QPLEX: Duplex Dueling Multi-Agent Q-Learning

Jianhao Wang[*1], **Zhizhou Ren**[*1], **Terry Liu**[1], **Yang Yu**[2], **Chongjie Zhang**[1]
[1]Institute for Interdisciplinary Information Sciences, Tsinghua University, China
[2]Nanjing University, China
{wjh19, rzz16, liudr18}@mails.tsinghua.edu.cn
yuy@nju.edu.cn
chongjie@tsinghua.edu.cn

Abstract

We explore value-based multi-agent reinforcement learning (MARL) in the popular paradigm of centralized training with decentralized execution (CTDE). CTDE requires the consistency of the optimal joint action selection with optimal individual action selections, which is called the IGM (*Individual-Global-Max*) principle. However, in order to achieve scalability, existing MARL methods either limit representation expressiveness of their value function classes or relax the IGM consistency, which may lead to poor policies or even divergence. This paper presents a novel MARL approach, called *duPLEX dueling multi-agent Q-learning* (QPLEX), that takes a duplex dueling network architecture to factorize the joint value function. This duplex dueling architecture transforms the IGM principle to easily realized constraints on advantage functions and thus enables efficient value function learning. Theoretical analysis shows that QPLEX solves a rich class of tasks. Empirical experiments on StarCraft II unit micromanagement tasks demonstrate that QPLEX significantly outperforms state-of-the-art baselines in both online and offline task settings, and also reveal that QPLEX achieves high sample efficiency and can benefit from offline datasets without additional exploration.

1 Introduction

Cooperative multi-agent reinforcement learning (MARL) has great promise for addressing many complex real-world problems, such as sensor networks [1], coordination of robot swarms [2] and autonomous cars [3]. However, cooperative MARL encounters two major challenges of scalability and partial observability in practical applications. The joint state-action space grows exponentially as the number of agents increases. The partial observability of the environment requires each agent to make its individual decisions based on the local observations. To address these challenges, a popular MARL paradigm, called *centralized training with decentralized execution* (CTDE) [4, 5], has recently attracted great attention, where agents' policies are trained with access to global information in a centralized way and executed only based on local observations in a decentralized way.

Many CTDE learning approaches have recently been proposed, among which value-based MARL algorithms [6–9] have shown state-of-the-art performance on the challenging unit micromanagement benchmark tasks of StarCraft II [10]. To enable effective CTDE for multi-agent Q-learning, it is essential that the optimal joint action should be equivalent to the collection of individual optimal actions of agents, which is called the IGM (*Individual-Global-Max*) principle [8]. Due to the exponential joint action space, the greedy action selection in the whole joint action space becomes intractable. To address this scalability issue, VDN [6] and QMIX [7] propose two sufficient conditions of IGM to factorize the joint action-value function. However, these two decomposition methods suffer

[*]Equal contribution.

Preprint. Under review.

from structural constraints and severely limit the joint action-value function class they can represent. The result of Wang et al. [11] indicates that the incompleteness of joint value function class may lead to poor performance or even unbounded divergence in off-policy training. To address this structural limitation, QTRAN [8] proposes a factorization method expressing the complete value function space induced by the IGM consistency, but its exact implementation is known to be computationally intractable and approximate versions have instable and poor performance in complex domains [12]. Therefore, achieving effective scalability remains an open problem for cooperative MARL.

To address this challenge, this paper presents a new MARL approach, called *duPLEX dueling multi-agent Q-learning* (QPLEX), that takes a duplex dueling network architecture to factorize the joint action-value function into individual action-value functions. QPLEX introduces the dueling structure $Q = V + A$ [13] for representing both joint and individual action-value functions and then reformalizes the IGM principle as an *advantage-based IGM*. This reformulation transforms the IGM consistency to easily realized constraints on the value range of advantage functions and thus facilitates the action-value function learning with linear decomposition structure. Unlike QTRAN that uses soft constraints and provides no guarantee for the exact IGM consistency [8], QPLEX takes advantage of a duplex dueling architecture to encode it into the network structure and provide a guaranteed IGM consistency. To our best knowledge, QPLEX is the first multi-agent Q-learning algorithm that effectively achieve high scalability with a full realization of the IGM principle.

We evaluate the performance of QPLEX in both didactic problems proposed by prior work [8, 11] and a range of unit micromanagement benchmark tasks in StarCraft II [10]. In these didactic problems, QPLEX demonstrates its full representation expressiveness and learns the optimal joint action-value function. Empirical results on more challenging StarCraft II tasks show that QPLEX significantly outperforms other multi-agent Q-learning baselines in both online and offline task settings. In particular, due to its full representation power for the IGM joint value function, QPLEX shows the ability of supporting off-policy training, which is not possessed by other baselines. This ability not only provides QPLEX with high stability and sample efficiency, but also with opportunities to efficiently utilize multi-source offline data without additional exploration [14–16].

2 Preliminaries

2.1 Decentralized Partially Observable MDP (Dec-POMDP)

We model a fully cooperative multi-agent task as a Dec-POMDP [17] defined by a tuple $\mathcal{M} = \langle \mathcal{N}, \mathcal{S}, \mathcal{A}, P, \Omega, O, r, \gamma \rangle$, where $\mathcal{N} \equiv \{1, 2, \ldots, n\}$ is a finite set of agents and $s \in \mathcal{S}$ is a finite set of global states. At each time step, every agent $i \in \mathcal{N}$ chooses an action $a_i \in \mathcal{A} \equiv \{\mathcal{A}^{(1)}, \ldots, \mathcal{A}^{(|\mathcal{A}|)}\}$ on a global state s, which forms a joint action $\boldsymbol{a} \equiv [a_i]_{i=1}^n \in \boldsymbol{\mathcal{A}} \equiv \mathcal{A}^n$. It results in a joint reward $r(s, \boldsymbol{a})$ and a transition to the next global state $s' \sim P(\cdot|s, \boldsymbol{a})$. $\gamma \in [0, 1)$ is a discount factor. We consider a *partially observable* setting, where each agent i receives an individual partial observation $o_i \in \Omega$ according to the observation probability function $O(o_i|s, a_i)$. Each agent i has an action-observation history $\tau_i \in \mathcal{T} \equiv (\Omega \times \mathcal{A})^*$ and constructs its individual policy $\pi_i(a|\tau_i)$ to jointly maximize team performance. We use $\boldsymbol{\tau} \in \boldsymbol{\mathcal{T}} \equiv \mathcal{T}^n$ to denote joint action-observation history. The formal objective function is to find a joint policy $\boldsymbol{\pi} = \langle \pi_1, \ldots, \pi_n \rangle$ that maximizes a joint value function $V^{\boldsymbol{\pi}}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, \boldsymbol{\pi}\right]$. Another quantity of interest in policy search is the joint action-value function $Q^{\boldsymbol{\pi}}(s, \boldsymbol{a}) = r(s, \boldsymbol{a}) + \gamma \mathbb{E}_{s'}[V^{\boldsymbol{\pi}}(s')]$.

2.2 Deep Multi-Agent Q-Learning in Dec-POMDP

Q-learning algorithms is a popular algorithm to find the optimal joint action-value function $Q^*(s, \boldsymbol{a}) = r(s, \boldsymbol{a}) + \gamma \mathbb{E}_{s'}\left[\max_{\boldsymbol{a'}} Q^*\left(s', \boldsymbol{a'}\right)\right]$. Deep Q-learning represents the action-value function with a deep neural network parameterized by $\boldsymbol{\theta}$. Mutli-agent Q-learning algorithms [6–9] use a replay memory D to store the transition tuple $(\boldsymbol{\tau}, \boldsymbol{a}, r, \boldsymbol{\tau'})$, where r is the reward for taking action $\boldsymbol{a}$ at joint action-observation history $\boldsymbol{\tau}$ with a transition to $\boldsymbol{\tau'}$. Due to partial observability, $Q\left(\boldsymbol{\tau}, \boldsymbol{a}; \boldsymbol{\theta}\right)$ is used in place of $Q\left(s, \boldsymbol{a}; \boldsymbol{\theta}\right)$. Thus, parameters $\boldsymbol{\theta}$ are learnt by minimizing the following expected TD error:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{\tau}, \boldsymbol{a}, r, \boldsymbol{\tau'}) \in D}\left[\left(r + \gamma V\left(\boldsymbol{\tau'}; \boldsymbol{\theta}^-\right) - Q\left(\boldsymbol{\tau}, \boldsymbol{a}; \boldsymbol{\theta}\right)\right)^2\right], \tag{1}$$

where $V\left(\boldsymbol{\tau'}; \boldsymbol{\theta}^-\right) = \max_{\boldsymbol{a'}} Q\left(\boldsymbol{\tau'}, \boldsymbol{a'}; \boldsymbol{\theta}^-\right)$ is the one-step expected future return of the TD target and $\boldsymbol{\theta}^-$ are the parameters of the target network, which will be periodically updated with $\boldsymbol{\theta}$.

