Towards Understanding Linear Value Decomposition in Cooperative Multi-Agent Q-Learning

Jianhao Wang*, **Zhizhou Ren***, **Beining Han**, **Chongjie Zhang**
Institute for Interdisciplinary Information Sciences
Tsinghua University, Beijing, China
{wjh19, rzz16, hbn18}@mails.tsinghua.edu.cn
chongjie@tsinghua.edu.cn

Abstract

Value decomposition 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 introduce a variant of the fitted Q-iteration framework for analyzing multi-agent Q-learning with value decomposition. Based on this framework, we derive a closed-form solution to the Bellman error minimization with linear value decomposition. With this novel solution, we further reveal two interesting insights: 1) linear value decomposition implicitly implements a classical multi-agent credit assignment called *counterfactual difference rewards*; and 2) multi-agent Q-learning with linear value decomposition requires on-policy data distribution to achieve numerical stability. In the empirical study, our experiments demonstrate the realizability of our theoretical implications in a broad set of complicated tasks. They show that most state-of-the-art deep multi-agent Q-learning algorithms using linear value decomposition cannot efficiently utilize off-policy samples, which may even lead to an unbounded divergence.

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 require MARL to learn decentralized policies for agents to jointly optimize a global cumulative reward signal, and post a number of challenges, including multi-agent credit assignment [4–6], non-stationarity [7, 8], and scalability [3, 9]. Recently, by leveraging the strength of deep learning techniques, cooperative MARL has made a series of great progress [10–19], particularly in value-based methods that demonstrate state-of-the-art performance on challenging tasks such as StarCraft unit micromanagement [20]. Sunehag et al. [11] proposed an approach called value-decomposition network (VDN) based on the paradigm of *centralized training with decentralized execution* [21]. VDN 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 decomposition 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, rather than on any reward specific to agent i. This decomposition technique significantly improves the scalability of multi-agent Q-learning algorithms, and fosters a series of subsequent work, including QMIX [13], QTRAN [14], and Qatten [19].

In spite of the empirical success in a broad class of tasks, multi-agent Q-learning with linear value decomposition has not been theoretically well-understood. Because of its limited representation

*Equal contribution.

Preprint. Under review.

complexity, this linear value decomposition corrupts the completeness of the joint value function class. The approximation error induced by this incompleteness is known as *inherent Bellman error* [22] which usually deviates Q-learning to an unexpected behavior. To develop deeper understanding of learning with value decomposition, this paper introduces a multi-agent variant of the fitted Q-iteration framework and derive a closed-form solution to the involved Bellman error minimization. To best of our knowledge, it is the first theoretical analysis characterizing the internal mechanism of linear value decomposition, which can serve as a powerful toolkit to explore more underlying insights from this popular structure.

By utilizing this novel closed-form solution, this paper reveals two interesting insights:

1. Learning linear value decomposition implicitly implements a classical multi-agent credit assignment method called *counterfactual difference rewards* [4], which draws a connection with COMA [12], a multi-agent policy-gradient method.
2. Multi-agent Q-learning with linear value decomposition requires on-policy data distribution to provide local stability around the optimal solution. This result is perversely counterintuitive, because single-agent Q-learning is an off-policy method in tabular settings [23].

Finally, we set up an extensive set of experiments to demonstrate the realizability of our theoretical implications. According to our measurement, the default implementation of most deep multi-agent Q-learning algorithms using linear value decomposition structure actually runs in a nearly on-policy mode and, when learning with a off-policy static dataset, their Q-values may diverge.

2 Notations and Preliminaries

2.1 Multi-agent Markov Decision Process (MMDP)

To support theoretical analyses on multi-agent Q-learning, we adopt the framework of MMDP [24], a special case of Dec-POMDP [25], to model fully cooperative multi-agent decision-making tasks. MMDP is defined as a tuple $\mathcal{M} = \langle \mathcal{N}, \mathcal{S}, \mathcal{A}, P, r, \gamma \rangle$. $\mathcal{N} \equiv \{1, \dots, n\}$ is a finite set of agents. $\mathcal{S}$ is a finite set of global states. $\mathcal{A}$ 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, the selected joint action $\mathbf{a}_t$ results in a transition $s_{t+1} \sim P(\cdot|s_t, \mathbf{a}_t)$ and a global reward signal $r(s_t, \mathbf{a}_t)$. $\gamma \in [0, 1)$ is a discount factor. The goal for MARL is to construct a joint policy $\boldsymbol{\pi} = \langle \pi_1, \dots, \pi_n \rangle$ maximizing expected discounted rewards $V^{\boldsymbol{\pi}}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, \boldsymbol{\pi}(s_t)) | s_0 = s\right]$, where $\pi_i : \mathcal{S} \mapsto \mathcal{A}$ denotes an individual policy of agent i. The corresponding action-value function is denoted as $Q^{\boldsymbol{\pi}}(s, \mathbf{a}) = r(s, \mathbf{a}) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,\mathbf{a})}[V^{\boldsymbol{\pi}}(s')]$. In addition, we use Q^* and V^* to denote the action-value function and the state-value function corresponding to the optimal policy $\boldsymbol{\pi}^*$, respectively.

2.2 Centralized Training with Decentralized Execution (CTDE)

Most deep multi-agent Q-learning algorithms with value decomposition adopt the paradigm of centralized training with decentralized execution [21]. In the training procedure, the centralized trainer can access to all global information, including the global states, the shared rewards, all agents' polices, and value functions. While in the decentralized execution, every agent needs to make individual decisions based on its local observations. Note that this paper considers MMDP as a simplified setting which rules out the concerns of partial observability. Thus our notations do not distinguish the concepts of states and observations. *Individual-Global-Max* (IGM) [14] is a common principle to meet the requirement for decentralized policy execution. It enforces the action selection consistency between the the global joint action-value Q_{tot} and individual action-values $[Q_i]_{i=1}^n$, which are specified as follows:

$$\forall s \in \mathcal{S},\ \arg\max_{\mathbf{a} \in \mathbf{A}} Q_{\text{tot}}(s, \mathbf{a}) = \left\langle \arg\max_{a_1 \in \mathcal{A}} Q_1(s, a_1), \dots, \arg\max_{a_n \in \mathcal{A}} Q_n(s, a_n) \right\rangle. \tag{1}$$

As stated in Eq. (2), the additivity constraint adopted by VDN [11] is a sufficient condition for the IGM constraint stated in Eq. (1). However, this linear decomposition structure is not a necessary condition and induces a limited joint value function class, because the linear number of individual

functions cannot represent a joint value function class exponential with the number of agents.

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

2.3 Fitted Q-Iteration (FQI) for Multi-Agent Q-Learning

For multi-agent Q-learning with value decomposition, we use Q_{tot} to denote a global but factorized value function, which can be factorized 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 to denote both of them. In the MMDP settings, the shared reward signal can only supervise the training of the joint value function Q_{tot}, which requires us to modify the notation of *Bellman optimality operator* $\mathcal{T}$ as follows:

$$(\mathcal{T}Q)_{\text{tot}}(s, \mathbf{a}) = r(s, \mathbf{a}) + \gamma \mathop{\mathbb{E}}_{s' \sim P(s'|s,\mathbf{a})} \left[\max_{\mathbf{a}' \in \mathbf{A}} Q_{\text{tot}}(s', \mathbf{a}') \right]. \tag{3}$$

Fitted Q-iteration (FQI) [26] provides a unified framework which extends the above operator to solve high-dimensional tasks using function approximation. It follows an iterative optimization framework based on a given dataset $D = \{(s, \mathbf{a}, r, s')\}$,

$$Q^{(t+1)} \leftarrow \mathop{\arg\min}_{Q \in \mathcal{Q}} \mathop{\mathbb{E}}_{(s,\mathbf{a},r,s') \sim D} \left[\left(r + \gamma \max_{\mathbf{a}' \in \mathbf{A}} Q_{\text{tot}}^{(t)}(s', \mathbf{a}') - Q_{\text{tot}}(s, \mathbf{a}) \right)^2 \right], \tag{4}$$

where an initial solution $Q^{(0)}$ is selected arbitrarily from the function class $\mathcal{Q}$. By constructing a specific function class $\mathcal{Q}$ that only contains instances satisfying the IGM condition stated in Eq. (1) [11, 13], the centralized training procedure in Eq. (4) will naturally produces suitable individual values $[Q_i]_{i=1}^n$, from which individual policies can be derived for decentralized execution.

3 Multi-Agent Q-Learning with Linear Value Decomposition

In the literature of deep MARL, constructing a specific function class $\mathcal{Q}$ satisfying the IGM condition is a critical step to realize centralized training with decentralized execution. Linear value decomposition proposed by VDN [11] is a simple method to implement this paradigm, which has become a foundation of deep multi-agent Q-learning with value decomposition.

In this section, we provide a theoretical analysis towards deeper understanding of this popular decomposition structure. Our result is based on a multi-agent variant of fitted Q-iteration (FQI) with linear value decomposition, named FQI-LVD. We drive the closed-form update rule of FQI-LVD, and then reveal the internal mechanism of credit assignment implicitly realized by linear value decomposition techniques, used by VDN and Qatten [19].

3.1 Multi-Agent Fitted Q-Iteration with Linear Value Decomposition (FQI-LVD)

To provide a clear perspective on the effects of linear value decomposition, we make two additional assumptions which both simplify the notations and facilitate the analysis.

Assumption 1 (Deterministic Dynamics)**.** *The transition function $P(\cdot|s, \mathbf{a})$ is deterministic.*

Assumption 1 considers an environment with deterministic transitions, which is a common simplification technique for theoretical analysis in reinforcement learning [27]. This assumption holds in many empirical scenarios, including popular StarCraft2 unit micromanagement benchmark tasks [20] with full observability.

Assumption 2 (Adequate and Factorizable Dataset)**.** *The dataset D contains all applicable state-action pairs $(s, \mathbf{a})$ whose empirical probability is factorizable with respect to the individual behaviors of multiple agents. Formally, let $p_D(\mathbf{a}|s)$ denote the empirical probability of joint action $\mathbf{a}$ executed on state s, which can be factorized to the production of individual components,*

$$p_D(\mathbf{a}|s) = \prod_{i \in \mathcal{N}} p_D(a_i|s), \quad \sum_{a_i \in \mathcal{A}} p_D(a_i|s) = 1, \quad p_D(a_i|s) > 0, \tag{5}$$

where $p_D(a_i|s)$ denotes the empirical probability of the event that agent i executes a_i on state s.

Assumption 2 is based on the fact that an adequate dataset is necessary for FQI algorithms to find an acceptable solution [28, 29]. In practice, the property of factorizable data distribution can be directly induced by a decentralized data collection procedure. When agents performs fully decentralized execution, the empirical probability of the event $(s, \mathbf{a})$ in the collected dataset D is naturally factorized.

Now we define FQI with linear value decomposition as follows.

Definition 1. *[FQI-LVD] Given a dataset D, FQI-LVD specifies the action-value function class*

$$\mathcal{Q}^{LVD} = \left\{ Q \;\middle|\; Q_{tot}(\cdot, \mathbf{a}) = \sum_{i=1}^{n} Q_i(\cdot, a_i), \forall \mathbf{a} \in \mathbf{A} \text{ and } \left[\forall Q_i \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}\right]_{i=1}^{n} \right\} \tag{6}$$

and induces the empirical Bellman operator $\mathcal{T}_D^{LVD} : \mathcal{Q}^{LVD} \mapsto \mathcal{Q}^{LVD}$ as follows.

$$Q^{(t+1)} \leftarrow \mathcal{T}_D^{LVD} Q^{(t)} \equiv \arg\min_{Q \in \mathcal{Q}^{LVD}} \sum_{(s,\mathbf{a}) \in \mathcal{S} \times \mathbf{A}} p_D(\mathbf{a}|s) \left(y^{(t)}(s, \mathbf{a}) - \sum_{i=1}^{n} Q_i(s, a_i) \right)^2, \tag{7}$$

where $y^{(t)}(s, \mathbf{a}) = r(s, \mathbf{a}) + \gamma \max_{\mathbf{a}'} Q_{tot}^{(t)}(s', \mathbf{a}')$ denotes the regression target derived by Bellman optimality operator. Q_{tot} and $[Q_i]_{i=1}^n$ refer to the interfaces of CTDE defined in section 2.3.

Value-decomposition network (VDN) [11] corresponds to an implementation of FQI-LVD, in which the individual value functions $[Q_i]_{i=1}^n$ are parameterized by deep neural networks. The joint value function Q_{tot} can be simply formed by the summation stated in Eq. (2) and thus does not require additional parameters. The specific action-value function class $\mathcal{Q}^{\text{LVD}}$ limits the number of parameters that fits the regression target $y^{(t)}$. We defer the empirical analysis of VQN and other deep multi-agent Q-learning algorithms to Section 6.

3.2 Implicit Credit Assignment in Linear Value Decomposition

In the formulation of FQI-LVD stated in Definition 1, the empirical Bellman error minimization in Eq. (7) can be regarded as a weighted linear least squares problem, which contains $n|\mathcal{S}||\mathcal{A}|$ variables to form individual value functions $[Q_i]_{i=1}^n$ and $|\mathcal{S}||\mathcal{A}|^n$ data points corresponding to all entries of the regression target $y^{(t)}(s, \mathbf{a})$. For this least squares problem, we derive a closed-form solution stated in Theorem 1, which can be verified through *Moore-Penrose inverse* [30] for weighted linear regression analysis. Proofs for all theorems, lemmas, and propositions in this paper are deferred to Appendix.

Theorem 1. *Let $Q^{(t+1)} = \mathcal{T}_D^{LVD} Q^{(t)}$ denote a single iteration of the empirical Bellman operator. Then $\forall i \in \mathcal{N}, \forall (s, \mathbf{a}) \in \mathcal{S} \times \mathbf{A}$, the individual action-value function $Q_i^{(t+1)}(s, a_i) =$*

$$\mathop{\mathbb{E}}_{a'_{-i} \sim p_D(\cdot|s)} \left[y^{(t)}\left(s, a_i \oplus a'_{-i}\right) \right] - \frac{n-1}{n} \mathop{\mathbb{E}}_{\mathbf{a}' \sim p_D(\cdot|s)} \left[y^{(t)}\left(s, \mathbf{a}'\right) \right] + w_i(s), \tag{8}$$

where we denote $a_i \oplus a'_{-i} = \langle a'_1, \ldots, a'_{i-1}, a_i, a'_{i+1}, \ldots, a'_n \rangle$. a'_{-i} denotes the action of all agents except agent i. The residue term $\mathbf{w} \equiv [w_i]_{i=1}^n$ is an arbitrary vector satisfying $\forall s$, $\sum_{i=1}^n w_i(s) = 0$.

As shown in Theorem 1, the local action-value function $Q_i^{(t+1)}$ consists of three terms. The first term is the expectation of one-step TD target value over 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 decomposition (see Eq. (2)).

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 given a joint policy, and the second term is the expected value of the default policy, which is considered as the *counterfactual baseline*. The difference between them corresponds to a credit assignment mechanism called *counterfactual difference rewards*, which has been used by counterfactual multi-agent policy gradient (COMA) [12].

