TOWARDS UNDERSTANDING LINEAR VALUE DECOMPOSITION IN COOPERATIVE MULTI-AGENT Q-LEARNING

Anonymous authors
Paper under double-blind review

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 or richer Q function classes to improve its trianing stability. In the empirical study, our experiments demonstrate the realizability of our theoretical closed-form formulation and implications in the didactic examples and a broad set of StarCraft II unit micromanagement tasks, respectively.

1 INTRODUCTION

Cooperative multi-agent reinforcement learning (MARL) has great promise for addressing coordination problems in a variety of applications, such as robotic systems (Hüttenrauch et al., 2017), autonomous cars (Cao et al., 2012), and sensor networks (Zhang & Lesser, 2011). 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 (Wolpert & Tumer, 2002; Nguyen et al., 2018), non-stationarity (Zhang & Lesser, 2010; Song et al., 2019), and scalability (Zhang & Lesser, 2011; Panait & Luke, 2005). Recently, by leveraging the strength of deep learning techniques, cooperative MARL has made a series of great progress (Sunehag et al., 2018; Baker et al., 2019; Wang et al., 2020b;a), particularly in value-based methods that demonstrate state-of-the-art performance on challenging tasks such as StarCraft unit micromanagement (Samvelyan et al., 2019). Sunehag et al. (2018) proposed a popular approach called value-decomposition network (VDN) based on the paradigm of *centralized training with decentralized execution* (CTDE; Foerster et al., 2016). 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 (Rashid et al., 2018), QTRAN (Son et al., 2019), and QPLEX (Wang et al., 2020a).

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 complexity, this linear value decomposition corrupts the completeness of the joint action-value function class. The approximation error induced by this incompleteness is known as *inherent Bellman error* (Munos & Szepesvári, 2008) 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 popluar Fitted Q-Iteration (FQI; Ernst et al., 2005; Levine et al., 2020) framework

and derive a closed-form solution to involved empirical Bellman error minimization. To best of our knowledge, it is the first theoretical analysis characterizing the underlying mechanism of linear value decomposition in cooperative multi-agent Q-learning, which can serve as a powerful toolkit to establish follow-up profound theory and explore potential insights from different perspectives in this popular structure.

By utilizing this novel closed-form solution, this paper reveals two interesting insights from our intriguing theorems: (1) Learning linear value decomposition implicitly implements a classical multi-agent credit assignment method called *counterfactual difference rewards* (Wolpert & Tumer, 2002), which draws a connection with COMA (Foerster et al., 2018), a multi-agent policy-gradient method. (2) Multi-agent Q-learning with linear value decomposition potentially suffers from the risk of unbounded divergence from any initialization. On-policy data distribution or richer Q function classes can improve its local or global convergence, respectively.

Finally, we set up an extensive set of experiments to demonstrate the realizability of our theoretical implications. Besides the FQI framework, we also consider the deep-learning-based implementations of different multi-agent value decomposition structures. Through didactic examples and the StarCraft II benchmark, we design several fair comparisons to illustrate the effectiveness of our closed-form formulation, and which online data distribution and richer Q function classes can significantly alleviate the limitations of VDN on offline training process (Ernst et al., 2005; Levine et al., 2020).

2 Related Works

Deep Q-learning algorithms that use neural networks as function approximators have shown great promise in solving complicated decision-making problems (Mnih et al., 2015). One of the core components of such methods is iterative Bellman error minimization which can be modeled by a classical framework called Fitted Q-Iteration (FQI; Ernst et al., 2005). Without considering the estimation error, FQI utilizes a specific Q function class to optimize iterative Bellman error on a dataset D. Numerous efforts have been made towards theoretically characterizing the behavior of FQI with finite samples and imperfect function classes (Munos & Szepesvári, 2008; Farahmand et al., 2010; Chen & Jiang, 2019). From a empirical perspective, there is also a growing trend to adopt FQI for empirical analysis of deep offline Q-learning algorithms (Fu et al., 2019; Levine et al., 2020). In MARL, joint Q function classes grow exponentially with the number of agents. In this paper, we extend FQI to a multi-agent variant as our grounding theoretical framework for analyzing cooperative multi-agent Q-learning with linear value decomposition (which has limited function expressiveness).

To achieve superior effectiveness and scalability in the multi-agent setting, centralized training with decentralized executing (CTDE) has become a popular MARL paradigm (Oliehoek et al., 2008; Kraemer & Banerjee, 2016). *Individual-Global-Max* (IGM) (Son et al., 2019) has become a critical concept for value-based CTDE (Mahajan et al., 2019), whose consistency between joint and local greedy action selections ensures effective performance in the training and execution phases. VDN (Sunehag et al., 2018) utilizes linear value decomposition to realize a sufficient condition of IGM. This simple additivity structure of VDN has achieved excellent scalability and has become a classic part of follow-up algorithms. QMIX (Rashid et al., 2018) proposes a monotonic Q network structure to improve the expressiveness of function class. QTRAN (Son et al., 2019) tries to implment the entire IGM function class, but his method is computationally intractable and requires two additional soft regularizations to approximate IGM (which actually loses the IGM guarantee). QPLEX (Wang et al., 2020a) encodes the IGM principle into the Q network architecture to achieve the entire IGM function class, but it may also have some potential limitations in scalability. Based on the advantages of VDN's simplicity and scalability, linear value decomposition becomes very popular in MARL (Son et al., 2019; Wang et al., 2020a;c). This paper focuses on the theoretical and empirical understanding of multi-agent Q-learning with linear value decomposition to explore its underlying implications.

3 Notations and Preliminaries

3.1 Multi-agent Markov Decision Process (MMDP)

To support theoretical analyses on multi-agent Q-learning, we adopt the framework of MMDP (Boutilier, 1996), a special case of Dec-POMDP (Oliehoek et al., 2016), to model fully cooperative

multi-agent decision-making tasks. MMDP is defined as a tuple $\mathcal{M} = \langle \mathcal{N}, \mathcal{S}, \mathcal{A}, P, r, \gamma \rangle$. $\mathcal{N} \equiv \{1, \ldots, n\}$ 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 $\boldsymbol{\pi} = \langle \pi_1, \ldots, \pi_n \rangle$ 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 $Q^{\boldsymbol{\pi}}(s, \mathbf{a}) = r(s, \mathbf{a}) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,\mathbf{a})}[V^{\boldsymbol{\pi}}(s')]$. We use Q^* and V^* to denote the action-value function and the state-value function corresponding to the optimal policy $\boldsymbol{\pi}^*$, respectively.

3.2 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 (Foerster et al., 2016). 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) (Son et al., 2019) 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:

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

As stated in Eq. (2), the additivity constraint adopted by VDN (Sunehag et al., 2018) 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 action-value function class, because the linear number of individual functions cannot represent a joint action-value function class exponential with the number of agents.

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

3.3 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) (Ernst et al., 2005) 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 $D = \{(s, \mathbf{a}, r, s')\}$,

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

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) (Sunehag et al., 2018; Rashid et al., 2018), 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.

4 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 decomposi-

tion proposed by VDN (Sunehag et al., 2018) 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 underlying mechanism of credit assignment implicitly realized by linear value decomposition techniques of VDN.

4.1 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 (Krishnamurthy et al., 2016).

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 (Farahmand et al., 2010; Chen & Jiang, 2019). 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

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

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 3.3.

Value-decomposition network (VDN) (Sunehag et al., 2018) 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 $y^{(t)}$.

4.2 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* (Moore, 1920) 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)), which indicates that joint action-value $Q_{\text{tot}}^{(t+1)} = \sum_i Q_i^{(t+1)}$ has a unique closed-form solution. We compare the theoretical analysis of FQI-LVD with the empirical results of VDN to demonstrate and verify the effectiveness of closed-form updating rule of FQI-LVD (see Eq. (8)) in Section 6.1.

Note that, if we regard the empirical probability $p_D(\mathbf{a}|s)$ within the dataset D as a *default policy*, the first term of Eq. (8) is the expected value of an individual action a_i, and the second term is the expected value of the default policy, which is considered as the *counterfactual baseline*. 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) (Foerster et al., 2018).

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 $(n-1)/n$ brings our derived credit assignment to be more consistent and meaningful in the sense that all amounts of global rewards should be assigned to agents. Consider a simple case where all joint actions generate the same reward signals, Eq. (9) will assign $1/n$ unit of rewards to each agent, but COMA will assign 0. Moreover, this gap will gradually close when n becomes sufficiently large.

5 Improving the Stability of Linear Value Decomposition Training

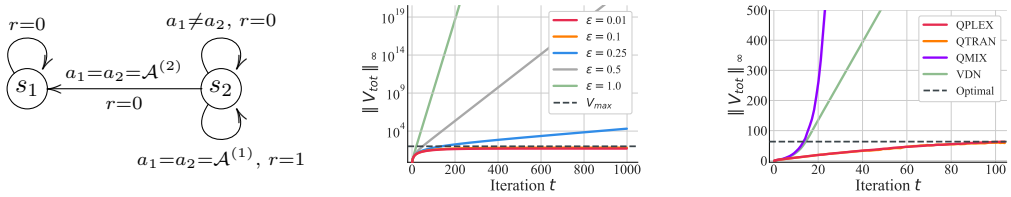
In the previous section, we have derived the closed-form update rule of FQI-LVD, which reveals the underlying credit assignment mechanism provided by linear value decomposition structure. This derivation also enables us to investigate more algorithmic functionalities of linear value decomposition in multi-agent Q-learning. Although linear value decomposition enjoys superior scalability in multi-agent settings, we reveal that FQI-LVD has a potential unbounded divergence risk from any initialization. To improve the stability of linear value decomposition training, we theoretically demonstrate that on-policy data distribution or richer Q function classes can realize some convergence guarantees. Moreover, we also utilize a concrete MMDP example to visualize our implications.

5.1 Unbounded Divergence in Offline Training

When offline training with a dataset D, we will provide an analysis on the convergence of FQI-LVD.

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.*

$$\forall Q_{tot}, Q'_{tot} \in \mathcal{Q}, \quad \|\mathcal{T} Q_{tot} - \mathcal{T} Q'_{tot}\|_\infty \leq \gamma \|Q_{tot} - Q'_{tot}\|_\infty \tag{10}$$



(a) A simple counterexample (b) On-policy FQI-LVD (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 $\|V_{\text{tot}}\|_\infty$ of on-policy FQI-LVD 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 the convergence proof for the standard *Bellman optimality operator* $\mathcal{T}$ (Sutton & Barto, 2018), Eq. (10) is a critical step to derive the theoretical guarantee on global convergence. In the context of FQI-LVD, the additivity constraint limits the joint action-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 induced by the negative important weight $(n-1)/n$ stated in Eq. (8) and is also known as *inherent Bellman error* (Munos & Szepesvári, 2008), 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. The learning curve of $\epsilon = 1.0$ (green one) in Figure 1b refers to an offline 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 $Q^{(0)}$. 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 (Sunehag et al., 2018), a deep-learning-based implementation of FQI-LVD, result in an unbounded divergence. We postpone the discussion of other deep-learning-based algorithms after Theorem 3.