2.3 Centralized Training with Decentralized Execution (CTDE)

CTDE is a popular paradigm of cooperative multi-agent deep reinforcement learning [6–9]. Agents are trained in a centralized way and granted access to other agents' information or the global states during the centralized training process. However, due to partial observability and communication constraints, each agent makes its own decision based on local action-observation history during the decentralized execution phase. A basic requirement of CTDE in multi-agent Q-learning is that the optimal joint action induced from the optimal centralized action-value function is equivalent to the collection of individual optimal actions of agents, which is called IGM (*Individual-Global-Max*) principle [8]. This principle asserts that a joint action-value function $Q_{tot}(\boldsymbol{\tau}, \boldsymbol{a})$ is factorizable if and only if there exists $[Q_i : \mathcal{T} \times \mathcal{A} \mapsto \mathbb{R}]_{i=1}^n$ such that $\forall \boldsymbol{\tau} \in \boldsymbol{\mathcal{T}}$:

$$\arg\max_{\boldsymbol{a} \in \boldsymbol{\mathcal{A}}} Q_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = \left(\arg\max_{a_1 \in \mathcal{A}} Q_1(\tau_1, a_1), \ldots, \arg\max_{a_n \in \mathcal{A}} Q_n(\tau_n, a_n) \right). \tag{2}$$

Two factorization structures, **additivity** and **monotonicity**, has been proposed by VDN [6] and QMIX [7], respectively, as shown below:

$$Q_{tot}^{\text{VDN}}(\boldsymbol{\tau}, \boldsymbol{a}) = \sum_{i=1}^{n} Q_i(\tau_i, a_i) \quad \text{and} \quad \forall i \in \mathcal{N}, \frac{\partial Q_{tot}^{\text{QMIX}}(\boldsymbol{\tau}, \boldsymbol{a})}{\partial Q_i(\tau_i, a_i)} > 0 \tag{3}$$

These two structures are sufficient conditions for the IGM constraint. However, they are not necessary conditions and limits their representation expressiveness of joint action-value functions [12]. There exist tasks whose factorizable joint action-value functions can not be represented by these decomposition methods, as shown in Section 4 in this paper. In contrast, QTRAN [8] transforms IGM to a linear constraint and uses it as soft regularization constrains. However, this relaxation may violate the exact IGM consistency and result in poor performance in complex problems.

3 QPLEX: Duplex Dueling Multi-Agent Q-Learning

In this section, we will first introduce advantage-based IGM, equivalent to the regular IGM principle, and, with this new definition, convert the IGM consistency of greedy action selection to simple constraints on advantage functions. We then present a novel deep MARL model, called *duPLEX dueling multi-agent Q-learning algorithm* (QPLEX), that realizes these constraints without sacrificing its scalability.

3.1 Advantage-Based IGM

To ensure the consistency of greedy action selection on the joint and local action-value functions, IGM principle constrains the relative order of Q values over action space. From the perspective of dueling decomposition structure $Q = V + A$ proposed by Dueling DQN [13], this consistency should only constrain the action-dependent advantage term A and be free of the state-value function V. This observation naturally motivates us to reformalize the IGM principle as an advantage-based IGM, which transforms the consistency constraint to advantage functions.

Definition 1. *[Advantage-based IGM] For a joint action-value function Q_{tot}: $\boldsymbol{\mathcal{T}} \times \boldsymbol{\mathcal{A}} \mapsto \mathbb{R}$, if there exists individual action-value functions $[Q_i : \mathcal{T} \times \mathcal{A} \mapsto \mathbb{R}]_{i=1}^n$, where $\forall \boldsymbol{\tau} \in \boldsymbol{\mathcal{T}}$, $\forall \boldsymbol{a} \in \boldsymbol{\mathcal{A}}$, $\forall i \in \mathcal{N}$,*

$$\textbf{\textit{(Joint Dueling)}} \quad Q_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = V_{tot}(\boldsymbol{\tau}) + A_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) \text{ and } V_{tot}(\boldsymbol{\tau}) = \max_{\boldsymbol{a'}} Q_{tot}(\boldsymbol{\tau}, \boldsymbol{a'}), \tag{4}$$

$$\textbf{\textit{(Individual Dueling)}} \quad Q_i(\tau_i, a_i) = V_i(\tau_i) + A_i(\tau_i, a_i) \text{ and } V_i(\tau_i) = \max_{a_i'} Q_i(\tau_i, a_i'), \tag{5}$$

such that the following holds

$$\arg\max_{\boldsymbol{a} \in \boldsymbol{\mathcal{A}}} A_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = \left(\arg\max_{a_1 \in \mathcal{A}} A_1(\tau_1, a_1), \ldots, \arg\max_{a_n \in \mathcal{A}} A_n(\tau_n, a_n) \right), \tag{6}$$

then, we can say that $[Q_i]_{i=1}^n$ satisfies advantage-based IGM for Q_{tot}.

As specified in Definition 1, the advantage-based IGM takes a duplex dueling architecture, *Joint Dueling* and *Individual Dueling*, which induces the joint and local advantage fucntions by $A = Q - V$. Compared to the regular IGM, the advantage-based IGM transfers the consistency constraint on action-value functions stated in Eq. (2) to that on advantage functions. This change is an equivalent transformation because the state-value terms V do not affect the action selection, as shown by Proposition 1.

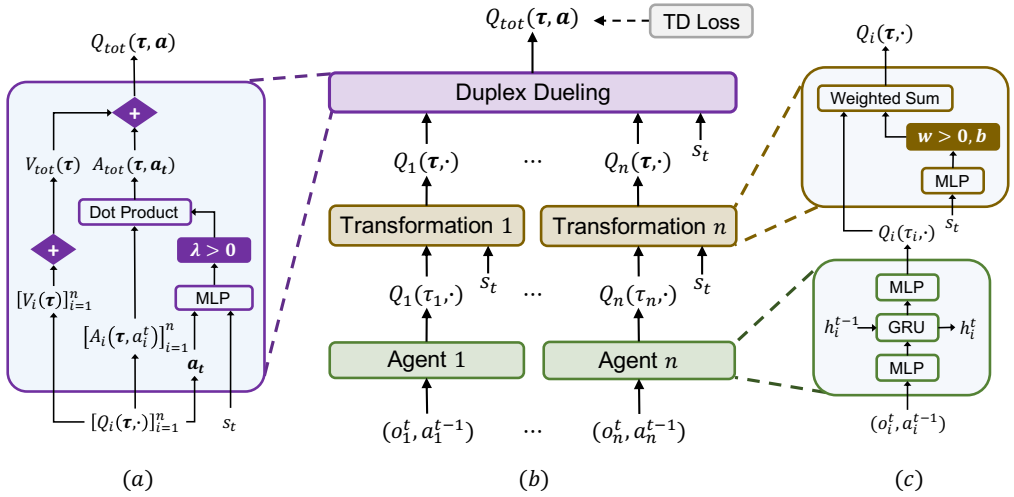


Figure 1: (a) Duplex dueling network structure. (b) The overall QPLEX architecture. (c) Agent network structure (bottom) and Transformation network structure (top).

Proposition 1. *The action-value function classes derived from advantage-based IGM and IGM are equivalent.*

One key benefit of using the advantage-based IGM is that its consistency constraint can be easily realized by limiting the value range of advantage functions, as indicated by the following fact.

Fact 1. *The constraint of advantage-based IGM stated in Eq.* (6) *is equivalent to that when* $\forall \boldsymbol{\tau} \in \boldsymbol{\mathcal{T}}$, $\forall \boldsymbol{a^*} \in \boldsymbol{\mathcal{A}}^*(\boldsymbol{\tau})$, $\forall \boldsymbol{a} \in \boldsymbol{\mathcal{A}} \setminus \boldsymbol{\mathcal{A}}^*(\boldsymbol{\tau})$, $\forall i \in \mathcal{N}$,

$$A_{tot}(\boldsymbol{\tau}, \boldsymbol{a^*}) = A_i(\tau_i, a_i^*) = 0 \quad \textit{and} \quad A_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) < 0, A_i(\tau_i, a_i) \leq 0, \tag{7}$$

where $\boldsymbol{\mathcal{A}}^*(\boldsymbol{\tau}) = \{\boldsymbol{a} | \boldsymbol{a} \in \boldsymbol{\mathcal{A}}, Q_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = V_{tot}(\boldsymbol{\tau})\}$.

Fact 1 enables us to develop an efficient MARL algorithm that allows the joint value function learning with any scalable decomposition structure and just imposes simple constraints limiting value ranges of advantage functions in order to achieve a full expressiveness power induced by advantage-based IGM or IGM. The next section will describe such a MARL algorithm.

3.2 The QPLEX Architecture

In this section, we present a novel multi-agent Q-learning algorithm with duplex dueling architecture, called QPLEX, that realizes the advantage-based IGM constraint by exploiting Fact 1. The overall architecture of QPLEX is illustrated in Figure 1, which consists of three main modules as follows: (i) an *Individual Action-Value Function* for each agent, (ii) a *Transformation* module that incorporates the information of global state or joint history into individual action-value functions during the centralized training process, and (iii) the *Duplex Dueling* network module that composes individual action-value functions into a joint action-value function under the advantage-based IGM constraint. During the centralized training, the whole network is learned in an end-to-end fashion to minimize TD loss as specified in Eq. (1) and, during the decentralized execution, the transformation and duplex dueling modules will be removed and each agent will select their action with its individual Q-function based on its local observation history.

Individual Action-Value Function is represented by a recurrent Q-network for each agent i, which takes last hidden states h_i^{t-1}, current local observations o_i^t, and last action a_i^{t-1} as inputs and outputs local $Q_i(\tau_i, a_i)$.

