $\mathcal{Q}^{\text{LVF}}$. In this section, we show that the divergence issues would not happen to FQI-IGM using $\mathcal{Q}^{\text{IGM}}$ in Dec-ROMDPs. Formally, the convergence guarantee of FQI-IGM can be derived based on the following assumption:

Assumption 2 (Exploratory Data Collection)**.** *The dataset D is collected by an exploratory policy $\boldsymbol{\pi}^D$ satisfying $\forall(\boldsymbol{x}\times\boldsymbol{a}) \in \boldsymbol{\mathcal{X}}\times\boldsymbol{A}, \boldsymbol{\pi}^D(\mathbf{a}|\boldsymbol{x}) > 0$.*

Theorem 3. *FQI-IGM globally converges to the optimal value function in arbitrary Dec-ROMDPs.*

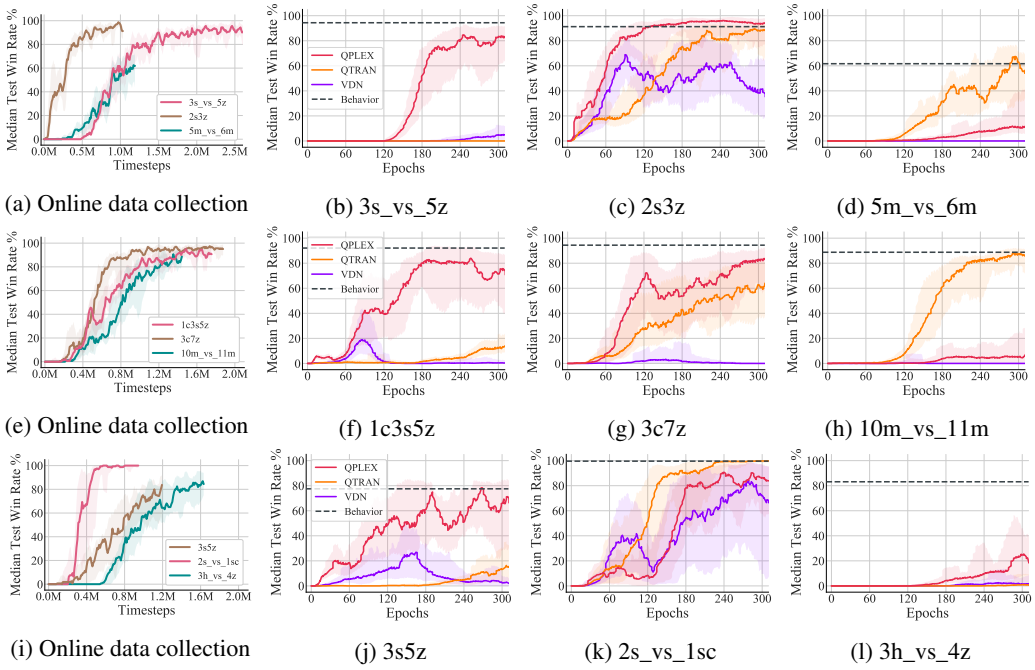
Theorem 3 relies on a fact that $\mathcal{Q}^{\text{IGM}}$ is closed under the Bellman operator, i.e., the empirical Bellman error minimization discussed in Eq. (6) can reach zero, and the empirical Bellman operator $\mathcal{T}_D^{\text{IGM}}$ satisfies the γ-contraction property. The global convergence presented in Theorem 3 is the same as the convergence property of tabular value iteration.

Remark on Assumptions. The only purpose to introduce the assumption of exploratory dataset as Assumption 2 is to make the outputs of empirical Bellman error minimization well-defined in analysis. This assumption is weaker than the assumptions used in Theorem 1 and Theorem 2 for analyzing linear value factorization. It indicates that multi-agent Q-learning with IGM value factorization is not sensitive to the training data distribution. In addition, we also consider the problem formulation of Dec-ROMDP for FQI-IGM, since prior work [38] suggests that, there is no algorithm can give convergence guarantees in general infinite-horizon Dec-POMDPs.