5.2 Local and Global Convergence Improvements

To improve the training stability of FQI-LVD, we investigate the local and global convergence improvements of FQI-LVD, which contributes to setting up a more detailed characterization for the algorithmic functionality of linear value decomposition.

Local Convergence Improvement. 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. Here, we switch to study the behavior of linear value decomposition in an on-policy setting, i.e., the dataset D is accumulated by running a ϵ-greedy policy (Mnih et al., 2015). Based on the on-policy data collection, we include an informal statement of local stability of FQI-LVD below, deferring the precise version, the proof, and the algorithm box of on-policy FQI-LVD with detailed discussion to Appendix C.1.

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

Theorem 2 indicates that multi-agent Q-learning with linear value decomposition has a convergent region, where the value function induces optimal actions. By combining this local stability with Brouwer's fixed-point theorem (Brouwer, 1911), we can further verify the existence of a fixed-point solution for the on-policy Bellman operator $\mathcal{T}_{D_t}^{\text{LVD}}$. Figure 1b visualizes the performance of on-policy FQI-LVD with different values of the hyper-parameter ϵ. With a smaller ϵ (such as 0.1 or 0.01),

on-policy FQI-LVD performs numerical stability and their corresponding collected datasets are more closer to an on-policy distribution.

Global Convergence Improvement. Linear value decomposition structure limits the joint action-value function class $\mathcal{Q}^{\text{LVD}}$, which is the origin of deviation of the empirical Bellman operator $\mathcal{T}_D^{\text{LVD}}$ discussed in Proposition 1. One direct way to improve training stability is to enrich the expressiveness of $\mathcal{Q}^{\text{LVD}}$. Consider a multi-agent fitted Q-iteration (FQI) with full action-value function class derived from IGM, named FQI-IGM, whose action-value function class

$$\mathcal{Q}^{\text{IGM}} = \left\{ Q \;\middle|\; Q_{\text{tot}} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|^n} \text{ and } \left[\forall Q_i \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}\right]_{i=1}^{n} \text{ with that Eq. (1) is satisfied} \right\}. \tag{11}$$

Noted that $\mathcal{Q}^{\text{LVD}} \subseteq \mathcal{Q}^{\text{IGM}}$ indicates that linear decomposition structure stated in Eq. (2) is a sufficient condition for the IGM constraint. The formal definition of FQI-IGM is deferred to Appendix C.2 and the global convergence stability can be derived in the following theorem now.

Theorem 3. *FQI-IGM will globally converge to optimal value function.*

Theorem 3 relies on a fact that $\mathcal{Q}^{\text{IGM}}$ is complete in MMDP setting, i.e., *inherent Bellman error* discussed in Proposition 1 can reach 0 and the empirical Bellman operator $\mathcal{T}_D^{\text{IGM}}$ is still a γ-contraction. Especially, when assuming universal function approximation of neural networks, QPLEX (Wang et al., 2020a) theoretically realize the complete IGM function class. Figure 1c shows that QPLEX, a deep-learning-based implementation of FQI-IGM, performs outstanding numerical stability. Another multi-agent Q-learning algorithm with richer expressiveness, QTRAN (Son et al., 2019), also converges in the given MMDP (see Figure 1a). However, QMIX (Rashid et al., 2018), which has a non-linear monotonic factorization but still underrepresent the IGM function class, also diverges to infinity in this task.

Combining the theoretical and empirical results in this section, we summarize the following insights.

Implication 2. *Multi-agent Q-learning with linear value decomposition potentially suffers from the risk of unbounded divergence from any initialization. On-policy data distribution or richer Q function classes can improve its local or global convergence, respectively.*

Although linear value decomposition is well-positioned in the literature of multi-agent Q-learning, its limited Q function class can induce potential training stability risks. To improve learning stability is critical for FQI-LVD and in this section we theoretically analyze that FQI-LVD equipped with on-policy data distribution or IGM function class can achieve local or global convergence guarantees, respectively. Moreover, we also empirically evaluate our theorems on a representative MMDP task (see Figure 1a).

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. We evaluate four state-of-the-art deep-learning-based methods, VDN (Sunehag et al., 2018), QMIX (Rashid et al., 2018), QTRAN (Son et al., 2019), QPLEX (Wang et al., 2020a) on the matrix game proposed by QTRAN and StarCraft Multi-Agent Challenge (SMAC) benchmark tasks (Samvelyan et al., 2019). In addition, empirical analysis of another didactic example, Two-State MMDP, has been illustrated in Section 5. The implementation details of four baselines and experimental settings are deferred to Appendix F. We evaluate all experiments with 6 random seeds and demonstrate them with median performance and 25-75% percentiles.

6.1 Is Our Closed-Form Update Rule of Linear Value Decomposition Effective in Deep-Learning-Based Implementation?

As shown in Theorem 1, we derive the closed-form update rule of FQI-LVD. From an optimization perspective, FQI-LVD and VDN share the same objective function (see Definition 1) but different optimization methods ($\arg\min$ vs. gradient descent). Starting from a common matrix game used by QTRAN (Son et al., 2019) and QPLEX (Wang et al., 2020a) stated in Table 1a, we will illustrste the effectiveness of our closed-form formulation. This matrix game describes a simple cooperative multi-agent, which includes two agents and three actions. Miscoordination penalties are also considered,

a_2 \ a_1	$\mathcal{A}^{(1)}$	$\mathcal{A}^{(2)}$	$\mathcal{A}^{(3)}$
$\mathcal{A}^{(1)}$	**8**	-12	-12
$\mathcal{A}^{(2)}$	-12	0	0
$\mathcal{A}^{(3)}$	-12	0	0

(a) Payoff of matrix game.

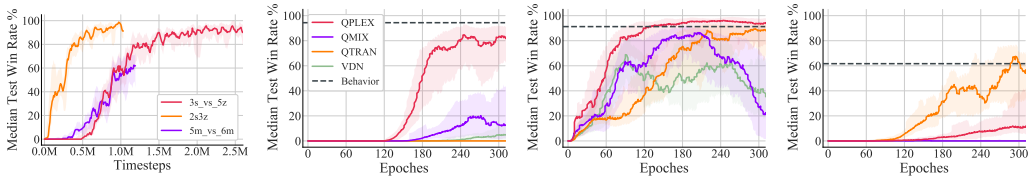
a_2 \ a_1	$\mathcal{A}^{(1)}$	$\mathcal{A}^{(2)}$	$\mathcal{A}^{(3)}$
$\mathcal{A}^{(1)}$	-6.22	-4.89	-4.89
$\mathcal{A}^{(2)}$	-4.89	**-3.56**	-3.56
$\mathcal{A}^{(3)}$	-4.89	-3.56	-3.56

(b) Q_{tot} of FQI-LVD.

a_2 \ a_1	$\mathcal{A}^{(1)}$	$\mathcal{A}^{(2)}$	$\mathcal{A}^{(3)}$
$\mathcal{A}^{(1)}$	-6.44	-4.98	-4.97
$\mathcal{A}^{(2)}$	-4.97	**-3.46**	-3.48
$\mathcal{A}^{(3)}$	-4.97	-3.49	-3.49

(c) Q_{tot} of VDN.

Table 1: (a) Payoff matrix of the one-step game. Boldface means the optimal joint action selection from payoff matrix. (b,c) Joint action-value functions Q_{tot} of FQI-LVD and VDN. Boldface means the greedy joint action selection from Q_{tot}.



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

Figure 2: (a) Constructing datasets using online data collection. (b-d) Evaluating the learning performance of deep multi-agent Q-learning algorithms with a given static dataset on three maps.

where the optimal strategy for two agents is to perform action $\mathcal{A}^{(1)}$ simultaneously. We adopt a full exploration strategy (i.e., ϵ-greedy exploration with $\epsilon = 1$) conducted over 100k steps.

Table 1b and 1c show the joint action-value functions of FQI-LVD and VDN, respectively. Comparing with these two joint action-value functions, we find that the estimation error of VDN is only $\|Q_{\text{tot}}^{\text{FQI-LVD}} - Q_{\text{tot}}^{\text{VDN}}\|_\infty = 0.22$, which strongly confirms the accuracy of our theorem. Moreover, as discussed by QTRAN and QPLEX, VDN with limited function class cannot learn the optimal policy in this didactic matrix game. The empirical results of QPLEX, QTRAN, and QMIX are deferred to Appendix F.4, where QPLEX and QTRAN can solve this task, but QMIX still cannot.

6.2 Is Linear Value Decomposition Limited in Offline Training?

In the literature of offline 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 (Gordon, 1995) or even unbounded divergence (Baird, 1995; Tsitsiklis & Van Roy, 1996). 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 needs to improve its training stability.

To investigate the performance of deep multi-agent Q-learning in an offline training mode, we adopt a similar experiment setting with Levine et al. (2020). This setting is named *offline reinforcement learning* in which the learning algorithm can only access to a given static dataset and cannot need extra online data collection. Offline reinforcement learning can equip with multi-source datasets and hold great promise for realizing powerful decision making engines (Levine et al., 2020). Differing with other related work studying distributional shift (Fujimoto et al., 2019; Levine et al., 2020; Yu et al., 2020), we aim to adopt a diverse dataset to investigate the effect of Q function expressiveness on offline training, i.e. which value decomposition structure is suitable for multi-agent offline reinforcement learning. The datasets we used are constructed by training a behavior policy and collecting all its experienced transitions during the whole training procedure.

We evaluate the learning curve of StarCraft II with offline data collection on 9 common maps. The results of 3 maps are shown in Figure 2 and the other 6 maps were deferred to Appendix F.5. To approximate the MMDP setting, we concatenate the global state with the local observations for each agent to handle partial observability. Figure 2(b-d) illustrates that VDN (Sunehag et al., 2018) and QMIX (Rashid et al., 2018) performs poorly and cannot utilize the off-policy dataset collected by an unfamiliar behavior policy. In contrast, QPLEX (Wang et al., 2020a) and QTRAN (Son et al., 2019) with richer Q function class perform pretty well and may reach *Behavior* line, which indicates that Q function expressiveness may dramatically affect multi-agent offline Q-learning. The learning

curves of *Behavior* line are shown in Figure 2a, which is implemented by VDN with ϵ-greedy online data collection. Figure 2a and deferred figures in Appendix F.5 claim that VDN with online data collection can solve these 9 tasks, but that with offline data collection cannot solve it, that is, there is a considerable gap between online and offline training with linear value decomposition. Although the distribution shift (Levine et al., 2020) can be a potential cause of this gap, the remarkable performance of QPLEX and QTRAN suggests that our datasets may be sufficient for offline training.