Transformation network module uses the centralized information to transform local action-value functions $[Q_i(\tau_i, a_i)]_{i=1}^n$ to $[Q_i(\boldsymbol{\tau}, a_i)]_{i=1}^n$ conditioned on the joint observation history, as shown below, for any agent i,

$$Q_i(\boldsymbol{\tau}, a_i) = w_i(\boldsymbol{\tau}) Q_i(\tau_i, a_i) + b_i(\boldsymbol{\tau}), \tag{8}$$

where $w_i(\boldsymbol{\tau}) > 0$ is the positive weight. This positive linear transformation maintains the consistency of the greedy action selection and alleviates the partial observability in Dec-POMDP [8, 9]. As used

by QMIX [7], QTRAN [8], and Qatten [9], the centralized information can be the global state s, if available, or the joint observation history $\boldsymbol{\tau}$.

Duplex Dueling network module takes the transformation network outputs as input, e.g., $[Q_i]_{i=1}^n$, and produces the values of joint Q_{tot}, as shown in Figure 1a. This duplex dueling network ensures the IGM consistency between individual action-value functions and the joint action-value function. It first derives the dueling structure for each agent i by computing the value function $V_i(\boldsymbol{\tau}) = \max_{a_i} Q_i(\boldsymbol{\tau}, a_i)$ and the advantage function $A_i(\boldsymbol{\tau}, a_i) = Q_i(\boldsymbol{\tau}, a_i) - V_i(\boldsymbol{\tau})$, then uses individual value and advantage functions to compute the joint value $V_{tot}(\boldsymbol{\tau})$ and the joint advantage $A_{tot}(\boldsymbol{\tau}, \boldsymbol{a})$, respectively, and finally outputs $Q_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = V_{tot}(\boldsymbol{\tau}) + A_{tot}(\boldsymbol{\tau}, \boldsymbol{a})$ by using the joint dueling structure.

Based on Fact 1, the advantage-based IGM principle imposes no constraints on value functions. Therefore, to enable efficient learning, we use a simple sum structure to compose the joint value:

$$V_{tot}(\boldsymbol{\tau}) = \sum_{i=1}^{n} V_i(\boldsymbol{\tau}) \tag{9}$$

To enforce the IGM consistency of the joint advantage and individual advantages, as specified by Eq. (7), QPLEX computes the joint advantage function as follows:

$$A_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = \sum_{i=1}^{n} \lambda_i(\boldsymbol{\tau}, \boldsymbol{a}) A_i(\boldsymbol{\tau}, a_i), \text{ where } \lambda_i(\boldsymbol{\tau}, \boldsymbol{a}) > 0. \tag{10}$$

The joint advantage function A_{tot} is the dot product of local advantage functions $[A_i]_{i=1}^t$ and positive importance weights $[\lambda_i]_{i=1}^n$. This positivity induced by λ_i will continue to maintain the consistency flow of the greedy action selection. To enable efficient learning of importance weights λ_i with joint observation history and action, QPLEX uses a small multi-head attention module [18]:

$$\lambda_i(\boldsymbol{\tau}, \boldsymbol{a}) = \sum_{k=1}^{K} \lambda_{i,k}(\boldsymbol{\tau}, \boldsymbol{a}) \phi_{i,k}(\boldsymbol{\tau}) \upsilon_k(\boldsymbol{\tau}), \tag{11}$$

where K is the number of attention heads, $\lambda_{i,k}(\boldsymbol{\tau}, \boldsymbol{a})$ and $\phi_{i,k}(\boldsymbol{\tau})$ are attention weights activated by a sigmoid regularizer, and $\upsilon_k(\boldsymbol{\tau}) > 0$ is a positive key of each head. This sigmoid activation of λ_i brings sparsity to the credit assignment of the joint advantage function to individuals, which enables efficient multi-agent learning [19].

With Eq. (9) and (10), the joint action-value function Q_{tot} can be reformulated as following:

$$Q_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = V_{tot}(\boldsymbol{\tau}) + A_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = \sum_{i=1}^{n} Q_i(\boldsymbol{\tau}, a_i) + \sum_{i=1}^{n} \left(\lambda_i(\boldsymbol{\tau}, \boldsymbol{a}) - 1\right) A_i(\boldsymbol{\tau}, a_i). \tag{12}$$

It can be seem that Q_{tot} consists of two terms. The first term is the sum of individual action-value functions $[Q_i]_{i=1}^n$, which is basically the joint action-value function Q_{tot}^{VDN} of VDN [6]. The second term corrects for the discrepancy between the centralized joint action-value function and Q_{tot}^{VDN}, which enables QPLEX with a full expressiveness power.

Proposition 2. *Assume neural networks provide universal function approximation, the joint action-value function class that QPLEX can realize is equivalent to what is induced by the IGM principle.*

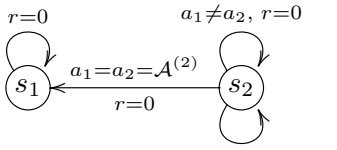
Proposition 2 assumes that the neural network of QPLEX can be large enough to achieve the full expressiveness of action-value functions by universal approximation theorem [20]. This full expressiveness of the action-value function class is very critical for multi-agent Q-learning algorithms. As shown by Wang et al. [11], insufficient representation complexity, like linear value decomposition used by VDN [6] and Qatten [9] and monotonic value decomposition used by QMIX [7], may result in learning divergence in some cases. In contrast, QPLEX shows stable and superior performance in both online and offline tasks settings, as demonstrated in Section 4.

4 Experiments

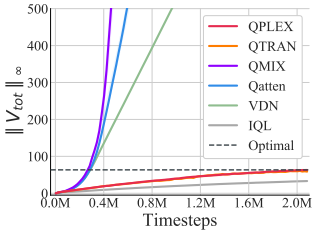
In this section, we will first consider two didactic examples proposed by prior work [8, 11] to investigate the optimality and convergence of QPLEX. To demonstrate the scalability on complex MARL domains, we evaluate the performance of QPLEX and other multi-agent Q-learning algorithms

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.



(b) Two-state MMDP.



(c) Infinity norm of V_{tot}.

Figure 2: (a) Payoff matrix of the one-step game. Boldface means the optimal joint action selection from payoff matrix. (b) An MMDP where mutli-agent Q-learning algorithms with linear or monotonic value decomposition may diverge to infinity. r is a shorthand for $r(s, \boldsymbol{a})$. (c) The learning curve of $\|V_{tot}\|_\infty$ while running several deep multi-agent Q-learning algorithms in the given MMDP.

a_2 \ a_1	$\mathcal{A}^{(1)}$	$\mathcal{A}^{(2)}$	$\mathcal{A}^{(3)}$
$\mathcal{A}^{(1)}$	**8.0**	-12.1	-12.1
$\mathcal{A}^{(2)}$	-12.2	-0.0	-0.0
$\mathcal{A}^{(3)}$	-12.1	-0.0	-0.0

(a) Q_{tot} of QPLEX.

a_2 \ a_1	$\mathcal{A}^{(1)}$	$\mathcal{A}^{(2)}$	$\mathcal{A}^{(3)}$
$\mathcal{A}^{(1)}$	**8.0**	-12.0	-12.0
$\mathcal{A}^{(2)}$	-12.0	-0.0	0.0
$\mathcal{A}^{(3)}$	-12.0	0.0	0.0

(b) Q_{tot} of QTRAN.

a_2 \ a_1	$\mathcal{A}^{(1)}$	$\mathcal{A}^{(2)}$	$\mathcal{A}^{(3)}$
$\mathcal{A}^{(1)}$	-7.8	-7.8	-7.8
$\mathcal{A}^{(2)}$	-7.8	-0.0	**-0.0**
$\mathcal{A}^{(3)}$	-7.8	-0.0	-0.0

(c) Q_{tot} of QMIX.

Table 1: Joint action-value functions Q_{tot} of QPLEX, QTRAN, and QMIX. Boldface means the greedy joint action selection from Q_{tot}.

on a range of StarCraft II unit micromanagement benchmark tasks [10]. We compare our method with five state-of-the-art baselines: QTRAN [8], Qatten [9], QMIX [7], VDN [6], and independent Q-learning (IQL) [21] in the didactic and StarCraft II tasks. VDN, QMIX, and QTRAN correspond to the most related baselines discussed in Section 1. Qatten uses a linear value decomposition structure with an attention-based *Transformation* module for incorporating centralized information. Every agent of IQL considers other agents as part of the environment to realize a single-agent setting with non-stationarity. The implementation details of QPLEX and five baselines are deferred to Appendix. Towards fair evaluation, all experimental results are illustrated with the median performance as well as 25-75% percentiles over 6 random seeds. The videos of our experiments on StarCraft II are available on an anonymous website[2].

