Multi-agent Determinantal Q-Learning

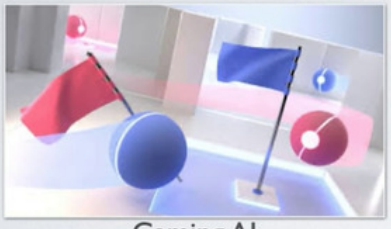
Yaodong Yang*[1,2], Ying Wen*[1,2], Liheng Chen[3], Jun Wang[2], Kun Shao[1], David Mguni[1], Weinan Zhang[3]

[1]*Huawei Technology R&D UK*
[2]*University College London*
[3]*Shanghai Jiao Tong University*

06/2020

Multi-Agent Reinforcement Learning for Real-World Applications

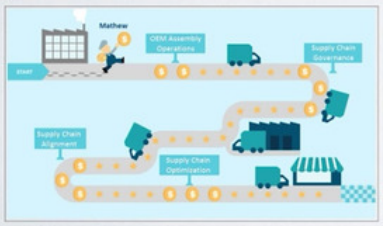
- MARL answers the question of how each agent should behave optimally when interacting with the others.



Gaming AI

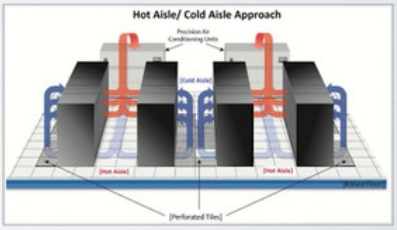
Robotics

Autonomous driving



Supply chain optimisation

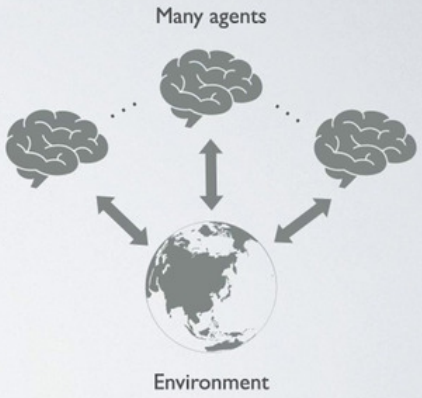
Network control



Data centre cooling

Problem Formulation: Multi-agent Reinforcement Learning

- **Modelled by a Stochastic Game** $(\mathcal{S}, \mathcal{A}^{\{1,\ldots,n\}}, \mathcal{R}^{\{1,\ldots,n\}}, \mathcal{T}, \mathcal{P}_0, \gamma)$
  - $\mathcal{S}$ denotes the state space,
  - $\boldsymbol{\mathcal{A}}$ is the joint-action space $\mathcal{A}^1 \times \ldots \times \mathcal{A}^n$,
  - $\mathcal{R}^i = \mathcal{R}^i(s, a^i, \boldsymbol{a}^{-i})$ is the reward function for the i-th agent,
  - $\mathcal{T} : \mathcal{S} \times \boldsymbol{\mathcal{A}} \times \mathcal{S} \to [0,1]$ is the transition function based on the joint action,
  - $\mathcal{P}_0$ is the distribution of the initial state, γ is a discount factor.
  - **Special case:** $n = 1 \to$ single-agent MDP, $|\mathcal{S}| = 1 \to$ normal-form game
  - **Dec-POMDP**: state is only partially observed, agents share the same reward.



- **Each agent tries to maximise its expected long-term reward:**

$$V_{i,\boldsymbol{\pi}}(s) = \sum_{t=0}^{\infty} \gamma^t \mathbf{E}_{\boldsymbol{\pi},\mathcal{P}} \left\{ R_{i,t} \,|\, s_0 = s, \boldsymbol{\pi} \right\}, \boldsymbol{\pi} = [\pi_1, \ldots, \pi_N]$$

$$Q_{j,\boldsymbol{\pi}}(s, \boldsymbol{a}) = R_j(s, \boldsymbol{a}) + \gamma \mathbf{E}_{s' \sim p} \left[v_{j,\boldsymbol{\pi}}(s') \right]$$

Solution to Multi-Agent RL

- **Value-based method:**
  - The sense of optimality changes, now it depends on other agents !