To bridge theoretical and empirical analyses in this paper, there may induce several approximations between FQI-based and deep-learning-based implementations. Even so, we still design fair ablation comparisons to visualize the limitations of linear value decomposition during offline training. We study the connection between multi-agent offline reinforcement learning and value function expressiveness, and find that the value decomposition structure of QPLEX and QTRAN may be the state-of-the-art choice for multi-agent offline training rather than linear value decomposition.

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 empirical 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. Besides convergence properties, we open the door to the theoretical study of multi-agent Q learning with linear value decomposition. Many interesting topics (such as credit assignment and scalability) can be further studied in the form of linear value decomposition. We also theoretically and empirically show the limitations of VDN (Sunehag et al., 2018) in the offline training process, whose offline data collection has been seen as a key step towards real-world RL applications (Dulac-Arnold et al., 2019; Levine et al., 2020). In addition to linear value decomposition, QPLEX (Wang et al., 2020a) and QTRAN (Son et al., 2019) can be regarded as reliable backends with different Q function structures, supporting effective multi-agent offline reinforcement learning.

References

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

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

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.

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

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

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.

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

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

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

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

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*, pp. 2137–2145, 2016.

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.

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

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

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

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

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

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

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

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

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.

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

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

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*, pp. 8102–8113, 2018.

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

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

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.

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*, pp. 4292–4301, 2018.

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*, pp. 2186–2188. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

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

Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 5887–5896, 2019.

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*, pp. 935–943. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

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*, pp. 2085–2087, 2018.

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

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

Jianhao Wang, Zhizhou Ren, Terry Liu, Yu Yang, and Chongjie Zhang. Qplex: Duplex dueling multi-agent q-learning. *arXiv preprint arXiv:2008.01062*, 2020a.

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

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

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

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *arXiv preprint arXiv:2005.13239*, 2020.

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

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

A OMITTED PROOFS IN SECTION 4

Lemma 1. *Considering following weighted linear regression problem*

$$\min_{\mathbf{x}} \parallel \sqrt{\mathbf{p}^\top} \cdot (\mathbf{A}\mathbf{x} - \mathbf{b}) \parallel_2^2 \tag{12}$$

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* $\forall i \in [m^n], j \in [mn]$

$$\mathbf{A}_{i,j} = \begin{cases} 1, & \textit{if} \quad \exists u \in [n], j = m \times u + (\lfloor i/m^u \rfloor \textit{ mod } m), \\ 0, & \textit{otherwise}. \end{cases} \tag{13}$$

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*

$$\mathbf{p}_j = \mathbf{p}(\vec{a}_j) = \prod_{u \in [n]} p_u(a_{u,j}), \textit{ where } p_u : [m] \to (0,1) \textit{ and } \sum_{a_u \in [m]} p_u(a_u) = 1, \forall u \in [n] \tag{14}$$

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* $\mathbf{w} \in \mathbb{R}^{mn}$

$$\mathbf{x}_i^* = \sum_{\vec{a}} \frac{\mathbf{p}(\vec{a})}{p_u(a_u)} \mathbf{b}_{\vec{a}} \cdot \mathbf{1}(a_u = v) - \frac{n-1}{n} \mathbf{p}(\vec{a}) \mathbf{b}_{\vec{a}} - \frac{1}{mn} \sum_{i' \in [mn]} \mathbf{w}_{i'} + \frac{1}{m} \sum_{v' \in [m]} \mathbf{w}_{um+v'} \tag{15}$$

Proof. For brevity, denote

$$\mathbf{A}^p = \sqrt{\mathbf{p}^\top} \cdot \mathbf{A}, \qquad \mathbf{b}^p = \sqrt{\mathbf{p}^\top} \cdot \mathbf{b} \tag{16}$$

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 (Moore, 1920):

$$(1)\ \mathbf{A}^p \mathbf{A}^{p,\dagger} \text{ and } \mathbf{A}^{p,\dagger} \mathbf{A}^p \text{ are self-adjoint} \tag{17}$$

$$(2)\ \mathbf{A}^p = \mathbf{A}^p \mathbf{A}^{p,\dagger} \mathbf{A}^p \tag{18}$$

$$(3)\ \mathbf{A}^{p,\dagger} = \mathbf{A}^{p,\dagger} \mathbf{A}^p \mathbf{A}^{p,\dagger} \tag{19}$$

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

$$\begin{aligned} \mathbf{A}_{i,j}^{p,\dagger} &= \mathbf{A}_{i,\vec{a}_j}^{p,\dagger} \\ &= \sqrt{\frac{\mathbf{p}(\vec{a}_{-u,j})}{p_u(a_{u,j})}} \cdot 1(a_{u,j} = v) - \frac{n-1}{n} \sqrt{\mathbf{p}(\vec{a}_j)} - \frac{1}{m} \sqrt{\frac{\mathbf{p}(\vec{a}_{-u,j})}{p_u(a_{u,j})}} + \frac{1}{mn} \sum_{u'=0}^{n-1} \sqrt{\frac{\mathbf{p}(\vec{a}_{-u',j})}{p_{u'}(a_{u',j})}} \end{aligned} \tag{20}$$

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, $O(\vec{a}_i, \vec{a}_j) = \{u | a_{u,i} = a_{u,j}, u \in [n]\}$.

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

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

$$
\begin{aligned}
(\mathbf{A}^{p,\dagger}\mathbf{A}^p)_{i,i} = & \sum_{\vec{a}\in O(i)} \sqrt{\mathbf{p}(\vec{a})}[\sqrt{\frac{\mathbf{p}(\vec{a}_{-u})}{p_u(a_u)}} \cdot \mathbf{1}(a_u = v) - \frac{n-1}{n}\sqrt{\mathbf{p}(\vec{a})} - \frac{1}{m}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u})}{p_u(a_u)}} \\
& + \frac{1}{mn}\sum_{u'=0}^{n-1}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u'})}{p_{u'}(a_{u'})}}] \\
= & \sum_{\vec{a}\in O(i)} \frac{\mathbf{p}(\vec{a})}{p_u(a_u)} - \frac{n-1}{n}\mathbf{p}(\vec{a}) - \frac{1}{m}\frac{\mathbf{p}(\vec{a})}{p_u(a_u)} + \frac{1}{mn}\sum_{u'=0}^{n-1}\frac{\mathbf{p}(\vec{a})}{p_{u'}(a_{u'})} \\
= & \sum_{\vec{a}\in O(i)} \left(\mathbf{p}(\vec{a}_{-u}) - \frac{1}{m}\mathbf{p}(\vec{a}_{-u}) + \frac{1}{mn}\sum_{u'=0}^{n-1}\mathbf{p}(\vec{a}_{-u'})\right) - \frac{n-1}{n}p_u(a_u = v) \\
= & 1 - \frac{1}{m} - \frac{n-1}{n}p_u(a_u = v) + \frac{1}{mn}\sum_{\substack{u'\in[n] \\ u'\neq u}}\sum_{\vec{a}\in O(i)}\mathbf{p}(\vec{a}_{-u'}) \\
& + \frac{1}{mn}\sum_{\vec{a}\in O(i)}\mathbf{p}(\vec{a}_{-u}) \\
= & 1 - \frac{1}{m} - \frac{n-1}{n}p_u(a_u = v) + \frac{1}{mn} + \frac{n-1}{mn}mp_u(a_u = v) \\
= & 1 - \frac{1}{m} + \frac{1}{mn}
\end{aligned} \tag{22}
$$

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

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

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

$$
\begin{aligned}
(\mathbf{A}^{p,\dagger}\mathbf{A}^p)_{i,i'} = & \sum_{\vec{a} \in O(i')} \sqrt{\mathbf{p}(\vec{a})}[\sqrt{\frac{\mathbf{p}(\vec{a}_{-u_1})}{p_{u_1}(a_{u_1})}} \cdot \mathbf{1}(a_{u_1} = v) - \frac{n-1}{n}\sqrt{\mathbf{p}(\vec{a})} \\
& - \frac{1}{m}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u_1})}{p_{u_1}(a_{u_1})}} + \frac{1}{mn}\sum_{u'=0}^{n-1}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u'})}{p_{u'}(a_{u'})}}] \\
= & \sum_{\vec{a} \in O(i) \cap O(i')} \frac{\mathbf{p}(\vec{a})}{p_{u_1}(a_{u_1})} - \frac{n-1}{n}\sum_{\vec{a} \in O(i')} \mathbf{p}(\vec{a}) - \frac{1}{m}\sum_{\vec{a} \in O(i')} \frac{\mathbf{p}(\vec{a})}{p_{u_1}(a_{u_1})} \\
& + \frac{1}{mn}\sum_{\substack{u' \in [n] \\ u' \neq u_2}} \sum_{\vec{a} \in O(i')} \frac{\mathbf{p}(\vec{a})}{p_{u'}(a_{u'})} + \frac{1}{mn}\sum_{\vec{a} \in O(i')} \frac{\mathbf{p}(\vec{a})}{p_{u_2}(a_{u_2})} \\
= & \, p_{u_2}(a_{u_2}) - \frac{n-1}{n} p_{u_2}(a_{u_2}) - p_{u_2}(a_{u_2}) + \frac{n-1}{mn} m p_{u_2}(a_{u_2}) + \frac{1}{mn} \\
= & \frac{1}{mn}
\end{aligned} \tag{24}
$$

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 $\mathbf{A}_0^p$

$$
\begin{aligned}
(\mathbf{A}^p\mathbf{A}_0^p)_{\vec{a},i} = & \sqrt{\mathbf{p}(\vec{a})} \sum_{u \in [n]} (\mathbf{A}_0^p)_{um + a_u, i} \\
= & \sqrt{\mathbf{p}(\vec{a})} \left(\mathbf{1}(\exists u \in [n], i = um + a_u) - \frac{1}{m} + \frac{1}{mn} + (n-1)\frac{1}{mn} \right) \\
= & \sqrt{\mathbf{p}(\vec{a})} \cdot \mathbf{1}(\exists u \in [n], i = um + a_u)
\end{aligned} \tag{25}
$$

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) \\
& -\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 (26)
\end{aligned}
$$

Clearly, we have the following relations

$$
\sum_{u\in[n]}[-\frac{1}{m}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u})}{p_u(a_u)}}+\frac{1}{mn}\sum_{u'=0}^{n-1}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u'})}{p_{u'}(a_{u'})}}]=0 \qquad (27)
$$

$$
\sum_{v\in[m]}(\mathbf{1}(v=v_0)-\frac{1}{m})=0 \qquad (28)
$$

Thus

$$
\begin{aligned}
(\mathbf{A}_0^p\mathbf{A}^{p,\dagger})_{i_0,\vec{a}} = & \frac{1}{mn}\sum_{u\in[n]}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u})}{p_u(a_u)}}-\frac{n-1}{n}\sqrt{\mathbf{p}(\vec{a})}+(\mathbf{1}(a_{u_0}=v_0)-\frac{1}{m})\sqrt{\frac{\mathbf{p}(\vec{a}_{-u_0})}{p_{u_0}(a_{u_0})}} \qquad (29) \\
= & \mathbf{A}_{i_0,\vec{a}}^{p,\dagger} \qquad (30)
\end{aligned}
$$

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 (Moore, 1920).