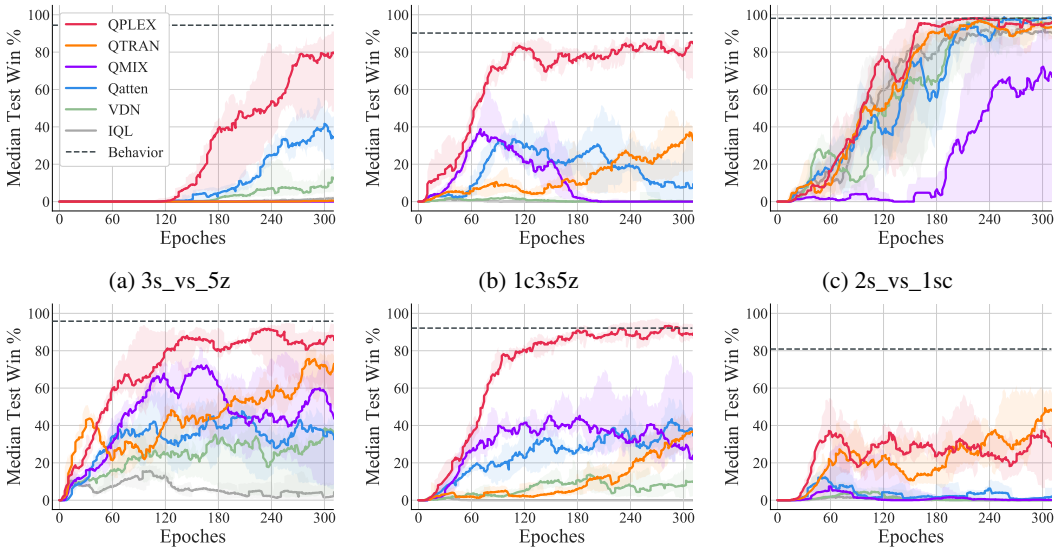
4.1 Didactic Examples

We first demonstrate our method in two didactic cases. To ensure the sufficient data collection in the joint action space, we adopts an uniform exploration strategy (*i.e.*, ϵ-greedy exploration with $\epsilon = 1$) for more than 100k or 2000k steps in the following two didactic problems, respectively.

Matrix Game. The matrix game proposed by QTRAN [8] with two agents and three actions is illustrated in Table 2a. This symmetric matrix game describes a simple cooperative multi-agent task with considerable miscoordination penalties, whose optimal joint strategy is to perform action $\mathcal{A}^{(1)}$ simultaneously. As shown in Table 1a and 1b, QPLEX and QTRAN with higher expressiveness power can achieve the optimal joint action-value functions, while the non-optimal joint action-value functions learned by other baselines (*e.g.*, QMIX) are either illustrated in Table 1c or deferred to Appendix. These joint action-value functions empirically reveal that multi-agent Q-learning algorithms with insufficient expressiveness power may fall into the local optimality induced by miscoordination penalties. QTRAN achieves superior performance in this matrix game but will suffer from its relaxation of IGM consistency on complex domains shown in the section 4.2 of StarCraft II.

Two-State MMDP. In this didactic example, we focus on Multi-agent Markov Decision Process (MMDP) [22] which is a fully cooperative multi-agent setting with full observability. Consider the two-state MMDP proposed by Wang et al. [11] with two agents and two actions (see Figure 2b). Two agents start at state s_2 and explore extrinsic rewards for 100 environment steps. The optimal policy of this MMDP is simply executing the action $\mathcal{A}^{(1)}$ at state s_2, which is the only coordination

[2]https://sites.google.com/view/qplex-marl/



(a) 3s_vs_5z (b) 1c3s5z (c) 2s_vs_1sc

(d) 2s3z (e) 3s5z (f) 2c_vs_64zg

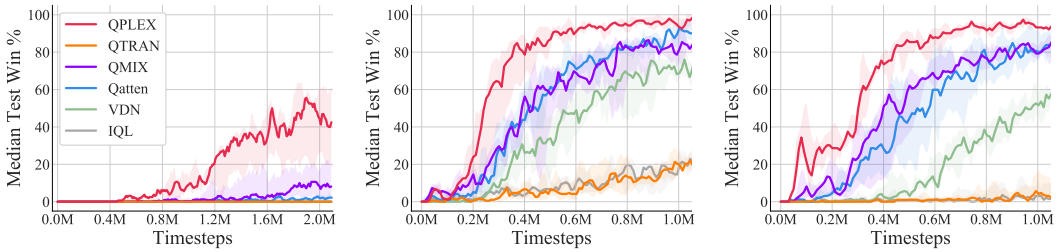
Figure 3: Learning curve of StarCraft II with offline data collection on six different maps.

pattern to obtain a positive reward. As shown in Figure 2c, the value function V_{tot} learned by baseline algorithms using limited function classes, including QMIX, VDN, and Qatten, will diverge to infinity. This divergence phenomenon has been investigated by Wang et al. [11]. By utilizing richer function classes, both QTRAN and QPLEX can address this instability issue and converge to the optimal joint value function.

4.2 StarCraft II

A more challenging set of empirical experiments are based on StarCraft Multi-Agent Challenge (SMAC) benchmark [10]. To demonstrate the off-policy nature of our proposed method, we adopt the offline data collection setting proposed by Fujimoto et al. [23], which can only be granted access to a given dataset without additional exploration. We also investigate the empirical performance in another popular experimental setting with ϵ-greedy exploration and a limited first-in first-out (FIFO) buffer [10], named online data collection setting. Ablation study of QPLEX investigates the influence of *Transformation* module mentioned in section 3.2, which is a popular alleviation trick for partial observability in Dec-POMDP [8, 9]. We pause training every 10k timesteps and evaluate 32 episodes with decentralised greedy action selection to measure *test win rate* of each algorithm. The detailed experimental setting of StarCraft II refers to Appendix.

Training with Offline Data Collection. We illustrate the learning curve of StarCraft II with offline data collection in Figure 3. We adopt a similar experiment setting with Fujimoto et al. [23] called *batch reinforcement learning*, which is a standard evaluation setting for off-policy training. Unlike other related work that study extrapolation error [14–16], we adopt a diverse dataset to ensure the sufficient data collection, which makes the expressiveness power of multi-agent Q-learning algorithms become the dominant factor to be investigated. To construct a diverse dataset, we train a behaviour policy and collect all its experienced transitions throughout the training process (see the details in Appendix). Notably, our method QPLEX significantly outperforms other multi-agent Q-learning baselines including QTRAN and may reach the *Behavior* line. Most of the baselines cannot utilize the off-policy dataset collected by an unfamiliar behaviour policy due to their limited expressiveness power. This argument has been justified by Wang et al. [11] through theoretical and empirical analysis of unexpected approximation error induced by limited expressiveness. Thus, our full representation expressiveness power will protect the off-policy nature of QPLEX, which induces excellent learning performance in this offline task setting. As shown in Figure 3, QMIX and Qatten cannot always maintain stable learning performance, meanwhile, VDN and IQL also suffer from offline data collection and lead to pretty weak empirical results. QTRAN may perform well in certain cases in which the soft constraints, two ℓ_2-penalty terms, are well minimized. However, this condition



(a) 5s10z (b) 1c3s5z (c) 3s5z

Figure 4: Learning curve of StarCraft II with online data collection on three different maps.

may not hold and this relaxation of QTRAN may lead to poor empirical performance on complicated domains, especially in the following online data collection tasks.

Training with Online Data Collection. Figure 4 shows the results of StarCraft II under the online data collection procedure. This popular experimental setting proposed by Samvelyan et al. [10] uses ϵ-greedy exploration strategy with a limited FIFO buffer to construct a dataset online and train the model based on it (see the details in Appendix). Obviously, our method QPLEX also outperforms other baselines by a large margin with higher sample efficiency during online data collection. On the super hard map 5s10z, the outperformance gap between QPLEX and five baselines exceeds 30% win rate after 2 million steps of training. With the online data collection, most multi-agent Q-learning baselines including Qatten, QMIX, VDN, and IQL can achieve reasonable performance. Compared with offline data collection, we conjecture that the limited representation expressiveness of such baselines may not cause a huge effect empirically. The theoretical benefits of online data collection in multi-agent Q-learning with linear value decomposition [11] may support our speculation. QTRAN shows pretty weak performance in this setting, which may be because its relaxation of IGM consistency affects the online data collection process and leads to unexpected deviations in the training dataset.

Ablation Study. We conduct the ablation study on several StarCraft II tasks to evaluate the importance of QPLEX's *Transformation* module described in section 3.2. We denote the QPLEX algorithm without *Transformation* module as QPLEX-NT for simplicity, whose implementation details are discussed in Appendix. Figure 5 presents the median win rates for QPLEX and QPLEX-NT with offline data collection on 1c3s5z map and other ablation experiments are deferred to Appendix. These plots show that QPLEX remarkably outperforms QPLEX-NT. These empirical performance gaps confirm that centralized information used in the training phase is indeed beneficial to improving sample efficiency and final performance, which has been widely studied in [7–9]. This ablation study reveals that the *Transformation* module is critical for QPLEX to achieve the highly scalable and efficient multi-agent Q-learning algorithm.

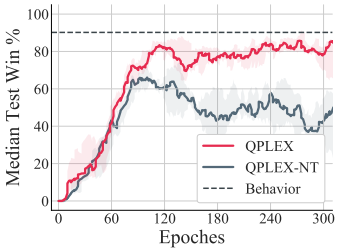


Figure 5: Win rates for QPLEX and its ablation QPLEX-NT with offline data collection on 1c3s5z map.

5 Conclusion

In this paper, we introduced QPLEX, a novel multi-agent Q-learning framework that allows centralized end-to-end training and efficiently learns to factorize a joint action-value function to enable decentralized execution. QPLEX takes advantage of a duplex dueling architecture that efficiently encodes the consistency constraint between the optimal joint action selection and optimal individual action selections. Our theoretical analysis shows that QPLEX solves a rich class of tasks. Empirical results demonstrate that it significantly outperforms state-of-the-art baselines in both online and offline task settings, including VDN, QMIX, QTRAN, and Qatten. In particular, unlike these baselines, QPLEX possesses a strong ability of supporting off-policy training. This ability provides QPLEX with high sample efficiency and opportunities of utilizing offline multi-source datasets, which has been seen as a key step for real-world RL applications [24]. It will be an interesting and valuable direction to study efficient off-policy multi-agent exploration for complementing QPLEX.