Implication 1. *As shown in Eq. (9), linear value decomposition implicitly implements a counterfactual credit assignment mechanism, which is similar to what is used by COMA.*

$$\underbrace{\mathop{\mathbb{E}}_{a'_{-i}\sim p_D(\cdot|s)}\left[y^{(t)}\left(s, a_i \oplus a'_{-i}\right)\right]}_{\textit{evaluation of the individual action } a_i} - \frac{n-1}{n}\underbrace{\mathop{\mathbb{E}}_{\mathbf{a}'\sim p_D(\cdot|s)}\left[y^{(t)}\left(s, \mathbf{a}'\right)\right]}_{\textit{counterfactual baseline}}. \tag{9}$$

Compared to COMA, this implicit credit assignment is naturally served by Bellman error minimization through linear value decomposition, which is much more scalable. The extra importance weight $(n-1)/n$ can be approximated to 1 if n is sufficiently large. In most practical scenarios, the dataset D is accumulated by running a greedy policy $\boldsymbol{\pi}^{(t)} = \arg\max_{\mathbf{a}} Q_{\text{tot}}^{(t)}(s, \mathbf{a})$ [31], which refers as on-policy learning that will be discussed in the next section.

4 Local Stability of Learning Linear Value Decomposition

In the previous section, we have derived the closed-form update rule of FQI-LVD, which reveals the internal credit assignment mechanism provided by linear value decomposition structure. This derivation also enables us to investigate more algorithmic functionalities of linear value decomposition in multi-agent Q-learning. In this section, we extend the analysis of FQI-LVD to provide some theoretical guarantees on the local convergent property in an on-policy learning mode.

4.1 Learning with On-Policy Data Distribution

As shown in Theorem 1, the choice of training data distribution is critical to the output of the empirical Bellman operator $\mathcal{T}_D^{\text{LVD}}$. From this perspective, it is natural to consider which data distribution can benefit multi-agent Q-learning with linear value decomposition. In this section, we switch to study the behavior of linear value decomposition in an on-policy setting, as depicted by Algorithm 1.

Algorithm 1 On-Policy Fitted Q-Iteration with ϵ-greedy Exploration

1: Initialize $Q^{(0)}$.
2: **for** $t = 0 \dots T-1$ **do** ▷ T denotes the computation budget
3: Construct an exploratory policy $\tilde{\boldsymbol{\pi}}_t$ based on $Q^{(t)}$. ▷ i.e., ϵ-greedy exploration

$$\tilde{\boldsymbol{\pi}}_t(a|s) = \prod_{i=1}^{n}\left(\frac{\epsilon}{|\mathcal{A}|} + (1-\epsilon)\mathbb{I}\left[a_i = \arg\max_{a'_i \in \mathcal{A}} Q_i^{(t)}(s, a'_i)\right]\right) \tag{10}$$

4: Collect a new dataset D_t by running $\tilde{\boldsymbol{\pi}}_t$.
5: Operate an on-policy Bellman operator $Q^{(t+1)} \leftarrow \mathcal{T}_\epsilon^{\text{LVD}} Q^{(t)} \equiv \mathcal{T}_{D_t}^{\text{LVD}} Q^{(t)}$.

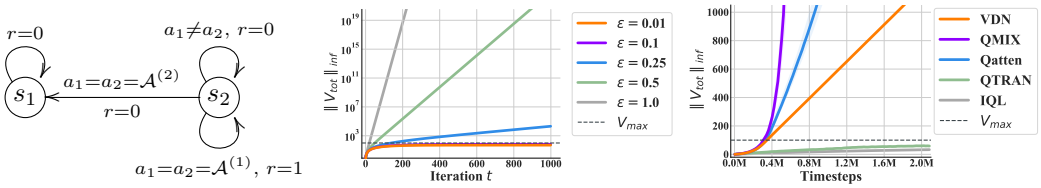
Algorithm 1 is a variant of fitted Q-iteration which adopts an on-policy sample distribution. At line 3, an exploratory noise is integrated into the greedy policy, since the function approximator generally requires an extensive set of samples to regularize extrapolation values. Particularly, we investigate a standard exploration module called ϵ-greedy, in which every agent takes a small probability to explore actions with non-maximum values. To make the underlying insights more accessible, we assume the data collection procedure at line 4 can obtain infinite samples, which makes the dataset D_t become a sufficient coverage over the state-action space (see Assumption 2). This algorithmic framework serves as a foundation for discussions on the local stability.

4.2 Local Stability Near the Optimal Solution

As a preliminary step, we investigate the local convergence of FQI-LVD, which contributes to setting up a more detailed characterization for the algorithmic functionality of linear value decomposition. In this section, we consider an additional assumption stated as follows.

Assumption 3 (Unique Optimal Policy)**.** *The optimal policy $\boldsymbol{\pi}^*$ is unique.*

The intuitive motivation of this assumption is to have the optimal policy $\boldsymbol{\pi}^*$ be a potential stable solution. In situations where the optimal policy is not unique, most Q-learning algorithms will oscillate around multiple optimal policies [32], and Assumption 3 helps us to rule out these non-interesting cases. Based on this setting, the local stability of FQI-LVD can be characterized by the following lemma.



(a) A simple counterexample (b) Algorithm 1 (c) Deep MARL

Figure 1: (a) An MMDP where FQI-LVD will diverge to infinity when $\gamma \in \left(\frac{4}{5}, 1\right)$. r is a shorthand for $r(s, \mathbf{a})$ and the action space for each agent $\mathcal{A} \equiv \left\{\mathcal{A}^{(1)}, \ldots, \mathcal{A}^{(|\mathcal{A}|)}\right\}$. (b) The learning curve of $\|V_{\text{tot}}\|_\infty$ while running Algorithm 1 with different hyper-parameters ϵ on the given MMDP. (c) The learning curve of $\|V_{\text{tot}}\|_\infty$ while running several deep multi-agent Q-learning algorithms. In addition, $V_{\text{max}} = \frac{1}{1-\gamma} = 10^2$ denotes the maximum possible value.

Lemma 1. *There exists a threshold $\delta > 0$ such that the on-policy Bellman operator $\mathcal{T}_\epsilon^{LVD}$ is closed in the following subspace $\mathcal{B} \subset \mathcal{Q}^{LVD}$, when the hyper-parameter ϵ is sufficiently small.*

$$\mathcal{B} = \left\{ Q \in \mathcal{Q}^{LVD} \;\middle|\; \boldsymbol{\pi}_Q = \boldsymbol{\pi}^*,\ \max_{s \in \mathcal{S}} |Q_{tot}(s, \boldsymbol{\pi}^*(s)) - V^*(s)| \leq \delta \right\}$$

Formally, $\exists \delta > 0$, $\exists \epsilon > 0$, $\forall Q \in \mathcal{B}$, there must be $\mathcal{T}_\epsilon^{LVD} Q \in \mathcal{B}$.

Lemma 1 indicates that, once the value function Q steps into the subspace $\mathcal{B}$, the induced policy $\boldsymbol{\pi}_Q$ will converge to the optimal policy $\boldsymbol{\pi}^*$. By combining this local stability with Brouwer's fixed-point theorem [33], we can further verify the existence of a fixed-point solution for the on-policy Bellman operator $\mathcal{T}_\epsilon^{\text{LVD}}$ (see Theorem 2).

Theorem 2. *Algorithm 1 will have a fixed point value function expressing the optimal policy if the hyper-parameter ϵ is sufficiently small.*

Theorem 2 and Lemma 1 indicate that, multi-agent Q-learning with linear value decomposition has a convergent region, where the value function induces optimal actions. Note that $\mathcal{Q}^{\text{LVD}}$ is a limited function class, which even cannot guarantee to contain the optimal joint value function Q^*. From this perspective, on-policy data distribution becomes necessary to make the one-step TD target projected to a small set of critical state-action pairs, which help construct the stable subspace $\mathcal{B}$ stated in Lemma 1. In Section 6, we will provide an empirical analysis to verify the connection between the algorithmic stability and the online data collection procedure.

5 Unbounded Divergence in Off-Policy Training

In this section, we will provide an analysis on the convergence of FQI-LVD in off-policy training. In general, its convergence guarantee is negative. Even the local stability guarantee discussed in the previous section does not hold for off-policy settings, which reveals a notable limitation of multi-agent Q-learning with linear value decomposition. The major reason is stated in the following proposition.

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

$$\forall Q, Q' \in \mathcal{Q}, \quad \|\mathcal{T}Q - \mathcal{T}Q'\| \leq \gamma \|Q - Q'\|_\infty \tag{11}$$

In the convergence proof for the standard *Bellman optimality operator* $\mathcal{T}$, Eq. (11) is a critical step to derive the theoretical guarantee on global convergence. In the context of FQI-LVD, the additivity constraint limits the joint value function class that it can represent, which deviates the empirical Bellman operator $\mathcal{T}_D^{\text{LVD}}$ from the original *Bellman optimality operator* $\mathcal{T}$ (see Theorem 1). This deviation is also known as *inherent Bellman error* [22], which corrupts a broad set of stability properties, including γ-contraction discussed in Proposition 1.

To serve a concrete example, we construct a simple MMDP with two agents, two global states, and two actions (see Figure 1a). 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 obtain a positive reward. Figure 1b visualizes the performance of Algorithm 1 with different values of the hyper-parameter ϵ. With a smaller ϵ, the collected dataset is closer to an on-policy distribution. The simulation results matches the implication

provided by section 4.2 that an on-policy sample distribution can benefit the numerical stability. As a special case, $\epsilon = 1.0$ refers to a completely off-policy setting with a uniform data distribution, in which an unbounded divergence can be observed as depicted by the following proposition.

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

Note that the unbounded divergence discussed in Propostion 2 would happen to an arbitrary initialization $Q^{(0)}$. In other words, the convergent region stated in Lemma 1 does not exist in the off-policy setting. To provide an implication for practical scenarios, we also investigate the performance of several deep multi-agent Q-learning algorithms in this MMDP. As shown in Figure 1c, VDN [11] and and Qatten [19], two deep-learning-based implementations of FQI-LVD, result in an unbounded divergence. QMIX [13], which has a non-linear monotonic factorization but still underrepresent the IGM function class, also diverge to infinity. As a positive case, QTRAN [14] and IQL [34], two well-known algorithms that can represent any joint value function under the IGM constraint, perform an outstanding numerical stability. These results imply the following insight.

Implication 2. *Multi-agent Q-learning with linear value decomposition structure requires on-policy samples to maintain numerical stability, which is counterintuitive to a conventional thought that Q-learning is an off-policy method.*

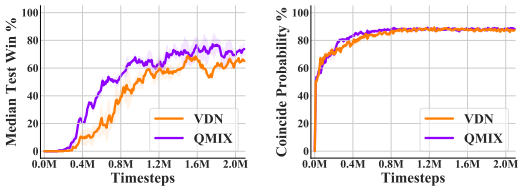
We hypothesize that multi-agent Q-learning with linear value decomposition is more like an on-policy algorithm, rather than an off-policy method as people thought before. In the next section, we will set up extensive experiments on a broad class of high-dimensional tasks to verify this hypothesis.

6 Empirical Analysis

In this section, we conduct an empirical study to connect our theoretical results with the practical scenarios of deep multi-agent Q-learning algorithms. Our experiments are based on StarCraft Multi-Agent Challenge (SMAC) benchmark tasks and open-source implementations of state-of-the-art algorithms provided by Samvelyan et al. [20]. Detailed experiment settings refer to Appendix.

6.1 To what extent does the training procedure of deep multi-agent Q-learning resemble on-policy learning?

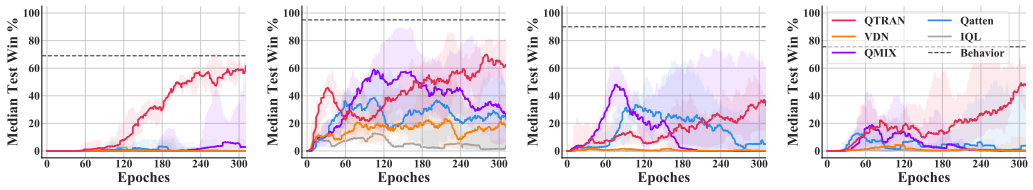
Most deep multi-agent Q-learning algorithms adopt the same buffer management mechanism as DQN [31], which uses a first-in first-out (FIFO) buffer to make learning focus on recently collected transitions. Combining with a modest exploration strategy, e.g., ϵ-greedy, the sample distribution in the replay buffer will be close to an on-policy distribution. The most widely-used open source implementations [20] of deep multi-agent Q-learning algorithms use a much lower training frequency comparing to DQN. They perform a single gradient step after collecting every episode, whereas DQN performs one update after collecting every transition. Moreover, these default implementations [20] also use a much smaller buffer containing only 5000 recent episodes, comparing to the replay buffer used by DQN with 10^6 recent transitions. The above comparison reveals that the default implementations of deep multi-agent Q-learning algorithms flushes the replay buffer in a higher frequency, which makes the training procedure more resemble on-policy learning.



(a) Test Win Rate % (b) Coincide Prob. %

Figure 2: Investigating the training procedure of VDN and QMIX in the StarCraft2 task 5m_vs_6m.

In the experiment shown in Figure 2, we investigate the training procedure of two state-of-the-art multi-agent Q-learning algorithms, VDN [11] and QMIX [13]. VDN corresponds to an advanced implementation of FQI-LVD stated in Definition 1, and QMIX extends its function class to a monotonic decomposition. The learning performance of these algorithms can reach nearly 70% *test win rate* shown in the right panel, where the replay buffer is managed by a FIFO rule. The hyper-parameter of ϵ-greedy exploration is $\epsilon = 0.05$. To measure the extent of on-policy distribution induced by a FIFO



(a) 5m_vs_6m (b) 2s3z (c) 1c3s5z (d) 2c_vs_64zg

Figure 3: Evaluating the learning performance of deep multi-agent Q-learning algorithms with a given static dataset. The detailed experiment setting is deferred to Appendix.

replay buffer, we design a special measurement named *coincide probability* $h(\cdot, \cdot)$ defined as follows:

$$h(D_t, \boldsymbol{\pi}^{(t)}) = \mathop{\mathbb{E}}_{(s,\mathbf{a})\sim D_t} \left[\frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\left[a_i = \pi_i^{(t)}(s) \right] \right] \tag{12}$$

where $\boldsymbol{\pi}^{(t)}$ and D_t denote the immediate joint policy and the content of the FIFO buffer at training iteration t, respectively. Recalling the analysis in section 4, if we consider the empirical probability in the replay buffer as a certain policy, the *coincide probability* measurement can be regarded as an approximation of the imaginary hyper-parameter, i.e., $\epsilon \approx 1 - h(D_t, \boldsymbol{\pi}^{(t)})$. As shown in Figure 2, any significant policy improvement can only be observed after reaching a high *coincide probability*, which resembles an on-policy training distribution. It suggests that collecting on-policy samples may be an important condition to achieve stable performance. To strengthen this argument, we will investigate an off-policy training mode in Section 6.2 to reveal more underlying insights.

6.2 Does deep multi-agent Q-learning really require on-policy samples?

In the literature of off-policy reinforcement learning with function approximation, there is a long history studying the behavior of Q-learning with non-universal function classes. Many counterexamples have been proposed to indicate certain function classes suffering from oscillation [35] or even unbounded divergence [36, 37]. Our analysis in section 5 corresponds to a case study of the function class $\mathcal{Q}^{\text{LVD}}$ with linear value decomposition structure. This function class $\mathcal{Q}^{\text{LVD}}$ is restricted by the additivity constraint stated in Eq. (2), which induces the functional incompleteness and may lead to unbounded divergence in some cases.