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

$$
\begin{aligned}
\mathbf{x}^p_i &= \sum_{\vec{a}} \mathbf{A}^{p,\dagger}_{i,\vec{a}} \sqrt{\mathbf{p}(\vec{a})}\mathbf{b}_{\vec{a}} \\
&= \sum_{\vec{a}} [\sqrt{\frac{\mathbf{p}(\vec{a}_{-u})}{p_u(a_u)}} \cdot \mathbf{1}(a_u = v) - \frac{n-1}{n}\sqrt{\mathbf{p}(\vec{a})} - \frac{1}{m}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u})}{p_u(a_u)}} \\
&\quad + \frac{1}{mn}\sum_{u'=0}^{n-1}\sqrt{\frac{\mathbf{p}(\vec{a}_{-u'})}{p_{u'}(a_{u'})}}]\sqrt{\mathbf{p}(\vec{a})}\mathbf{b}_{\vec{a}} \\
&= \sum_{\vec{a}} \left[\frac{\mathbf{p}(\vec{a})}{p_u(a_u)} \cdot \mathbf{1}(a_u = v) - \frac{n-1}{n}\mathbf{p}(\vec{a}) - \frac{1}{m}\frac{\mathbf{p}(\vec{a})}{p_u(a_u)} + \frac{1}{mn}\sum_{u'=0}^{n-1}\frac{\mathbf{p}(\vec{a})}{p_{u'}(a_{u'})}\right]\mathbf{b}_{\vec{a}} \qquad (31)
\end{aligned}
$$

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

$$
(\mathbf{I} - \mathbf{A}^{p,\dagger}\mathbf{A}^p)_{i,i'} = \begin{cases} \frac{1}{m} - \frac{1}{mn} & \textit{if } u = u' \\ -\frac{1}{mn} & \textit{if } u \neq u' \end{cases} \qquad (32)
$$

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

$$
\mathbf{w}_{i_0} = \sum_{\vec{a}\in O(i_0)} \frac{\mathbf{p}(\vec{a})}{p_{u_0}(a_{u_0})}\mathbf{b}_{\vec{a}} \qquad (33)
$$

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

$$
\begin{aligned}
((\mathbf{I} - \mathbf{A}^{p,\dagger}\mathbf{A}^p)\mathbf{w})_i &= \sum_{\substack{i_0\in[mn] \\ u\neq u_0}} -\frac{1}{mn}\mathbf{w}_{i_0} + \sum_{i_0:u_0=u}(\frac{1}{m} - \frac{1}{mn})\mathbf{w}_{i_0} \qquad (34) \\
&= \sum_{\vec{a}} -\frac{1}{mn}\sum_{u'\in[n]}\frac{\mathbf{p}(\vec{a})}{p_{u'}(a_{u'})}\mathbf{b}_{\vec{a}} + \frac{1}{m}\sum_{\vec{a}}\frac{\mathbf{p}(\vec{a})}{p_u(a_u)}\mathbf{b}_{\vec{a}} \qquad (35)
\end{aligned}
$$

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 $\mathbf{w} \in \mathrm{R}^{mn}$.

$$
\mathbf{x}^*_i = \sum_{\vec{a}} \frac{\mathbf{p}(\vec{a})}{p_u(a_u)}\mathbf{b}_{\vec{a}} \cdot \mathbf{1}(a_u = v) - \frac{n-1}{n}\mathbf{p}(\vec{a})\mathbf{b}_{\vec{a}} - \frac{1}{mn}\sum_{i'\in[mn]}\mathbf{w}_{i'} + \frac{1}{m}\sum_{v'\in[m]}\mathbf{w}_{um+v'} \qquad (36)
$$

This completes the proof. □

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

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

and induces the empirical Bellman operator

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

where $y^{(t)}(s,\mathbf{a}) = r(s,\mathbf{a}) + \gamma \max_{\mathbf{a}'} Q^{(t)}_{tot}(s',\mathbf{a}')$ 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 3.3.

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

$$\min_{\mathbf{x}} \parallel \sqrt{\mathbf{p}^\top} \cdot (\mathbf{A}\mathbf{x} - \mathbf{b}) \parallel_2^2 \tag{37}$$

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 $\forall i \in [m^n], j \in [mn]$

$$\mathbf{A}_{i,j} = \begin{cases} 1, & \text{if} \quad \exists u \in [n], j = m \times u + (\lfloor i/m^u \rfloor \; mod \; m), \\ 0, & \text{otherwise.} \end{cases} \tag{38}$$

For simplicity, j^{th} row of $\mathbf{A}$ corresponds to a m-ary number $\vec{a}_j = (j)_m$ where $\vec{a} = a_0 a_1 \ldots a_{n-1}$, 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

$$\mathbf{p}_j = \mathbf{p}(\vec{a}_j) = \prod_{u \in [n]} p_u(a_{u,j}), \; \textit{where } p_u : [m] \to (0, 1) \textit{ and } \sum_{a_u \in [m]} p_u(a_u) = 1, \forall u \in [n] \tag{39}$$

According to Lemma 1, 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 $\mathbf{w} \in \mathbb{R}^{mn}$

$$\mathbf{x}_i^* = \sum_{\vec{a}} \frac{\mathbf{p}(\vec{a})}{p_u(a_u)} \mathbf{b}_{\vec{a}} \cdot \mathbf{1}(a_u = v) - \frac{n-1}{n} \mathbf{p}(\vec{a}) \mathbf{b}_{\vec{a}} - \frac{1}{mn} \sum_{i' \in [mn]} \mathbf{w}_{i'} + \frac{1}{m} \sum_{v' \in [m]} \mathbf{w}_{um+v'} \tag{40}$$

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

$$\mathop{\mathbb{E}}_{a'_{-i} \sim p_D(\cdot|s)} \left[y^{(t)} \left(s, a_i \oplus a'_{-i} \right) \right] - \frac{n-1}{n} \mathop{\mathbb{E}}_{\mathbf{a}' \sim p_D(\cdot|s)} \left[y^{(t)} \left(s, \mathbf{a}' \right) \right] + w_i(s), \tag{41}$$

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 $\forall s$, $\sum_{i=1}^n w_i(s) = 0$. □

B Omitted Proofs in Section 5.1

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.*

$$\forall Q_{tot}, Q'_{tot} \in \mathcal{Q}, \quad \|\mathcal{T} Q_{tot} - \mathcal{T} Q'_{tot}\|_\infty \leq \gamma \|Q_{tot} - Q'_{tot}\|_\infty \tag{10}$$

Proof. Assume the empirical Bellman operator $\mathcal{T}_D^{\mathrm{LVD}}$ is a γ-contraction. For any MMDPs, when using a uniform data distribution, the value function of FQI-LVD will converge (Ernst et al., 2005)

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 $\mathbf{a} = \langle a_1, a_2 \rangle$.

$$r(s_1) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \quad r(s_2) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \tag{42}$$

Besides, the transition is deterministic.

$$T(s_1) = \begin{pmatrix} s_1 & s_1 \\ s_1 & s_1 \end{pmatrix} \quad T(s_2) = \begin{pmatrix} s_2 & s_2 \\ s_2 & s_1 \end{pmatrix} \tag{43}$$

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, 0th 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 1. $\mathbf{b}$ is target one-step TD-value w.r.t the Q-value of the last iteration. Through described in Lemma 1, 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 31

$$-\frac{1}{m}\frac{\mathbf{p}(\vec{a})}{p_u(a_u)} + \frac{1}{mn}\sum_{u'=0}^{n-1}\frac{\mathbf{p}(\vec{a})}{p_{u'}(a_{u'})} = 0 \tag{44}$$

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

$$Q_1^t(s_j, a_1) = \frac{1}{2}\sum_{a_2 \in \mathcal{A}}\left(r(s_j, \mathbf{a}) + \gamma V_{\text{tot}}^{t-1}(T(s_j, \mathbf{a})\right) - \frac{1}{2}\sum_{\mathbf{a} \in \boldsymbol{\mathcal{A}}}\frac{1}{4}\left(r(s_j, \mathbf{a}) + \gamma V_{\text{tot}}^{t-1}(T(s_j, \mathbf{a}))\right) \tag{45}$$

$$= Q_2^t(s_j, a_2) \tag{46}$$

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 45, $\forall a \in \mathcal{A}, t \geq 1$

$$Q^t(s_1, a) = \frac{\gamma}{2} V_{\text{tot}}^{t-1}(s_1) \tag{47}$$

$$= \frac{\gamma}{2}\max_{a_1, a_2 \in \mathcal{A}}\left(Q^{t-1}(s_1, a_1) + Q^{t-1}(s_1, a_2)\right) \tag{48}$$

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 45

$$\begin{aligned} Q^t\left(s_2, \mathcal{A}^{(1)}\right) &= \frac{1}{2}(1 + 2\gamma V_{\text{tot}}^{t-1}(s_2)) - \frac{1}{8}[1 + \gamma(3V_{\text{tot}}^{t-1}(s_2) + V_{\text{tot}}^{t-1}(s_1))] \\ &= \frac{5\gamma}{8} V_{\text{tot}}^{t-1}(s_2) + \frac{3}{8} - \frac{1}{4}\gamma^t q_1 \\ &= \frac{5\gamma}{4} \max_{a \in \mathcal{A}} Q^{t-1}(s_2, a) + \frac{3}{8} - \frac{1}{4}\gamma^t q_1 \end{aligned} \tag{49}$$

$$\begin{aligned} Q^t\left(s_2, \mathcal{A}^{(2)}\right) &= \frac{1}{2}(\gamma V_{\text{tot}}^{t-1}(s_2) + \gamma V_{\text{tot}}^{t-1}(s_1)) - \frac{1}{8}[1 + \gamma(3V_{\text{tot}}^{t-1}(s_2) + V_{\text{tot}}^{t-1}(s_1))] \\ &= \frac{\gamma}{8} V_{\text{tot}}^{t-1}(s_2) - \frac{1}{8} + \frac{3}{4}\gamma^t q_1 \\ &= \frac{\gamma}{4} \max_{a \in \mathcal{A}} Q^{t-1}(s_2, a) - \frac{1}{8} + \frac{3}{4}\gamma^t q_1 \end{aligned} \tag{50}$$

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

$$Q^t\left(s_2, \mathcal{A}^{(2)}\right) \geq \frac{\gamma}{4} \max_{a \in \mathcal{A}} Q^{t-1}(s_2, a) - \frac{1+\delta}{8} \geq \frac{\gamma}{4} Q^{t-1}\left(s_2, \mathcal{A}^{(2)}\right) - \frac{1+\delta}{8} \tag{51}$$