Broader Impact

In recent years, compared to prosperity in development of MARL algorithms, real-world applications of MARL are still limited. Major constraints include learning from a real system with limited samples and insufficient explorations of possible collaborations. Our proposed algorithm, QPLEX, to a great extent relieves these constraints as it is an off-policy MARL algorithm which could effectively use off-policy data and learn from off-policy exploration. It could provide solutions to many empirical multi-agent problems such as warehouse robots coordination, autonomous vehicle traffic planning, Unmanned Aerial Vehicles (UAVs) control and etc.

Nevertheless, real-world applications of MARL would require data collection from the task environment. This may cause invasion of people's privacy to different extents based on the applied scenarios. A simple solution is to acquire permissions from people whose data can be used for training. Since QPLEX is an off-policy MARL algorithm, it could effectively use all data as long as consent has been granted by their providers.

Moreover, QPLEX not only fills the gap between deep MARL and real-world applications, it could also bring impacts on human society, which can be regarded as a large multi-agent system. QPLEX, as a cooperative MARL algorithm, could help to reveal insights on how human beings can work together to foster the development of mankind. QPLEX can learn from past experience (history) and offer a potentially more effective cooperation strategy that society can collaborate to maximize social well-being in the future.

References

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

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

[3] 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.

[4] 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.

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

[6] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 2085–2087, 2018.

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

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

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

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

[11] Jianhao Wang, Zhizhou Ren, Beining Han, and Chongjie Zhang. Towards understanding linear value decomposition in cooperative multi-agent q-learning. *arXiv preprint arXiv:2006.00587*, 2020.

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

[13] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. Dueling network architectures for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1995–2003, 2016.

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

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

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

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

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

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

[20] Balázs Csanád Csáji et al. Approximation with artificial neural networks. *Faculty of Sciences, Etvs Lornd University, Hungary*, 24(48):7, 2001.

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

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

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

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

A Omitted Proofs in Section 3

Definition 1. *[Advantage-based IGM] For a joint action-value function Q_{tot}: $\boldsymbol{\mathcal{T}} \times \boldsymbol{\mathcal{A}} \mapsto \mathbb{R}$, if there exists individual action-value functions $[Q_i : \mathcal{T} \times \mathcal{A} \mapsto \mathbb{R}]_{i=1}^n$, where $\forall \boldsymbol{\tau} \in \boldsymbol{\mathcal{T}}$, $\forall \boldsymbol{a} \in \boldsymbol{\mathcal{A}}$, $\forall i \in \mathcal{N}$,*

$$\textbf{\textit{(Joint Dueling)}} \quad Q_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = V_{tot}(\boldsymbol{\tau}) + A_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) \text{ and } V_{tot}(\boldsymbol{\tau}) = \max_{\boldsymbol{a'}} Q_{tot}(\boldsymbol{\tau}, \boldsymbol{a'}), \quad (4)$$

$$\textbf{\textit{(Individual Dueling)}} \quad Q_i(\tau_i, a_i) = V_i(\tau_i) + A_i(\tau_i, a_i) \text{ and } V_i(\tau_i) = \max_{a_i'} Q_i(\tau_i, a_i'), \quad (5)$$

such that the following holds

$$\arg\max_{\boldsymbol{a} \in \boldsymbol{\mathcal{A}}} A_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = \left(\arg\max_{a_1 \in \mathcal{A}} A_1(\tau_1, a_1), \ldots, \arg\max_{a_n \in \mathcal{A}} A_n(\tau_n, a_n) \right), \quad (6)$$

then, we can say that $[Q_i]_{i=1}^n$ satisfies advantage-based IGM for Q_{tot}.

Let the action-value function class derived from IGM is denoted by

$$\widetilde{\mathcal{Q}} = \left\{ \left(\widetilde{Q}_{tot} \in \mathbb{R}^{|\boldsymbol{\mathcal{T}}||\mathcal{A}|^n}, \left[\widetilde{Q}_i \in \mathbb{R}^{|\boldsymbol{\mathcal{T}}||\mathcal{A}|} \right]_{i=1}^n \right) \,\middle|\, \text{Eq. (2) is satisfied} \right\}, \quad (13)$$

where $\widetilde{Q}_{tot}$ and $\left[\widetilde{Q}_i \right]_{i=1}^n$ denote the joint and individual action-value functions induced by IGM, respectively. Similarly, let

$$\widehat{\mathcal{Q}} = \left\{ \left(\widehat{Q}_{tot} \in \mathbb{R}^{|\boldsymbol{\mathcal{T}}||\mathcal{A}|^n}, \left[\widehat{Q}_i \in \mathbb{R}^{|\boldsymbol{\mathcal{T}}||\mathcal{A}|} \right]_{i=1}^n \right) \,\middle|\, \text{Eq. (4), (5), (6) are satisfied} \right\} \quad (14)$$

denote the action-value function class derived from advantage-based IGM. $\widetilde{V}_{tot}$ and $\widetilde{A}_{tot}$ denote the joint state-value and advantage functions, respectively. $\left[\widetilde{V}_i \right]_{i=1}^n$ and $\left[\widetilde{A}_i \right]_{i=1}^n$ denote the individual state-value and advantage functions induced by advantage-IGM, respectively. According to the duplex dueling architecture $Q = V + A$ stated in advantage-based IGM (see Definition 1), we derive the joint and individual action-value functions as following: $\forall \boldsymbol{\tau} \in \boldsymbol{\mathcal{T}}$, $\forall \boldsymbol{a} \in \boldsymbol{\mathcal{A}}$, $\forall i \in \mathcal{N}$,

$$\widehat{Q}_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = \widehat{V}_{tot}(\boldsymbol{\tau}) + \widehat{A}_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) \quad \text{and} \quad \widehat{Q}_i(\tau_i, a_i) = \widehat{V}_i(\tau_i) + \widehat{A}_i(\tau_i, a_i). \quad (15)$$

Proposition 1. *The action-value function classes derived from advantage-based IGM and IGM are equivalent.*

Proof. We will prove $\widetilde{\mathcal{Q}} \equiv \widehat{\mathcal{Q}}$ in the following two directions.

$\widetilde{\mathcal{Q}} \subseteq \widehat{\mathcal{Q}}$ For any $\left(\widetilde{Q}_{tot}, \left[\widetilde{Q}_i \right]_{i=1}^n \right) \in \widetilde{\mathcal{Q}}$, we construct $\widehat{Q}_{tot} = \widetilde{Q}_{tot}$ and $\left[\widehat{Q}_i \right]_{i=1}^n = \left[\widetilde{Q}_i \right]_{i=1}^n$. The joint and individual state-value/advantage functions induced by advantage-IGM

$$\widehat{V}_{tot}(\boldsymbol{\tau}) = \max_{\boldsymbol{a'}} \widehat{Q}_{tot}(\boldsymbol{\tau}, \boldsymbol{a'}) \quad \text{and} \quad \widehat{A}_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = \widehat{Q}_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) - \widehat{V}_{tot}(\boldsymbol{\tau}), \quad (16)$$

$$\widehat{V}_i(\tau_i) = \max_{a_i'} \widehat{Q}_i(\tau_i, a') \quad \text{and} \quad \widehat{A}_i(\tau_i, \boldsymbol{a}) = \widehat{Q}_i(\tau_i, a') - \widehat{V}_i(\tau_i), \quad \forall i \in \mathcal{N}, \quad (17)$$

are derived by Eq. (4) and Eq. (5), respectively. Because state-value functions do not affect the greedy action selection, $\forall \boldsymbol{\tau} \in \boldsymbol{\mathcal{T}}$, $\forall \boldsymbol{a} \in \boldsymbol{\mathcal{A}}$,

$$\arg\max_{\boldsymbol{a} \in \boldsymbol{\mathcal{A}}} \widetilde{Q}_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = \left(\arg\max_{a_1 \in \mathcal{A}} \widetilde{Q}_1(\tau_1, a_1), \ldots, \arg\max_{a_n \in \mathcal{A}} \widetilde{Q}_n(\tau_n, a_n) \right) \quad (18)$$

$$\Rightarrow \arg\max_{\boldsymbol{a} \in \boldsymbol{\mathcal{A}}} \widehat{Q}_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = \left(\arg\max_{a_1 \in \mathcal{A}} \widehat{Q}_1(\tau_1, a_1), \ldots, \arg\max_{a_n \in \mathcal{A}} \widehat{Q}_n(\tau_n, a_n) \right) \quad (19)$$

$$\Rightarrow \arg\max_{\boldsymbol{a} \in \boldsymbol{\mathcal{A}}} \left(\widehat{Q}_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) - \widehat{V}_{tot}(\boldsymbol{\tau}) \right) = \quad (20)$$

$$\left(\arg\max_{a_1 \in \mathcal{A}} \left(\widehat{Q}_1(\tau_1, a_1) - \widehat{V}_1(\tau_1) \right), \ldots, \arg\max_{a_n \in \mathcal{A}} \left(\widehat{Q}_n(\tau_n, a_n) - \widehat{V}_n(\tau_n) \right) \right) \quad (21)$$

$$\Rightarrow \arg\max_{\boldsymbol{a} \in \boldsymbol{\mathcal{A}}} \widehat{A}_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = \left(\arg\max_{a_1 \in \mathcal{A}} \widehat{A}_1(\tau_1, a_1), \ldots, \arg\max_{a_n \in \mathcal{A}} \widehat{A}_n(\tau_n, a_n) \right). \quad (22)$$

$$(23)$$

Thus, $\left(\widehat{Q}_{tot}, \left[\widehat{Q}_i \right]_{i=1}^n \right) \in \widehat{\mathcal{Q}}$, which means that $\widetilde{\mathcal{Q}} \subseteq \widehat{\mathcal{Q}}$.