When combined with deep neural networks, FQI-LVD will have a possibly further reduced function class $\mathcal{Q}^{\text{LVD}}$ because of the limited function representation of certain network architectures. To investigate the performance of deep multi-agent Q-learning in an off-policy training mode, we adopt a similar experiment setting with Fujimoto et al. [38]. This setting is named *batch reinforcement learning* in which the learning algorithm can only access to a given static dataset and cannot perform extra exploration. Differing with other related work studying extrapolation error [38–40], we aim to adopt a diverse dataset simulating a uniform data distribution, which makes the function expressiveness of value decomposition become the dominant factor to be investigated. This dataset is constructed by training a behavior policy and collecting all its experienced transitions during the whole training procedure. Notice that we need not consider the problem of exploration while using this static dataset, because the behavior policy has been well trained, which will achieve high sample efficiency by this off-policy nature.

Based on the above experiment setting, we investigate a number of advanced algorithms, including VDN [11], QMIX [13], Qatten [19], QTRAN [14], and IQL [34]. As shown in Figure 3, most of these algorithms cannot utilize the off-policy dataset collected by an unfamiliar behavior policy. As a comparative experiment on the 5m_vs_6m task, Figure 2a and Figure 3a demonstrate that VDN and QMIX can achieve superior performance during on-policy training but perform poorly with off-policy data collection, which matches the implication derived by section 5. In particular, QTRAN uses the richest function class among these algorithms. Its function class is directly restricted by the IGM constraint through two ℓ_2-penalties, which induces a superset of $\mathcal{Q}^{\text{LVD}}$. This superior of function expressivity definitely leads to a better performance in off-policy training, but still has a considerable gap from the performance of behavior policies (e.g., see Figure 3c).

Our experiments of off-policy training indicate a fact that the on-policy data collection procedure is extremely important for multi-agent Q-learning with a limited function class. The functionality

of on-policy data collection not only serves for exploration but also contributes a lot to maintaining algorithmic stability.

7 Conclusion

In this paper, we make an initial effort to provide theoretical analyses on multi-agent Q-learning with value decomposition. We derive a closed-form solution to the Bellman error minimization with linear value decomposition. Based on this novel result, we reveal the implicit credit assignment mechanism of linear value decomposition and provide an formal analysis on its learning stability and convergence. Our results indicate that linear value decomposition structure induces the incompleteness of the joint value function class. This incompleteness would hurt the numerical stability for off-policy learning. Benefiting from a richer value function class, it is shown that QTRAN [14] achieves a little more stable performance while training with a static dataset. However, empirical results show that, to address computation complexity, QTRAN adopts approximated techniques that result in instability and poor performance in complicated domains [20, 41]. Therefore, it is a valuable future direction to construct a novel multi-agent reinforcement learning architecture with a rich value function class that achieves both learning efficiency and scalability.

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] Adrian K Agogino and Kagan Tumer. Unifying temporal and structural credit assignment problems. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 980–987. IEEE Computer Society, 2004.

[6] 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.

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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] Tonghan Wang, Jianhao Wang, Yi Wu, and Chongjie Zhang. Influence-based multi-agent exploration. *arXiv preprint arXiv:1910.05512*, 2019.

[16] Tonghan Wang, Jianhao Wang, Chongyi Zheng, and Chongjie Zhang. Learning nearly decomposable value functions via communication minimization. *arXiv preprint arXiv:1910.05366*, 2019.

[17] Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. Emergent tool use from multi-agent autocurricula. *arXiv preprint arXiv:1909.07528*, 2019.

[18] Tonghan Wang, Heng Dong, Victor Lesser, and Chongjie Zhang. Multi-agent reinforcement learning with emergent roles. *arXiv preprint arXiv:2003.08039*, 2020.

[19] 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.

[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] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8(3-4):279–292, 1992.

[24] 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.

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

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

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

[28] 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.

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

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

[31] 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.

[32] 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.

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

[34] Ming Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 330–337, 1993.

[35] Geoffrey J Gordon. Stable function approximation in dynamic programming. In *Machine Learning Proceedings 1995*, pages 261–268. Elsevier, 1995.

[36] Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*, pages 30–37. Elsevier, 1995.

[37] John N Tsitsiklis and Benjamin Van Roy. Feature-based methods for large scale dynamic programming. *Machine Learning*, 22(1-3):59–94, 1996.

[38] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pages 2052–2062, 2019.

[39] Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. In *Advances in Neural Information Processing Systems*, pages 11761–11771, 2019.

[40] Scott Fujimoto, Edoardo Conti, Mohammad Ghavamzadeh, and Joelle Pineau. Benchmarking batch deep reinforcement learning algorithms. *arXiv preprint arXiv:1910.01708*, 2019.

[41] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *arXiv preprint arXiv:2003.08839*, 2020.

A Omitted Proofs in Section 3

Lemma 2. *Considering following weighted linear regression problem*

$$\min_{\mathbf{x}} \parallel \sqrt{\mathbf{p}^\top} \cdot (\mathbf{A}\mathbf{x} - \mathbf{b}) \parallel_2^2 \tag{13}$$

where $\mathbf{A} \in \mathbb{R}^{m^n \times mn}, \mathbf{x} \in \mathbb{R}^{mn}, \mathbf{b}, \mathbf{p} \in \mathbb{R}^{m^n}$, $m, n \in \mathrm{Z}^+$. *Besides,* $\mathbf{A}$ *is m-ary encoding matrix namely* $\forall i \in [m^n], j \in [mn]$

$$\mathbf{A}_{i,j} = \begin{cases} 1, & \textit{if} \quad \exists u \in [n], j = m \times u + (\lfloor i/m^u \rfloor \textit{ mod } m), \\ 0, & \textit{otherwise}. \end{cases} \tag{14}$$

For simplicity, j^{th} *row of* $\mathbf{A}$ *corresponds to a m-ary number* $\vec{a}_j = (j)_m$ *where* $\vec{a} = a_0 a_1 \dots a_{n-1}$, *with* $a_u \in [m], \forall u \in [n]$. *Assume* $\mathbf{p}$ *is a positive vector which follows that*

$$\mathbf{p}_j = \mathbf{p}(\vec{a}_j) = \prod_{u \in [n]} p_u(a_{u,j}), \textit{ where } p_u : [m] \to (0,1) \textit{ and } \sum_{a_u \in [m]} p_u(a_u) = 1, \forall u \in [n] \tag{15}$$

The optimal solution of this problem is the following. Denote $i = u \times m + v, v \in [m], u \in [n]$ *and an arbitrary vector* $\mathbf{w} \in \mathbb{R}^{mn}$

$$\mathbf{x}_i^* = \sum_{\vec{a}} \frac{\mathbf{p}(\vec{a})}{p_u(a_u)} \mathbf{b}_{\vec{a}} \cdot \mathbf{1}(a_u = v) - \frac{n-1}{n} \mathbf{p}(\vec{a}) \mathbf{b}_{\vec{a}} - \frac{1}{mn} \sum_{i' \in [mn]} \mathbf{w}_{i'} + \frac{1}{m} \sum_{v' \in [m]} \mathbf{w}_{um+v'} \tag{16}$$

Proof. For brevity, denote

$$\mathbf{A}^p = \sqrt{\mathbf{p}^\top} \cdot \mathbf{A}, \qquad \mathbf{b}^p = \sqrt{\mathbf{p}^\top} \cdot \mathbf{b} \tag{17}$$

Then the weighted linear regression becomes a standard Linear regression problem w.r.t $\mathbf{A}^p, \mathbf{b}^p$. To compute the optimal solutions, we need to calculate the Moore-Penrose inverse of $\mathbf{A}^p$. The sufficient and necessary condition of this inverse matrix $\mathbf{A}^{p,\dagger} \in \mathbb{R}^{mn \times m^n}$ is the following three statements [30]:

$$(1)\ \mathbf{A}^p \mathbf{A}^{p,\dagger} \text{ and } \mathbf{A}^{p,\dagger} \mathbf{A}^p \text{ are self-adjoint} \tag{18}$$

$$(2)\ \mathbf{A}^p = \mathbf{A}^p \mathbf{A}^{p,\dagger} \mathbf{A}^p \tag{19}$$

$$(3)\ \mathbf{A}^{p,\dagger} = \mathbf{A}^{p,\dagger} \mathbf{A}^p \mathbf{A}^{p,\dagger} \tag{20}$$

We consider the following matrix as $\mathbf{A}^{p,\dagger}$ and we prove that it satisfies all three statements. For $\forall i \in [mn], i = u \times m + v, u \in [n], v \in [m], j \in [m^n]$

$$\begin{aligned} \mathbf{A}_{i,j}^{p,\dagger} &= \mathbf{A}_{i,\vec{a}_j}^{p,\dagger} \\ &= \sqrt{\frac{\mathbf{p}(\vec{a}_{-u,j})}{p_u(a_{u,j})}} \cdot 1(a_{u,j} = v) - \frac{n-1}{n} \sqrt{\mathbf{p}(\vec{a}_j)} - \frac{1}{m} \sqrt{\frac{\mathbf{p}(\vec{a}_{-u,j})}{p_u(a_{u,j})}} + \frac{1}{mn} \sum_{u'=0}^{n-1} \sqrt{\frac{\mathbf{p}(\vec{a}_{-u',j})}{p_{u'}(a_{u',j})}} \end{aligned} \tag{21}$$

where $\mathbf{p}(\vec{a}_{-u}) = \prod_{u' \neq u} p_{u'}(a_{u'})$.

First, we verify that $\mathbf{A}^p \mathbf{A}^{p,\dagger}$ is a $m^n \times m^n$ self-adjoint matrix in statement (1). For simplicity, $O(\vec{a}_i, \vec{a}_j) = \{u | a_{u,i} = a_{u,j}, u \in [n]\}$.

$$\begin{aligned} (\mathbf{A}^p \mathbf{A}^{p,\dagger})_{i,j} &= \sum_{u \in [n]} \sqrt{\mathbf{p}(\vec{a}_i)} [\sqrt{\frac{\mathbf{p}(\vec{a}_{-u,j})}{p_u(a_{u,j})}} \cdot \mathbf{1}(a_{u,j} = a_{u,i}) - \frac{n-1}{n} \sqrt{\mathbf{p}(\vec{a}_j)} - \frac{1}{m} \sqrt{\frac{\mathbf{p}(\vec{a}_{-u,j})}{p_u(a_{u,j})}} \\ &\quad + \frac{1}{mn} \sum_{u'=0}^{n-1} \sqrt{\frac{\mathbf{p}(\vec{a}_{-u',j})}{p_{u'}(a_{u',j})}}] \\ &= \sum_{u \in O(\vec{a}_i, \vec{a}_j)} \frac{\sqrt{\mathbf{p}(\vec{a}_j)\mathbf{p}(\vec{a}_i)}}{p_u(a_{u,j})} - \frac{n-1}{n} \sum_{u \in [n]} \sqrt{\mathbf{p}(\vec{a}_i)\mathbf{p}(\vec{a}_j)} - \frac{1}{m} \sum_{u \in [n]} \frac{\sqrt{\mathbf{p}(\vec{a}_j)\mathbf{p}(\vec{a}_i)}}{p_u(a_{u,j})} \end{aligned}$$

$$
\begin{aligned}
&+\sum_{u\in[n]}\frac{1}{mn}\sum_{u'=0}^{n-1}\frac{\sqrt{\mathbf{p}(\vec{a}_j)\mathbf{p}(\vec{a}_i)}}{p_{u'}(a_{u',j})}\\
=&\sum_{u\in O(\vec{a}_i,\vec{a}_j)}\frac{\sqrt{\mathbf{p}(\vec{a}_{j'})\mathbf{p}(\vec{a}_i)}}{p_u(a_{u,j})}-(n-1)\sqrt{\mathbf{p}(\vec{a}_i)\mathbf{p}(\vec{a}_j)}-\frac{1}{m}\sum_{u\in[n]}\frac{\sqrt{\mathbf{p}(\vec{a}_j)\mathbf{p}(\vec{a}_i)}}{p_u(a_{u,j})}\\
&+\frac{1}{m}\sum_{u\in[n]}\frac{\sqrt{\mathbf{p}(\vec{a}_j)\mathbf{p}(\vec{a}_i)}}{p_u(a_{u,j})}\\
=&\sum_{u\in O(\vec{a}_i,\vec{a}_j)}\frac{\sqrt{\mathbf{p}(\vec{a}_j)\mathbf{p}(\vec{a}_i)}}{p_u(a_{u,j})}-(n-1)\sqrt{\mathbf{p}(\vec{a}_i)\mathbf{p}(\vec{a}_j)} \qquad (22)
\end{aligned}
$$

Observe that $p_u(a_{u,j})=p_u(a_{u,i})$ if $a_{u,i}=a_{u,j}$, thus $(\mathbf{A}^p\mathbf{A}^{p,\dagger})_{i,j}=(\mathbf{A}^p\mathbf{A}^{p,\dagger})_{j,i}$ for any $i,j\in[m^n]$. This proves that $\mathbf{A}^p\mathbf{A}^{p,\dagger}$ is self-adjoint.

Second, we prove that $\mathbf{A}^{p,\dagger}\mathbf{A}^p$ is a $mn\times mn$ self-adjoint matrix and has surprisingly succinct form. Let $i=u\times m+v, u\in[n], v\in[m]$.

