Multi-Agent Determinantal Q-Learning

Yaodong Yang [*1 2] **Ying Wen** [*1 2] **Liheng Chen** [3] **Jun Wang** [2] **Kun Shao** [1] **David Mguni** [1] **Weinan Zhang** [3]

Abstract

Centralized training with decentralized execution has become an important paradigm in multi-agent learning. Though practical, current methods rely on restrictive assumptions to decompose the centralized value function across agents for execution. In this paper, we eliminate this restriction by proposing multi-agent determinantal Q-learning. Our method is established on Q-DPP, an extension of determinantal point process (DPP) with partition-matroid constraint to multi-agent setting. Q-DPP promotes agents to acquire diverse behavioral models; this allows a natural factorization of the joint Q-functions with no need for *a priori* structural constraints on the value function or special network architectures. We demonstrate that Q-DPP generalizes major solutions including VDN, QMIX, and QTRAN on decentralizable cooperative tasks. To efficiently draw samples from Q-DPP, we adopt an existing linear-time sampler with theoretical approximation guarantee. The sampler also benefits exploration by coordinating agents to cover orthogonal directions in the state space during multi-agent training. We evaluate our algorithm on various cooperative benchmarks; its effectiveness has been demonstrated when compared with the state-of-the-art.

1. Introduction

Multi-agent reinforcement learning (MARL) methods hold great potential to solve a variety of real-world problems, such as mastering multi-player video games (Peng et al., 2017), dispatching taxi orders (Li et al., 2019), and studying population dynamics (Yang et al., 2018). In this work, we consider the multi-agent cooperative setting (Panait & Luke, 2005) where a team of agents collaborate to achieve one common goal in a partially observed environment.

[*]Equal contribution [1]Huawei Technology R&D UK. [2]University College London. [3]Shanghai Jiaotong University. Correspondence to: Yaodong Yang <yaodong.yang@huawei.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

A full spectrum of MARL algorithms has been developed to solve cooperative tasks (Panait & Luke, 2005); the two endpoints of the spectrum are independent and centralized learning (see Fig. 1). Independent learning (IL) (Tan, 1993) merely treats other agents' influence to the system as part of the environment. The learning agent not only faces a non-stationary environment, but also suffers from *spurious* rewards (Sunehag et al., 2017; Du et al., 2019). Centralized learning (CL), at the other end, treats a multi-agent problem as a single-agent problem despite the fact that many real-world applications require local autonomy. Importantly, the CL approaches exhibit combinatorial complexity and can hardly scale to more than tens of agents (Yang et al., 2019).

Another paradigm typically considered is a hybrid of centralized training and decentralized execution (CTDE) (Oliehoek et al., 2008). For value-based approaches in the framework of CTDE, a fundamental challenge is how to correctly decompose the centralized value function among agents for decentralized execution. For a cooperative task to be deemed decentralizable, it is required that local maxima on the value function per every agent should amount to the global maximum on the joint value function. In enforcing such a condition, current state-of-the-art methods rely on restrictive structural constraints and/or network architectures. For instance, Value Decomposition Network (VDN) (Sunehag et al., 2017) and Factorized-Q (Zhou et al., 2019) propose to directly factorize the joint value function into a summation of individual value functions. QMIX (Rashid et al., 2018) augments the summation to be non-linear aggregations, while maintaining a monotonic relationship between centralized and individual value functions. QTRAN (Son et al., 2019) introduces a refined learning objective on top of QMIX along with specific network designs.

Unsurprisingly, the structural constraints put forward by VDN / QMIX / QTRAN inhibit the representational power of the centralized value function (Son et al., 2019); as a result, the class of decentralizable cooperative tasks these methods can tackle is limited. For example, poor empirical results of QTRAN have been reported on multiple multi-agent cooperative benchmarks (Mahajan et al., 2019).

Apart from the aforementioned problems, structural constraints also hinder efficient explorations when applied to value function decomposition. In fact, since agents are

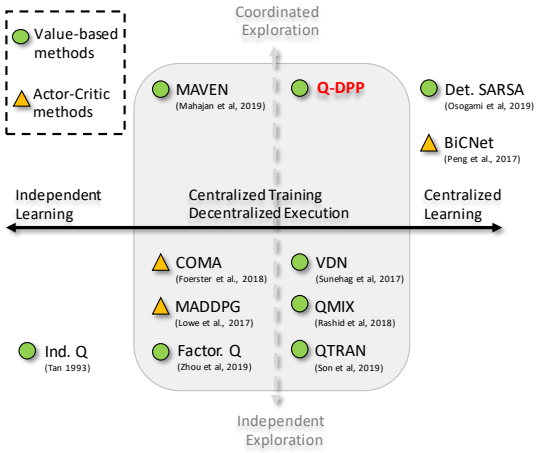


Figure 1: Spectrum of MARL methods on cooperative tasks.

treated independently during the execution stage, CTDE methods inevitably lack a principled exploration strategy (Matignon et al., 2007). Clearly, an increasing per-agent exploration rate of ϵ-greedy in the single-agent setting can help exploration; however, it has been proved (Mahajan et al., 2019) that due to the structural constraints (e.g. the monotonicity assumption in QMIX), in the multi-agent setting, increasing ϵ will only lower the probability of obtaining the optimal value function. As a treatment, MAVEN (Mahajan et al., 2019) introduces a hierarchical model to coordinate diverse explorations among agents. Yet, a principled exploration strategy with minor structural constraints on the value function is still missing for value-based CTDE methods.

To eliminate restrictive constraints on the value function decomposition, one reasonable solution is to make agents acquire a **diverse** set of behavioral models during training so that the optimal action of one agent does not depend on the actions of the other agents. In such scenario, the equivalence between the local maxima on the per-agent value function and the global maximum on the joint value function can be automatically achieved. As a result, the centralized value function can enjoy a natural factorization with no need for any structural constraints beforehand.

In this paper, we present a new value-based solution in the CTDE paradigm to multi-agent cooperative tasks. We establish Q-DPP, an extension of determinantal point process (DPP) (Macchi, 1977) with partition constraint, and apply it to multi-agent learning. DPPs are elegant probabilistic models on sets that can capture both quality and diversity when a subset is sampled from a ground set; this makes them ideal for modeling the set that contains different agents' observation-action pairs in the multi-agent learning context. We adopt Q-DPP as a function approximator for the centralized value function. An attractive property of using Q-DPP is that, when reaching the optimum, it can offer a natural factorization on the centralized value function, assuming agents have acquired a diverse set of behaviors. Our method eliminates the need for *a priori* structural constraints or bespoke neural architectures. In fact, we demonstrate that Q-DPP generalizes current solvers including VDN, QMIX, and QTRAN, where all these methods can be derived as special cases from Q-DPP. As an additional contribution, we adopt a tractable sampler, based on the idea of sample-by-projection in P-DPP (Celis et al., 2018), for Q-DPP with theoretical approximation guarantee. Our sampler makes agents explore in a sequential manner; agents who act later are coordinated to visit only the orthogonal areas in the state space that previous agents haven't explored. Such coordinated way of explorations effectively boosts the sampling efficiency in the CTDE setting. Building upon these advantages, finally, we demonstrate that our proposed Q-DPP algorithm is superior to the existing state-of-the-art solutions on a variety of multi-agent cooperation benchmarks.

2. Preliminaries: Determinantal Point Process

DPP is a probabilistic framework that characterizes how likely a subset is going to be sampled from a ground set. Originated from quantum physics for modeling repulsive Fermion particles (Macchi, 1977), DPP has recently been introduced to the machine learning community due to its probabilistic nature (Kulesza et al., 2012).