$\widehat{\mathcal{Q}} \subseteq \widetilde{\mathcal{Q}}$ We will prove this direction in the same way. For any $\left(\widehat{Q}_{tot}, \left[\widehat{Q}_i\right]_{i=1}^n\right) \in \widehat{\mathcal{Q}}$, we construct $\widetilde{Q}_{tot} = \widehat{Q}_{tot}$ and $\left[\widetilde{Q}_i\right]_{i=1}^n = \left[\widehat{Q}_i\right]_{i=1}^n$. Because state-value functions do not affect the greedy action selection, $\forall \boldsymbol{\tau} \in \boldsymbol{\mathcal{T}}$, $\forall \boldsymbol{a} \in \boldsymbol{\mathcal{A}}$,

$$\arg\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}} \widehat{A}_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = \left(\arg\max_{a_1\in\mathcal{A}} \widehat{A}_1(\tau_1, a_1), \ldots, \arg\max_{a_n\in\mathcal{A}} \widehat{A}_n(\tau_n, a_n)\right) \tag{24}$$

$$\Rightarrow \arg\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}} \left(\widehat{A}_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) + \widehat{V}_{tot}(\boldsymbol{\tau})\right) = \tag{25}$$

$$\left(\arg\max_{a_1\in\mathcal{A}} \left(\widehat{A}_1(\tau_1, a_1) + \widehat{V}_1(\tau_1)\right), \ldots, \arg\max_{a_n\in\mathcal{A}} \left(\widehat{A}_n(\tau_n, a_n) + \widehat{V}_n(\tau_n)\right)\right) \tag{26}$$

$$\Rightarrow \arg\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}} \widehat{Q}_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = \left(\arg\max_{a_1\in\mathcal{A}} \widehat{Q}_1(\tau_1, a_1), \ldots, \arg\max_{a_n\in\mathcal{A}} \widehat{Q}_n(\tau_n, a_n)\right) \tag{27}$$

$$\Rightarrow \arg\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}} \widetilde{Q}_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = \left(\arg\max_{a_1\in\mathcal{A}} \widetilde{Q}_1(\tau_1, a_1), \ldots, \arg\max_{a_n\in\mathcal{A}} \widetilde{Q}_n(\tau_n, a_n)\right). \tag{28}$$

$$\tag{29}$$

Thus, $\left(\widetilde{Q}_{tot}, \left[\widetilde{Q}_i\right]_{i=1}^n\right) \in \widetilde{\mathcal{Q}}$, which means that $\widehat{\mathcal{Q}} \subseteq \widetilde{\mathcal{Q}}$. The action-value function classes derived from advantage-based IGM and IGM are equivalent. □

Fact 1. *The constraint of advantage-based IGM stated in Eq.* (6) *is equivalent to that when* $\forall \boldsymbol{\tau} \in \boldsymbol{\mathcal{T}}$, $\forall \boldsymbol{a^*} \in \boldsymbol{\mathcal{A}}^*(\boldsymbol{\tau})$, $\forall \boldsymbol{a} \in \boldsymbol{\mathcal{A}} \setminus \boldsymbol{\mathcal{A}}^*(\boldsymbol{\tau})$, $\forall i \in \mathcal{N}$,

$$A_{tot}(\boldsymbol{\tau}, \boldsymbol{a^*}) = A_i(\tau_i, a_i^*) = 0 \quad \textit{and} \quad A_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) < 0, A_i(\tau_i, a_i) \leq 0, \tag{7}$$

where $\boldsymbol{\mathcal{A}}^*(\boldsymbol{\tau}) = \{\boldsymbol{a} | \boldsymbol{a} \in \boldsymbol{\mathcal{A}}, Q_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = V_{tot}(\boldsymbol{\tau})\}$.

Proof. We derive that $\forall \boldsymbol{\tau} \in \boldsymbol{\mathcal{T}}$, $\forall \boldsymbol{a} \in \boldsymbol{\mathcal{A}}$, $\forall i \in \mathcal{N}$, $\widehat{A}_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) \leq 0$ and $\widehat{A}_i(\tau_i, a_i) \leq 0$ from Eq. (4) and Eq. (5) of Definition 1, respectively. According to the definition of $\arg\max$ operator, Eq. (4), and Eq. (5), $\forall \boldsymbol{\tau} \in \boldsymbol{\mathcal{T}}$, let $\widehat{\boldsymbol{\mathcal{A}}}^*(\boldsymbol{\tau})$ denote $\arg\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}} \widehat{A}_{tot}(\boldsymbol{\tau}, \boldsymbol{a})$ as follows:

$$\widehat{\boldsymbol{\mathcal{A}}}^*(\boldsymbol{\tau}) = \arg\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}} \widehat{A}_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = \arg\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}} \widehat{Q}_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) \tag{30}$$

$$= \left\{\boldsymbol{a} | \boldsymbol{a} \in \boldsymbol{\mathcal{A}}, \widehat{Q}_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = \widehat{V}_{tot}(\boldsymbol{\tau})\right\} \tag{31}$$

$$= \left\{\boldsymbol{a} | \boldsymbol{a} \in \boldsymbol{\mathcal{A}}, \widehat{Q}_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) - \widehat{V}_{tot}(\boldsymbol{\tau}) = 0\right\} \tag{32}$$

$$= \left\{\boldsymbol{a} | \boldsymbol{a} \in \boldsymbol{\mathcal{A}}, \widehat{A}_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = 0\right\}. \tag{33}$$

Similarly, $\forall \boldsymbol{\tau} \in \boldsymbol{\mathcal{T}}$, $\forall i \in \mathcal{N}$, let $\widehat{\mathcal{A}}_i^*(\tau_i)$ denote $\arg\max_{a_i\in\mathcal{A}} \widehat{A}_i(\tau_i, a_i)$ as follows:

$$\widehat{\mathcal{A}}_i^*(\tau_i) = \arg\max_{a_i\in\mathcal{A}} \widehat{A}_i(\tau_i, a_i) = \arg\max_{a_i\in\mathcal{A}} \widehat{Q}_i(\tau_i, a_i) \tag{34}$$

$$= \left\{a_i | a_i \in \mathcal{A}, \widehat{Q}_i(\tau_i, a_i) = \widehat{V}_i(\tau_i)\right\} \tag{35}$$

$$= \left\{a_i | a_i \in \mathcal{A}, \widehat{A}_i(\tau_i, a_i) = 0\right\}. \tag{36}$$

Thus, $\forall \boldsymbol{\tau} \in \boldsymbol{\mathcal{T}}$, $\forall \boldsymbol{a^*} \in \widehat{\boldsymbol{\mathcal{A}}}^*(\boldsymbol{\tau})$, $\forall \boldsymbol{a} \in \boldsymbol{\mathcal{A}} \setminus \widehat{\boldsymbol{\mathcal{A}}}^*(\boldsymbol{\tau})$,

$$\widehat{A}_{tot}(\boldsymbol{\tau}, \boldsymbol{a^*}) = 0 \quad \text{and} \quad \widehat{A}_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) < 0; \tag{37}$$

$\forall \boldsymbol{\tau} \in \boldsymbol{\mathcal{T}}$, $\forall i \in \mathcal{N}$, $\forall a_i^* \in \widehat{\mathcal{A}}^*(\tau_i)$, $\forall a_i \in \mathcal{A} \setminus \widehat{\mathcal{A}}^*(\tau_i)$,

$$\widehat{A}_i(\tau_i, a_i^*) = 0 \quad \text{and} \quad \widehat{A}_i(\tau_i, a_i) < 0. \tag{38}$$

Recall the constraint stated in Eq. 6, $\forall \boldsymbol{\tau} \in \boldsymbol{\mathcal{T}}$,

$$\arg\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}} \widehat{A}_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = \left(\arg\max_{a_1\in\mathcal{A}} \widehat{A}_1(\tau_1, a_1), \ldots, \arg\max_{a_n\in\mathcal{A}} \widehat{A}_n(\tau_n, a_n)\right). \tag{39}$$

We can rewrite the constraint of advantage-based IGM stated in Eq. (6) as $\forall \boldsymbol{\tau} \in \boldsymbol{\mathcal{T}}$,

$$\widehat{\boldsymbol{\mathcal{A}}}^*(\boldsymbol{\tau}) = \left\{ \langle a_1, \ldots, a_n \rangle \middle| a_i \in \widehat{\mathcal{A}}_i^*(\tau_i), \forall i \in \mathcal{N} \right\}. \tag{40}$$