1. $i=i'$. Besides, $O(i)=\{\vec{a}\in[m^n]|a_u=v\}$

$$
\begin{aligned}
(\mathbf{A}^{p,\dagger}\mathbf{A}^p)_{i,i}=&\sum_{\vec{a}\in O(i)}\sqrt{\mathbf{p}(\vec{a})}[\sqrt{\frac{\mathbf{p}(\vec{a}_{-u})}{p_u(a_u)}}\cdot\mathbf{1}(a_u=v)-\frac{n-1}{n}\sqrt{\mathbf{p}(\vec{a})}-\frac{1}{m}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u})}{p_u(a_u)}}\\
&+\frac{1}{mn}\sum_{u'=0}^{n-1}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u'})}{p_{u'}(a_{u'})}}]\\
=&\sum_{\vec{a}\in O(i)}\frac{\mathbf{p}(\vec{a})}{p_u(a_u)}-\frac{n-1}{n}\mathbf{p}(\vec{a})-\frac{1}{m}\frac{\mathbf{p}(\vec{a})}{p_u(a_u)}+\frac{1}{mn}\sum_{u'=0}^{n-1}\frac{\mathbf{p}(\vec{a})}{p_{u'}(a_{u'})}\\
=&\sum_{\vec{a}\in O(i)}\left(\mathbf{p}(\vec{a}_{-u})-\frac{1}{m}\mathbf{p}(\vec{a}_{-u})+\frac{1}{mn}\sum_{u'=0}^{n-1}\mathbf{p}(\vec{a}_{-u'})\right)-\frac{n-1}{n}p_u(a_u=v)\\
=&1-\frac{1}{m}-\frac{n-1}{n}p_u(a_u=v)+\frac{1}{mn}\sum_{\substack{u'\in[n]\\u'\neq u}}\sum_{\vec{a}\in O(i)}\mathbf{p}(\vec{a}_{-u'})\\
&+\frac{1}{mn}\sum_{\vec{a}\in O(i)}\mathbf{p}(\vec{a}_{-u})\\
=&1-\frac{1}{m}-\frac{n-1}{n}p_u(a_u=v)+\frac{1}{mn}+\frac{n-1}{mn}mp_u(a_u=v)\\
=&1-\frac{1}{m}+\frac{1}{mn} \qquad (23)
\end{aligned}
$$

2. $i=u\times m+v, i'=u\times m+v', v\neq v'$. This implies that $Q(i)\cap O(i')=\emptyset$

$$
\begin{aligned}
(\mathbf{A}^{p,\dagger}\mathbf{A}^p)_{i,i'}=&\sum_{\vec{a}\in O(i')}\sqrt{\mathbf{p}(\vec{a})}[\sqrt{\frac{\mathbf{p}(\vec{a}_{-u})}{p_u(a_u)}}\cdot\mathbf{1}(a_u=v)-\frac{n-1}{n}\sqrt{\mathbf{p}(\vec{a})}\\
&-\frac{1}{m}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u})}{p_u(a_u)}}+\frac{1}{mn}\sum_{u'=0}^{n-1}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u'})}{p_{u'}(a_{u'})}}]\\
=&\sum_{\vec{a}\in O(i)\cap O(i')}\frac{\mathbf{p}(\vec{a})}{p_u(a_u)}-\frac{n-1}{n}\sum_{\vec{a}\in O(i')}\mathbf{p}(\vec{a})-\frac{1}{m}\sum_{\vec{a}\in O(i')}\frac{\mathbf{p}(\vec{a})}{p_u(a_u)}\\
&+\frac{1}{mn}\sum_{\substack{u'\in[n]\\u'\neq u}}\sum_{\vec{a}\in O(i')}\frac{\mathbf{p}(\vec{a})}{p_{u'}(a_{u'})}+\frac{1}{mn}\sum_{\vec{a}\in O(i')}\frac{\mathbf{p}(\vec{a})}{p_u(a_u)}
\end{aligned}
$$

$$= -\frac{n-1}{n}p_u(a_u = v') - \frac{1}{m} + \frac{n-1}{mn}\sum_{\vec{a}\in O(i')}\mathbf{p}(\vec{a}_{-u'}) + \frac{1}{mn}$$
$$= -\frac{1}{m} + \frac{1}{mn} \tag{24}$$

3. $i = u_1 \times m + v_1, i' = u_2 \times m + v_2, u_1 \neq u_2$.

$$\begin{aligned}(\mathbf{A}^{p,\dagger}\mathbf{A}^p)_{i,i'} = &\sum_{\vec{a}\in O(i')}\sqrt{\mathbf{p}(\vec{a})}[\sqrt{\frac{\mathbf{p}(\vec{a}_{-u_1})}{p_{u_1}(a_{u_1})}}\cdot\mathbf{1}(a_{u_1} = v) - \frac{n-1}{n}\sqrt{\mathbf{p}(\vec{a})}\\
&- \frac{1}{m}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u_1})}{p_{u_1}(a_{u_1})}} + \frac{1}{mn}\sum_{u'=0}^{n-1}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u'})}{p_{u'}(a_{u'})}}]\\
= &\sum_{\vec{a}\in O(i)\cap O(i')}\frac{\mathbf{p}(\vec{a})}{p_{u_1}(a_{u_1})} - \frac{n-1}{n}\sum_{\vec{a}\in O(i')}\mathbf{p}(\vec{a}) - \frac{1}{m}\sum_{\vec{a}\in O(i')}\frac{\mathbf{p}(\vec{a})}{p_{u_1}(a_{u_1})}\\
&+ \frac{1}{mn}\sum_{\substack{u'\in[n]\\u'\neq u_2}}\sum_{\vec{a}\in O(i')}\frac{\mathbf{p}(\vec{a})}{p_{u'}(a_{u'})} + \frac{1}{mn}\sum_{\vec{a}\in O(i')}\frac{\mathbf{p}(\vec{a})}{p_{u_2}(a_{u_2})}\\
= &\, p_{u_2}(a_{u_2}) - \frac{n-1}{n}p_{u_2}(a_{u_2}) - p_{u_2}(a_{u_2}) + \frac{n-1}{mn}mp_{u_2}(a_{u_2}) + \frac{1}{mn}\\
= &\,\frac{1}{mn}\end{aligned} \tag{25}$$

Observe that $\mathbf{A}^{p,\dagger}\mathbf{A}^p$ is self-adjoint by equation (2,3,4) and the expression is succinct.

Third, we verify statement (2). Since we have compute $\mathbf{A}^{p,\dagger}\mathbf{A}^p$, the verification is straightforward. For brevity, denote $\mathbf{A}^{p,\dagger}\mathbf{A}^p$ as $\mathbf{A}_0^p$

$$\begin{aligned}(\mathbf{A}^p\mathbf{A}_0^p)_{\vec{a},i} &= \sqrt{\mathbf{p}(\vec{a})}\sum_{u\in[n]}(\mathbf{A}_0^p)_{um+a_u,i}\\
&= \sqrt{\mathbf{p}(\vec{a})}\left(\mathbf{1}(\exists u\in[n], i = um + a_u) - \frac{1}{m} + \frac{1}{mn} + (n-1)\frac{1}{mn}\right)\\
&= \sqrt{\mathbf{p}(\vec{a})}\cdot\mathbf{1}(\exists u\in[n], i = um + a_u)\end{aligned} \tag{26}$$

Thus, $\mathbf{A}^p\mathbf{A}^{p,\dagger}\mathbf{A}^p = \mathbf{A}^p$.

Similarly, we can verify statement (3). Suppose $i_0 = u_0 \times m + v_0$, we have

$$\begin{aligned}(\mathbf{A}_0^p\mathbf{A}^{p,\dagger})_{i_0,\vec{a}} = &\frac{1}{mn}\sum_{\substack{u\neq u_0\\u\in[n]}}\sum_{v\in[m]}[\sqrt{\frac{\mathbf{p}(\vec{a}_{-u})}{p_u(a_u)}}\cdot\mathbf{1}(a_u = v)\\
&- \frac{n-1}{n}\sqrt{\mathbf{p}(\vec{a})} - \frac{1}{m}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u})}{p_u(a_u)}} + \frac{1}{mn}\sum_{u'=0}^{n-1}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u'})}{p_{u'}(a_{u'})}}]\\
&+ \sum_{v\in[m]}(\mathbf{1}(v = v_0) - \frac{1}{m} + \frac{1}{mn})[\sqrt{\frac{\mathbf{p}(\vec{a}_{-u_0})}{p_{u_0}(a_{u_0})}}\cdot\mathbf{1}(a_{u_0} = v)\\
&- \frac{n-1}{n}\sqrt{\mathbf{p}(\vec{a})} - \frac{1}{m}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u_0})}{p_{u_0}(a_{u_0})}} + \frac{1}{mn}\sum_{u'=0}^{n-1}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u'})}{p_{u'}(a_{u'})}}]\\
= &\frac{1}{mn}\sum_{u\in[n]}\sum_{v\in[m]}[\sqrt{\frac{\mathbf{p}(\vec{a}_{-u})}{p_u(a_u)}}\cdot\mathbf{1}(a_u = v)\end{aligned}$$

$$
\begin{aligned}
&- \frac{n-1}{n}\sqrt{\mathbf{p}(\vec{a})} - \frac{1}{m}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u})}{p_u(a_u)}} + \frac{1}{mn}\sum_{u'=0}^{n-1}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u'})}{p_{u'}(a_{u'})}}] \\
&+ \sum_{v\in[m]}(\mathbf{1}(v=v_0) - \frac{1}{m})[-\frac{n-1}{n}\sqrt{\mathbf{p}(\vec{a})} - \frac{1}{m}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u_0})}{p_{u_0}(a_{u_0})}} \\
&+ \frac{1}{mn}\sum_{u'=0}^{n-1}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u'})}{p_{u'}(a_{u'})}}] + \sum_{v\in[m]}(\mathbf{1}(v=v_0) - \frac{1}{m})\sqrt{\frac{\mathbf{p}(\vec{a}_{-u_0})}{p_{u_0}(a_{u_0})}}\cdot\mathbf{1}(a_{u_0}=v) \\
=& \frac{1}{mn}\sum_{u\in[n]}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u})}{p_u(a_u)}} - \frac{n-1}{n}\sqrt{\mathbf{p}(\vec{a})} \\
&+ \frac{1}{n}\sum_{u\in[n]}[-\frac{1}{m}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u})}{p_u(a_u)}} + \frac{1}{mn}\sum_{u'=0}^{n-1}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u'})}{p_{u'}(a_{u'})}}] \\
&+ \left(\sum_{v\in[m]}(\mathbf{1}(v=v_0) - \frac{1}{m})\right)[-\frac{n-1}{n}\sqrt{\mathbf{p}(\vec{a})} - \frac{1}{m}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u_0})}{p_{u_0}(a_{u_0})}} \\
&+ \frac{1}{mn}\sum_{u'=0}^{n-1}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u'})}{p_{u'}(a_{u'})}}] + (\mathbf{1}(a_{u_0}=v_0) - \frac{1}{m})\sqrt{\frac{\mathbf{p}(\vec{a}_{-u_0})}{p_{u_0}(a_{u_0})}} \qquad (27)
\end{aligned}
$$

Clearly, we have the following relations

$$
\sum_{u\in[n]}[-\frac{1}{m}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u})}{p_u(a_u)}} + \frac{1}{mn}\sum_{u'=0}^{n-1}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u'})}{p_{u'}(a_{u'})}}] = 0 \qquad (28)
$$

$$
\sum_{v\in[m]}(\mathbf{1}(v=v_0) - \frac{1}{m}) = 0 \qquad (29)
$$

Thus

$$
(\mathbf{A}_0^p\mathbf{A}^{p,\dagger})_{i_0,\vec{a}} = \frac{1}{mn}\sum_{u\in[n]}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u})}{p_u(a_u)}} - \frac{n-1}{n}\sqrt{\mathbf{p}(\vec{a})} + (\mathbf{1}(a_{u_0}=v_0) - \frac{1}{m})\sqrt{\frac{\mathbf{p}(\vec{a}_{-u_0})}{p_{u_0}(a_{u_0})}} \qquad (30)
$$

$$
= \mathbf{A}^{p,\dagger}_{i_0,\vec{a}} \qquad (31)
$$

This proves $\mathbf{A}^{p,\dagger} = \mathbf{A}^{p,\dagger}\mathbf{A}^p\mathbf{A}^{p,\dagger}$ in statement (3) and $\mathbf{A}^{p,\dagger}$ is the Moore-Penrose inverse of $\mathbf{A}^p$. Since the optimal solution $\mathbf{x}^* = \mathbf{A}^{p,\dagger}\mathbf{b}^p + (\mathbf{I}_{mn\times mn} - \mathbf{A}^{p,\dagger}\mathbf{A}^p)\mathbf{w}$ where $w \in \mathrm{R}^{mn}$ is any vector [30].

Denote $\mathbf{x}^p = \mathbf{A}^{p,\dagger}\mathbf{b}^p$. We have $\forall i = u\times m + v$

$$
\begin{aligned}
\mathbf{x}_i^p &= \sum_{\vec{a}}\mathbf{A}^{p,\dagger}_{i,\vec{a}}\sqrt{\mathbf{p}(\vec{a})}\mathbf{b}_{\vec{a}} \\
&= \sum_{\vec{a}}[\sqrt{\frac{\mathbf{p}(\vec{a}_{-u})}{p_u(a_u)}}\cdot\mathbf{1}(a_u=v) - \frac{n-1}{n}\sqrt{\mathbf{p}(\vec{a})} - \frac{1}{m}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u})}{p_u(a_u)}} \\
&\quad + \frac{1}{mn}\sum_{u'=0}^{n-1}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u'})}{p_{u'}(a_{u'})}}]\sqrt{\mathbf{p}(\vec{a})}\mathbf{b}_{\vec{a}} \\
&= \sum_{\vec{a}}\left[\frac{\mathbf{p}(\vec{a})}{p_u(a_u)}\cdot\mathbf{1}(a_u=v) - \frac{n-1}{n}\mathbf{p}(\vec{a}) - \frac{1}{m}\frac{\mathbf{p}(\vec{a})}{p_u(a_u)} + \frac{1}{mn}\sum_{u'=0}^{n-1}\frac{\mathbf{p}(\vec{a})}{p_{u'}(a_{u'})}\right]\mathbf{b}_{\vec{a}} \qquad (32)
\end{aligned}
$$

From equation (2, 3, 4), we have $i = u\times m + v, i' = u'\times m + v'$

$$
(\mathbf{I} - \mathbf{A}^{p,\dagger}\mathbf{A}^p)_{i,i'} = \begin{cases} \frac{1}{m} - \frac{1}{mn} & \text{if } u = u' \\ -\frac{1}{mn} & \text{if } u \neq u' \end{cases} \qquad (33)
$$

If we consider $\mathbf{w}$ as the following $i_0 = u_0 \times m + v_0$

$$\mathbf{w}_{i_0} = \sum_{\vec{a} \in O(i_0)} \frac{\mathbf{p}(\vec{a})}{p_{u_0}(a_{u_0})} \mathbf{b}_{\vec{a}} \tag{34}$$

Then for $i = u \times m + v$

$$((\mathbf{I} - \mathbf{A}^{p,\dagger}\mathbf{A}^p)\mathbf{w})_i = \sum_{\substack{i_0 \in [mn] \\ u \neq u_0}} -\frac{1}{mn}\mathbf{w}_{i_0} + \sum_{i_0 : u_0 = u} (\frac{1}{m} - \frac{1}{mn})\mathbf{w}_{i_0} \tag{35}$$

$$= \sum_{\vec{a}} -\frac{1}{mn} \sum_{u' \in [n]} \frac{\mathbf{p}(\vec{a})}{p_{u'}(a_{u'})} \mathbf{b}_{\vec{a}} + \frac{1}{m} \sum_{\vec{a}} \frac{\mathbf{p}(\vec{a})}{p_u(a_u)} \mathbf{b}_{\vec{a}} \tag{36}$$

Notice that this is exactly the last two terms in equation (5). Therefore, the optimal solutions of this weighted linear regression problem can be written as: $i = u \times m + v, v \in [m], u \in [n]$ and an arbitrary vector $\mathbf{w} \in \mathrm{R}^{mn}$.

$$\mathbf{x}_i^* = \sum_{\vec{a}} \frac{\mathbf{p}(\vec{a})}{p_u(a_u)} \mathbf{b}_{\vec{a}} \cdot \mathbf{1}(a_u = v) - \frac{n-1}{n} \mathbf{p}(\vec{a}) \mathbf{b}_{\vec{a}} - \frac{1}{mn} \sum_{i' \in [mn]} \mathbf{w}_{i'} + \frac{1}{m} \sum_{v' \in [m]} \mathbf{w}_{um+v'} \tag{37}$$

This completes the proof. □

Definition 1. *[FQI-LVD] Given a dataset D, FQI-LVD specifies the action-value function class*

$$\mathcal{Q}^{LVD} = \left\{ Q \;\middle|\; Q_{tot}(\cdot, \mathbf{a}) = \sum_{i=1}^{n} Q_i(\cdot, a_i), \forall \mathbf{a} \in \mathbf{A} \text{ and } \left[\forall Q_i \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} \right]_{i=1}^{n} \right\} \tag{6}$$

and induces the empirical Bellman operator $\mathcal{T}_D^{LVD} : \mathcal{Q}^{LVD} \mapsto \mathcal{Q}^{LVD}$ as follows.

$$Q^{(t+1)} \leftarrow \mathcal{T}_D^{LVD} Q^{(t)} \equiv \underset{Q \in \mathcal{Q}^{LVD}}{\arg\min} \sum_{(s,\mathbf{a}) \in \mathcal{S} \times \mathbf{A}} p_D(\mathbf{a}|s) \left(y^{(t)}(s, \mathbf{a}) - \sum_{i=1}^{n} Q_i(s, a_i) \right)^2, \tag{7}$$

where $y^{(t)}(s, \mathbf{a}) = r(s, \mathbf{a}) + \gamma \max_{\mathbf{a}'} Q_{tot}^{(t)}(s', \mathbf{a}')$ denotes the regression target derived by Bellman optimality operator. Q_{tot} and $[Q_i]_{i=1}^n$ refer to the interfaces of CTDE defined in section 2.3.