Definition 1 (DPP)**.** *For a ground set of items $\mathcal{Y} = \{1, 2, \ldots, M\}$, a DPP, denoted by $\mathbb{P}$, is a probability measure on the set of all subsets of $\mathcal{Y}$, i.e., $2^{\mathcal{Y}}$. Given an $M \times M$ positive semi-definite (PSD) kernel $\boldsymbol{\mathcal{L}}$ that measures similarity for any pairs of items in $\mathcal{Y}$, let $\boldsymbol{Y}$ be a random subset drawn according to $\mathbb{P}$, then we have, $\forall Y \subseteq \mathcal{Y}$,*

$$\mathbb{P}_{\boldsymbol{\mathcal{L}}}\big(\boldsymbol{Y} = Y\big) \propto \det\big(\boldsymbol{\mathcal{L}}_Y\big) = \mathrm{Vol}^2\big(\{\boldsymbol{w}_i\}_{i \in Y}\big), \quad (1)$$

where $\boldsymbol{\mathcal{L}}_Y := [\boldsymbol{\mathcal{L}}_{i,j}]_{i,j \in Y}$ denotes the submatrix of $\boldsymbol{\mathcal{L}}$ whose entries are indexed by the items included in Y. If we write $\boldsymbol{\mathcal{L}} := \boldsymbol{\mathcal{W}}\boldsymbol{\mathcal{W}}^\top$ with $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{M \times P}$, $P \leq M$, and rows of $\boldsymbol{\mathcal{W}}$ being $\{\boldsymbol{w}_i\}$, then the determinant value is essentially the squared $|Y|$-dimensional volume of parallelepiped spanned by the rows of $\boldsymbol{\mathcal{W}}$ corresponding to elements in Y.

A PSD matrix ensures all principal minors of $\boldsymbol{\mathcal{L}}$ are non-negative $\det(\boldsymbol{\mathcal{L}}_Y) \geq 0$; it thus suffices to be a proper probability distribution. The normalizer can be computed as: $\sum_{Y \subseteq \mathcal{Y}} \det(\boldsymbol{\mathcal{L}}_Y) = \det(\boldsymbol{\mathcal{L}} + \boldsymbol{I})$, where $\boldsymbol{I}$ is an $M \times M$ identity matrix. Intuitively, one can think of a diagonal entry $\boldsymbol{\mathcal{L}}_{i,i}$ as capturing the quality of item i, while an off-diagonal entry $\boldsymbol{\mathcal{L}}_{i,j}$ measures the similarity between items i and j. DPP models the **repulsive** connections among **multiple** items in a sampled subset. In the example of two items, $\mathbb{P}_{\boldsymbol{\mathcal{L}}}(\{i,j\}) \propto \begin{vmatrix} \boldsymbol{\mathcal{L}}_{i,i} & \boldsymbol{\mathcal{L}}_{i,j} \\ \boldsymbol{\mathcal{L}}_{j,i} & \boldsymbol{\mathcal{L}}_{j,j} \end{vmatrix} = \boldsymbol{\mathcal{L}}_{i,i}\boldsymbol{\mathcal{L}}_{j,j} - \boldsymbol{\mathcal{L}}_{i,j}\boldsymbol{\mathcal{L}}_{j,i}$, which suggests, if item i and item j are perfectly similar, such that $\boldsymbol{\mathcal{L}}_{i,j} = \sqrt{\boldsymbol{\mathcal{L}}_{i,i}\boldsymbol{\mathcal{L}}_{j,j}}$, then we know these two items will almost surely not co-occur, hence such two-item subset of $\{i,j\}$ from the ground set will never be sampled.

DPPs are attractive in that they only require training the kernel matrix $\mathcal{L}$, which can be learned via maximum likelihood (Affandi et al., 2014). A trainable DPP favors many supervised learning tasks where diversified outcomes are desired, such as image generation (Elfeki et al., 2019), video summarization (Sharghi et al., 2018), model ensemble (Pang et al., 2019), and recommender system (Chen et al., 2018). It is, however, non-trivial to adapt DPPs to a multi-agent setting since additional restrictions are required to put on the ground set so that valid samples can be drawn for the purpose of multi-agent training. This leads to our Q-DPPs.

3. Multi-Agent Determinantal Q-Learning

We offer a new value-based solution to multi-agent cooperative tasks. In particular, we introduce Q-DPPs as general function approximators for the centralized value functions, similar to neural networks in deep Q-learning (Mnih et al., 2015). We start from the problem formulation.

3.1. Problem Formulation of Multi-Agent Cooperation

Multi-agent cooperation in a partially-observed environment is usually modeled as a Dec-POMDP (Oliehoek et al., 2016) denoted by a tuple $\mathcal{G} =< \mathcal{S}, \mathcal{N}, \mathcal{A}, \mathcal{O}, \mathcal{Z}, \mathcal{P}, \mathcal{R}, \gamma >$. Within $\mathcal{G}$, $s \in \mathcal{S}$ denotes the global environmental state. At every time-step $t \in \mathbb{Z}^+$, each agent $i \in \mathcal{N} = \{1, \dots, N\}$ selects an action $a_i \in \mathcal{A}$ where a joint action stands for $\boldsymbol{a} := (a_i)_{i\in\mathcal{N}} \in \mathcal{A}^N$. Since the environment is partially observed, each agent only has access to its local observation $o \in \mathcal{O}$ that is acquired through an observation function $\mathcal{Z}(s, a) : \mathcal{S} \times \mathcal{A} \to \mathcal{O}$. The state transition dynamics are determined by $\mathcal{P}(s'|s, \boldsymbol{a}) := \mathcal{S} \times \mathcal{A}^N \times \mathcal{S} \to [0, 1]$. Agents optimize towards one shared goal whose performance is measured by $\mathcal{R}(s, \boldsymbol{a}) : \mathcal{S} \times \mathcal{A}^N \to \mathbb{R}$, and $\gamma \in [0, 1)$ discounts the future rewards. Each agent recalls an observation-action history $\tau_i \in \mathcal{T} := (\mathcal{O} \times \mathcal{A})^t$, and executes a stochastic policy $\pi_i(a_i|\tau_i) : \mathcal{T} \times \mathcal{A} \to [0, 1]$ which is conditioned on τ_i. All of the agents histories is defined as $\boldsymbol{\tau} := (\tau_i)_{i\in\mathcal{N}} \in \mathcal{T}^N$. Given a joint policy $\boldsymbol{\pi} := (\pi_i)_{i\in\mathcal{N}}$, the joint action-value function at time t stands as $Q^{\boldsymbol{\pi}}(\boldsymbol{\tau}^t, \boldsymbol{a}^t) = \mathbb{E}_{s^{t+1:\infty}, \boldsymbol{a}^{t+1:\infty}}[G^t|\boldsymbol{\tau}^t, \boldsymbol{a}^t]$, where $G^t = \sum_{i=0}^{\infty} \gamma^i \mathcal{R}^{t+i}$ is the total accumulative rewards.

The **goal** is to find an optimal value function $Q^* = \max_{\boldsymbol{\pi}} Q^{\boldsymbol{\pi}}(\boldsymbol{\tau}^t, \boldsymbol{a}^t)$ and the corresponding policy $\boldsymbol{\pi}^*$. A direct centralized approach is to learn the joint value function, parameterized by θ, by minimizing the squared temporal-difference error $\mathcal{L}(\theta)$ (Watkins & Dayan, 1992) from a sampled mini-batch of transition data $\{\langle \boldsymbol{\tau}, \boldsymbol{a}, \mathcal{R}, \boldsymbol{\tau}' \rangle\}_{j=1}^{E}$, i.e.,

$$\mathcal{L}(\theta) = \sum_{j=1}^{E} \left\| \mathcal{R} + \gamma \max_{\boldsymbol{a}'} Q(\boldsymbol{\tau}', \boldsymbol{a}'; \theta^-) - Q^{\boldsymbol{\pi}}(\boldsymbol{\tau}, \boldsymbol{a}; \theta) \right\|^2, \quad (2)$$

where θ^- denotes the target parameters that can be periodically copied from θ during training.

In our work, apart from the joint value function, we also focus on obtaining a decentralized policy for each agent. CTDE is a paradigm for solving Dec-POMDP (Oliehoek et al., 2008) where it allows the algorithm access to all of the agents local histories $\boldsymbol{\tau}$ during training. During testing, however, the algorithm uses each of the agent's own history τ_i for execution. CTDE methods provide valid solutions to multi-agent cooperative tasks that are *decentralizable*, which is formally defined as below.