6.3 IGM Value Factorization Provides Better Learning Stability in Offline Deep MARL

In this subsection, we conduct an empirical study to connect our theoretical implications to practical scenarios of deep multi-agent Q-learning algorithms. We evaluate three deep-learning-based counterparts of FQI-LVF and FQI-IGM, i.e., VDN [10], QTRAN [13], and QPLEX [19], respectively. VDN can be regarded as a deep-learning-based implementation of FQI-LVF. QTRAN and QPLEX correspond to two different implementations of FQI-IGM. As suggested by Proposition 2 that linear value factorization may suffer from training stability in offline setting, we utilize StarCraft Multi-Agent Challenge (SMAC) benchmark [20] with offline data collection to investigate the effect of the expressiveness of a factorized value function, i.e., which value factorization is suitable for multi-agent offline reinforcement learning.

We evaluate the performance of VDN, QTRAN, and QPLEX on nine common maps of StarCraft II. The results are shown in Figure 2. The datasets are collected by training a VDN with ϵ-greedy online data collection, which correspond to the learning curves of *Behavior* line shown in Figure 2(a,e,i). Figure 2(b-d,f-h,j-l) illustrate that QPLEX and QTRAN with $\mathcal{Q}^{\text{IGM}}$ function class achieve the state-of-the-art performance, but offline VDN performs poorly and cannot utilize well the offline dataset collected by an unfamiliar behavior policy. This considerable performance gap between deep multi-agent Q-learning with IGM and linear value factorization indicates that the expressiveness of value factorization structures dramatically affects the performance, and IGM value factorization is a promising choice of value factorization for offline multi-agent reinforcement learning.



(a) Online data collection (b) 3s_vs_5z (c) 2s3z (d) 5m_vs_6m

(e) Online data collection (f) 1c3s5z (g) 3c7z (h) 10m_vs_11m

(i) Online data collection (j) 3s5z (k) 2s_vs_1sc (l) 3h_vs_4z

Figure 2: (a,c,i) The learning curve of VDN using online data collection when constructing the static datasets. (b-d,f-h,j-l) Evaluating the performance of deep multi-agent Q-learning algorithms with a given static dataset.

Remark on Experiment Settings. Similar experiment results are observed in Dec-POMDP settings in prior work [19]. To make the comparison rigorous, we slightly modify the observation emission rule of SMAC benchmark to establish a suite of Dec-ROMDP environments. Formally, we concatenate the local observations with the global latent state vector to ensure rich observations. A detailed description of the experiment setting is deferred to Appendix E.3.

7 Conclusion

This paper proposes a unified framework for analyzing cooperative multi-agent Q-learning with value factorization, which makes an initial effort to provide theoretical understanding of this branch of methods. We analyze two classes of value factorization and investigate their algorithmic properties. The derived implications of our theoretical results are supported by experiments with deep-learning-based implementations. To close this paper, we connect our results with additional related literature as a discussion of future work.

Other value factorization approaches. In addition to linear and IGM value decomposition discussed in this paper, there are a variety of value factorization methods have been proposed to trade off the computational complexity and function expressiveness. e.g., the monotonic value decomposition used by QMIX [12] and the second-order value factorization used by deep coordination graph (DCG) [49]. These factorization mechanisms aim to capture some special structures of multi-agent learning, and investigating their algorithmic properties would provide deeper insights to understand state-of-the-art MARL algorithms.

Sample complexity analysis. Finite-sample analysis of fully centralized multi-agent algorithms has been studied in the literature of PAC reinforcement learning. Several prior work [50, 51] proposed provably efficient reinforcement learning algorithms for factored MDPs in terms of regret and sample complexity. However, these algorithms are established in an information-theoretic manner and are not computational efficient. From this perspective, introducing the concept of value factorization is a promising way to improve the computational efficiency of multi-agent algorithms in PAC learning settings.

References

[1] Maximilian Hüttenrauch, Adrian Šošić, and Gerhard Neumann. Guided deep reinforcement learning for swarm systems. *arXiv preprint arXiv:1709.06011*, 2017.

[2] Yongcan Cao, Wenwu Yu, Wei Ren, and Guanrong Chen. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial informatics*, 9(1):427–438, 2012.

[3] Chongjie Zhang and Victor Lesser. Coordinated multi-agent reinforcement learning in networked distributed pomdps. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[4] David H Wolpert and Kagan Tumer. Optimal payoff functions for members of collectives. In *Modeling Complexity in Economic and Social Systems*, pages 355–369. World Scientific, 2002.