Theorem 1. *Let $Q^{(t+1)} = \mathcal{T}_D^{LVD} Q^{(t)}$ denote a single iteration of the empirical Bellman operator. Then $\forall i \in \mathcal{N}, \forall (s, \mathbf{a}) \in \mathcal{S} \times \mathbf{A}$, the individual action-value function $Q_i^{(t+1)}(s, a_i) =$*

$$\mathop{\mathbb{E}}_{a'_{-i} \sim p_D(\cdot|s)} \left[y^{(t)}\left(s, a_i \oplus a'_{-i}\right) \right] - \frac{n-1}{n} \mathop{\mathbb{E}}_{\mathbf{a}' \sim p_D(\cdot|s)} \left[y^{(t)}\left(s, \mathbf{a}'\right) \right] + w_i(s), \tag{8}$$

where we denote $a_i \oplus a'_{-i} = \langle a'_1, \ldots, a'_{i-1}, a_i, a'_{i+1}, \ldots, a'_n \rangle$. a'_{-i} denotes the action of all agents except agent i. The residue term $\mathbf{w} \equiv [w_i]_{i=1}^n$ is an arbitrary vector satisfying $\forall s$, $\sum_{i=1}^n w_i(s) = 0$.

Proof. In the formulation of FQI-LVD stated in Definition 1, the empirical Bellman error minimization in Eq. (7) can be regarded as a weighted linear least squares problem as follows: $\forall s \in \mathcal{S}$,

$$\min_{\mathbf{x}} \parallel \sqrt{\mathbf{p}^\top} \cdot (\mathbf{A}\mathbf{x} - \mathbf{b}) \parallel_2^2 \tag{38}$$

where let $m, n \in \mathrm{Z}^+$ denote the size of action space $|\mathcal{A}|$ and the number of agents, respectively; $\mathbf{A} \in \mathbb{R}^{m^n \times mn}$ denotes the multi-agent credit assignment coefficient matrix of action-value functions with linear value decomposition; $\mathbf{x} \in \mathbb{R}^{mn}$ denotes individual action-value functions $\left[Q_i^{(t)}(s, \cdot) \in \mathbb{R}^m \right]_{i=1}^n$ under the empirical Bellman error minimization; $\mathbf{b} \in \mathbb{R}^{m^n}$ denotes the regression target $y^{(t)}(s, \cdot)$ derived by *Bellman optimality operator*; $\mathbf{p} \in \mathbb{R}^{m^n}$ denotes the empirical probability of joint action $\mathbf{a}$ executed on state s, $p_D(\mathbf{a}|s)$, which can be factorized to the production of individual components illustrated in Assumption 2.

Besides, $\mathbf{A}$ is m-ary encoding matrix namely $\forall i \in [m^n], j \in [mn]$

$$\mathbf{A}_{i,j} = \begin{cases} 1, & \text{if} \quad \exists u \in [n], j = m \times u + (\lfloor i/m^u \rfloor \ \textit{mod}\ m), \\ 0, & \text{otherwise.} \end{cases} \tag{39}$$

For simplicity, j^{th} row of $\mathbf{A}$ corresponds to a m-ary number $\vec{a}_j = (j)_m$ where $\vec{a} = a_0 a_1 \dots a_{n-1}$, with $a_u \in [m], \forall u \in [n]$. According to the factorizable empirical probability p_D shown in Assumption 2, $\mathbf{p}$ is a corresponding positive vector which follows that

$$\mathbf{p}_j = \mathbf{p}(\vec{a}_j) = \prod_{u \in [n]} p_u(a_{u,j}), \ \textit{where}\ p_u : [m] \to (0,1)\ \textit{and} \sum_{a_u \in [m]} p_u(a_u) = 1, \forall u \in [n] \tag{40}$$

According to Lemma 2, we derive the optimal solution of this problem is the following. Denote $i = u \times m + v, v \in [m], u \in [n]$ and an arbitrary vector $\mathbf{w} \in \mathbb{R}^{mn}$

$$\mathbf{x}_i^* = \sum_{\vec{a}} \frac{\mathbf{p}(\vec{a})}{p_u(a_u)} \mathbf{b}_{\vec{a}} \cdot \mathbf{1}(a_u = v) - \frac{n-1}{n} \mathbf{p}(\vec{a}) \mathbf{b}_{\vec{a}} - \frac{1}{mn} \sum_{i' \in [mn]} \mathbf{w}_{i'} + \frac{1}{m} \sum_{v' \in [m]} \mathbf{w}_{um+v'} \tag{41}$$

which means $\forall i \in \mathcal{N}, \forall (s, \mathbf{a}) \in \mathcal{S} \times \mathbf{A}$, the individual action-value function $Q_i^{(t+1)}(s, a_i) =$

$$\mathop{\mathbb{E}}_{a'_{-i} \sim p_D(\cdot|s)} \left[y^{(t)} \left(s, a_i \oplus a'_{-i} \right) \right] - \frac{n-1}{n} \mathop{\mathbb{E}}_{\mathbf{a}' \sim p_D(\cdot|s)} \left[y^{(t)} \left(s, \mathbf{a}' \right) \right] + w_i(s), \tag{42}$$

where we denote $a_i \oplus a'_{-i} = \langle a'_1, \dots, a'_{i-1}, a_i, a'_{i+1}, \dots, a'_n \rangle$. a'_{-i} denotes the action of all agents except agent i. The residue term $\mathbf{w} \equiv [w_i]_{i=1}^n$ is an arbitrary vector satisfying $\forall s, \sum_{i=1}^n w_i(s) = 0$. □

B Omitted Proofs in Section 4

B.1 Some Notations

In this section, we only consider the data distribution generated by the optimal joint policy $\boldsymbol{\pi}^*$.

To simplify the notations, we use $\varepsilon = \frac{\epsilon}{|\mathcal{A}|}$ to reformulate the exploratory policy generated by ϵ-greedy exploration as follows

$$\tilde{\boldsymbol{\pi}}(\mathbf{a}|s) = \prod_{i=1}^n \left(\varepsilon + (1 - \hat{\varepsilon}) \mathbb{I} \left[a_i = \arg\max_{a'_i \in \mathcal{A}} Q_i^*(s, a'_i) \right] \right) \tag{43}$$

where $\hat{\varepsilon} = (|\mathcal{A}| - 1)\varepsilon$.

In addition, we use $f(s, \cdot, \cdot)$ to denote the corresponding coefficient in the closed-form updating

$$(\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \mathbf{a}) = \sum_{\mathbf{a}' \in \mathcal{A}^n} f(s, \mathbf{a}, \mathbf{a}') (\mathcal{T} Q)_{\text{tot}}(s, \mathbf{a}') \tag{44}$$

where $(\mathcal{T}Q)_{\text{tot}} = r(s, \mathbf{a}') + \gamma V_{\text{tot}}(s')$ denote the precise target values derived by Bellman optimality equation.

Formally,

$$f(s, \mathbf{a}, \mathbf{a}') = \left(\frac{h^{(1)}(s, \mathbf{a}, \mathbf{a}')}{1 - \hat{\varepsilon}} + \frac{h^{(0)}(s, \mathbf{a}, \mathbf{a}')}{\varepsilon} - (n-1) \right) (1 - \hat{\varepsilon})^{h^{\hat{\boldsymbol{\pi}}}(s, \mathbf{a}')} \varepsilon^{n - h^{\hat{\boldsymbol{\pi}}}(s, \mathbf{a}')}, \tag{45}$$

in which

$$h^{\boldsymbol{\pi}^*}(s, \mathbf{a}) = \sum_{i=1}^n \mathbb{I}[a_i = \boldsymbol{\pi}_i^*(s)] \tag{46}$$

$$h^{(1)}(s, \mathbf{a}, \mathbf{a}') = \sum_{i=1}^n \mathbb{I}[a_i = \boldsymbol{\pi}_i^*(s)] \mathbb{I}[a_i = a'_i] \tag{47}$$

$$h^{(0)}(s, \mathbf{a}, \mathbf{a}') = \sum_{i=1}^{n} \mathbb{I}[a_i \neq \boldsymbol{\pi}_i^*(s)]\mathbb{I}[a_i = a_i'] \tag{48}$$

As a reference for the values of optimal action selections, we denote

$$\mathcal{E}(Q) = \max_{(s,\mathbf{a}) \in \mathcal{S} \times (\mathcal{A}^n \setminus \{\boldsymbol{\pi}^*(s)\})} (Q_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - Q_{\text{tot}}(s, \mathbf{a})) \tag{49}$$

Notice that $\mathcal{E}(Q)$ might be negative for a non-optimal value function Q.

B.2 Omitted Proofs

Lemma 3. *Given a dataset D generated by the optimal policy $\boldsymbol{\pi}^*$ with ϵ-greedy exploration, for any target value function Q,*

$$\forall \delta > 0,\ \forall 0 < \varepsilon \leq \frac{\delta}{n^2 |\mathcal{A}|^n 2^{n+1} (R_{max} + \gamma \|V_{tot}\|_\infty)}, \tag{50}$$

we have

$$\forall s \in \mathcal{S},\ \left| (\mathcal{T}_D^{LVD} Q)_{tot}(s, \boldsymbol{\pi}^*(s)) - (\mathcal{T}Q)_{tot}(s, \boldsymbol{\pi}^*(s)) \right| \leq \delta, \tag{51}$$

where $(\mathcal{T}Q)_{tot}(s, \mathbf{a}) = r(s, \mathbf{a}) + \gamma V(s')$ denotes the regression target generated by Q.

Proof. $\forall s \in \mathcal{S}$,

$$\begin{aligned}
&\left| (\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) \right| \\
\leq\ & |(f(s, \boldsymbol{\pi}^*(s), \boldsymbol{\pi}^*(s)) - 1)(\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s))| + \left| \sum_{a' \in \mathcal{A}^n \setminus \{\boldsymbol{\pi}^*(s)\}} f(s, \boldsymbol{\pi}^*(s), \mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s, \mathbf{a}') \right| \\
\leq\ & \left(|f(s, \boldsymbol{\pi}^*(s), \boldsymbol{\pi}^*(s)) - 1| + \sum_{a' \in \mathcal{A}^n \setminus \{\boldsymbol{\pi}^*(s)\}} |f(s, \boldsymbol{\pi}^*(s), \mathbf{a}')| \right) \|(\mathcal{T}Q)_{\text{tot}}\|_\infty.
\end{aligned} \tag{52}$$

In the first term, $\forall s \in \mathcal{S}$,

$$\begin{aligned}
|f(s, \boldsymbol{\pi}^*(s), \boldsymbol{\pi}^*(s)) - 1| &= \left| \left(\frac{n}{1-\hat{\varepsilon}} - (n-1) \right) (1-\hat{\varepsilon})^n - 1 \right| \\
&= \left| (n - (n-1)(1-\hat{\varepsilon}))(1-\hat{\varepsilon})^{n-1} - 1 \right| \\
&= \left| (1 + (n-1)\hat{\varepsilon})(1-\hat{\varepsilon})^{n-1} - 1 \right| \\
&= \left| (1 + (n-1)\hat{\varepsilon}) \left(\sum_{\ell=0}^{n-1} \binom{n-1}{\ell} (-1)^\ell \hat{\varepsilon}^\ell \right) - 1 \right| \\
&= \left| (1 + (n-1)\hat{\varepsilon}) \left(1 - (n-1)\hat{\varepsilon} + \sum_{\ell=2}^{n-1} \binom{n-1}{\ell} (-1)^\ell \hat{\varepsilon}^\ell \right) - 1 \right| \\
&= \left| 1 - (n-1)^2 \hat{\varepsilon}^2 + (1 + (n-1)\hat{\varepsilon}) \left(\sum_{\ell=2}^{n-1} \binom{n-1}{\ell} (-1)^\ell \hat{\varepsilon}^\ell \right) - 1 \right| \\
&= \left| \hat{\varepsilon}^2 \left((n-1)^2 - (1 + (n-1)\hat{\varepsilon}) \sum_{\ell=2}^{n-1} \binom{n-1}{\ell} (-1)^\ell \hat{\varepsilon}^{\ell-2} \right) \right| \\
&\leq |\mathcal{A}|^2 \varepsilon^2 \left(n^2 + n \sum_{\ell=2}^{n-1} \binom{n-1}{\ell} \right) \\
&\leq |\mathcal{A}|^2 \varepsilon^2 \left(n^2 + n 2^{n-1} \right) \\
&\leq \varepsilon^2 n^2 |\mathcal{A}|^2 2^n.
\end{aligned} \tag{53}$$

In the second term, $\forall s \in \mathcal{S}$,

$$\begin{aligned}
&\sum_{a' \in \mathcal{A}^n \setminus \{\boldsymbol{\pi}^*(s)\}} |f(s, \boldsymbol{\pi}^*(s), \mathbf{a}')| \\
\leq &\sum_{a' \in \mathcal{A}^n \setminus \{\boldsymbol{\pi}^*(s)\}} \left| \left(\frac{h^{\boldsymbol{\pi}^*}(s, \mathbf{a}')}{1-\hat{\varepsilon}} - (n-1) \right) (1-\hat{\varepsilon})^{h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')} \varepsilon^{n-h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')} \right| \\
= &\sum_{a' \in \mathcal{A}^n \setminus \{\boldsymbol{\pi}^*(s)\}} \left| \left(h^{\boldsymbol{\pi}^*}(s, \mathbf{a}') - (n-1)(1-\hat{\varepsilon}) \right) (1-\hat{\varepsilon})^{h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')-1} \varepsilon^{n-h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')} \right| \\
\leq &\sum_{a' \in \mathcal{A}^n \setminus \{\boldsymbol{\pi}^*(s)\}} \left| 2n(1-\hat{\varepsilon})^{h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')-1} \varepsilon^{n-h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')} \right| \\
\leq &\sum_{a' \in \mathcal{A}^n \setminus \{\boldsymbol{\pi}^*(s)\}} 2n\varepsilon \\
\leq &\, 2n\varepsilon |\mathcal{A}|^n. \qquad (54)
\end{aligned}$$

Thus $\forall s \in \mathcal{S}$,

$$\begin{aligned}
&\left| (\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) \right| \\
\leq &\left(|f(s, \boldsymbol{\pi}^*(s), \boldsymbol{\pi}^*(s)) - 1| + \sum_{a' \in \mathcal{A}^n \setminus \{\boldsymbol{\pi}^*(s)\}} |f(s, \boldsymbol{\pi}^*(s), \mathbf{a}')| \right) \|(\mathcal{T}Q)_{\text{tot}}\|_\infty \\
\leq &(\varepsilon^2 n^2 |\mathcal{A}|^2 2^n + 2n\varepsilon |\mathcal{A}|^n) \|(\mathcal{T}Q)_{\text{tot}}\|_\infty \\
\leq &\, \varepsilon n^2 |\mathcal{A}|^n 2^{n+1} \|(\mathcal{T}Q)_{\text{tot}}\|_\infty \\
\leq &\, \varepsilon n^2 |\mathcal{A}|^n 2^{n+1} (R_{\max} + \gamma \|V\|_\infty) \\
\leq &\, \delta. \qquad (55)
\end{aligned}$$

□

Lemma 4. *Given a dataset D generated by the optimal policy $\boldsymbol{\pi}^*$ with ϵ-greedy exploration, for any target value function Q,*

$$\forall 0 < \varepsilon \leq \frac{(1-\gamma)\|V^{\boldsymbol{\pi}^*} - V^*\|_\infty}{n^2 |\mathcal{A}|^n 2^{n+1} (R_{max}/(1-\gamma) + \gamma \|V^{\boldsymbol{\pi}^*} - V^*\|_\infty)}, \qquad (56)$$

we have

$$\forall s \in \mathcal{S}, \ \left| (\mathcal{T}_D^{LVD} Q)_{tot}(s, \boldsymbol{\pi}^*(s)) - V^*(s) \right| \leq \|V^{\boldsymbol{\pi}^*} - V^*\|_\infty, \qquad (57)$$

where $V^{\boldsymbol{\pi}^*}(s) = Q_{tot}(s, \boldsymbol{\pi}^*(s))$.