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 51. Since $t \geq t_\delta$

$$\widehat{Q}^t\left(s_2, \mathcal{A}^{(2)}\right) = \left(\frac{\gamma}{4}\right)^{t-t_\delta}\left(Q^{t_\delta}\left(s_2, \mathcal{A}^{(2)}\right) - \frac{1+\delta}{2\gamma - 8}\right) + \frac{1+\delta}{2\gamma - 8} \tag{52}$$

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

$$Q^{T_\delta}\left(s_2, \mathcal{A}^{(2)}\right) \geq \widehat{Q}^{T_\delta}\left(s_2, \mathcal{A}^{(2)}\right) \geq \frac{1+2\delta}{2\gamma - 8} > -\frac{1+2\delta}{6} \tag{53}$$

According to equation 49 and let $\delta < \frac{1}{11}$.

$$Q^{T_\delta+1}\left(s_2, \mathcal{A}^{(1)}\right) \geq \frac{5\gamma}{4} Q^{T_\delta}\left(s_2, \mathcal{A}^{(2)}\right) + \frac{3}{8} - \frac{1}{4}\gamma^t q_1 \tag{54}$$

$$> -\frac{5+10\delta}{24} + \frac{3}{8} - \frac{1}{24}\delta \tag{55}$$

$$> \frac{1}{8} \tag{56}$$

Similar to equation 51, we observer from equation 49 that $\forall t > T_{\delta=\frac{1}{11}} + 1$

$$Q^t\left(s_2, \mathcal{A}^{(1)}\right) \geq \frac{5\gamma}{4} Q^{t-1}\left(s_2, \mathcal{A}^{(1)}\right) + \frac{1}{4} \tag{57}$$

and

$$V_{\text{tot}}^t(s_2) = 2Q^t\left(s_2, \mathcal{A}^{(1)}\right) \tag{58}$$

$$\geq 2\left(\frac{5\gamma}{4} Q^{t-1}\left(s_2, \mathcal{A}^{(1)}\right) + \frac{1}{4}\right) \tag{59}$$

$$= \frac{5\gamma}{4} V_{\text{tot}}^{t-1}(s_2) + \frac{1}{4} \tag{60}$$

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 $V_{\text{tot}}^t(s_2)$ 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. □

C OMITTED ALGORITHM BOX, THEOREM, AND DEFINITION IN SECTION 5.2

C.1 LOCAL CONVERGENCE IMPROVEMENT

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

$$\tilde{\boldsymbol{\pi}}_t(a|s) = \prod_{i=1}^{n} \left(\frac{\epsilon}{|\mathcal{A}|} + (1-\epsilon)\mathbb{I}\left[a_i = \arg\max_{a_i' \in \mathcal{A}} Q_i^{(t)}(s, a_i') \right] \right) \tag{61}$$

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.

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 (Simchowitz & Jamieson, 2019), 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.

Lemma 2. *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.*

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

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

Lemma 2 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 (Brouwer, 1911), we can further verify the existence of a fixed-point solution for the on-policy Bellman operator $\mathcal{T}_\epsilon^{\text{LVD}}$ (see Theorem 4).

Theorem 4 (Formal version of Theorem 2)**.** *Besides Lemma 2, Algorithm 1 will have a fixed point value function expressing the optimal policy if the hyper-parameter ϵ is sufficiently small.*

Theorem 4 indicates 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 one-step TD target $\mathcal{T}_D^{\text{LVD}} Q$. 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 2.

C.2 GLOBAL CONVERGENCE IMPROVEMENT

Definition 2 (FQI-IGM)**.** *Given a dataset D, FQI-IGM specifies the action-value function class*

$$\mathcal{Q}^{IGM} = \left\{ Q \;\middle|\; Q_{tot} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|^n} \text{ and } \left[\forall Q_i \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} \right]_{i=1}^{n} \text{ with that Eq. (1) is satisfied} \right\}. \tag{62}$$

and induces the empirical Bellman operator

$$Q^{(t+1)} \leftarrow \mathcal{T}_D^{IGM} Q^{(t)} \equiv \arg\min_{Q \in \mathcal{Q}^{IGM}} \sum_{(s,\mathbf{a}) \in \mathcal{S} \times \mathbf{A}} p_D(\mathbf{a}|s) \left(y^{(t)}(s,\mathbf{a}) - Q_{tot}(s,\mathbf{a}) \right)^2, \tag{63}$$

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 3.3.

Compared with FQI-LVD stated in Definition 1, the differences are the Q function class, i.e, $\mathcal{Q}^{\text{IGM}}$ vs. $\mathcal{Q}^{\text{LVD}}$.

D Omitted Proofs of Theorem 3

Lemma 3. *The empirical Bellman operator $\mathcal{T}_D^{IGM}$ stated in Definition 2 is a γ-contraction, i.e., the following important property of the standard Bellman optimality operator $\mathcal{T}$ will hold for $\mathcal{T}_D^{IGM}$.*

$$\forall Q_{tot}, Q'_{tot} \in \mathcal{Q}, \quad \|\mathcal{T} Q_{tot} - \mathcal{T} Q'_{tot}\|_\infty \leq \gamma \|Q_{tot} - Q'_{tot}\|_\infty \tag{64}$$

Proof. We want to prove

$$\left(\mathcal{T}_D^{\text{IGM}} Q\right)_{\text{tot}} = r(s,\mathbf{a}) + \gamma \left\langle P(s,\mathbf{a}), V_Q \right\rangle, \tag{65}$$

where P is transition function, $V_Q(\cdot) = \max_{\mathbf{a} \in \mathbf{A}} Q_{\text{tot}}(\cdot,\mathbf{a})$, and $\langle \cdot, \cdot \rangle$ is inner product. According to Eq. (65) and Lemma 1.5 in RL textbook (Agarwal et al., 2019), we can prove that $\mathcal{T}_D^{\text{IGM}}$ is a γ-contraction. Eq. (65) indicates that the empirical Bellman error

$$err_D^{\text{IGM}} \equiv \min_{Q \in \mathcal{Q}^{\text{IGM}}} \sum_{(s,\mathbf{a}) \in \mathcal{S} \times \mathbf{A}} p_D(\mathbf{a}|s) \left(y^{(t)}(s,\mathbf{a}) - Q_{\text{tot}}(s,\mathbf{a}) \right)^2 = 0. \tag{66}$$

Let $\mathbf{a}^{*,(t)} = \left[a_i^{*,(t)}\right]_{i=1}^n = \arg\max_{\mathbf{a} \in \mathbf{A}} y^{(t)}(s,\mathbf{a})$. Then, $\forall y^{(t)}(s,\cdot)$, we construct $Q_{\text{tot}}(s,\mathbf{a}) = y^{(t)}(s,\mathbf{a})$ and its corresponding local action-value functions $[Q_i]_{i=1}^n$ satisfying IGM principle:

$$Q_i(s,a_i) = \begin{cases} 1, & \text{when } a_i = a_i^{*,(t)}, \\ 0, & \text{when } a_i \neq a_i^{*,(t)}. \end{cases} \tag{67}$$

To avoid the multiple solutions of $\arg\max$ operator in $\mathbf{a}^{*,(t)}$, we consider the lexicographic order of joint actions as the second priority. Thus, we illustrate the completeness of IGM function class in MMDP setting from our construction. Then, Eq. (65) is held and $\mathcal{T}_D^{\text{IGM}}$ is a γ-contraction in MMDP framework. □

Theorem 3. *FQI-IGM will globally converge to optimal value function.*

Proof. Let $Q^*(s,\mathbf{a}) = max_{\boldsymbol{\pi} \in \Pi} Q^{\boldsymbol{\pi}}(s,\mathbf{a})$ where Π is the space of all policies. According to Lemma 3 and Theorem 1.4 in RL textbook (Agarwal et al., 2019), we have that

- There exists a stationary and deterministic policy $\boldsymbol{\pi}$ such that $Q_{\text{tot}}^{\boldsymbol{\pi}} = Q_{\text{tot}}^*$.
- A vector $Q_{\text{tot}} \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|^n}$ is equal to Q_{tot}^* if and only if it satisfies $Q_{\text{tot}} = \left(\mathcal{T}_D^{\text{IGM}} Q\right)_{\text{tot}}$.
- $\forall Q'_{\text{tot}} \in \mathcal{Q}^{\text{IGM}}$,

$$\left\| Q_{\text{tot}}^* - \left(\mathcal{T}_D^{\text{IGM}} Q'\right)_{\text{tot}} \right\|_\infty = \left\| \left(\mathcal{T}_D^{\text{IGM}} Q^*\right)_{\text{tot}} - \left(\mathcal{T}_D^{\text{IGM}} Q'\right)_{\text{tot}} \right\|_\infty \tag{68}$$

$$\leq \gamma \left\| Q_{\text{tot}}^* - Q'_{\text{tot}} \right\|_\infty. \tag{69}$$

Thus, FQI-IGM will globally converge to optimal value function. □

E OMITTED PROOFS OF APPENDIX C.1

E.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

$$\tilde{\boldsymbol{\pi}}(\mathbf{a}|s) = \prod_{i=1}^{n} \left(\varepsilon + (1-\hat{\varepsilon})\mathbb{I}\left[a_i = \arg\max_{a_i' \in \mathcal{A}} Q_i^*(s, a_i') \right] \right) \tag{70}$$

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

$$(\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \mathbf{a}) = \sum_{\mathbf{a}' \in \mathcal{A}^n} f(s, \mathbf{a}, \mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s, \mathbf{a}') \tag{71}$$

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, according to Eq. (8),

$$f(s, \mathbf{a}, \mathbf{a}') = \left(\frac{h^{(1)}(s, \mathbf{a}, \mathbf{a}')}{1-\hat{\varepsilon}} + \frac{h^{(0)}(s, \mathbf{a}, \mathbf{a}')}{\varepsilon} - (n-1) \right) (1-\hat{\varepsilon})^{h^{\boldsymbol{\pi}^*}(s, \mathbf{a}')} \varepsilon^{n - h^{\boldsymbol{\pi}^*}(s, \mathbf{a}')}, \tag{72}$$

in which

$$h^{\boldsymbol{\pi}^*}(s, \mathbf{a}) = \sum_{i=1}^{n} \mathbb{I}[a_i = \pi_i^*(s)] \tag{73}$$

$$h^{(1)}(s, \mathbf{a}, \mathbf{a}') = \sum_{i=1}^{n} \mathbb{I}[a_i = \pi_i^*(s)]\mathbb{I}[a_i = a_i'] \tag{74}$$

$$h^{(0)}(s, \mathbf{a}, \mathbf{a}') = \sum_{i=1}^{n} \mathbb{I}[a_i \neq \pi_i^*(s)]\mathbb{I}[a_i = a_i'] \tag{75}$$

As a reference indicating whether the learned value function produces the optimal policy, we denote

$$\mathcal{E}(Q) = \max_{s \in \mathcal{S}} \left[\max_{\mathbf{a} \in (\mathcal{A}^n \setminus \{\boldsymbol{\pi}^*(s)\})} (Q_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - Q_{\text{tot}}(s, \mathbf{a})) \right] \tag{76}$$

Notice that $\boldsymbol{\pi}^*$ denotes the optimal policy of the given MDP, so $\mathcal{E}(Q)$ might be negative for a non-optimal or inaccurate value function Q.