[5] Duc Thien Nguyen, Akshat Kumar, and Hoong Chuin Lau. Credit assignment for collective multiagent rl with global rewards. In *Advances in Neural Information Processing Systems*, pages 8102–8113, 2018.

[6] Chongjie Zhang and Victor Lesser. Multi-agent learning with policy prediction. In *Twenty-fourth AAAI conference on Artificial Intelligence*, 2010.

[7] Xinliang Song, Tonghan Wang, and Chongjie Zhang. Convergence of multi-agent learning with a finite step size in general-sum games. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 935–943. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

[8] Liviu Panait and Sean Luke. Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-agent Systems*, 11(3):387–434, 2005.

[9] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, pages 6379–6390, 2017.

[10] 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 based on team reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 2085–2087, 2018.

[11] Jakob N Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[12] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 4292–4301, 2018.

[13] 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*, pages 5887–5896, 2019.

[14] Tonghan Wang, Jianhao Wang, Yi Wu, and Chongjie Zhang. Influence-based multi-agent exploration. In *International Conference on Learning Representations*, 2020.

[15] Tonghan Wang, Jianhao Wang, Chongyi Zheng, and Chongjie Zhang. Learning nearly decomposable value functions via communication minimization. In *International Conference on Learning Representations*, 2020.

[16] Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. Emergent tool use from multi-agent autocurricula. In *International Conference on Learning Representations*, 2020.

[17] Tonghan Wang, Heng Dong, Victor Lesser, and Chongjie Zhang. Multi-agent reinforcement learning with emergent roles. In *International Conference on Machine Learning*, 2020.

[18] Yaodong Yang, Jianye Hao, Ben Liao, Kun Shao, Guangyong Chen, Wulong Liu, and Hongyao Tang. Qatten: A general framework for cooperative multiagent reinforcement learning. *arXiv preprint arXiv:2002.03939*, 2020.

[19] Jianhao Wang, Zhizhou Ren, Terry Liu, Yu Yang, and Chongjie Zhang. Qplex: Duplex dueling multi-agent q-learning. In *International Conference on Learning Representations*, 2021.

[20] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. The starcraft multi-agent challenge. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 2186–2188. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

[21] Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2137–2145, 2016.

[22] Rémi Munos and Csaba Szepesvári. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9(May):815–857, 2008.

[23] Amir-massoud Farahmand, Csaba Szepesvári, and Rémi Munos. Error propagation for approximate policy and value iteration. In *Advances in Neural Information Processing Systems*, pages 568–576, 2010.

[24] Jinglin Chen and Nan Jiang. Information-theoretic considerations in batch reinforcement learning. In *International Conference on Machine Learning*, pages 1042–1051, 2019.

[25] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[26] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[27] Damien Ernst, Pierre Geurts, and Louis Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6(Apr):503–556, 2005.

[28] Justin Fu, Aviral Kumar, Matthew Soh, and Sergey Levine. Diagnosing bottlenecks in deep q-learning algorithms. In *International Conference on Machine Learning*, pages 2021–2030, 2019.

[29] Frans A Oliehoek, Matthijs TJ Spaan, and Nikos Vlassis. Optimal and approximate q-value functions for decentralized pomdps. *Journal of Artificial Intelligence Research*, 32:289–353, 2008.

[30] Landon Kraemer and Bikramjit Banerjee. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing*, 190:82–94, 2016.

[31] Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. Maven: Multi-agent variational exploration. In *Advances in Neural Information Processing Systems*, pages 7611–7622, 2019.

[32] Yihan Wang, Beining Han, Tonghan Wang, Heng Dong, and Chongjie Zhang. Off-policy multi-agent decomposed policy gradients. *arXiv preprint arXiv:2007.12322*, 2020.

[33] Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Pac reinforcement learning with rich observations. In *Advances in Neural Information Processing Systems*, pages 1840–1848, 2016.

[34] Kamyar Azizzadenesheli, Alessandro Lazaric, and Animashree Anandkumar. Reinforcement learning in rich-observation mdps using spectral methods. *arXiv preprint arXiv:1611.03907*, 2016.

[35] Frans A Oliehoek, Christopher Amato, et al. *A concise introduction to decentralized POMDPs*, volume 1. Springer, 2016.