Therefore, combining Eq. (37), Eq. (38), and Eq. (40), we can derive $\forall \boldsymbol{\tau} \in \boldsymbol{\mathcal{T}}$, $\forall \boldsymbol{a^*} \in \widehat{\boldsymbol{\mathcal{A}}}^*(\boldsymbol{\tau})$, $\forall \boldsymbol{a} \in \boldsymbol{\mathcal{A}} \setminus \widehat{\boldsymbol{\mathcal{A}}}^*(\boldsymbol{\tau}), \forall i \in \mathcal{N}$,

$$\widehat{A}_{tot}(\boldsymbol{\tau}, \boldsymbol{a^*}) = \widehat{A}_i(\tau_i, a_i^*) = 0 \quad \text{and} \quad \widehat{A}_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) < 0, \widehat{A}_i(\tau_i, a_i) \leq 0. \tag{41}$$

In another way, combining Eq. (37), Eq. (38), and Eq. (41), we can derive Eq. (40) by the definition of $\widehat{\boldsymbol{\mathcal{A}}}^*$ and $\left[\widehat{\mathcal{A}}^*\right]_{i=1}^n$ (see Eq. (33) and Eq. (36)). In more detail, the closed set property of Cartesian product of $[a_i^*]_{i=1}^n$ has been encoded into the Eq. (40) and Eq. (41) simultaneously. □

Proposition 2. *Assume neural networks provide universal function approximation, the joint action-value function class that QPLEX can realize is equivalent to what is induced by the IGM principle.*

Proof. We assume that the neural network of QPLEX can be large enough to achieve the universal function approximation by corresponding theorem [20]. Let the action-value function class that QPLEX can realize is denoted by

$$\overline{\mathcal{Q}} = \left\{ \left(\overline{Q}_{tot} \in \mathbb{R}^{|\boldsymbol{\mathcal{T}}||\mathcal{A}|^n}, \left[\overline{Q}_i \in \mathbb{R}^{|\boldsymbol{\mathcal{T}}||\mathcal{A}|} \right]_{i=1}^n \right) \middle| \text{ Eq. (8), (9), (10), (12) are satisfied} \right\}. \tag{42}$$

In addition, $\overline{Q}_{tot}$, $\overline{V}_{tot}$, $\overline{A}_{tot}$, $\left[\overline{Q}_i'\right]_{i=1}^n$, $\left[\overline{V}_i'\right]_{i=1}^n$, $\left[\overline{A}_i'\right]_{i=1}^n$, $\left[\overline{Q}_i\right]_{i=1}^n$, $\left[\overline{V}_i\right]_{i=1}^n$, and $\left[\overline{A}_i\right]_{i=1}^n$ denote the corresponding (joint, transformed, and individual) (action-value, state-value, and advantage) functions, respectively. In the implmentation of QPLEX, we ensure the positivity of important weights of *Transformation* and joint advantage function, $[w_i]_{i=1}^n$ and $[\lambda_i]_{i=1}^n$, which maintains the greedy action selection flow and rules out these non-interesting points (zeros) on optimization. We will prove $\widehat{\mathcal{Q}} \equiv \overline{\mathcal{Q}}$ in the following two directions.

$\widehat{\mathcal{Q}} \subseteq \overline{\mathcal{Q}}$ For any $\left(\widehat{Q}_{tot}, \left[\widehat{Q}_i\right]_{i=1}^n\right) \in \widehat{\mathcal{Q}}$, we construct $\overline{Q}_{tot} = \widehat{Q}_{tot}$ and $\left[\overline{Q}_i\right]_{i=1}^n = \left[\widehat{Q}_i\right]_{i=1}^n$ and derive $\overline{V}_{tot}$, $\overline{A}_{tot}$, $\left[\overline{V}_i\right]_{i=1}^n$, and $\left[\overline{A}_i\right]_{i=1}^n$ by Eq.(4) and Eq. (5), respectively. In addition, we construct transformed functions connecting joint and individual functions as follows: $\forall \boldsymbol{\tau} \in \boldsymbol{\mathcal{T}}$, $\forall \boldsymbol{a} \in \boldsymbol{\mathcal{A}}$, $\forall i \in \mathcal{N}$,

$$\overline{Q}_i'(\boldsymbol{\tau}, \boldsymbol{a}) = \frac{\overline{Q}_{tot}(\boldsymbol{\tau}, \boldsymbol{a})}{n}, \ \overline{V}_i'(\boldsymbol{\tau}) = \arg\max_{\boldsymbol{a} \in \boldsymbol{\mathcal{A}}} \overline{Q}_i'(\boldsymbol{\tau}, \boldsymbol{a}), \text{ and } \overline{A}_i'(\boldsymbol{\tau}, \boldsymbol{a}) = \overline{Q}_i'(\boldsymbol{\tau}, \boldsymbol{a}) - \overline{V}_i'(\boldsymbol{\tau}), \tag{43}$$

which means that according to Fact 1,

$$w_i(\boldsymbol{\tau}) = 1, \ b_i(\boldsymbol{\tau}) = \overline{V}_i'(\boldsymbol{\tau}) - \overline{V}_i(\tau_i), \text{ and } \lambda_i(\boldsymbol{\tau}, \boldsymbol{a}) = \begin{cases} \dfrac{\overline{A}_i'(\boldsymbol{\tau}, \boldsymbol{a})}{\overline{A}_i(\tau_i, a_i)} > 0, & \text{when } \overline{A}_i(\tau_i, a_i) < 0, \\ 1, & \text{when } \overline{A}_i(\tau_i, a_i) = 0. \end{cases} \tag{44}$$

Thus, $\left(\overline{Q}_{tot}, \left[\overline{Q}_i\right]_{i=1}^n\right) \in \overline{\mathcal{Q}}$, which means that $\widehat{\mathcal{Q}} \subseteq \overline{\mathcal{Q}}$.

$\overline{\mathcal{Q}} \subseteq \widehat{\mathcal{Q}}$ For any $\left(\overline{Q}_{tot}, \left[\overline{Q}_i\right]_{i=1}^n\right) \in \overline{\mathcal{Q}}$, with the similar discussion of Fact 1, $\forall \boldsymbol{\tau} \in \boldsymbol{\mathcal{T}}$, $\forall i \in \mathcal{N}$, let $\overline{\mathcal{A}}_i^*(\tau_i)$ denote $\arg\max_{a_i \in \mathcal{A}} \overline{A}_i(\tau_i, a_i)$, where

$$\overline{\mathcal{A}}_i^*(\tau_i) = \left\{ a_i | a_i \in \mathcal{A}, \overline{A}_i(\tau_i, a_i) = 0 \right\}. \tag{45}$$

Combining the positivity of $[w_i]_{i=1}^n$ and $[\lambda_i]_{i=1}^n$ with Eq. (8),(9),(10),(12), we can derive $\forall \boldsymbol{\tau} \in \boldsymbol{\mathcal{T}}$, $\forall i \in \mathcal{N}$, $\forall a_i^* \in \overline{\mathcal{A}}^*(\tau_i)$, $\forall a_i \in \mathcal{A} \setminus \overline{\mathcal{A}}^*(\tau_i)$,

$$\overline{A}_i(\tau_i, a_i^*) = 0 \quad \text{and} \quad \overline{A}_i(\tau_i, a_i) < 0 \tag{46}$$

$$\Rightarrow \quad \overline{A}_i'(\boldsymbol{\tau}, a_i^*) = w_i(\boldsymbol{\tau}) \overline{A}_i(\tau_i, a_i^*) = 0 \quad \text{and} \quad \overline{A}_i'(\boldsymbol{\tau}, a_i) = w_i(\boldsymbol{\tau}) \overline{A}_i(\tau_i, a_i) < 0 \tag{47}$$

$$\Rightarrow \quad \overline{A}_{tot}(\boldsymbol{\tau}, \boldsymbol{a}^*) = \lambda_i(\boldsymbol{\tau}, \boldsymbol{a}^*) \overline{A}_i'(\tau_i, a_i^*) = 0 \quad \text{and} \quad \overline{A}_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = \lambda_i(\boldsymbol{\tau}, \boldsymbol{a}) \overline{A}_i'(\tau_i, a_i^*) < 0, \tag{48}$$

QPLEX's architecuture configurations	Didactic Examples	StarCraft II
The number of layers in $w, b, \lambda, \phi, \upsilon$	2	1
Unit number in hidden layer of $w, b, \lambda, \phi, \upsilon$	64	$\varnothing$
Activation after hidden layer 1 of w, υ	Relu	Absolute
Activation after hidden layer 2 of w, υ	Absolute	$\varnothing$
Activation after hidden layer 1 of b	Relu	None
Activation after hidden layer 2 of b	None	$\varnothing$
Activation after hidden layer 1 of λ, ϕ	Relu	Sigmoid
Activation after hidden layer 2 of λ, ϕ	Sigmoid	$\varnothing$

Table 2: The network configurations of QPLEX's architecture.

where $\boldsymbol{a}^* = \langle a_1^*, \ldots, a_n^* \rangle$ and $\boldsymbol{a} = \langle a_1, \ldots, a_n \rangle$. Notably, these $\boldsymbol{a}^*$ forms

$$\overline{\boldsymbol{\mathcal{A}}}^*(\boldsymbol{\tau}) = \left\{ \langle a_1, \ldots, a_n \rangle \middle| a_i \in \overline{\mathcal{A}}_i^*(\tau_i), \forall i \in \mathcal{N} \right\} \tag{49}$$

which is similar with Eq. (40) in the proof of Fact 1. We construct $\widehat{Q}_{tot} = \overline{Q}_{tot}$ and $\left[\widehat{Q}_i\right]_{i=1}^n = \left[\overline{Q}_i\right]_{i=1}^n$. According to Eq. (49), the constraints of advantage-based IGM stated in Fact 1 (Eq. (4), Eq. (5), and Eq. (7)) are satisfied, which means that $\left(\widehat{Q}_{tot}, \left[\widehat{Q}_i\right]_{i=1}^n\right) \in \widehat{\mathcal{Q}}$ and $\overline{\mathcal{Q}} \subseteq \widehat{\mathcal{Q}}$.