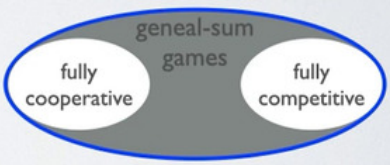
$$\pi_{i,t}(s,\cdot) = \mathbf{solve}_i\{Q_{\cdot,t}(s_t,\cdot)\}$$

$$Q_{i,t+1}(s_k,\boldsymbol{\pi}_t) = Q_{i,t}(s_t,\boldsymbol{\pi}_t) + \alpha\left[R_{i,t+1} + \gamma \cdot \mathbf{eval}_i\{Q_{\cdot,t}(s_{t+1},\cdot)\} - Q_{i,t}(s_t,\boldsymbol{\pi}_t)\right]$$

  - **Fully-cooperative game**: agents share the same reward function

$$\mathbf{solve}_i\{Q_{\cdot,t}(s_t,\cdot)\} = \arg\max_{a_i}\left(\max_{a^{-i}} Q_{i,t}(s_t,a^i,a^{-i})\right)$$

$$\mathbf{eval}_i\{Q_{\cdot,t}(s_{t+1},\cdot)\} = \max_{\boldsymbol{a}} Q_{i,t}(s_{t+1},\boldsymbol{a})$$



  - **Fully-competitive game**: sum of agents' reward is zero

$$\mathbf{solve}_i\{Q_{\cdot,t}(s_t,\cdot)\} = \arg\max_{a_i}\left(\min_{a^{-i}} Q_{i,t}(s_t,a^i,a^{-i})\right)$$

$$\mathbf{eval}_i\{Q_{\cdot,t}(s_{t+1},\cdot)\} = \max_{a_i}\min_{a^{-i}} Q_{i,t}(s_t,a^i,a^{-i})$$

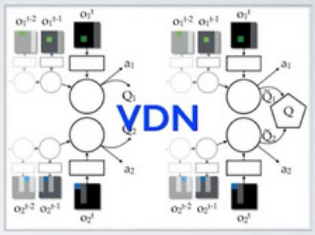
  - In this work, we focus on cooperative MARL with partial observability.

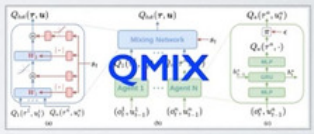
Cooperative Multi-agent Reinforcement Learning

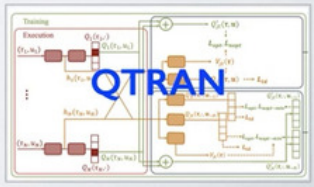
- Decentralising the joint Q-function is an essential approach to scale MARL.
  - A task is decentralisable if

$$\arg\max_{\boldsymbol{a}} Q^{\pi}(\boldsymbol{s}, \boldsymbol{a}) = \begin{bmatrix} \arg\max_{a_1} Q_1\left(s_1, a_1\right) \\ \vdots \\ \arg\max_{a_N} Q_N\left(s_N, a_N\right) \end{bmatrix}$$

  - VDN [Sunehag 2017]: $Q^{\pi}(s, \boldsymbol{a}) := \sum_{i=1}^{N} Q_i\left(s_i, a_i\right)$
  - QMIX [Rashid 2018]: $\frac{\partial Q^{\pi}(s, \boldsymbol{a})}{\partial Q_i\left(s_i, a_i\right)} \geq 0, \forall i \in \mathcal{N}$
  - QTRAN [Kyunghwan 2019]: $\sum_{i=1}^{N} Q_i\left(s_i, a_i\right) - Q^{\pi}(s, \boldsymbol{a}) + V_{\boldsymbol{\pi}}(a) = 0$ if $\boldsymbol{a} = \boldsymbol{a}^*$
    ≥ 0 if $\boldsymbol{a} \neq \boldsymbol{a}^*$