E.2 OMITTED PROOFS

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

$$\forall \delta > 0,\ \forall 0 < \varepsilon \leq \frac{\delta}{n^2 |\mathcal{A}|^n 2^{n+1} (R_{max} + \gamma \|V_{tot}\|_\infty)}, \tag{77}$$

we have

$$\forall s \in \mathcal{S},\ \left| (\mathcal{T}_D^{LVD} Q)_{tot}(s, \boldsymbol{\pi}^*(s)) - (\mathcal{T}Q)_{tot}(s, \boldsymbol{\pi}^*(s)) \right| \leq \delta, \tag{78}$$

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

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

$$\begin{aligned}
&\left|(\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s))\right| \\
\leq\ & |(f(s, \boldsymbol{\pi}^*(s), \boldsymbol{\pi}^*(s)) - 1)(\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s))| + \left| \sum_{a' \in \mathcal{A}^n \setminus \{\boldsymbol{\pi}^*(s)\}} f(s, \boldsymbol{\pi}^*(s), \mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s, \mathbf{a}') \right| \\
\leq\ & \left(|f(s, \boldsymbol{\pi}^*(s), \boldsymbol{\pi}^*(s)) - 1| + \sum_{a' \in \mathcal{A}^n \setminus \{\boldsymbol{\pi}^*(s)\}} |f(s, \boldsymbol{\pi}^*(s), \mathbf{a}')| \right) \|(\mathcal{T}Q)_{\text{tot}}\|_\infty. \qquad (79)
\end{aligned}$$

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

$$\begin{aligned}
|f(s, \boldsymbol{\pi}^*(s), \boldsymbol{\pi}^*(s)) - 1| &= \left| \left(\frac{n}{1-\hat{\varepsilon}} - (n-1) \right) (1-\hat{\varepsilon})^n - 1 \right| \\
&= \left| (n - (n-1)(1-\hat{\varepsilon}))(1-\hat{\varepsilon})^{n-1} - 1 \right| \\
&= \left| (1 + (n-1)\hat{\varepsilon})(1-\hat{\varepsilon})^{n-1} - 1 \right| \\
&= \left| (1 + (n-1)\hat{\varepsilon}) \left(\sum_{\ell=0}^{n-1} \binom{n-1}{\ell} (-1)^\ell \hat{\varepsilon}^\ell \right) - 1 \right| \\
&= \left| (1 + (n-1)\hat{\varepsilon}) \left(1 - (n-1)\hat{\varepsilon} + \sum_{\ell=2}^{n-1} \binom{n-1}{\ell} (-1)^\ell \hat{\varepsilon}^\ell \right) - 1 \right| \\
&= \left| 1 - (n-1)^2 \hat{\varepsilon}^2 + (1 + (n-1)\hat{\varepsilon}) \left(\sum_{\ell=2}^{n-1} \binom{n-1}{\ell} (-1)^\ell \hat{\varepsilon}^\ell \right) - 1 \right| \\
&= \left| \hat{\varepsilon}^2 \left((n-1)^2 - (1 + (n-1)\hat{\varepsilon}) \sum_{\ell=2}^{n-1} \binom{n-1}{\ell} (-1)^\ell \hat{\varepsilon}^{\ell-2} \right) \right| \\
&\leq |\mathcal{A}|^2 \varepsilon^2 \left(n^2 + 2 \sum_{\ell=2}^{n-1} \binom{n-1}{\ell} \right) \\
&\leq |\mathcal{A}|^2 \varepsilon^2 \left(n^2 + 2^n \right) \\
&\leq \varepsilon^2 n^2 |\mathcal{A}|^2 2^n. \qquad (80)
\end{aligned}$$

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

$$\begin{aligned}
&\sum_{a' \in \mathcal{A}^n \setminus \{\boldsymbol{\pi}^*(s)\}} |f(s, \boldsymbol{\pi}^*(s), \mathbf{a}')| \\
\leq\ & \sum_{a' \in \mathcal{A}^n \setminus \{\boldsymbol{\pi}^*(s)\}} \left| \left(\frac{h^{\boldsymbol{\pi}^*}(s, \mathbf{a}')}{1-\hat{\varepsilon}} - (n-1) \right) (1-\hat{\varepsilon})^{h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')} \varepsilon^{n - h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')} \right| \\
=\ & \sum_{a' \in \mathcal{A}^n \setminus \{\boldsymbol{\pi}^*(s)\}} \left| \left(h^{\boldsymbol{\pi}^*}(s, \mathbf{a}') - (n-1)(1-\hat{\varepsilon}) \right) (1-\hat{\varepsilon})^{h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')-1} \varepsilon^{n - h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')} \right| \\
\leq\ & \sum_{a' \in \mathcal{A}^n \setminus \{\boldsymbol{\pi}^*(s)\}} \left| 2n (1-\hat{\varepsilon})^{h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')-1} \varepsilon^{n - h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')} \right| \\
\leq\ & \sum_{a' \in \mathcal{A}^n \setminus \{\boldsymbol{\pi}^*(s)\}} 2n\varepsilon \\
\leq\ & 2n\varepsilon |\mathcal{A}|^n. \qquad (81)
\end{aligned}$$

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

$$\begin{aligned}
&\left|(\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s))\right| \\
\leq &\left(|f(s, \boldsymbol{\pi}^*(s), \boldsymbol{\pi}^*(s)) - 1| + \sum_{a' \in \mathcal{A}^n \setminus \{\boldsymbol{\pi}^*(s)\}} |f(s, \boldsymbol{\pi}^*(s), \mathbf{a}')| \right) \|(\mathcal{T}Q)_{\text{tot}}\|_\infty \\
\leq &(\varepsilon^2 n^2 |\mathcal{A}|^2 2^n + 2n\varepsilon |\mathcal{A}|^n) \|(\mathcal{T}Q)_{\text{tot}}\|_\infty \\
\leq &\varepsilon n^2 |\mathcal{A}|^n 2^{n+1} \|(\mathcal{T}Q)_{\text{tot}}\|_\infty \\
\leq &\varepsilon n^2 |\mathcal{A}|^n 2^{n+1} (R_{\max} + \gamma \|V_{\text{tot}}\|_\infty) \\
\leq &\delta. \qquad (82)
\end{aligned}$$

□

Lemma 5. *Given a dataset D generated by the optimal policy $\boldsymbol{\pi}^*$ with ϵ-greedy exploration, for any target value function Q,*

$$\forall 0 < \varepsilon \leq \frac{(1-\gamma)\mathcal{E}(Q^*)}{\gamma n^3 |\mathcal{A}|^n 2^{n+4} (R_{max}/(1-\gamma) + \gamma \|V_{tot}^{\boldsymbol{\pi}^*} - V^*\|_\infty)}, \qquad (83)$$

we have

$$\forall s \in \mathcal{S}, \ \left|(\mathcal{T}_D^{LVD} Q)_{tot}(s, \boldsymbol{\pi}^*(s)) - V^*(s)\right| \leq \gamma \|V_{tot}^{\boldsymbol{\pi}^*} - V^*\|_\infty + \frac{1-\gamma}{8n\gamma} \mathcal{E}(Q^*), \qquad (84)$$

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

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

$$\begin{aligned}
&\left|(\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - V^*(s)\right| \\
\leq &\left|(\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s))\right| + |(\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - V^*(s)| \\
= &\left|(\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s))\right| + |(\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - Q^*(s, \boldsymbol{\pi}^*(s))| \\
= &\left|(\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s))\right| + |(\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - (\mathcal{T}Q^*)(s, \boldsymbol{\pi}^*(s))| \\
\leq &\left|(\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s))\right| + \gamma |V_{\text{tot}}(s') - V^*(s')| \\
\leq &\left|(\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s))\right| + \gamma |Q_{\text{tot}}(s', \boldsymbol{\pi}^*(s')) - V^*(s')| \\
\leq &\left|(\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s))\right| + \gamma \|V_{\text{tot}}^{\boldsymbol{\pi}^*} - V^*\|_\infty \qquad (85)
\end{aligned}$$

Let $\delta = \frac{1-\gamma}{8n\gamma}\mathcal{E}(Q^*)$. According to Lemma 4, with the condition

$$0 < \varepsilon \leq \frac{\delta}{n^2 |\mathcal{A}|^n 2^{n+1} (R_{\max} + \gamma \|V_{\text{tot}}\|_\infty)} = \frac{(1-\gamma)\mathcal{E}(Q^*)/(8n\gamma)}{n^2 |\mathcal{A}|^n 2^{n+1} (R_{\max} + \gamma \|V_{\text{tot}}\|_\infty)}, \qquad (86)$$

we have

$$\left|(\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s))\right| \leq \delta = \frac{1-\gamma}{8n\gamma}\mathcal{E}(Q^*). \qquad (87)$$

Notice that

$$\begin{aligned}
\|V_{\text{tot}}\|_\infty &\leq \|V^*\|_\infty + \|V_{\text{tot}} - V^*\|_\infty \qquad (88) \\
&\leq \frac{R_{\max}}{1-\gamma} + \|V_{\text{tot}}^{\boldsymbol{\pi}^*} - V^*\|_\infty. \qquad (89)
\end{aligned}$$

The overall statement is

$$\forall 0 < \varepsilon \leq \frac{(1-\gamma)\mathcal{E}(Q^*)}{\gamma n^3 |\mathcal{A}|^n 2^{n+4} (R_{\max}/(1-\gamma) + \gamma \|V_{\text{tot}}^{\boldsymbol{\pi}^*} - V^*\|_\infty)} \leq \frac{(1-\gamma)\mathcal{E}(Q^*)/(8n\gamma)}{n^2 |\mathcal{A}|^n 2^{n+1} (R_{\max} + \gamma \|V_{\text{tot}}\|_\infty)} \qquad (90)$$

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

$$\begin{aligned}
&\left|(\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - V^*(s)\right| \\
\leq &\left|(\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s))\right| + \gamma \|V_{\text{tot}}^{\boldsymbol{\pi}^*} - V^*\|_\infty \\
\leq &\gamma \|V_{\text{tot}}^{\boldsymbol{\pi}^*} - V^*\|_\infty + \frac{1-\gamma}{8n\gamma}\mathcal{E}(Q^*).
\end{aligned} \tag{91}$$