Proof. $\forall s \in \mathcal{S}$,

$$\begin{aligned}
&\left| (\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - V^*(s) \right| \\
\leq &\left| (\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) \right| + |(\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - V^*(s)| \\
= &\left| (\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) \right| + |(\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - Q^*(s, \boldsymbol{\pi}^*(s))| \\
= &\left| (\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) \right| + |(\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - (\mathcal{T}Q^*)(s, \boldsymbol{\pi}^*(s))| \\
\leq &\left| (\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) \right| + \gamma |V_{\text{tot}}(s') - V^*(s')| \\
\leq &\left| (\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) \right| + \gamma |Q_{\text{tot}}(s', \boldsymbol{\pi}^*(s')) - V^*(s')| \\
\leq &\left| (\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) \right| + \gamma \|V^{\boldsymbol{\pi}^*} - V^*\|_\infty \qquad (58)
\end{aligned}$$

Let $\delta = (1-\gamma)\|V^{\boldsymbol{\pi}^*} - V^*\|_\infty$. According to Lemma 3, with the condition

$$0 < \varepsilon \leq \frac{\delta}{n^2 |\mathcal{A}|^n 2^{n+1} (R_{\max} + \gamma \|V\|_\infty)} = \frac{(1-\gamma)\|V^{\boldsymbol{\pi}^*} - V^*\|_\infty}{n^2 |\mathcal{A}|^n 2^{n+1} (R_{\max} + \gamma \|V\|_\infty)}, \qquad (59)$$

we have

$$\left|(\mathcal{T}_D^{\text{LVD}}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s)) - (\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s))\right| \leq \delta = (1-\gamma)\|V^{\boldsymbol{\pi}^*} - V^*\|_\infty. \tag{60}$$

Notice that

$$\begin{aligned}\|V_{\text{tot}}\|_\infty &\leq \|V^*\|_\infty + \|V_{\text{tot}} - V^*\|_\infty &(61)\\ &\leq \frac{R_{\max}}{1-\gamma} + \|V^{\boldsymbol{\pi}^*} - V^*\|_\infty. &(62)\end{aligned}$$

The overall statement is

$$\forall 0 < \varepsilon \leq \frac{(1-\gamma)\|V^{\boldsymbol{\pi}^*} - V^*\|_\infty}{n^2|\mathcal{A}|^n 2^{n+1}(R_{\max}/(1-\gamma) + \gamma\|V^{\boldsymbol{\pi}^*} - V^*\|_\infty)} \leq \frac{(1-\gamma)\|V^{\boldsymbol{\pi}^*} - V^*\|_\infty}{n^2|\mathcal{A}|^n 2^{n+1}(R_{\max} + \gamma\|V_{\text{tot}}\|_\infty)} \tag{63}$$

we have $\forall s \in \mathcal{S}$,

$$\begin{aligned}\left|(\mathcal{T}_D^{\text{LVD}}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s)) - V^*(s)\right| \leq\ & \left|(\mathcal{T}_D^{\text{LVD}}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s)) - (\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s))\right| \\ &+ \gamma\|V^{\boldsymbol{\pi}^*} - V^*\|_\infty \\ &\leq (1-\gamma)\|V^{\boldsymbol{\pi}^*} - V^*\|_\infty + \gamma\|V^{\boldsymbol{\pi}^*} - V^*\|_\infty \\ &\leq \|V^{\boldsymbol{\pi}^*} - V^*\|_\infty. &(64)\end{aligned}$$

□

Lemma 5. *For any value function Q, the corresponding sub-optimality gap satisfies*

$$\mathcal{E}(\mathcal{T}Q) \geq \mathcal{E}(Q^*) - 2\gamma\|V_{tot} - V^*\|_\infty \tag{65}$$

Proof. With a slight abuse of notation, let s_1 and s_2 denote the states to transit while taking actions $\boldsymbol{\pi}^*(s)$ and a at the state s, respectively. According to the definition,

$$\begin{aligned}\mathcal{E}(\mathcal{T}Q) =& \max_{(s,\mathbf{a})\in\mathcal{S}\times(\mathcal{A}^n\setminus\{\boldsymbol{\pi}^*(s)\})}((\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s)) - (\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a})) \\ \geq& \max_{(s,\mathbf{a})\in\mathcal{S}\times(\mathcal{A}^n\setminus\{\boldsymbol{\pi}^*(s)\})}\left((\mathcal{T}Q^*)(s,\boldsymbol{\pi}^*(s)) - (\mathcal{T}Q^*)(s,\mathbf{a}) - \gamma\left(|V_{\text{tot}}(s_1) - V^*(s_1)| + |V_{\text{tot}}(s_2) - V^*(s_2)|\right)\right) \\ \geq& \max_{(s,\mathbf{a})\in\mathcal{S}\times(\mathcal{A}^n\setminus\{\boldsymbol{\pi}^*(s)\})}\left((\mathcal{T}Q^*)(s,\boldsymbol{\pi}^*(s)) - (\mathcal{T}Q^*)(s,\mathbf{a}) - 2\gamma\|V_{\text{tot}} - V^*\|_\infty\right) \\ =& \max_{(s,\mathbf{a})\in\mathcal{S}\times(\mathcal{A}^n\setminus\{\boldsymbol{\pi}^*(s)\})}\left(Q^*(s,\boldsymbol{\pi}^*(s)) - Q^*(s,\mathbf{a}) - 2\gamma\|V_{\text{tot}} - V^*\|_\infty\right) \\ =& \mathcal{E}(Q^*) - 2\gamma\|V_{\text{tot}} - V^*\|_\infty &(66)\end{aligned}$$

□

Lemma 6. *Given a dataset D generated by the optimal policy $\boldsymbol{\pi}^*$ with ϵ-greedy exploration, for any target value function Q,*

$$\forall\delta > 0,\ \forall 0 < \varepsilon \leq \frac{\delta}{n^2|\mathcal{A}|^n 2^n(R_{max}/(1-\gamma) + \gamma\|V_{tot} - V^*\|_\infty)}, \tag{67}$$

we have $\forall s \in \mathcal{S}$, $\forall a \in \mathcal{A}^n \setminus \{\boldsymbol{\pi}^*(s)\}$,

$$(\mathcal{T}_D^{LVD}Q)_{tot}(s,\mathbf{a}) \leq (\mathcal{T}Q)_{tot}(s,\boldsymbol{\pi}^*(s)) - \mathcal{E}(Q^*) + 2n\gamma\|V_{tot} - V^*\|_\infty + \delta \tag{68}$$

where $(\mathcal{T}Q)_{tot}(s,\mathbf{a}) = r(s,\mathbf{a}) + \gamma V_{tot}(s')$ denotes the regression target generated by Q.

Proof. $\forall s \in \mathcal{S}, \forall a \in \mathcal{A}^n \setminus \{\boldsymbol{\pi}^*(s)\}$,

$$\begin{aligned}(\mathcal{T}_D^{\text{LVD}}Q)_{\text{tot}}(s,\mathbf{a}) =& \sum_{a'\in\mathcal{A}^n} f(s,\mathbf{a},\mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}') \\ =& f(s,\mathbf{a},\boldsymbol{\pi}^*(s))(\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s)) \\ &+ \sum_{a'\in\mathcal{A}^n : h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')=n-1} f(s,\mathbf{a},\mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}')\end{aligned}$$

$$+ \sum_{a' \in \mathcal{A}^n : h^{\boldsymbol{\pi}^*}(s,\mathbf{a}') < n-1} f(s,\mathbf{a},\mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}') \tag{69}$$

In the first term,

$$\begin{aligned}
& f(s,\mathbf{a},\boldsymbol{\pi}^*(s))(\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s)) \\
&= \left(\frac{h^{\boldsymbol{\pi}^*}(s,\mathbf{a})}{1-\hat{\varepsilon}} - (n-1)\right)(1-\hat{\varepsilon})^n (\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s)) \\
&= \left(h^{\boldsymbol{\pi}^*}(s,\mathbf{a}) - (n-1)(1-\hat{\varepsilon})\right)(1-\hat{\varepsilon})^{n-1} (\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s)) \\
&= \left(h^{\boldsymbol{\pi}^*}(s,\mathbf{a}) - (n-1) + (n-1)(|\mathcal{A}|-1)\varepsilon\right)(1-\hat{\varepsilon})^{n-1} (\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s)) \\
&\le \left(h^{\boldsymbol{\pi}^*}(s,\mathbf{a}) - (n-1)\right)(1-\hat{\varepsilon})^{n-1} (\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s)) + \varepsilon n|\mathcal{A}|\|(\mathcal{T}Q)_{\text{tot}}\|_\infty \\
&= \left(h^{\boldsymbol{\pi}^*}(s,\mathbf{a}) - (n-1)\right)(1+(1-\hat{\varepsilon})^{n-1}-1)(\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s)) + \varepsilon n|\mathcal{A}|\|(\mathcal{T}Q)_{\text{tot}}\|_\infty \\
&\le \left(h^{\boldsymbol{\pi}^*}(s,\mathbf{a}) - (n-1)\right)(\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s)) + \left|h^{\boldsymbol{\pi}^*}(s,\mathbf{a}) - (n-1)\right|((1-\hat{\varepsilon})^{n-1}-1)\|(\mathcal{T}Q)_{\text{tot}}\|_\infty + \varepsilon n|\mathcal{A}|\|(\mathcal{T}Q)_{\text{tot}}\|_\infty \\
&\le \left(h^{\boldsymbol{\pi}^*}(s,\mathbf{a}) - (n-1)\right)(\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s)) + 2n\left(\sum_{\ell=1}^{n-1}\binom{n-1}{\ell}(-1)^\ell \hat{\varepsilon}^\ell\right)\|(\mathcal{T}Q)_{\text{tot}}\|_\infty + \varepsilon n|\mathcal{A}|\|(\mathcal{T}Q)_{\text{tot}}\|_\infty \\
&\le \left(h^{\boldsymbol{\pi}^*}(s,\mathbf{a}) - (n-1)\right)(\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s)) + 2n\hat{\varepsilon}\left(\sum_{\ell=1}^{n-1}\binom{n-1}{\ell}\right)\|(\mathcal{T}Q)_{\text{tot}}\|_\infty + \varepsilon n|\mathcal{A}|\|(\mathcal{T}Q)_{\text{tot}}\|_\infty \\
&\le \left(h^{\boldsymbol{\pi}^*}(s,\mathbf{a}) - (n-1)\right)(\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s)) + \hat{\varepsilon}n2^n\|(\mathcal{T}Q)_{\text{tot}}\|_\infty + \varepsilon n|\mathcal{A}|\|(\mathcal{T}Q)_{\text{tot}}\|_\infty \\
&\le \left(h^{\boldsymbol{\pi}^*}(s,\mathbf{a}) - (n-1)\right)(\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s)) + \varepsilon n2^n|\mathcal{A}|\|(\mathcal{T}Q)_{\text{tot}}\|_\infty + \varepsilon n|\mathcal{A}|\|(\mathcal{T}Q)_{\text{tot}}\|_\infty
\end{aligned} \tag{70}$$

In the second term,

$$\begin{aligned}
& \sum_{a' \in \mathcal{A}^n : h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')=n-1} f(s,\mathbf{a},\mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}') \\
&= \sum_{a' \in \mathcal{A}^n : h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')=n-1} \left(\frac{h^{(1)}(s,\mathbf{a},\mathbf{a}')}{1-\hat{\varepsilon}} + \frac{h^{(0)}(s,\mathbf{a},\mathbf{a}')}{\varepsilon} - (n-1)\right)(1-\hat{\varepsilon})^{n-1}\varepsilon(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}') \\
&= \sum_{a' \in \mathcal{A}^n : h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')=n-1} \left(h^{(0)}(s,\mathbf{a},\mathbf{a}')(1-\hat{\varepsilon})^{n-1}(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}') + \left(\frac{h^{(1)}(s,\mathbf{a},\mathbf{a}')}{1-\hat{\varepsilon}} - (n-1)\right)(1-\hat{\varepsilon})^{n-1}\varepsilon(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}')\right) \\
&\le \sum_{a' \in \mathcal{A}^n : h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')=n-1} \left(h^{(0)}(s,\mathbf{a},\mathbf{a}')(1-\hat{\varepsilon})^{n-1}(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}') + \left|\frac{h^{(1)}(s,\mathbf{a},\mathbf{a}')}{1-\hat{\varepsilon}} - (n-1)\right|(1-\hat{\varepsilon})^{n-1}\varepsilon\|(\mathcal{T}Q)_{\text{tot}}\|_\infty\right) \\
&\le \sum_{a' \in \mathcal{A}^n : h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')=n-1} \left(h^{(0)}(s,\mathbf{a},\mathbf{a}')(1-\hat{\varepsilon})^{n-1}(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}') + 2n\varepsilon\|(\mathcal{T}Q)_{\text{tot}}\|_\infty\right) \\
&= \sum_{a' \in \mathcal{A}^n : h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')=n-1} \left(h^{(0)}(s,\mathbf{a},\mathbf{a}')\left(\sum_{\ell=0}^{n-1}\binom{n-1}{\ell}(-1)^\ell\hat{\varepsilon}^\ell\right)(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}') + 2n\varepsilon\|(\mathcal{T}Q)_{\text{tot}}\|_\infty\right) \\
&= \sum_{a' \in \mathcal{A}^n : h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')=n-1} \left(h^{(0)}(s,\mathbf{a},\mathbf{a}')\left(1+\sum_{\ell=1}^{n-1}\binom{n-1}{\ell}(-1)^\ell\hat{\varepsilon}^\ell\right)(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}') + 2n\varepsilon\|(\mathcal{T}Q)_{\text{tot}}\|_\infty\right) \\
&\le \sum_{a' \in \mathcal{A}^n : h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')=n-1} \left(h^{(0)}(s,\mathbf{a},\mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}') + \left|\sum_{\ell=1}^{n-1}\binom{n-1}{\ell}(-1)^\ell\hat{\varepsilon}^\ell\right|\|(\mathcal{T}Q)_{\text{tot}}\|_\infty + 2n\varepsilon\|(\mathcal{T}Q)_{\text{tot}}\|_\infty\right) \\
&= \sum_{a' \in \mathcal{A}^n : h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')=n-1} \left(h^{(0)}(s,\mathbf{a},\mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}') + \hat{\varepsilon}\left|\sum_{\ell=1}^{n-1}\binom{n-1}{\ell}(-1)^\ell\hat{\varepsilon}^{\ell-1}\right|\|(\mathcal{T}Q)_{\text{tot}}\|_\infty + 2n\varepsilon\|(\mathcal{T}Q)_{\text{tot}}\|_\infty\right) \\
&\le \sum_{a' \in \mathcal{A}^n : h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')=n-1} \left(h^{(0)}(s,\mathbf{a},\mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}') + \hat{\varepsilon}\left(\sum_{\ell=1}^{n-1}\binom{n-1}{\ell}\right)\|(\mathcal{T}Q)_{\text{tot}}\|_\infty + 2n\varepsilon\|(\mathcal{T}Q)_{\text{tot}}\|_\infty\right) \\
&\le \sum_{a' \in \mathcal{A}^n : h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')=n-1} \left(h^{(0)}(s,\mathbf{a},\mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}') + \varepsilon|\mathcal{A}|2^{n-1}\|(\mathcal{T}Q)_{\text{tot}}\|_\infty + 2n\varepsilon\|(\mathcal{T}Q)_{\text{tot}}\|_\infty\right) \\
&= \left(\sum_{a' \in \mathcal{A}^n : h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')=n-1} h^{(0)}(s,\mathbf{a},\mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}')\right) + \varepsilon n|\mathcal{A}|2^{n-1}\|(\mathcal{T}Q)_{\text{tot}}\|_\infty + 2n^2\varepsilon\|(\mathcal{T}Q)_{\text{tot}}\|_\infty \\
&\le \left(\sum_{a' \in \mathcal{A}^n : h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')=n-1} h^{(0)}(s,\mathbf{a},\mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}')\right) + \varepsilon n^2|\mathcal{A}|2^n\|(\mathcal{T}Q)_{\text{tot}}\|_\infty
\end{aligned} \tag{71}$$