[36] Michael L Littman. Memoryless policies: theoretical limitations and practical results. In *Proceedings of the third international conference on Simulation of adaptive behavior: from animals to animats 3: from animals to animats 3*, pages 238–245, 1994.

[37] Nicolas Meuleau, Leonid Peshkin, Kee-Eung Kim, and Leslie Pack Kaelbling. Learning finite-state controllers for partially observable environments. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 427–436, 1999.

[38] Omid Madani, Steve Hanks, and Anne Condon. On the undecidability of probabilistic planning and infinite-horizon partially observable markov decision problems. In *AAAI/IAAI*, pages 541–548, 1999.

[39] Simon Du, Akshay Krishnamurthy, Nan Jiang, Alekh Agarwal, Miroslav Dudik, and John Langford. Provably efficient rl with rich observations via latent state decoding. In *International Conference on Machine Learning*, pages 1665–1674. PMLR, 2019.

[40] Dipendra Misra, Mikael Henaff, Akshay Krishnamurthy, and John Langford. Kinematic state abstraction and provably efficient rich-observation reinforcement learning. In *International conference on machine learning*, pages 6961–6971. PMLR, 2020.

[41] Eliakim H Moore. On the reciprocal of the general algebraic matrix. *Bulletin of the American Mathematical Society*, 26:394–395, 1920.

[42] Tabish Rashid, Gregory Farquhar, Bei Peng, and Shimon Whiteson. Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020.

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

[44] Craig Boutilier. Planning, learning and coordination in multiagent decision processes. In *Proceedings of the 6th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 195–210. Morgan Kaufmann Publishers Inc., 1996.

[45] Richard S Sutton. Generalization in reinforcement learning: Successful examples using sparse coarse coding. In *Advances in Neural Information Processing Systems*, pages 1038–1044, 1996.

[46] J. N. Tsitsiklis and B. Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5):674–690, 1997.

[47] Hado Van Hasselt, Yotam Doron, Florian Strub, Matteo Hessel, Nicolas Sonnerat, and Joseph Modayil. Deep reinforcement learning and the deadly triad. *arXiv preprint arXiv:1812.02648*, 2018.

[48] Luitzen Egbertus Jan Brouwer. Über abbildung von mannigfaltigkeiten. *Mathematische Annalen*, 71(1):97–115, 1911.

[49] Wendelin Böhmer, Vitaly Kurin, and Shimon Whiteson. Deep coordination graphs. In *International Conference on Machine Learning*, 2020.

[50] Wen Sun, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Model-based rl in contextual decision processes: Pac bounds and exponential improvements over model-free approaches. In *Conference on Learning Theory*, pages 2898–2933. PMLR, 2019.

[51] Xiaoyu Chen, Jiachen Hu, Lihong Li, and Liwei Wang. Efficient reinforcement learning in factored mdps with application to constrained rl. In *International Conference on Learning Representations*, 2021.

[52] Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. Contextual decision processes with low bellman rank are pac-learnable. In *International Conference on Machine Learning*, pages 1704–1713. PMLR, 2017.

[53] Max Simchowitz and Kevin G Jamieson. Non-asymptotic gap-dependent regret bounds for tabular mdps. In *Advances in Neural Information Processing Systems*, pages 1151–1160, 2019.

[54] Alekh Agarwal, Nan Jiang, and Sham M Kakade. Reinforcement learning: Theory and algorithms. Technical report, Technical Report, Department of Computer Science, University of Washington, 2019.

[55] Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. Challenges of real-world reinforcement learning. *arXiv preprint arXiv:1904.12901*, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]

(b) Did you describe the limitations of your work? [Yes] We discuss the purpose of all involved assumptions in analyses.

(c) Did you discuss any potential negative societal impacts of your work? [N/A] This paper is a theory paper and does not discuss any direct applications in real world.

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [Yes]

(b) Did you include complete proofs of all theoretical results? [Yes] Please refer to Appendix.

3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We provide codes in the supplemental material.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Please refer to Appendix.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We plot confidence interval in learning curves.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Please refer to Appendix.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes]

(b) Did you mention the license of the assets? [N/A]

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A] We do not create new asserts.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We only use simulators for video games.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] We only use simulators for video games.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]