Definition 2 (Decentralizable Cooperative Tasks, a.k.a. IGM Condition (Son et al., 2019))**.** *A cooperative task is decentralizable if* $\exists \{Q_i\}_{i=1}^N$ *such that* $\forall \boldsymbol{\tau} \in \tau^N, \boldsymbol{a} \in \mathcal{A}^N$,

$$\arg\max_{\boldsymbol{a}} Q^{\boldsymbol{\pi}}(\boldsymbol{\tau}, \boldsymbol{a}) = \begin{bmatrix} \arg\max_{a_1} Q_1(\tau_1, a_1) \\ \vdots \\ \arg\max_{a_N} Q_N(\tau_N, a_N) \end{bmatrix}. \quad (3)$$

Eq. 3 suggests that local maxima on the extracted value function per every agent needs to amount to the global maximum on the joint value function. A key challenge for CTDE methods is, then, how to correctly extract each of the agent's individual Q-function $\{Q_i\}_{i=1}^N$, and as such an executable policy, from a centralized Q-function $Q^{\boldsymbol{\pi}}$.

To satisfy Eq. 3, current solutions rely on restrictive assumptions that enforce structural constraints on the factorization of the joint Q-function. For example, VDN (Sunehag et al., 2017) adopts the additivity assumption by assuming $Q^{\boldsymbol{\pi}}(\boldsymbol{\tau}, \boldsymbol{a}) := \sum_{i=1}^{N} Q_i(\tau_i, a_i)$. QMIX (Rashid et al., 2018) applies the monotonicity assumption to ensure $\frac{\partial Q^{\boldsymbol{\pi}}(\boldsymbol{\tau}, \boldsymbol{a})}{\partial Q_i(\tau_i, a_i)} \geq 0, \forall i \in \mathcal{N}$. QTRAN (Son et al., 2019) introduces a refined factorizable learning objective in addition to QMIX. Nonetheless, structural constraints harm the representational power of the centralized value function, and also hinder efficient explorations (Son et al., 2019). To mitigate these problems, we propose Q-DPP as an alternative that naturally factorizes the joint Q-function by learning a diverse set of behavioral models among agents.

3.2. Q-DPP: A Constrained DPP for MARL

Our method is established on Q-DPP which is an extension of DPP that suits MARL. We assume that local observation o_i encodes all history information τ_i at each time-step. We model the ground set of all agents' observation-action pairs by a DPP, i.e., $\mathcal{Y} = \left\{(o_1^1, a_1^1), \dots, (o_N^{|\mathcal{O}|}, a_N^{|\mathcal{A}|})\right\}$ with the size of the ground set being $|\mathcal{Y}| = N|\mathcal{O}||\mathcal{A}|$.

In the context of multi-agent learning, each agent takes one valid action depending on its local observation. A valid sample from DPP, therefore, is expected to include one valid observation-action pair for each agent, and the observations from the sampled pairs must match the true observations that agents receive at every time step. To meet such requirements, we propose a new type of DPP, named **Q-DPP**.

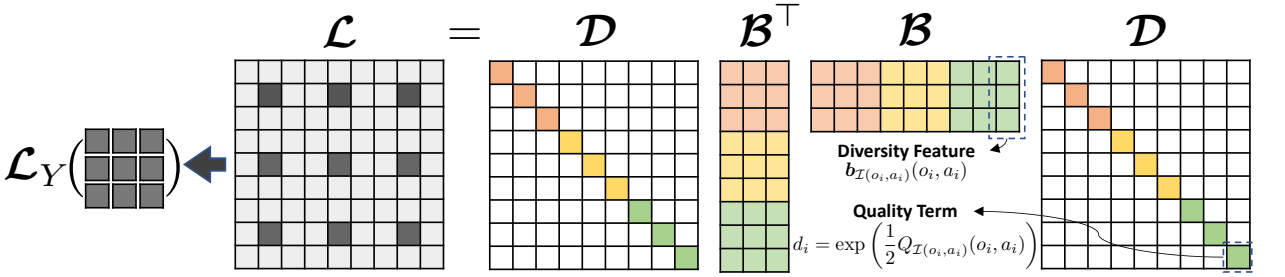


Figure 2: Example of Q-DPP with quality-diversity kernel decomposition in a single-state three-player learning task, each agent has three actions to choose. The size of the ground set is $|\mathcal{Y}| = 9$, and the size of valid subsets $|\mathcal{C}(\boldsymbol{o})|$ is $3^3 = 27$. Different colors represent different partitions of each agent's observation-action pairs. Suppose all three agents select the 2nd action, then the Q-value of the joint action according to Eq. 5 is $Q^{\boldsymbol{\pi}}(\boldsymbol{o}, \boldsymbol{a}) = \det\left([\boldsymbol{\mathcal{L}}_{[i,j],i,j\in\{2,5,8\}}]\right)$.

Definition 3 (Q-DPP). *Given a ground set $\mathcal{Y}$ of size M that includes N agents' all possible observation-action pairs $\mathcal{Y} = \{(o_1^1, a_1^1), \ldots, (o_N^{|\mathcal{O}|}, a_N^{|\mathcal{A}|})\}$, we partition $\mathcal{Y}$ into N disjoint parts, i.e., $\mathcal{Y} = \bigcup_{i=1}^N \mathcal{Y}_i$ and $\sum_{i=1}^N |\mathcal{Y}_i| = M = N|\mathcal{O}||\mathcal{A}|$, where each partition represents each individual agent's all possible observation-action pairs. At every time-step, given agents' observations, $\boldsymbol{o} = (o_i)_{i\in\mathcal{N}}$, we define $\mathcal{C}(\boldsymbol{o}) \subseteq \mathcal{Y}$ to be a set of valid subsets including only observation-action pairs that agents are allowed to take,*

$$\mathcal{C}(\boldsymbol{o}) := \left\{Y \subseteq \mathcal{Y} : |Y \cap \mathcal{Y}_i(o_i)| = 1, \forall i \in \{1, \ldots, N\}\right\},$$

with $|\mathcal{C}(\boldsymbol{o})| = |\mathcal{A}|^N$, and $\mathcal{Y}_i(o_i)$ of size $|\mathcal{A}|$ denotes the set of pairs in partition $\mathcal{Y}_i$ with only o_i as the observation,

$$\mathcal{Y}_i(o_i) = \left\{(o_i, a_i^1), \ldots, (o_i, a_i^{|\mathcal{A}|})\right\}.$$

Q-DPP, denoted by $\tilde{\mathbb{P}}$, defines a probability measure over the valid subsets $Y \in \mathcal{C}(\boldsymbol{o}) \subseteq \mathcal{Y}$. Let $\boldsymbol{Y}$ be a random subset drawn according to $\tilde{\mathbb{P}}$, its probability distribution is defined:

$$\tilde{\mathbb{P}}_{\boldsymbol{\mathcal{L}}}\left(\boldsymbol{Y} = Y | \boldsymbol{Y} \in \mathcal{C}(\boldsymbol{o})\right) := \frac{\det\left(\boldsymbol{\mathcal{L}}_Y\right)}{\sum_{Y' \in \mathcal{C}(\boldsymbol{o})} \det\left(\boldsymbol{\mathcal{L}}_{Y'}\right)}. \quad (4)$$

In addition, given a valid sample $Y \in \mathcal{C}(\boldsymbol{o})$, we define an identifying function $\mathcal{I} : Y \to \mathcal{N}$ that specifies the agent number for each valid pair in Y, and an index function $\mathcal{J} : \mathcal{Y} \to \{1, \ldots, M\}$ that specifies the cardinality of each item in Y in the ground set $\mathcal{Y}$.

The construction of Q-DPP is inspired by P-DPP (Celis et al., 2018). However, the partitioned sets in P-DPP stay fixed, while in Q-DPP, $\mathcal{C}(\boldsymbol{o_t})$ changes at every time-step with the new observation, and the kernel $\boldsymbol{\mathcal{L}}$ is learned through the process of reinforcement learning rather than being given. More differences are listed in *Appendix A.3*.