In the third term,

$$
\begin{aligned}
&\sum_{a'\in\mathcal{A}^n:h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')<n-1} f(s,\mathbf{a},\mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}') \\
\leq &\sum_{a'\in\mathcal{A}^n:h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')<n-1} \left|f(s,\mathbf{a},\mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}')\right| \\
= &\sum_{a'\in\mathcal{A}^n:h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')<n-1} \left|\frac{h^{(1)}(s,\mathbf{a},\mathbf{a}')}{1-\hat{\varepsilon}}+\frac{h^{(0)}(s,\mathbf{a},\mathbf{a}')}{\varepsilon}-(n-1)\right|(1-\hat{\varepsilon})^{h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')}\varepsilon^{n-h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')}\left|(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}')\right| \\
\leq &\sum_{a'\in\mathcal{A}^n:h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')<n-1} \left|\frac{h^{(1)}(s,\mathbf{a},\mathbf{a}')}{1-\hat{\varepsilon}}+\frac{h^{(0)}(s,\mathbf{a},\mathbf{a}')}{\varepsilon}+(n-1)\right|(1-\hat{\varepsilon})^{h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')}\varepsilon^{n-h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')}\left|(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}')\right| \\
\leq &\sum_{a'\in\mathcal{A}^n:h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')<n-1} n\left(1+\frac{1}{1-\hat{\varepsilon}}+\frac{1}{\varepsilon}\right)(1-\hat{\varepsilon})^{h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')}\varepsilon^{n-h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')}\left|(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}')\right| \\
\leq &\sum_{a'\in\mathcal{A}^n:h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')<n-1} n\left(1+\frac{2}{\varepsilon}\right)(1-\hat{\varepsilon})^{h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')}\varepsilon^{n-h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')}\left|(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}')\right| \\
\leq &\sum_{a'\in\mathcal{A}^n:h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')<n-1} 3n\varepsilon^{n-h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')-1}\left|(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}')\right| \\
\leq &\sum_{a'\in\mathcal{A}^n:h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')<n-1} 3n\varepsilon\|(\mathcal{T}Q)_{\text{tot}}\|_\infty \\
\leq &\, 3n\varepsilon|\mathcal{A}|^n\|(\mathcal{T}Q)_{\text{tot}}\|_\infty
\end{aligned}
\tag{72}
$$

Combining the above terms, we can get

$$
\begin{aligned}
&(\mathcal{T}_D^{\text{LVD}}Q)_{\text{tot}}(s,\mathbf{a}) \\
&= f(s,\mathbf{a},\boldsymbol{\pi}^*(s))(\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s)) + \sum_{a'\in\mathcal{A}^n:h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')=n-1} f(s,\mathbf{a},\mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}') \\
&\quad + \sum_{a'\in\mathcal{A}^n:h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')<n-1} f(s,\mathbf{a},\mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}') \\
&\leq \left(h^{\boldsymbol{\pi}^*}(s,\mathbf{a})-(n-1)\right)(\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s)) + \varepsilon n 2^n|\mathcal{A}|\|(\mathcal{T}Q)_{\text{tot}}\|_\infty + \varepsilon n|\mathcal{A}|\|(\mathcal{T}Q)_{\text{tot}}\|_\infty \\
&\quad + \left(\sum_{a'\in\mathcal{A}^n:h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')=n-1} h^{(0)}(s,\mathbf{a},\mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}')\right) + \varepsilon n^2|\mathcal{A}|2^n\|(\mathcal{T}Q)_{\text{tot}}\|_\infty + 3n\varepsilon|\mathcal{A}|^n\|(\mathcal{T}Q)_{\text{tot}}\|_\infty \\
&\leq \left(h^{\boldsymbol{\pi}^*}(s,\mathbf{a})-(n-1)\right)(\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s)) + \left(\sum_{a'\in\mathcal{A}^n:h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')=n-1} h^{(0)}(s,\mathbf{a},\mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}')\right) \\
&\quad + \varepsilon n^2|\mathcal{A}|^n 2^n\|(\mathcal{T}Q)_{\text{tot}}\|_\infty
\end{aligned}
\tag{73}
$$

in which

$$
\begin{aligned}
&\sum_{a'\in\mathcal{A}^n:h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')=n-1} h^{(0)}(s,\mathbf{a},\mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}') \\
&\leq \left(\sum_{a'\in\mathcal{A}^n:h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')=n-1} h^{(0)}(s,\mathbf{a},\mathbf{a}')\right)\max_{a'\in\mathcal{A}^n:h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')=n-1}(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}') \\
&= \left(n-h^{\boldsymbol{\pi}^*}(s,\mathbf{a})\right)\max_{a'\in\mathcal{A}^n:h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')=n-1}(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}') \\
&\leq \left(n-h^{\boldsymbol{\pi}^*}(s,\mathbf{a})\right)\max_{a'\in\mathcal{A}^n\setminus\{\boldsymbol{\pi}^*(s)\}}(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}') \\
&= \left(n-h^{\boldsymbol{\pi}^*}(s,\mathbf{a})\right)\left((\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*)-\mathcal{E}(\mathcal{T}Q)\right)
\end{aligned}
\tag{74}
$$

Thus $\forall s\in\mathcal{S}, \forall a\in\mathcal{A}^n\setminus\{\boldsymbol{\pi}^*(s)\}$,

$(\mathcal{T}_D^{\text{LVD}}Q)_{\text{tot}}(s,\mathbf{a})$

$$
\begin{aligned}
&\leq \left(h^{\boldsymbol{\pi}^*}(s,\mathbf{a})-(n-1)\right)(\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s))+\left(\sum_{a'\in\mathcal{A}^n:h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')=n-1} h^{(0)}(s,\mathbf{a},\mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}')\right)\\
&\quad+\varepsilon n^2|\mathcal{A}|^n 2^n\|(\mathcal{T}Q)_{\text{tot}}\|_\infty\\
&\leq \left(h^{\boldsymbol{\pi}^*}(s,\mathbf{a})-(n-1)\right)(\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s))+(n-h^{\boldsymbol{\pi}^*}(s,\mathbf{a}))\left((\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*)-\mathcal{E}(\mathcal{T}Q)\right)+\varepsilon n^2|\mathcal{A}|^n 2^n\|(\mathcal{T}Q)_{\text{tot}}\|_\infty\\
&=(\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s))-(n-h^{\boldsymbol{\pi}^*}(s,\mathbf{a}))\mathcal{E}(\mathcal{T}Q)+\varepsilon n^2|\mathcal{A}|^n 2^n\|(\mathcal{T}Q)_{\text{tot}}\|_\infty \qquad (75)
\end{aligned}
$$

According to Lemma 5, $\mathcal{E}(\mathcal{T}Q)\geq\mathcal{E}(Q^*)-2\gamma\|V_{\text{tot}}-V^*\|_\infty$. So $\forall s\in\mathcal{S}$, $\forall a\in\mathcal{A}^n\setminus\{\boldsymbol{\pi}^*(s)\}$,

$$
\begin{aligned}
(\mathcal{T}_D^{\text{LVD}}Q)_{\text{tot}}(s,\mathbf{a}) &\leq (\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s))-(n-h^{\boldsymbol{\pi}^*}(s,\mathbf{a}))\mathcal{E}(\mathcal{T}Q)+\varepsilon n^2|\mathcal{A}|^n 2^n\|(\mathcal{T}Q)_{\text{tot}}\|_\infty\\
&\leq (\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s))-(n-h^{\boldsymbol{\pi}^*}(s,\mathbf{a}))\left(\mathcal{E}(Q^*)-2\gamma\|V_{\text{tot}}-V^*\|_\infty\right)+\varepsilon n^2|\mathcal{A}|^n 2^n\|(\mathcal{T}Q)_{\text{tot}}\|_\infty\\
&\leq (\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s))-\mathcal{E}(Q^*)+2n\gamma\|V_{\text{tot}}-V^*\|_\infty+\varepsilon n^2|\mathcal{A}|^n 2^n\|(\mathcal{T}Q)_{\text{tot}}\|_\infty\\
&\leq (\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s))-\mathcal{E}(Q^*)+2n\gamma\|V_{\text{tot}}-V^*\|_\infty+\varepsilon n^2|\mathcal{A}|^n 2^n(R_{\max}+\gamma\|V_{\text{tot}}\|_\infty)\\
&\leq (\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s))-\mathcal{E}(Q^*)+2n\gamma\|V_{\text{tot}}-V^*\|_\infty+\varepsilon n^2|\mathcal{A}|^n 2^n(R_{\max}+\gamma\|V^*\|_\infty+\gamma\|V_{\text{tot}}-V^*\|_\infty)\\
&\leq (\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s))-\mathcal{E}(Q^*)+2n\gamma\|V_{\text{tot}}-V^*\|_\infty+\varepsilon n^2|\mathcal{A}|^n 2^n(R_{\max}/(1-\gamma)+\gamma\|V_{\text{tot}}-V^*\|_\infty)\\
&\leq (\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s))-\mathcal{E}(Q^*)+2n\gamma\|V_{\text{tot}}-V^*\|_\infty+\delta \qquad (76)
\end{aligned}
$$

□

Lemma 7. *Let $\mathcal{B}$ denote a subspace of value functions*

$$
\mathcal{B}=\left\{Q\in\mathcal{Q}^{LVD}\,\middle|\,\mathcal{E}(Q)\geq 0,\ \|V_{tot}-V^*\|_\infty\leq\frac{1}{6n\gamma}\mathcal{E}(Q^*)\right\} \qquad (77)
$$

Given a dataset D generated by the optimal policy $\boldsymbol{\pi}^*$ with ϵ-greedy exploration,

$$
\forall 0<\varepsilon\leq\frac{(1-\gamma)\mathcal{E}(Q^*)}{6n^3|\mathcal{A}|^n 2^{n+1}(R_{max}/(1-\gamma)+\mathcal{E}(Q^*)/(6n))} \qquad (78)
$$

we have $\forall Q\in\mathcal{B}$, $\mathcal{T}_D^{LVD}Q\in\mathcal{B}$.

Proof. According to Lemma 3, with the condition

$$
0<\varepsilon\leq\frac{\mathcal{E}(Q^*)}{3n^2|\mathcal{A}|^n 2^{n+1}(R_{\max}/(1-\gamma)+\mathcal{E}(Q^*)/(6n))}\leq\frac{\mathcal{E}(Q^*)}{3n^2|\mathcal{A}|^n 2^{n+1}(R_{\max}+\gamma\|V\|_\infty)} \qquad (79)
$$

we have $\forall Q\in\mathcal{B}$, $\forall s\in\mathcal{S}$,

$$
\left|(\mathcal{T}_D^{\text{LVD}}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s))-(\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s))\right|\leq\frac{1}{3}\mathcal{E}(Q^*) \qquad (80)
$$

which implies $\forall Q\in\mathcal{B}$, $\forall s\in\mathcal{S}$,

$$
(\mathcal{T}_D^{\text{LVD}}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s))\geq(\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s))-\frac{1}{3}\mathcal{E}(Q^*). \qquad (81)
$$

According to Lemma 6, with the condition

$$
0<\varepsilon\leq\frac{\mathcal{E}(Q^*)}{3n^2|\mathcal{A}|^n 2^n(R_{\max}/(1-\gamma)+\mathcal{E}(Q^*)/(6n))} \qquad (82)
$$

we have $\forall Q\in\mathcal{B}$, $\forall s\in\mathcal{S}$, $\forall a\in\mathcal{A}^n\setminus\{\boldsymbol{\pi}^*(s)\}$,

$$
\begin{aligned}
(\mathcal{T}_D^{\text{LVD}}Q)_{\text{tot}}(s,\mathbf{a}) &\leq (\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s))-\mathcal{E}(Q^*)+2n\gamma\|V_{\text{tot}}-V^*\|_\infty+\frac{1}{3}\mathcal{E}(Q^*)\\
&\leq (\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s))-\mathcal{E}(Q^*)+\frac{1}{3}\mathcal{E}(Q^*)+\frac{1}{3}\mathcal{E}(Q^*)\\
&=(\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s))-\frac{1}{3}\mathcal{E}(Q^*)\\
&\leq (\mathcal{T}_D^{\text{LVD}}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s)) \qquad (83)
\end{aligned}
$$

which implies $\mathcal{E}(\mathcal{T}_D^{\text{LVD}}Q)\geq 0$.

According to Lemma 4, with the condition

$$0 < \varepsilon \leq \frac{(1-\gamma)\mathcal{E}(Q^*)}{6\gamma n^3 |\mathcal{A}|^n 2^{n+1}(R_{\max}/(1-\gamma) + \mathcal{E}(Q^*)/(6n))}, \tag{84}$$

we have $\forall Q \in \mathcal{B}, \forall s \in \mathcal{S}$,

$$\left|(\mathcal{T}_D^{\text{LVD}} V)(s) - V^*(s)\right| = \left|(\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - V^*(s)\right| \leq \frac{1}{6n\gamma}\mathcal{E}(Q^*). \tag{85}$$

The overall condition is

$$0 < \varepsilon \leq \frac{(1-\gamma)\mathcal{E}(Q^*)}{6n^3 |\mathcal{A}|^n 2^{n+1}(R_{\max}/(1-\gamma) + \mathcal{E}(Q^*)/(6n))} \tag{86}$$

□

Lemma 1. *There exists a threshold $\delta > 0$ such that the on-policy Bellman operator $\mathcal{T}_\epsilon^{LVD}$ is closed in the following subspace $\mathcal{B} \subset \mathcal{Q}^{LVD}$, when the hyper-parameter ϵ is sufficiently small.*

$$\mathcal{B} = \left\{ Q \in \mathcal{Q}^{LVD} \;\middle|\; \boldsymbol{\pi}_Q = \boldsymbol{\pi}^*,\ \max_{s \in \mathcal{S}} |Q_{tot}(s, \boldsymbol{\pi}^*(s)) - V^*(s)| \leq \delta \right\}$$

Formally, $\exists \delta > 0$, $\exists \epsilon > 0$, $\forall Q \in \mathcal{B}$, there must be $\mathcal{T}_\epsilon^{LVD} Q \in \mathcal{B}$.