□

Lemma 6. *For any value function Q, the corresponding sub-optimality gap satisfies*

$$\mathcal{E}(\mathcal{T}Q) \geq \mathcal{E}(Q^*) - 2\gamma \|V_{tot} - V^*\|_\infty \tag{92}$$

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

$$\begin{aligned}
\mathcal{E}(\mathcal{T}Q) = &\max_{(s,\mathbf{a}) \in \mathcal{S} \times (\mathcal{A}^n \setminus \{\boldsymbol{\pi}^*(s)\})} ((\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - (\mathcal{T}Q)_{\text{tot}}(s, \mathbf{a})) \\
\geq &\max_{(s,\mathbf{a}) \in \mathcal{S} \times (\mathcal{A}^n \setminus \{\boldsymbol{\pi}^*(s)\})} \left((\mathcal{T}Q^*)(s, \boldsymbol{\pi}^*(s)) - (\mathcal{T}Q^*)(s, \mathbf{a}) - \gamma \left(|V_{\text{tot}}(s_1) - V^*(s_1)| + |V_{\text{tot}}(s_2) - V^*(s_2)|\right)\right) \\
\geq &\max_{(s,\mathbf{a}) \in \mathcal{S} \times (\mathcal{A}^n \setminus \{\boldsymbol{\pi}^*(s)\})} \left((\mathcal{T}Q^*)(s, \boldsymbol{\pi}^*(s)) - (\mathcal{T}Q^*)(s, \mathbf{a}) - 2\gamma \|V_{\text{tot}} - V^*\|_\infty\right) \\
= &\max_{(s,\mathbf{a}) \in \mathcal{S} \times (\mathcal{A}^n \setminus \{\boldsymbol{\pi}^*(s)\})} \left(Q^*(s, \boldsymbol{\pi}^*(s)) - Q^*(s, \mathbf{a}) - 2\gamma \|V_{\text{tot}} - V^*\|_\infty\right) \\
= &\mathcal{E}(Q^*) - 2\gamma \|V_{\text{tot}} - V^*\|_\infty
\end{aligned} \tag{93}$$

□

Lemma 7. *Given a dataset D generated by the optimal policy $\boldsymbol{\pi}^*$ with ϵ-greedy exploration, for any target value function Q,*

$$\forall \delta > 0,\ \forall 0 < \varepsilon \leq \frac{\delta}{n^2 |\mathcal{A}|^n 2^n (R_{max}/(1-\gamma) + \gamma \|V_{tot} - V^*\|_\infty)}, \tag{94}$$

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

$$(\mathcal{T}_D^{LVD} Q)_{tot}(s, \mathbf{a}) \leq (\mathcal{T}Q)_{tot}(s, \boldsymbol{\pi}^*(s)) - \mathcal{E}(Q^*) + 2n\gamma \|V_{tot} - V^*\|_\infty + \delta \tag{95}$$

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

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

$$\begin{aligned}
(\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \mathbf{a}) = &\sum_{a' \in \mathcal{A}^n} f(s, \mathbf{a}, \mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s, \mathbf{a}') \\
= &f(s, \mathbf{a}, \boldsymbol{\pi}^*(s))(\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) \\
&+ \sum_{a' \in \mathcal{A}^n : h^{\boldsymbol{\pi}^*}(s, \mathbf{a}') = n-1} f(s, \mathbf{a}, \mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s, \mathbf{a}') \\
&+ \sum_{a' \in \mathcal{A}^n : h^{\boldsymbol{\pi}^*}(s, \mathbf{a}') < n-1} f(s, \mathbf{a}, \mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s, \mathbf{a}')
\end{aligned} \tag{96}$$

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{97}
$$

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{98}
$$

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{99}
$$

Combining the above terms, we can get

$$
\begin{aligned}
&(\mathcal{T}_D^{\text{LVD}}Q)_{\text{tot}}(s,\mathbf{a}) \\
&= f(s,\mathbf{a},\boldsymbol{\pi}^*(s))(\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s)) + \sum_{a'\in\mathcal{A}^n:h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')=n-1} f(s,\mathbf{a},\mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}') \\
&\quad + \sum_{a'\in\mathcal{A}^n:h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')<n-1} f(s,\mathbf{a},\mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}') \\
&\leq \left(h^{\boldsymbol{\pi}^*}(s,\mathbf{a})-(n-1)\right)(\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s)) + \varepsilon n2^n|\mathcal{A}|\|(\mathcal{T}Q)_{\text{tot}}\|_\infty + \varepsilon n|\mathcal{A}|\|(\mathcal{T}Q)_{\text{tot}}\|_\infty \\
&\quad + \left(\sum_{a'\in\mathcal{A}^n:h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')=n-1} h^{(0)}(s,\mathbf{a},\mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}')\right) + \varepsilon n^2|\mathcal{A}|2^n\|(\mathcal{T}Q)_{\text{tot}}\|_\infty + 3n\varepsilon|\mathcal{A}|^n\|(\mathcal{T}Q)_{\text{tot}}\|_\infty \\
&\leq \left(h^{\boldsymbol{\pi}^*}(s,\mathbf{a})-(n-1)\right)(\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*(s)) + \left(\sum_{a'\in\mathcal{A}^n:h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')=n-1} h^{(0)}(s,\mathbf{a},\mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}')\right) \\
&\quad + \varepsilon n^2|\mathcal{A}|^n2^n\|(\mathcal{T}Q)_{\text{tot}}\|_\infty
\end{aligned}
\tag{100}
$$

in which

$$
\begin{aligned}
&\sum_{a'\in\mathcal{A}^n:h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')=n-1} h^{(0)}(s,\mathbf{a},\mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}') \\
&\leq \left(\sum_{a'\in\mathcal{A}^n:h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')=n-1} h^{(0)}(s,\mathbf{a},\mathbf{a}')\right)\max_{a'\in\mathcal{A}^n:h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')=n-1}(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}') \\
&= \left(n-h^{\boldsymbol{\pi}^*}(s,\mathbf{a})\right)\max_{a'\in\mathcal{A}^n:h^{\boldsymbol{\pi}^*}(s,\mathbf{a}')=n-1}(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}') \\
&\leq \left(n-h^{\boldsymbol{\pi}^*}(s,\mathbf{a})\right)\max_{a'\in\mathcal{A}^n\setminus\{\boldsymbol{\pi}^*(s)\}}(\mathcal{T}Q)_{\text{tot}}(s,\mathbf{a}') \\
&= \left(n-h^{\boldsymbol{\pi}^*}(s,\mathbf{a})\right)\left((\mathcal{T}Q)_{\text{tot}}(s,\boldsymbol{\pi}^*)-\mathcal{E}(\mathcal{T}Q)\right)
\end{aligned}
\tag{101}
$$

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

$$
\begin{aligned}
&(\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \mathbf{a}) \\
\leq\ & \left(h^{\boldsymbol{\pi}^*}(s, \mathbf{a}) - (n-1)\right)(\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) + \left(\sum_{a' \in \mathcal{A}^n : h^{\boldsymbol{\pi}^*}(s, \mathbf{a}') = n-1} h^{(0)}(s, \mathbf{a}, \mathbf{a}')(\mathcal{T}Q)_{\text{tot}}(s, \mathbf{a}')\right) \\
&+ \varepsilon n^2 |\mathcal{A}|^n 2^n \|(\mathcal{T}Q)_{\text{tot}}\|_\infty \\
\leq\ & \left(h^{\boldsymbol{\pi}^*}(s, \mathbf{a}) - (n-1)\right)(\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) + (n - h^{\boldsymbol{\pi}^*}(s, \mathbf{a}))\left((\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*) - \mathcal{E}(\mathcal{T}Q)\right) + \varepsilon n^2 |\mathcal{A}|^n 2^n \|(\mathcal{T}Q)_{\text{tot}}\|_\infty \\
=\ & (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - (n - h^{\boldsymbol{\pi}^*}(s, \mathbf{a}))\mathcal{E}(\mathcal{T}Q) + \varepsilon n^2 |\mathcal{A}|^n 2^n \|(\mathcal{T}Q)_{\text{tot}}\|_\infty
\end{aligned} \tag{102}
$$

According to Lemma 6, $\mathcal{E}(\mathcal{T}Q) \geq \mathcal{E}(Q^*) - 2\gamma\|V_{\text{tot}} - V^*\|_\infty$. So $\forall s \in \mathcal{S}, \forall a \in \mathcal{A}^n \setminus \{\boldsymbol{\pi}^*(s)\}$,

$$
\begin{aligned}
(\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \mathbf{a}) &\leq (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - (n - h^{\boldsymbol{\pi}^*}(s, \mathbf{a}))\mathcal{E}(\mathcal{T}Q) + \varepsilon n^2 |\mathcal{A}|^n 2^n \|(\mathcal{T}Q)_{\text{tot}}\|_\infty \\
&\leq (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - (n - h^{\boldsymbol{\pi}^*}(s, \mathbf{a}))\left(\mathcal{E}(Q^*) - 2\gamma\|V_{\text{tot}} - V^*\|_\infty\right) + \varepsilon n^2 |\mathcal{A}|^n 2^n \|(\mathcal{T}Q)_{\text{tot}}\|_\infty \\
&\leq (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - \mathcal{E}(Q^*) + 2n\gamma\|V_{\text{tot}} - V^*\|_\infty + \varepsilon n^2 |\mathcal{A}|^n 2^n \|(\mathcal{T}Q)_{\text{tot}}\|_\infty \\
&\leq (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - \mathcal{E}(Q^*) + 2n\gamma\|V_{\text{tot}} - V^*\|_\infty + \varepsilon n^2 |\mathcal{A}|^n 2^n (R_{\max} + \gamma\|V_{\text{tot}}\|_\infty) \\
&\leq (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - \mathcal{E}(Q^*) + 2n\gamma\|V_{\text{tot}} - V^*\|_\infty + \varepsilon n^2 |\mathcal{A}|^n 2^n (R_{\max} + \gamma\|V^*\|_\infty + \gamma\|V_{\text{tot}} - V^*\|_\infty) \\
&\leq (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - \mathcal{E}(Q^*) + 2n\gamma\|V_{\text{tot}} - V^*\|_\infty + \varepsilon n^2 |\mathcal{A}|^n 2^n (R_{\max}/(1-\gamma) + \gamma\|V_{\text{tot}} - V^*\|_\infty) \\
&\leq (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - \mathcal{E}(Q^*) + 2n\gamma\|V_{\text{tot}} - V^*\|_\infty + \delta
\end{aligned} \tag{103}
$$

□

Lemma 8. *Let $\mathcal{B}$ denote a subspace of value functions*

$$
\mathcal{B} = \left\{ Q \in \mathcal{Q}^{LVD} \,\middle|\, \mathcal{E}(Q) \geq 0,\ \|V_{tot} - V^*\|_\infty \leq \frac{1}{8n\gamma}\mathcal{E}(Q^*) \right\} \tag{104}
$$