Given Q-DPPs, we can represent the centralized value function by adopting Q-DPPs as general function approximators:

$$Q^{\boldsymbol{\pi}}(\boldsymbol{o}, \boldsymbol{a}) := \log\det\left(\boldsymbol{\mathcal{L}}_{Y=\left\{(o_1,a_1),\ldots,(o_N,a_N)\right\}\in\mathcal{C}(\boldsymbol{o}^t)}\right), \quad (5)$$

where $\boldsymbol{\mathcal{L}}_Y$ denotes the sub-matrix of $\boldsymbol{\mathcal{L}}$ whose entries are indexed by the pairs included in Y. Q-DPP embeds the connection between the joint action and each agent's individual actions into a subset-sampling process, and the Q-value is quantified by the determinant of a kernel matrix whose elements are indexed by the associated observation-action pairs. The goal of multi-agent learning is to learn an optimal joint Q-function. Eq. 5 states $\det(\boldsymbol{\mathcal{L}}_Y) = \exp(Q^\pi(\boldsymbol{o}, \boldsymbol{a}))$, meaning Q-DPP actually assigns large probability to the subsets that have large Q-values. Given $\det(\boldsymbol{\mathcal{L}}_Y)$ is always positive, the $\log$ operator ensures Q-DPPs, as general function approximators, can recover any real Q-functions.

DPPs can capture both the quality and diversity of a sampled subset; the joint Q-function represented by Q-DPP in theory should not only acknowledge the quality of each agent's individual action towards a large reward, but the diversification of agents' actions as well. The remaining question is, then, how to obtain such quality-diversity representation.

3.3. Representation of Q-DPP Kernels

For any PSD matrix $\boldsymbol{\mathcal{L}}$, such a $\boldsymbol{\mathcal{W}}$ can always be found so that $\boldsymbol{\mathcal{L}} = \boldsymbol{\mathcal{W}}\boldsymbol{\mathcal{W}}^\top$ where $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{M\times P}, P \leq M$. Since the diagonal and off-diagonal entries of $\boldsymbol{\mathcal{L}}$ represent *quality* and *diversity* respectively, we adopt an interpretable decomposition by expressing each row of $\boldsymbol{\mathcal{W}}$ as a product of a **quality** term $d_i \in \mathbb{R}^+$ and a **diversity** feature term $\boldsymbol{b}_i \in \mathbb{R}^{P\times 1}$ with $\|\boldsymbol{b}_i\| \leq 1$, i.e., $\boldsymbol{w}_i = d_i \boldsymbol{b}_i^\top$. An example of such decomposition is visualized in Fig. 2 where we define $\boldsymbol{\mathcal{B}} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_M]$ and $\boldsymbol{\mathcal{D}} = \mathrm{diag}(d_1, \ldots, d_M)$. Note that both $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{B}}$ are free parameters that can be learned from the environment during the Q-learning process in Eq. 2.

If we denote the quality term as each agent's individual Q-value for a given observation-action pair, i.e., $\forall(o_i, a_i) \in \mathcal{Y}, i = \{1, \ldots, M\}, d_i := \exp\left(\frac{1}{2}Q_{\mathcal{I}(o_i,a_i)}(o_i, a_i)\right)$, then Eq. 5 can be further written into

$$\begin{aligned} Q^{\boldsymbol{\pi}}(\boldsymbol{o}, \boldsymbol{a}) &= \log\det\left(\boldsymbol{\mathcal{W}}_Y \boldsymbol{\mathcal{W}}_Y^\top\right) \\ &= \log\left(\mathrm{tr}\left(\boldsymbol{\mathcal{D}}_Y^\top \boldsymbol{\mathcal{D}}_Y\right)\det\left(\boldsymbol{\mathcal{B}}_Y^\top \boldsymbol{\mathcal{B}}_Y\right)\right) \\ &= \sum_{i=1}^N Q_{\mathcal{I}(o_i,a_i)}(o_i, a_i) + \log\det\left(\boldsymbol{\mathcal{B}}_Y^\top \boldsymbol{\mathcal{B}}_Y\right). \end{aligned} \quad (6)$$

Since a determinant value only reaches the maximum when the associated vectors in $\boldsymbol{\mathcal{B}}_Y$ are mutually orthogonal (Noble et al., 1988), Eq. 6 essentially stipulates that Q-DPP represents the joint value function by taking into account not only the quality of each agent's contribution towards reward maximization, more importantly, from a holistic perspective, the orthogonalization of agents' actions.

In fact, the inclusion of diversifying agents' behaviors is an important factor in satisfying the condition in Eq. 3. Intuitively, in a decentralizable task with a shared goal, promoting orthogonality between agent's actions can help clarify the functionality and responsibility of each agent, which in return leads to a better instantiation of Eq. 3. On the other hand, diversity does not means that agents have to take different actions all the time. Since the goal is still to achieve large reward via optimizing Eq. 2, certain scenarios, such as agents need to take identical actions to accomplish a task, will not be excluded as a result of promoting diversity.

3.4. Connections to Current Methods

Based on the quality-diversity representation, one can draw a key connection between Q-DPP and the existing methods. It turns out that, under the sufficient condition that if the learned diversity features that correspond to the optimal actions are mutually orthogonal, then Q-DPP degenerates to VDN (Sunehag et al., 2017), QMIX (Rashid et al., 2018), and QTRAN (Son et al., 2019) respectively.

To elaborate such condition, let us denote $a_i^* = \arg\max Q_i(o_i, a_i)$, $\boldsymbol{a}^* = (a_i^*)_{i\in\mathcal{N}}$, $Y^* = \{(o_i, a_i^*)\}_{i=1}^N$, with $\|\boldsymbol{b}_i\| = 1$ and $\boldsymbol{b}_i^\top \boldsymbol{b}_j = 0, \forall i \neq j$, then we have

$$\det\left(\boldsymbol{\mathcal{B}}_{Y^*}^\top \boldsymbol{\mathcal{B}}_{Y^*}\right) = 1. \tag{7}$$

Connection to VDN. When $\{\boldsymbol{b}_j\}_{j=1}^M$ are pairwise orthognal, by plugging Eq. 7 into Eq. 6, we can obtain

$$Q^{\boldsymbol{\pi}}(\boldsymbol{o}, \boldsymbol{a}^*) = \sum_{i=1}^N Q_{\mathcal{I}(o_i, a_i^*)}(o_i, a_i^*). \tag{8}$$

Eq. 8 recovers the exact additivity constraint that VDN applies to factorize the joint value function in meeting Eq. 3.

Connection to QMIX. Q-DPP also generalizes QMIX, which adopts a monotonic constraint on the centralized value function to meet Eq. 3. Under the special condition when $\{\boldsymbol{b}_j\}_{j=1}^M$ are mutually orthogonal, we can easily show that Q-DPP meets the monotonicity condition because

$$\frac{\partial Q^{\boldsymbol{\pi}}(\boldsymbol{o}, \boldsymbol{a}^*)}{\partial Q_{\mathcal{I}(o_i, a_i^*)}(o_i, a_i^*)} = 1 \geq 0, \quad \forall \mathcal{I}(o_i, a_i^*) \in \mathcal{N}. \tag{9}$$

Connection to QTRAN. Q-DPP also meets the sufficient conditions that QTRAN proposes for meeting Eq. 3, that is,

$$\sum_{i=1}^N Q_i(o_i, a_i) - Q^{\boldsymbol{\pi}}(\boldsymbol{o}, \boldsymbol{a}) + V(\boldsymbol{o}) = \begin{cases} 0 & \boldsymbol{a} = \boldsymbol{a}^* \\ \geq 0 & \boldsymbol{a} \neq \boldsymbol{a}^* \end{cases}, \tag{10}$$

where $V(\boldsymbol{o}) = \max_{\boldsymbol{a}} Q^{\boldsymbol{\pi}}(\boldsymbol{o}, \boldsymbol{a}) - \sum_{i=1}^N Q_i(o_i, a_i^*)$. Through Eq. 6, we know Q-DPP can have Eq. 10 written as

$$-\log\det\left(\boldsymbol{\mathcal{B}}_Y^\top \boldsymbol{\mathcal{B}}_Y\right) + \max_{\boldsymbol{a}} Q^{\boldsymbol{\pi}}(\boldsymbol{o}, \boldsymbol{a}) - \sum_{i=1}^N Q_i(o_i, a_i^*). \tag{11}$$