Thus, when assuming neural networks provide universal function approximation, the joint action-value function class that QPLEX can realize is equivalent to what is induced by the IGM principle. □

B Implementation Details and Experiment Settings

B.1 Implmentation Details

We adopt the PyMARL [10] implementation of five state-of-the-art baselines: QTRAN [8], Qatten [9], QMIX [7], VDN [6], and IQL [21]. The hyper-parameters of these algorithms are the same as that in SMAC [10] and referred in their source codes. QPLEX is also based on PyMARL, whose special hyper-parameters are illustrated in Table 2 and other common hyper-parameters are adopted by the default implementation of PyMARL [10]. Especially, in the online task settings, we take the advanced implementation of *Transmaformation* of Qatten in QPLEX. To ensure the positivity of important weights of *Transformation* and joint advantage function, we add a sufficient small amount $\epsilon' = 10^{-10}$ on $[w_i]_{i=1}^n$ and $[\lambda_i]_{i=1}^n$. In addition, we stop gradients of local advantage function A_i to increase the optimization stability of the max operator of dueling structure. This instability consideration about max operator has been justified by Dueling DQN [13]. We approximate the joint action-value function as

$$Q_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) \approx \sum_{i=1}^n Q_i(\boldsymbol{\tau}, a_i) + \sum_{i=1}^n \left(\lambda_i(\boldsymbol{\tau}, \boldsymbol{a}) - 1\right) \widetilde{A}_i(\boldsymbol{\tau}, a_i), \tag{50}$$

where $\widetilde{A}_i$ denotes a variant of the local advantage function A_i by stoping gradients.

Our training time on an NVIDIA RTX 2080TI GPU of each task is about 6 hours to 20 hours, depending on the agent number and the episode length limit of each map. The percentage of episodes in which RL agents defeat all enemy units within the time limit is called *test win rate*.

Training with Offline Data Collection To construct a diverse dataset, we train a behaviour policy of QMIX [7] or VDN [6] and collect its 20k or 50k experienced episodes throughout the training process. The dataset configurations are shown in Table 3. We evaluate QPLEX and five baselines over 6 random seeds, which includes 3 different datasets and tests two seeds on each dataset. We train 300 epochs to demonstrate our learning performance, where each epoch trains 160k transitions with a batch of 32 episodes. Moreover, the training process of behaviour policy is the same as that discussed in PyMARL [10].

Map Name	Replay Buffer Size	Behaviour Test Win Rate	Behaviour Policy
2s3z	20k episodes	95.8%	QMIX
3s5z	20k episodes	92.0%	QMIX
1c3s5z	20k episodes	90.2%	QMIX
2s_vs_1sc	20k episodes	98.1%	QMIX
3s_vs_5z	20k episodes	94.4%	VDN
2c_vs_64zg	50k episodes	80.9%	QMIX

Table 3: The dataset configurations of offline data collection setting.

Training with Online Data Collection We have collected a total of 2 million timestep data for each task and test the model every 10 thousand steps. We use ϵ-greedy exploration and a limited first-in first-out (FIFO) replay buffer of size 5000 episodes, where ϵ is linearly annealed from 1.0 to 0.05 over 50k timesteps and keep it constant for the rest training process. During training, we perform gradient update twice with a batch of 32 episodes after collecting every episode.

B.2 StarCraft II

We consider the combat scenario of StarCraft II unit micromanagement tasks, where the enemy units are controlled by the built-in AI, and each ally unit is controlled by the reinforcement learning agent. The units of the two groups can be asymmetric but the units of each group should belong to the same race. At each timestep, every agent takes an action from the discrete action space, which includes the following actions: noop, move [direction], attack [enemy id], and stop. Under the control of these actions, agents move and attack in continuous maps. At each time step, MARL agents will get a global reward equal to the total damage done to enemy units. Killing each enemy unit and winning the combat will bring additional bonuses of 10 and 200, respectively. We briefly introduce the SMAC challenges of our paper in Table 4.

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
1c3s8z_vs_1c3s9z	1 Colossus, 3 Stalkers & 8 Zealots	1 Colossus, 3 Stalkers & 9 Zealots
7sz	7 Stalkers & 7 Zealots	7 Stalkers & 7 Zealots
5s10z	5 Stalkers & 10 Zealots	5 Stalkers & 10 Zealots
2s_vs_1sc	2 Stalkers	1 Spine Crawler
3s_vs_5z	3 Stalkers	5 Zealots
2c_vs_64zg	2 Colossi	64 Zerglings

Table 4: SMAC challenges.

C Deferred Figures and Tables in Section 4

C.1 Deferred Tables in Section 4.1

a_2 \ a_1	$\mathcal{A}^{(1)}$	$\mathcal{A}^{(2)}$	$\mathcal{A}^{(3)}$
$\mathcal{A}^{(1)}$	-6.5	-4.9	-4.9
$\mathcal{A}^{(2)}$	-5.0	-3.5	**-3.4**
$\mathcal{A}^{(3)}$	-5.0	-3.5	-3.5

(a) Q_{tot} of Qatten.

a_2 \ a_1	$\mathcal{A}^{(1)}$	$\mathcal{A}^{(2)}$	$\mathcal{A}^{(3)}$
$\mathcal{A}^{(1)}$	-6.5	-5.0	-5.0
$\mathcal{A}^{(2)}$	-5.0	-3.5	-3.5
$\mathcal{A}^{(3)}$	-5.0	**-3.5**	-3.5

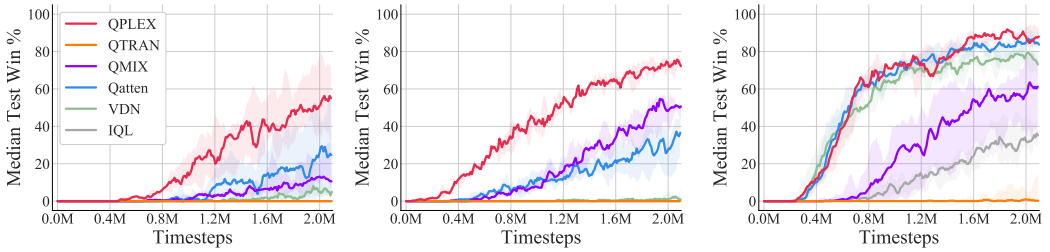
(b) Q_{tot} of VDN.

a_2 \ a_1	$\mathcal{A}^{(1)}$	$\mathcal{A}^{(2)}$	$\mathcal{A}^{(3)}$
$\mathcal{A}^{(1)}$	-5.3	-4.6	-4.7
$\mathcal{A}^{(2)}$	-4.7	-4.0	-4.0
$\mathcal{A}^{(3)}$	-4.7	**-4.0**	-4.0

(c) Averaged individual Q of IQL.

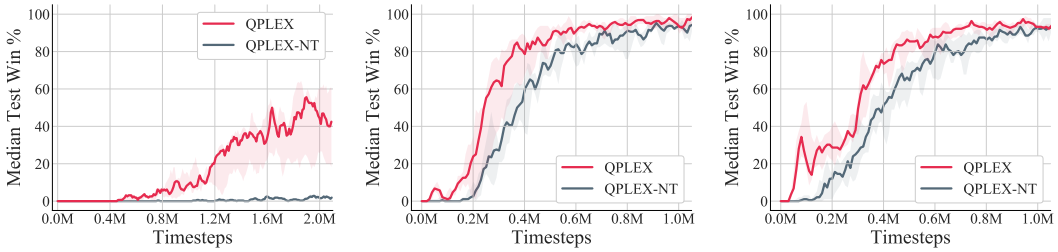
Table 5: Deferred joint action-value functions Q_{tot} or averaged individual Q of Qatten, VDN, and IQL. Boldface means greedy joint action selection from individual or joint action-value functions.

C.2 Deferred Figures in Section 4.2

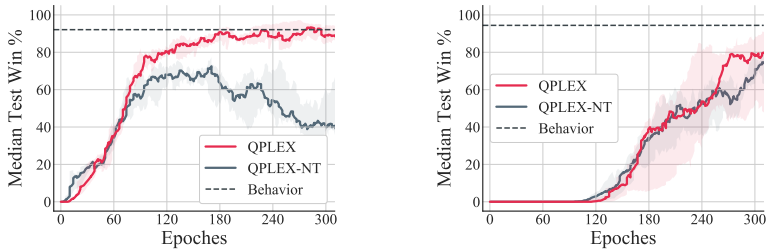


(a) 1c3s8z_vs_1c3s9z (b) 7sz (c) 3s_vs_5z

Figure 6: Deferred learning curve of StarCraft II with online data collection on other three different maps.



(a) Online data collection of 5s10z (b) Online data collection of 1c3s5z (c) Online data collection of 3s5z



(d) Offline data collection of 3s5z (e) Offline data collection of 3s_vs_5z

Figure 7: Deferred figures of win rates for QPLEX and its ablation QPLEX-NT in both online and offline data collection on three different maps.