Proof. It refers to an alternative description of Lemma 7. □

Theorem 2. *Algorithm 1 will have a fixed point value function expressing the optimal policy if the hyper-parameter ϵ is sufficiently small.*

Proof. Notice that the subspace $\mathcal{B}$ defined in Lemma 7 is a compact and convex space. The operator $\mathcal{T}_D^{\text{LVD}}$ is a continuous mapping because it only involves elementary functions. According to Brouwer's Fixed Point Theorem [33], there exist $Q \in \mathcal{B}$ satisfying $\mathcal{T}_D^{\text{LVD}} Q$. □

C Omitted Proofs in Section 5

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

$$\forall Q, Q' \in \mathcal{Q}, \quad \|\mathcal{T}Q - \mathcal{T}Q'\| \leq \gamma \|Q - Q'\|_\infty \tag{11}$$

Proof. Assume the empirical Bellman operator $\mathcal{T}_D^{\text{LVD}}$ is a γ-contraction. For any MMDPs, when using a uniform data distribution, the value function of FQI-LVD will converge [26] because of the contraction of the distance (infinity norm) between any pair of Q. However, one counterexample is indicated in Proposition 2, which shows that there exists MMDPs such that, when using a uniform data distribution, the value function of FQI-LVD diverges to infinity from an arbitrary initialization $Q^{(0)}$. The assumption of γ-contraction is not hold and the empirical Bellman operator $\mathcal{T}_D^{\text{LVD}}$ is not a γ-contraction. □

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

Proof. We consider the following MMDP with 2 agents, 2 states (s_1, s_2) and each agent $(i = 1, 2)$ has 2 actions $\mathcal{A} \equiv \left\{\mathcal{A}^{(1)}, \mathcal{A}^{(2)}\right\}$. The reward function is listed below which $r(s_j, \mathbf{a})$ denotes the reward of $(s_j, \mathbf{a})$, where $\mathbf{a} = \langle a_1, a_2 \rangle$.

$$r(s_1) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \quad r(s_2) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \tag{87}$$

Besides, the transition is deterministic.

$$T(s_1) = \begin{pmatrix} s_1 & s_1 \\ s_1 & s_1 \end{pmatrix} \quad T(s_2) = \begin{pmatrix} s_2 & s_2 \\ s_2 & s_1 \end{pmatrix} \tag{88}$$

Furthermore, $\gamma \in (\frac{4}{5}, 1)$. (In practice, γ is usually chosen as 0.99 or 0.95.) The following proves that this MMDP will diverge for any initialization.

Denote $Q_i^t(s_j, a_i)$ as the decomposed Q-value of agent i after t^{th} value-iteration at state s_j with action a_i. Then, the total Q-value can be described as $Q_{\text{tot}}^t(s_j, \mathbf{a}) = Q_1^t(s_j, a_1) + Q_2^t(s_j, a_2)$. For brevity, 0^{th} Q-value is its initialization.

First, we clarify the process of each iteration. Since the value-iteration for linear decomposed function class is solving the MSE problem in Lemma 2. $\mathbf{b}$ is target one-step TD-value w.r.t the Q-value of the last iteration. Through described in Lemma 2, the optimal solution of this MSE problem is not unique. We can ignore the term of an arbitrary vector $\mathbf{w}$ when considering the joint action-value functions, because $\mathbf{w}$ does not affect the local action selection of each agent and will be eliminated in the summation operator of linear value decomposition. In addition, under uniformed sampling, we observe that $p_u(a_u) = \frac{1}{2}$ for any $\vec{a}, u$. Then, in equation 32

$$-\frac{1}{m}\frac{\mathbf{p}(\vec{a})}{p_u(a_u)} + \frac{1}{mn}\sum_{u'=0}^{n-1}\frac{\mathbf{p}(\vec{a})}{p_{u'}(a_{u'})} = 0 \tag{89}$$

Second, we denote $V_{\text{tot}}^t(s_j) = \max_{\mathbf{a}} Q_{\text{tot}}^t(s_j, \mathbf{a})$ and observe that $\forall t \geq 1, s_j$

$$Q_1^t(s_j, a_1) = \frac{1}{2}\sum_{a_2 \in \mathcal{A}}\left(r(s_j, \mathbf{a}) + \gamma V_{\text{tot}}^{t-1}(T(s_j, \mathbf{a})\right) - \frac{1}{2}\sum_{\mathbf{a} \in \boldsymbol{\mathcal{A}}}\frac{1}{4}\left(r(s_j, \mathbf{a}) + \gamma V_{\text{tot}}^{t-1}(T(s_j, \mathbf{a}))\right) \tag{90}$$

$$= Q_2^t(s_j, a_2) \tag{91}$$

The second equation holds because the transition T and the reward R are symmetric for both agents. Thus, we omit the subscript of local Q-values as $Q^t(s_j, a)$ when $t \geq 1$.

Third, we analyze the Q-values on state s_1. Clearly, its iteration is irrelevant with s_2. According to equation 90, $\forall a \in \mathcal{A}, t \geq 1$

$$Q^t(s_1, a) = \frac{\gamma}{2}V_{\text{tot}}^{t-1}(s_1) \tag{92}$$

$$= \frac{\gamma}{2}\max_{a_1, a_2 \in \mathcal{A}}\left(Q^{t-1}(s_1, a_1) + Q^{t-1}(s_1, a_2)\right) \tag{93}$$

Clearly, when $t \geq 1, Q^t\left(s_1, \mathcal{A}^{(1)}\right) = Q^t\left(s_1, \mathcal{A}^{(2)}\right)$. Therefore, we observe that $Q^t(s_1, \cdot) = \gamma^t q_1, \forall t \geq 1$ where q_1 is determined by the initialization $Q_{\text{tot}}^0(s_1, \mathbf{a}), \forall \mathbf{a} \in \boldsymbol{\mathcal{A}}$.

Last, we consider state s_2. It is straightforward to observe the following recursion for $t \geq 2$ from equation 90

$$\begin{aligned} Q^t\left(s_2, \mathcal{A}^{(1)}\right) &= \frac{1}{2}(1 + 2\gamma V_{\text{tot}}^{t-1}(s_2)) - \frac{1}{8}[1 + \gamma(3V_{\text{tot}}^{t-1}(s_2) + V_{\text{tot}}^{t-1}(s_1))] \\ &= \frac{5\gamma}{8}V_{\text{tot}}^{t-1}(s_2) + \frac{3}{8} - \frac{1}{4}\gamma^t q_1 \\ &= \frac{5\gamma}{4}\max_{a \in \mathcal{A}} Q^{t-1}(s_2, a) + \frac{3}{8} - \frac{1}{4}\gamma^t q_1 \end{aligned} \tag{94}$$

$$\begin{aligned} Q^t\left(s_2, \mathcal{A}^{(2)}\right) &= \frac{1}{2}(\gamma V_{\text{tot}}^{t-1}(s_2) + \gamma V_{\text{tot}}^{t-1}(s_1)) - \frac{1}{8}[1 + \gamma(3V_{\text{tot}}^{t-1}(s_2) + V_{\text{tot}}^{t-1}(s_1))] \\ &= \frac{\gamma}{8}V_{\text{tot}}^{t-1}(s_2) - \frac{1}{8} + \frac{3}{4}\gamma^t q_1 \\ &= \frac{\gamma}{4}\max_{a \in \mathcal{A}} Q^{t-1}(s_2, a) - \frac{1}{8} + \frac{3}{4}\gamma^t q_1 \end{aligned} \tag{95}$$

We consider some $\delta > 0$ and $t_\delta = \left\lceil \log_\gamma \frac{\delta}{6|q_1|} \right\rceil$. Then, $t > t_\delta$

$$Q^t\left(s_2, \mathcal{A}^{(2)}\right) \geq \frac{\gamma}{4}\max_{a \in \mathcal{A}} Q^{t-1}(s_2, a) - \frac{1+\delta}{8} \geq \frac{\gamma}{4}Q^{t-1}\left(s_2, \mathcal{A}^{(2)}\right) - \frac{1+\delta}{8} \tag{96}$$

Denote $\widehat{Q}^t\left(s_2, \mathcal{A}^{(2)}\right) = \frac{\gamma}{4}\widehat{Q}^{t-1}\left(s_2, \mathcal{A}^{(2)}\right) - \frac{1+\delta}{8}, \forall t > t_\delta$ and $\widehat{Q}^{t_\delta}\left(s_2, \mathcal{A}^{(2)}\right) = Q^{t_\delta}\left(s_2, \mathcal{A}^{(2)}\right)$. Consequently, $Q^t(s_2, a_2) \geq \widehat{Q}^{t_\delta}\left(s_2, \mathcal{A}^{(2)}\right), \forall t \geq t_\delta$ by equation 96. Since $t \geq t_\delta$

$$\widehat{Q}^t\left(s_2, \mathcal{A}^{(2)}\right) = \left(\frac{\gamma}{4}\right)^{t-t_\delta}\left(Q^{t_\delta}\left(s_2, \mathcal{A}^{(2)}\right) - \frac{1+\delta}{2\gamma-8}\right) + \frac{1+\delta}{2\gamma-8} \tag{97}$$

Furthermore, $\gamma \in (\frac{4}{5}, 1)$. There exists some $T_\delta \geq t_\delta$ which

$$Q^{T_\delta}\left(s_2, \mathcal{A}^{(2)}\right) \geq \widehat{Q}^{T_\delta}\left(s_2, \mathcal{A}^{(2)}\right) \geq \frac{1+2\delta}{2\gamma-8} > -\frac{1+2\delta}{6} \tag{98}$$

According to equation 94 and let $\delta < \frac{1}{11}$.

$$Q^{T_\delta+1}\left(s_2, \mathcal{A}^{(1)}\right) \geq \frac{5\gamma}{4}Q^{T_\delta}\left(s_2, \mathcal{A}^{(2)}\right) + \frac{3}{8} - \frac{1}{4}\gamma^t q_1 \tag{99}$$

$$> -\frac{5+10\delta}{24} + \frac{3}{8} - \frac{1}{24}\delta \tag{100}$$

$$> \frac{1}{8} \tag{101}$$

Similar to equation 96, we observer from equation 94 that $\forall t > T_{\delta=\frac{1}{11}} + 1$

$$Q^t\left(s_2, \mathcal{A}^{(1)}\right) \geq \frac{5\gamma}{4}Q^{t-1}\left(s_2, \mathcal{A}^{(1)}\right) + \frac{1}{4} \tag{102}$$

and

$$V_{\text{tot}}^t\left(s_2\right) = 2Q^t\left(s_2, \mathcal{A}^{(1)}\right) \tag{103}$$

$$\geq 2\left(\frac{5\gamma}{4}Q^{t-1}\left(s_2, \mathcal{A}^{(1)}\right) + \frac{1}{4}\right) \tag{104}$$

$$= \frac{5\gamma}{4}V_{\text{tot}}^{t-1}\left(s_2\right) + \frac{1}{4} \tag{105}$$

Since $\frac{5\gamma}{4} > 1$ and the initial point at $T_{\delta=\frac{1}{11}} + 1$ is larger than $\frac{1}{8}$, this suggests that $V_{\text{tot}}^t\left(s_2\right)$ will eventually diverge.

Noticing that our proof holds with respect to any $\left\{Q_{\text{tot}}^0(s_j, \mathbf{a}) | \forall j \in \mathcal{S}, \mathbf{a} \in \boldsymbol{\mathcal{A}}\right\}$. Thus, value-iteration on linear decomposed function class w.r.t this MDP will diverge evnetually under any circumstances. □

D Experiment Settings and Implementation Details

D.1 Implmentation Details

We adopt the PyMARL [20] implementation with default hyper-parameters to investigate five state-of-the-art multi-agent Q-learning algorithms: QTRAN [14], Qatten [19], QMIX [13], VDN [11], and IQL [34]. The training time of these algorithms on an NVIDIA RTX 2080TI GPU is about 4 hours to 12 hours, which is depended on the number of agents and the episode length limit of each map. The performance measure of StarCraft II tasks is the percentage of episodes in which RL agents defeat all enemy units within the limited time constraints, called *test win rate*. The dataset providing off-policy exploration is constructed by training a behaviour policy of QMIX [13] and collecting its 20k or 50k experienced episodes. The dataset configurations are shown in Table 1. We investigate five multi-agent Q-learning algorithms over 6 random seeds, which includes 3 different datasets and evaluates two seeds on each dataset. We train 300 epochs to evaluate the learning performance with a given static dataset, of which 32 episodes are trained in each update and 160k transitions are trained for each epoch totally. Moreover, the training process of behaviour policy is the same as that discussed in PyMARL [20], which has collected a total of 2 million timestep data and anneals the hyper-parameter ϵ of ϵ-greedy exploration strategy linearly from 1.0 to 0.05 over 50k timesteps. In the two-state MMDP shown in Figure 1a, due to the GRU-based implementation of

Map Name	Replay Buffer Size	Behaviour Test Win Rate	Behaviour Policy
5m_vs_6m	50k episodes	69%	QMIX
2s3z	20k episodes	95%	QMIX
1c3s5z	20k episodes	90%	QMIX
2c_vs_64zg	50k episodes	76%	QMIX

Table 1: The dataset configurations of offline data collection setting.

Map Name	Ally Units	Enemy Units
5m_vs_6m	5 Marines	6 Marines
2s3z	2 Stalkers & 3 Zealots	2 Stalkers & 3 Zealots
1c3s5z	1 Colossus, 3 Stalkers & 5 Zealots	1 Colossus, 3 Stalkers & 5 Zealots
2c_vs_64zg	2 Colossi	64 Zerglings

Table 2: SMAC challenges.

the finite-horizon paradigm in the above five deep multi-agent Q-learning algorithms, we assume that two agents starting from state s_2 have 100 environmental steps executed by a uniform ϵ-greedy exploration strategy (*i.e.*, $\epsilon = 1$). We use this long-term horizon pattern and uniform ϵ-greedy exploration methods to approximate an infinite-horizon MMDP paradigm , which is discussed with the uniform data distribution in our theoretical framework.

D.2 StarCraft II

StarCraft II unit micromanagement tasks consider a combat game of two groups of agents, where StarCraft II takes built-in AI to control enemy units, and MARL algorithms can control each ally unit to fight the enemies. Units in two groups can contain different types of soldiers, but these soldiers in the same group should belong to the same race. The action space of each agent includes noop, move [direction], attack [enemy id], and stop. At each timestep, agents choose to move or attack in continuous maps. MARL agents will get a global reward equal to the amount of damage done to enemy units. Moreover, killing one enemy unit and winning the combat will bring additional bonuses of 10 and 200, respectively. The maps of SMAC challenges in this paper are introduced in Table 2.