When $\boldsymbol{a} = \boldsymbol{a}^*$, for pairwise orthogonal $\{\boldsymbol{b}_j\}_{j=1}^M$, Q-DPP satisfies the first condition since Eq. 11 equals to zero due to $\log\det(\boldsymbol{\mathcal{B}}_{Y^*}^\top \boldsymbol{\mathcal{B}}_{Y^*}) = 0$. When $\boldsymbol{a} \neq \boldsymbol{a}^*$, Eq. 11 equals to $-\log\det\left(\boldsymbol{\mathcal{B}}_Y^\top \boldsymbol{\mathcal{B}}_Y\right) + \log\det\left(\boldsymbol{\mathcal{B}}_{Y^*}^\top \boldsymbol{\mathcal{B}}_Y^*\right)$, which is always positive since $\det(\boldsymbol{\mathcal{B}}_Y^\top \boldsymbol{\mathcal{B}}_Y) < 1, \forall Y \neq Y^*$; Q-DPP thereby meets the second condition of Eq. 10 and recovers QTRAN.

Other Related Work. Determinantal SARSA (Osogami & Raymond, 2019) applies a normal DPP to model the ground set of the joint state-action pairs $\left\{(s^0, a_1^0, \ldots, a_N^0), \ldots, (s^{|\mathcal{S}|}, a_1^{|\mathcal{A}|}, \ldots, a_N^{|\mathcal{A}|})\right\}$. It fails to consider at all a proper ground set that suits multi-agent problems, which leads to the size of subsets being $2^{|\mathcal{S}||\mathcal{A}|^N}$ that is double-exponential to the number of agents. Furthermore, unlike Q-DPP that learns decentralized policies, Det. SARSA learns the centralized joint-action policy, which strongly limits its applicability for scalable real-world tasks.

3.5. Sampling from Q-DPP

Agents need to explore the environment effectively during training; however, how to sample from Q-DPPs defined in Eq. 4 is still unknown. In fact, sampling from the DPPs with partition-matroid constraint is a non-trivial task. So far, the best known exact sampler for partitioned DPPs has $\mathcal{O}(m^p)$ time complexity with m being the ground-set size and p being the number of partitions (Li et al., 2016; Celis et al., 2017). Nonetheless, these samplers still pose great computational challenges for multi-agent learning tasks and cannot scale to large number of agents because we have $m = |\mathcal{C}(\boldsymbol{o})| = |\mathcal{A}|^N$ for multi-agent learning tasks.

In this work, we instead adopt a biased yet tractable sampler for Q-DPP. Our sampler is an application of the sampling-by-projection idea in Celis et al. (2018) and Chen et al. (2018) which leverages the property that Gram-Schmidt process preserves the determinant. One benefit of our sampler is that it promotes efficient explorations among agents during training. Importantly, it enjoys only linear-time complexity *w.r.t.* the number of agents. The intuition is as follows.

Additional Notations. In a Euclidean space $\mathbb{R}^n$ equipped with an inner product $\langle\cdot,\cdot\rangle$, let $\mathcal{U} \subseteq \mathbb{R}^n$ be any linear subspace, and $\mathcal{U}^\perp$ be its orthogonal complement $\mathcal{U}^\perp := \{x \in \mathbb{R}^n | \langle x, y\rangle = 0, \forall y \in \mathcal{U}\}$. We define an orthogonal projection operator, $\Pi_{\mathcal{U}} : \mathbb{R}^n \to \mathbb{R}^n$, such that $\forall \boldsymbol{u} \in \mathbb{R}^n$, if $\boldsymbol{u} = \boldsymbol{u}_1 + \boldsymbol{u}_2$ with $\boldsymbol{u}_1 \in \mathcal{U}$ and $\boldsymbol{u}_2 \in \mathcal{U}^\perp$, then $\Pi_{\mathcal{U}}(\boldsymbol{u}) = \boldsymbol{u}_2$.

Gram-Schmidt (Noble et al., 1988) is a process for orthogonalizing a set of vectors; given a set of linearly in-

Algorithm 1 Multi-Agent Determinantal Q-Learning

DEF Orthogonalizing-Sampler $(\mathcal{Y}, \boldsymbol{\mathcal{D}}, \boldsymbol{\mathcal{B}}, \boldsymbol{o})$:
Init: $\boldsymbol{b}_j \leftarrow \boldsymbol{\mathcal{B}}_{[:,j]}, Y \leftarrow \emptyset, B \leftarrow \emptyset, J \leftarrow \emptyset$.
for each partition $\mathcal{Y}_i$ **do**
  Define $\forall (o,a) \in \mathcal{Y}_i(o_i)$
  $q(o,a) := \left\|\boldsymbol{b}_{\mathcal{J}(o,a)}\right\|^2 \exp\left(\boldsymbol{\mathcal{D}}_{\mathcal{J}(o,a),\mathcal{J}(o,a)}\right)$.
  Sample $(\tilde{o}_i, \tilde{a}_i) \in \mathcal{Y}_i(o_i)$ from the distribution:

$$\left\{\frac{q(o,a)}{\sum_{(\hat{o},\hat{a})\in\mathcal{Y}_i(o_i)} q(\hat{o},\hat{a})}\right\}_{(o,a)\in\mathcal{Y}_i(o_i)}.$$

  Let $Y \leftarrow Y \cup (\tilde{o}_i, \tilde{a}_i), B \leftarrow B \cup \boldsymbol{b}_{\mathcal{J}(\tilde{o}_i,\tilde{a}_i)}$, $J \leftarrow J \cup \mathcal{J}(\tilde{o}_i, \tilde{a}_i)$.
  // Gram-Schmidt orthogonalization
  Set $\boldsymbol{b}_j = \amalg_{\text{span}\{B\}}(\boldsymbol{b}_j), \forall j \in \{1, ..., M\} - J$
end for
Return: Y.

DEF Determinantal-Q-Learning $(\theta = [\theta_{\boldsymbol{\mathcal{D}}}, \theta_{\boldsymbol{\mathcal{B}}}], \mathcal{Y})$:
Init: $\theta^- \leftarrow \theta$, $D \leftarrow \emptyset$.
for each time-step **do**
  Collect observations $\boldsymbol{o} = [o_1, \ldots, o_N]$ for all agents.
  $\boldsymbol{a} =$ **Orthogonalizing-Sampler**$(\mathcal{Y}, \theta_{\boldsymbol{\mathcal{D}}}, \theta_{\boldsymbol{\mathcal{B}}}, \boldsymbol{o})$.
  Execute $\boldsymbol{a}$, store the transition $\langle \boldsymbol{o}, \boldsymbol{a}, \mathcal{R}, \boldsymbol{o}' \rangle$ in D.
  Sample a mini-batch of $\{\langle \boldsymbol{o}, \boldsymbol{a}, \mathcal{R}, \boldsymbol{o}' \rangle\}_{j=1}^{E}$ from D.
  Compute for each transition in the mini-batch
  $\max_{\boldsymbol{a}'} Q(\boldsymbol{o}', \boldsymbol{a}'; \theta^-)$
  $= \log\det\left(\boldsymbol{\mathcal{L}}_{Y=\{(o'_1,a^*_1),\ldots,(o'_N,a^*_N)\}}\right)$
  where *// off-policy decentralized execution*
  $a^*_i = \arg\max_{a_i\in\mathcal{A}_i}\left[\left\|\theta^-_{\boldsymbol{b}_{\mathcal{J}(o'_i,a_i)}}\right\|^2 \cdot \exp\left(\theta^-_{\boldsymbol{\mathcal{D}}_{\mathcal{J}(o'_i,a_i),\mathcal{J}(o'_i,a_i)}}\right)\right]$.
  // centralized training
  Update θ by minimizing $\mathcal{L}(\theta)$ defined in Eq. 2.
  Update target $\theta^- = \theta$ periodically.
end for
Return: $\theta_{\boldsymbol{\mathcal{D}}}, \theta_{\boldsymbol{\mathcal{B}}}$.

dependent vectors $\{\boldsymbol{w}_i\}$, it outputs a mutually orthogonal set of vectors $\{\hat{\boldsymbol{w}}_i\}$ by computing $\hat{\boldsymbol{w}}_i := \amalg_{\mathcal{U}_i}(\boldsymbol{w}_i)$ where $\mathcal{U}_i = \text{span}\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{i-1}\}$. Note that we neglect the normalizing step of Gram-Schmidt in this work. Finally, if the rows $\{\boldsymbol{w}_i\}$ of a matrix $\boldsymbol{\mathcal{W}}$ are mutually orthogonal, we can compute the determinant by $\det(\boldsymbol{\mathcal{W}}\boldsymbol{\mathcal{W}}^\top) = \prod \|\boldsymbol{w}_i\|^2$. The Q-DPP sampler is built upon the following property.