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

$$
\forall 0 < \varepsilon \leq \frac{(1-\gamma)\mathcal{E}(Q^*)}{n^3 |\mathcal{A}|^n 2^{n+4}(R_{max}/(1-\gamma) + \mathcal{E}(Q^*)/(8n))} \tag{105}
$$

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

$$
\hat{\mathcal{B}} = \left\{ Q \in \mathcal{Q}^{LVD} \,\middle|\, \mathcal{E}(Q) > 0,\ \|V_{tot} - V^*\|_\infty \leq \frac{1}{8n\gamma}\mathcal{E}(Q^*) \right\} \tag{106}
$$

Proof. According to Lemma 4, with the condition

$$
0 < \varepsilon \leq \frac{\mathcal{E}(Q^*)/4}{n^2 |\mathcal{A}|^n 2^{n+1}(R_{\max}/(1-\gamma) + \mathcal{E}(Q^*)/(8n))} \leq \frac{\mathcal{E}(Q^*)/4}{n^2 |\mathcal{A}|^n 2^{n+1}(R_{\max} + \gamma\|V_{\text{tot}}\|_\infty)} \tag{107}
$$

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

$$
\left|(\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s))\right| \leq \frac{1}{4}\mathcal{E}(Q^*) \tag{108}
$$

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

$$
(\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) \geq (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - \frac{1}{4}\mathcal{E}(Q^*). \tag{109}
$$

According to Lemma 7, with the condition

$$
\begin{aligned}
0 < \varepsilon &\leq \frac{\mathcal{E}(Q^*)/4}{n^2 |\mathcal{A}|^n 2^n (R_{\max}/(1-\gamma) + \mathcal{E}(Q^*)/(8n))} \\
&\leq \frac{\mathcal{E}(Q^*)/4}{n^2 |\mathcal{A}|^n 2^n (R_{\max}/(1-\gamma) + \gamma\|V_{\text{tot}} - V^*\|_\infty)}
\end{aligned} \tag{110}
$$

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

$$\begin{aligned}(\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \mathbf{a}) &\leq (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - \mathcal{E}(Q^*) + 2n\gamma \|V_{\text{tot}} - V^*\|_\infty + \frac{1}{4}\mathcal{E}(Q^*) \\ &\leq (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - \mathcal{E}(Q^*) + \frac{1}{4}\mathcal{E}(Q^*) + \frac{1}{4}\mathcal{E}(Q^*) \\ &= (\mathcal{T}Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - \frac{1}{2}\mathcal{E}(Q^*) \\ &< (\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) \end{aligned} \tag{111}$$

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

According to Lemma 5, with the condition

$$\begin{aligned} 0 < \varepsilon &\leq \frac{(1-\gamma)\mathcal{E}(Q^*)}{\gamma n^3 |\mathcal{A}|^n 2^{n+4}(R_{\max}/(1-\gamma) + \mathcal{E}(Q^*)/(8n))} \\ &\leq \frac{(1-\gamma)\mathcal{E}(Q^*)}{\gamma n^3 |\mathcal{A}|^n 2^{n+4}(R_{\max}/(1-\gamma) + \gamma\|V_{\text{tot}}^{\boldsymbol{\pi}^*} - V^*\|_\infty)}, \end{aligned} \tag{112}$$

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

$$\left|(\mathcal{T}_D^{\text{LVD}} V)(s) - V^*(s)\right| = \left|(\mathcal{T}_D^{\text{LVD}} Q)_{\text{tot}}(s, \boldsymbol{\pi}^*(s)) - V^*(s)\right| \tag{113}$$

$$\leq \gamma \|V_{\text{tot}}^{\boldsymbol{\pi}^*} - V^*\|_\infty + \frac{1-\gamma}{8n\gamma}\mathcal{E}(Q^*) \leq \frac{1}{8n\gamma}\mathcal{E}(Q^*). \tag{114}$$

Combing Eq. (107), (110), and (112), the overall condition is

$$0 < \varepsilon \leq \frac{(1-\gamma)\mathcal{E}(Q^*)}{n^3 |\mathcal{A}|^n 2^{n+4}(R_{\max}/(1-\gamma) + \mathcal{E}(Q^*)/(8n))} \tag{115}$$

□

Lemma 2. *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 is implied by Lemma 8. □

Theorem 4 (Formal version of Theorem 2)**.** *Besides Lemma 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 state value function is sufficient to determine target values, so the subspace $\mathcal{B}$ defined in Lemma 8 is a compact and convex space in terms of V_{tot}. The operator $\mathcal{T}_D^{\text{LVD}}$ is a continuous mapping because it only involves elementary functions. According to Brouwer's Fixed Point Theorem (Brouwer, 1911), there exist $Q \in \mathcal{B}$ satisfying $\mathcal{T}_D^{\text{LVD}} Q \in \mathcal{B}$. In addition, according to the definition stated in Eq. (106), the fixed point must represent the unique optimal policy since it cannot lie on the boundary with $\mathcal{E}(Q) = 0$. □

F Experiment Settings and Implementation Details

F.1 Implmentation Details

We adopt the PyMARL (Samvelyan et al., 2019) implementation with default hyper-parameters to investigate four state-of-the-art multi-agent Q-learning algorithms: VDN (Sunehag et al., 2018), QMIX (Rashid et al., 2018), QTRAN (Son et al., 2019), and QPLEX (Wang et al., 2020a). The training time of these algorithms on an NVIDIA RTX 2080TI GPU is about 4 hours to 12 hours, which

Map Name	Replay Buffer Size	Behaviour Test Win Rate	Behaviour Policy
2s3z	20k episodes	91.2%	VDN
3s5z	20k episodes	77.5%	VDN
1c3s5z	20k episodes	79.8%	VDN
2s_vs_1sc	20k episodes	99.6%	VDN
3s_vs_5z	20k episodes	94.2%	VDN
3c7z	20k episodes	92.9%	VDN
5m_vs_6m	50k episodes	61.7%	VDN
10m_vs_11m	50k episodes	88.7%	VDN
3h_vs_4z	50k episodes	83.1%	VDN

Table 2: The dataset configurations of offline data collection setting.

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 VDN and collecting its 20k, 30k or 50k experienced episodes. The dataset configurations are shown in Table 2. 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 (Samvelyan et al., 2019), 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.

F.2 TWO-STATE MMDP

In the two-state MMDP shown in Figure 1a, due to the GRU-based implementation of 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 with the uniform data distribution. We adopt $\gamma = 0.9$ to implment FQI-LVD and deep MARL algorithms. In the FQI-LVD framework, $V_{max} = \frac{1}{1-\gamma} = 100$ as shown in Figure 1b. Figure 1c demonstrates that *Optimal* line is approximately $\sum_{i=0}^{99} \gamma^i = 63.4$ in one episode of 100 timesteps.

F.3 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 3 in the episodes of 100 timesteps.

Map Name	Ally Units	Enemy Units
2s3z	2 Stalkers & 3 Zealots	2 Stalkers & 3 Zealots
3s5z	3 Stalkers & 5 Zealots	3 Stalkers & 5 Zealots
1c3s5z	1 Colossus, 3 Stalkers & 5 Zealots	1 Colossus, 3 Stalkers & 5 Zealots
2s_vs_1sc	2 Stalkers	1 Spine Crawler
3s_vs_5z	3 Stalkers	5 Zealots
3c7z	3 Colossi & 7 Zealots	3 Colossi & 7 Zealots
5m_vs_6m	5 Marines	6 Marines
10m_vs_11m	10 Marines	11 Marines
3h_vs_4z	3 Hydralisks	4 Zealots

Table 3: SMAC challenges.

F.4 Deferred Tables in Section 6.1

a_2 \ a_1	$\mathcal{A}^{(1)}$	$\mathcal{A}^{(2)}$	$\mathcal{A}^{(3)}$
$\mathcal{A}^{(1)}$	**7.98**	-12.09	-12.10
$\mathcal{A}^{(2)}$	-12.18	-0.02	-0.02
$\mathcal{A}^{(3)}$	-12.11	-0.03	-0.03

(a) Q_{tot} of QPLEX.

a_2 \ a_1	$\mathcal{A}^{(1)}$	$\mathcal{A}^{(2)}$	$\mathcal{A}^{(3)}$
$\mathcal{A}^{(1)}$	**8.00**	-12.00	-12.00
$\mathcal{A}^{(2)}$	-12.00	-0.00	0.00
$\mathcal{A}^{(3)}$	-12.00	0.00	0.00

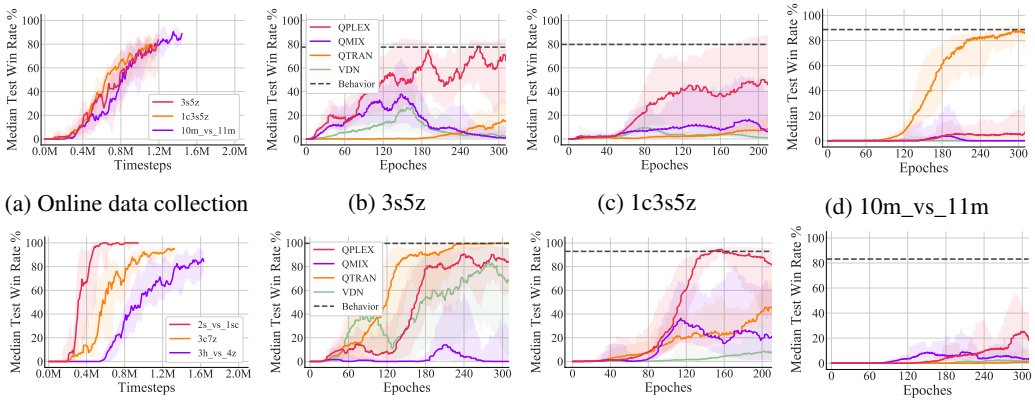
(b) Q_{tot} of QTRAN.

a_2 \ a_1	$\mathcal{A}^{(1)}$	$\mathcal{A}^{(2)}$	$\mathcal{A}^{(3)}$
$\mathcal{A}^{(1)}$	-7.88	-7.88	-7.88
$\mathcal{A}^{(2)}$	-7.88	-0.00	**-0.00**
$\mathcal{A}^{(3)}$	-7.88	-0.00	-0.00

(c) Q_{tot} of QMIX.

Table 4: (a-c) Joint action-value functions Q_{tot} of QPLEX, QTRAN, and QMIX. Boldface means the greedy joint action selection from Q_{tot}.

F.5 Deferred Figures in Section 6.2



(a) Online data collection (b) 3s5z (c) 1c3s5z (d) 10m_vs_11m

(e) Online data collection (f) 2s_vs_1sc (g) 3c7z (h) 3h_vs_4z

Figure 3: (a,e) Constructing datasets using online data collection. (b-d,f-h) Evaluating the learning performance of deep multi-agent Q-learning algorithms with a given static dataset on six maps.