Proposition 1 (Volume preservation of Gram-Schmidt, see Chapter 7 in Shafarevich & Remizov (2012), also Lemma 3.1 in Celis et al. (2018).)**.** *Let $\mathcal{U}_i = \text{span}\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{i-1}\}$ and $\boldsymbol{w}_i \in \mathbb{R}^P$ be the i-th row of $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{M\times P}$, then $\prod_{i=1}^{M} \| \amalg_{\mathcal{U}_i}(\boldsymbol{w}_i)\|^2 = \det(\boldsymbol{\mathcal{W}}\boldsymbol{\mathcal{W}}^\top)$.*

We also provide an intuition by Gaussian elimination in *Appendix A.1*. Proposition 1 suggests that the determinant of a Gram matrix is invariant to applying the Gram-Schmidt orthogonalization on the rows of that Gram matrix. In Q-DPP's case, a kernel matrix with mutually orthogonal rows can largely simplify the sampling process. In such scenario, an effective sampler can be that, from each partition $\mathcal{Y}_i$, sample an item $i \in \mathcal{Y}_i$ with $\mathbb{P}(i) \propto \|d_i \boldsymbol{b}_i^\top\|^2$, then add i to the output sample Y and move to the next partition; the above steps iterate until all partitions are covered. It is effortless to see that the probability of obtaining sample Y in such a way is

$$\mathbb{P}(Y) \propto \prod_{i\in Y} \|d_i \boldsymbol{b}_i^\top\|^2 = \prod_{i\in Y} \|\boldsymbol{w}_i\|^2 = \det(\boldsymbol{\mathcal{W}}_Y \boldsymbol{\mathcal{W}}_Y^\top) \propto \det(\boldsymbol{\mathcal{L}}_Y). \tag{12}$$

We formally describe the orthogonalizing sampling procedures in Algorithm 1. As it is suggested in Celis et al. (2018), the time complexity of the sampling function is $\mathcal{O}(NMP)$ (see also the breakdown analysis for each step in *Appendix A.4*), given the input size being $\mathcal{O}(MP)$, our sampler thus enjoys linear-time complexity *w.r.t* the agent number.

Though the Gram-Schmidt process can preserve the determinant and simply the sampling process, it comes at a prize of introducing **bias** on the normalization term. Specifically, the normalization in our proposed sampler is conducted at each agent/partition level $\mathcal{Y}_i(o_i)$ (see the red in line 5) which does not match Eq. 4 that suggests normalizing by listing all valid samples considering all partitions $\mathcal{C}(\boldsymbol{o})$; this directly leads to a sampled subset from our sampler having *larger* probability than what Q-DPP defines. Interestingly, it turns out that such violation can be controlled through bounding the singular values of each partition in the kernel matrix (see Assumption 1), a technique also known as the β-balance condition introduced in P-DPP (Celis et al., 2018).

Assumption 1 (Singular-Value Constraint on Partitions)**.** *For a Q-DPP defined in Definition 1, which is parameterized by $\boldsymbol{\mathcal{D}} \in \mathbb{R}^{M\times M}, \boldsymbol{\mathcal{B}} \in \mathbb{R}^{P\times M}$ and $\boldsymbol{\mathcal{W}} := \boldsymbol{\mathcal{D}}\boldsymbol{\mathcal{B}}^\top$, let $\sigma_1 \geq \ldots \geq \sigma_P$ represent the singular values of $\boldsymbol{\mathcal{W}}$, and $\hat{\sigma}_{i,1} \geq \ldots \geq \hat{\sigma}_{i,P}$ denote the singular values of $\boldsymbol{\mathcal{W}}_{\mathcal{Y}_i}$ that is the submatrix of $\boldsymbol{\mathcal{W}}$ with the rows and columns corresponding to the i-th partition $\mathcal{Y}_i$, we assume $\forall j \in \{1, \ldots, P\}$, $\exists\, \delta \in (0, 1]$, s.t., $\min_{i\in\{1,\ldots,N\}} \hat{\sigma}^2_{i,j}/\delta \geq \sigma^2_j$ holds.*

Theorem 1 (Approximation Guarantee of Orthogonalizing Sampler)**.** *For a Q-DPP defined in Definition 1, under Assumption 1, the Orthogonalizing Sampler described in Algorithm 1 returns a sampled subset $Y \in \mathcal{C}(\boldsymbol{o})$ with probability $\mathbb{P}(Y) \leq 1/\delta^N \cdot \tilde{\mathbb{P}}(\boldsymbol{Y} = Y)$ where N is the number of agents, $\tilde{\mathbb{P}}$ is defined in Eq. 4, δ is defined in Assumption 1.*

Proof. The proof is in *Appendix A.2*. It can also be taken as a special case of Theorem 3.2 in Celis et al. (2018) when the number of sample from each partition is one. ■

Theorem 1 effectively suggests a way to bound the error between our sampler and the true distribution of Q-DPP through minimizing the difference between σ^2_j and $\hat{\sigma}^2_{i,j}$.

3.6. Determinantal Q-Learning

We present the full learning procedures in Algorithm 1. Determinantal Q-Learning is a CTDE method. During training, agents' explorations are conducted through the orthogonalizing-sampler. The parameters of $\mathcal{B}$ and $\mathcal{D}$ are updated through Q-learning in a centralized way by following Eq. 2. To meet Assumption 1, one can implement an auxiliary loss function of $\max(0, \sigma_j^2 - \hat{\sigma}_{i,j}^2/\delta)$ in addition to where δ is a hyper-parameter. Given Theorem 1, for large N, we know δ should be set close to 1 to make the bound tight. In fact, it is worth mentioning that the Gram-Schmidt process adopted in the sampler can boost the sampling efficiency for multi-agent training. Since agents' diversity features of observation-action pairs are orthogonalized every time after a partition is visited, agents who act later are essentially coordinated to explore the observation-action space that is distinctive to all previous agents. This speeds up training in early stages.

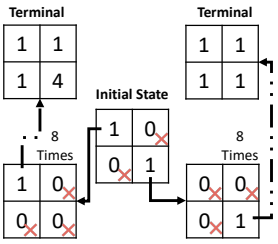
During execution, agents only need to access the parameters in their own partitions to compute the greedy action (see line 19). Note that neural networks can be seamlessly applied to represent both $\mathcal{B}$ and $\mathcal{D}$ to tackle continuous states. Though a full treatment of deep Q-DPP needs substantial future work, we show a proof of concept in *Appendix C*. Hereafter, we use Q-DPP to represent our proposed algorithm.

4. Experiments

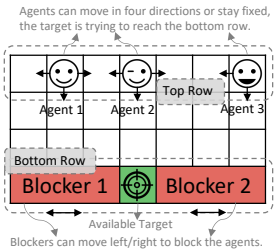
We compare Q-DPP with state-of-the-art CTDE solvers for multi-agent cooperative tasks, including COMA (Foerster et al., 2018), VDN (Sunehag et al., 2017), QMIX (Rashid et al., 2018), QTRAN (Son et al., 2019), and MAVEN (Mahajan et al., 2019). All baselines are imported from PyMARL (Samvelyan et al., 2019). Detailed settings are in *Appendix B*. Code is released in `https://github.com/QDPP-GitHub/QDPP`. We consider four cooperative tasks in Fig. 3, all of which require non-trivial value function decomposition to achieve the largest reward.

Pathological Stochastic Game. The optimal policy of this game is to let both agents keep acting top left until the 10-th step to change to bottom right, which results in the optimal reward of 13. The design of such stochastic game intends to be pathological. First, it is non-monotonic (thus QMIX surely fails), second, it demonstrates *relative overgeneralization* (Wei et al., 2018) because both agents playing the 1st action on average offer a higher reward 10 when matched with arbitrary actions from the other agent. We allow agent to observe the current step number and the joint action in the last time-step. Zero reward leads to immediate termination. Fig. 4a shows Q-DPP can converge to the global optimal in only $20K$ steps while other baselines struggle.

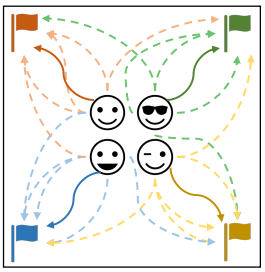
Blocker Game & **Coordinated Navigation**. Blocker game



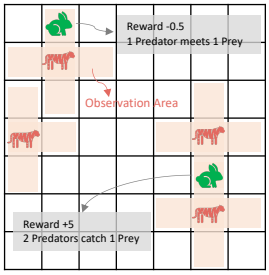
(a) Pathological Stochastic Game



(b) Blocker Game



(c) Coordinated Navigation



(d) Predator-Prey World

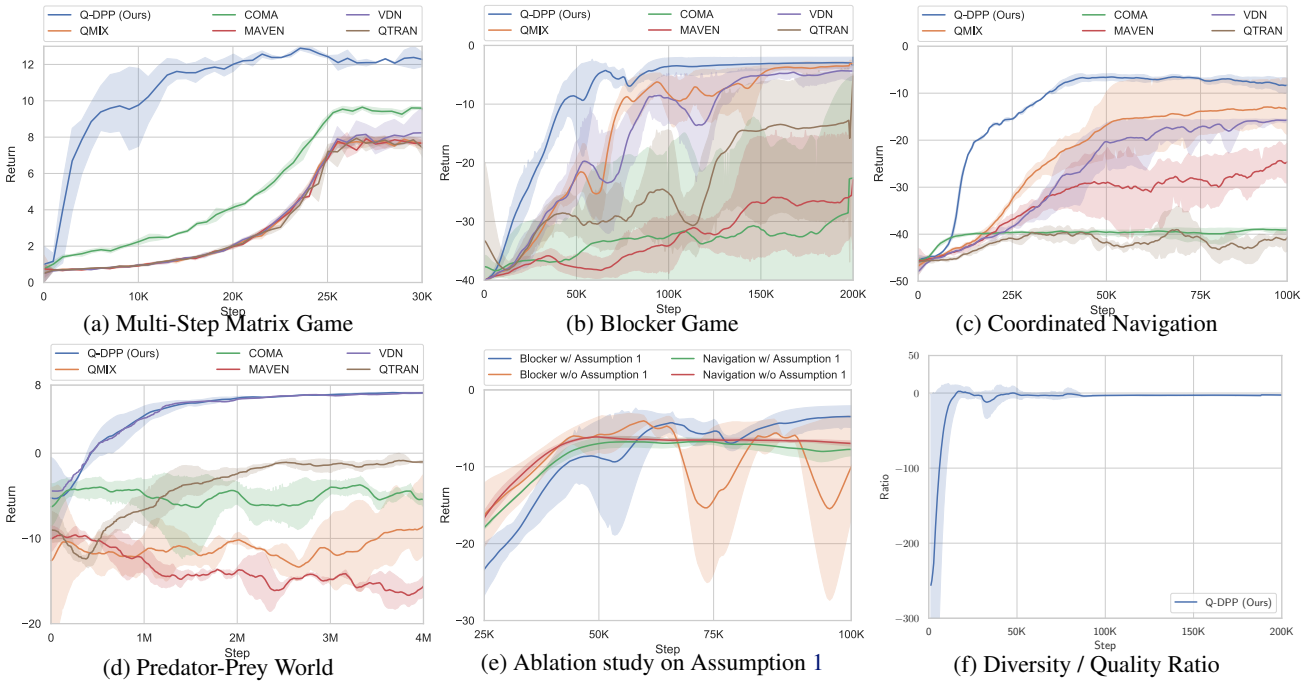
Figure 3: Multi-agent cooperative tasks. The size of the ground set for each task is a) 176, b) 420, c) 720, d) 3920.

(Heess et al., 2012) requires agents to reach the bottom row by coordinating with its teammates to deceive the blockers that can move left/right to block them. The navigation game requires four agents to reach four different landmarks. For both tasks, it costs all agents -1 reward per time-step before they all reach the destination. Depending on the starting points, the largest reward of the game are -3 and -6 respectively. Both tasks are challenging in the sense that coordination is rather challenging for agents that only have decentralized policies and local observations. Fig. 4b & 4c suggest Q-DPP still achieves the best performance.

Predator-Prey World. In this task, four predators attempt to capture two randomly-moving preys. Each predator can move in four directions but they only have local views. The predators get a team reward of 1 if two or more predators are capturing the same prey at the same time, and they are penalized for -0.5 if only one of them captures a prey. The game terminates when all preys are caught. Fig. 4d shows Q-DPP's superior performance than all other baselines.

Apart from the best performance in terms of rewards, here we offer more insights of why and how Q-DPP works well.

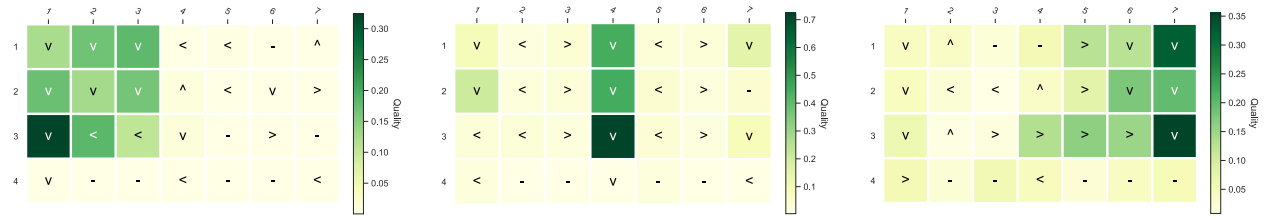
The Importance of Assumption 1. Assumption 1 is the premise for the correctness of Q-DPP sampler to hold. To investigate its impact in practice, we conduct the ablation study on Blocker and Navigation games. We implement such assumption via an auxiliary loss function of $\max(0, \sigma_j^2 - \hat{\sigma}_{i,j}^2/\delta)$ that penalizes the violation of the assumption, we set $\delta = 0.5$. Fig. 4e presents the performance comparisons of the Q-DPPs with and without such additional loss function. We can tell that maintaining such a condition, though not helping improve the performance,



(a) Multi-Step Matrix Game (b) Blocker Game (c) Coordinated Navigation

(d) Predator-Prey World (e) Ablation study on Assumption 1 (f) Diversity / Quality Ratio

Figure 4: **(a)-(d)**:Performance over time on different tasks. **(e)**: Ablation study on Assumption 1 on Blocker game. **(f)**: The ratio of diversity to quality, i.e., $\log\det(\boldsymbol{\mathcal{B}}_Y^\top \boldsymbol{\mathcal{B}}_Y) / \sum_{i=1}^N Q_{\mathcal{I}(o_i,a_i)}(o_i,a_i)$, during training on Blocker game.



(a) Agent 1. (b) Agent 2. (c) Agent 3.

Figure 5: **(a)-(c)**: Each of the agent's decentralized policy, i.e., $\arg\max_a Q_i(o_i, a)$, during execution on Blocker game.

stablizes the training process by significantly reducing the variance of the rewards. We believe this is because violating Assumption 1 leads to over-estimating the probability of certain observation-action pairs in the partition where the violation happens, such over-estimation can make the agent stick to a poor local observation-action pair for some time.

The Satisfaction of Eq. 3. We show empirical evidence on Blocker game that the natural factorization that Q-DPP offers indeed satisfy Eq. 3. Intuitively, Q-DPPs encourage agents to acquire diverse behavorial models during training so that the optimal action of one agent does not depend on the actions of the other agents during the decentralized execution stage, as a result, Eq. 3 can be satisfied. Fig. 5 (a-c) justify such intuition by showing Q-DPP learns mutually orthogonal behavioral models. Given the distinction among agents' individual policies, one can tell that the joint optimum is reached through individual optima.

Quality versus Diversity. We investigate the change of the relative importance of quality versus diversity during training. On Blocker game, we show the ratio of $\log\det\left(\boldsymbol{\mathcal{B}}_Y^\top \boldsymbol{\mathcal{B}}_Y\right) / \sum_{i=1}^N Q_{\mathcal{I}(o_i,a_i)}\left(o_i,a_i\right)$, which reflects how the learning algorithm balances maximizing reward against encouraging diverse behaviors. In Fig. 4f, we can see that the ratio gradually converges to 0. The diversity term plays a less important role with the development of training; this is also expected since explorations tend to be rewarded more at the early stage of a task.

5. Conclusion

We proposed Q-DPP, a new type of value-function approximator for cooperative multi-agent reinforcement learning. Q-DPP, as a probabilistic way of modeling sets, considers not only the quality of agents' actions towards reward maximization, but the diversity of agents' behaviors as well. We have demonstrated that Q-DPP addresses the limitation of current major solutions including VDN, QMIX, and QTRAN by learning the value function decomposition without structural constraints. In the future, we plan to investigate other kernel representations for Q-DPPs to tackle the tasks with continuous states and continuous actions.

Acknowledgement

We sincerely thank Dr. Haitham Bou Ammar for his constructive comments. Weinan Zhang thanks the support of "New Generation of AI 2030" Major Project 2018AAA0100900 and NSFC (61702327, 61632017).

References

Affandi, R. H., Fox, E., Adams, R., and Taskar, B. Learning the parameters of determinantal point process kernels. In *International Conference on Machine Learning*, pp. 1224–1232, 2014.

Celis, L. E., Deshpande, A., Kathuria, T., Straszak, D., and Vishnoi, N. K. On the complexity of constrained determinantal point processes. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

Celis, L. E., Keswani, V., Straszak, D., Deshpande, A., Kathuria, T., and Vishnoi, N. K. Fair and diverse dpp-based data summarization. *arXiv preprint arXiv:1802.04023*, 2018.

Chen, L., Zhang, G., and Zhou, E. Fast greedy map inference for determinantal point process to improve recommendation diversity. In *Advances in Neural Information Processing Systems*, pp. 5622–5633, 2018.

Du, Y., Han, L., Fang, M., Liu, J., Dai, T., and Tao, D. Liir: Learning individual intrinsic reward in multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 4403–4414, 2019.

Elfeki, M., Couprie, C., Riviere, M., and Elhoseiny, M. Gdpp: Learning diverse generations using determinantal point processes. In *International Conference on Machine Learning*, pp. 1774–1783, 2019.

Foerster, J. N., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. Counterfactual multi-agent policy gradients. In *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

Heess, N., Silver, D., and Teh, Y. W. Actor-critic reinforcement learning with energy-based policies. In *EWRL*, pp. 43–58, 2012.

Kulesza, A., Taskar, B., et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012.

Li, C., Sra, S., and Jegelka, S. Fast mixing markov chains for strongly rayleigh measures, dpps, and constrained sampling. In *Advances in Neural Information Processing Systems*, pp. 4188–4196, 2016.

Li, M., Qin, Z., Jiao, Y., Yang, Y., Wang, J., Wang, C., Wu, G., and Ye, J. Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning. In *The World Wide Web Conference*, pp. 983–994. ACM, 2019.

Macchi, O. The fermion process—a model of stochastic point process with repulsive points. In *Transactions of the Seventh Prague Conference on Information Theory, Statistical Decision Functions, Random Processes and of the 1974 European Meeting of Statisticians*, pp. 391–398. Springer, 1977.

Mahajan, A., Rashid, T., Samvelyan, M., and Whiteson, S. Maven: Multi-agent variational exploration. In *Advances in Neural Information Processing Systems*, pp. 7611–7622, 2019.

Matignon, L., Laurent, G. J., and Le Fort-Piat, N. Hysteretic q-learning: an algorithm for decentralized reinforcement learning in cooperative multi-agent teams. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 64–69. IEEE, 2007.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533, 2015.

Noble, B., Daniel, J. W., et al. *Applied linear algebra*, volume 3. Prentice-Hall New Jersey, 1988.

Oliehoek, F. A., Spaan, M. T., and Vlassis, N. Optimal and approximate q-value functions for decentralized pomdps. *Journal of Artificial Intelligence Research*, 32:289–353, 2008.

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

Osogami, T. and Raymond, R. Determinantal reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 4659–4666, 2019.

Panait, L. and Luke, S. Cooperative multi-agent learning: The state of the art. *Autonomous agents and multi-agent systems*, 11(3):387–434, 2005.

Pang, T., Xu, K., Du, C., Chen, N., and Zhu, J. Improving adversarial robustness via promoting ensemble diversity. In *International Conference on Machine Learning*, pp. 4970–4979, 2019.

Peng, P., Wen, Y., Yang, Y., Yuan, Q., Tang, Z., Long, H., and Wang, J. Multiagent bidirectionally-coordinated nets: Emergence of human-level coordination in learning to play starcraft combat games. *arXiv preprint arXiv:1703.10069*, 2017.

Rashid, T., Samvelyan, M., De Witt, C. S., Farquhar, G., Foerster, J., and Whiteson, S. Qmix: monotonic value function factorisation for deep multi-agent reinforcement learning. *arXiv preprint arXiv:1803.11485*, 2018.

Samvelyan, M., Rashid, T., de Witt, C. S., Farquhar, G., Nardelli, N., Rudner, T. G. J., Hung, C.-M., Torr, P. H. S., Foerster, J., and Whiteson, S. The StarCraft Multi-Agent Challenge. *CoRR*, abs/1902.04043, 2019.

Shafarevich, I. R. and Remizov, A. O. *Linear algebra and geometry*. Springer Science & Business Media, 2012.

Sharghi, A., Borji, A., Li, C., Yang, T., and Gong, B. Improving sequential determinantal point processes for supervised video summarization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 517–533, 2018.

Son, K., Kim, D., Kang, W. J., Hostallero, D. E., and Yi, Y. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:1905.05408*, 2019.

Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zambaldi, V., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J. Z., Tuyls, K., et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017.

Tan, M. Multi-agent reinforcement learning: independent versus cooperative agents. In *International Conference on Machine Learning (ICML)*, pp. 330–337, 1993.

Watkins, C. J. and Dayan, P. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

Wei, E., Wicke, D., Freelan, D., and Luke, S. Multiagent soft q-learning. In *2018 AAAI Spring Symposium Series*, 2018.

Yang, Y., Yu, L., Bai, Y., Wen, Y., Zhang, W., and Wang, J. A study of ai population dynamics with million-agent reinforcement learning. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2133–2135. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

Yang, Y., Tutunov, R., Sakulwongtana, P., Ammar, H. B., and Wang, J. Alpha-alpha-rank: Scalable multi-agent evaluation through evolution. *AAMAS 2020*, 2019.

Zhou, M., Chen, Y., Wen, Y., Yang, Y., Su, Y., Zhang, W., Zhang, D., and Wang, J. Factorized q-learning for large-scale multi-agent systems. In *Proceedings of the First International Conference on Distributed Artificial Intelligence*, pp. 1–7, 2019.