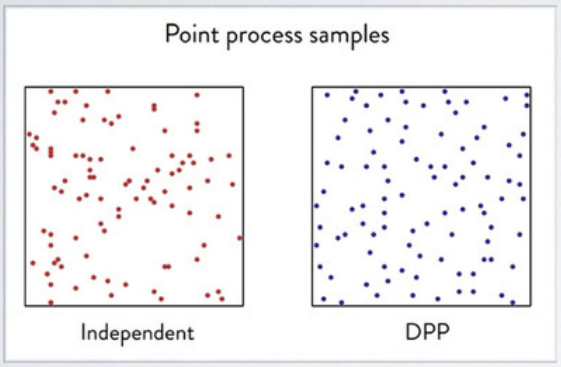


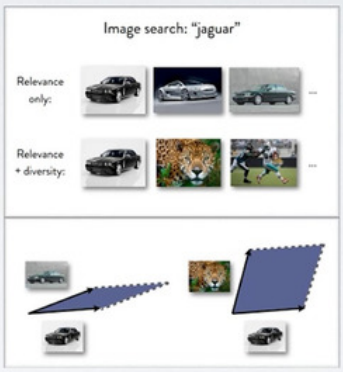


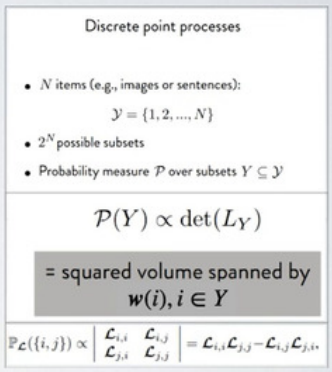
Can we remove the these special structural constraints on $Q^{\pi}(s, \boldsymbol{a})$?

Determinantal Point Process: A Probabilistic Framework for Diversity

- Intuition: if the agents learn diverse behaviours, then $Q^{\pi}(s, \boldsymbol{a})$ can be naturally decentralised.
  - Determinantal Point Process (DPP) [Alex Kulesza 2013] : a point process parameterised by a distance kernel.







$$\mathbb{P}_{\mathscr{L}}(\boldsymbol{Y} = Y) \propto \det(\mathscr{L}_Y) = \mathrm{Vol}^2\left(\{\boldsymbol{w}_i\}_{i \in Y}\right)$$

Q-DPP: A New Function Approximator for Cooperative MARL

- **Q-Determinantal Point Process (Q-DPP): a new function approximator for the joint Q-function.**
  - The DPP kernel $\mathscr{L}$ can represent the joint Q-function $Q^{\pi}(\boldsymbol{o}, \boldsymbol{a})$.

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,*

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

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,

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

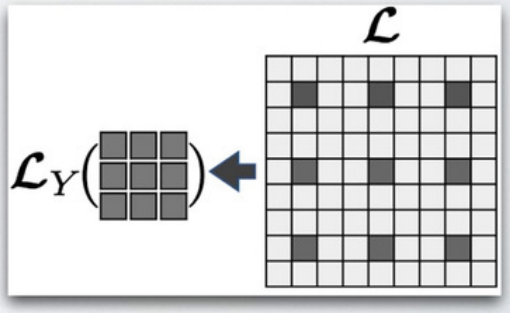
Q-DPP, denoted by $\tilde{\mathbb{P}}$, defines a probability measure over the valid subsets $Y \in \mathcal{C}(\boldsymbol{o}) \subseteq \mathcal{Y}$. Let $\boldsymbol{Y}$ be a random subset drawn according to $\tilde{\mathbb{P}}$, its probability distribution is defined:

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

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 $\mathcal{Y}$.

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

Example: *single-state three-player three-action task, the probability of all agents select their second action is represented by the following sampling process of DPP.*



Q-DPP: A New Function Approximator for Cooperative MARL

- **Q-Determinantal Point Process (Q-DPP): a new function approximator for the joint Q-function.**
  - The DPP kernel $\mathscr{L}$ can represent both quality (maximising reward) & diversity (different behaviours).

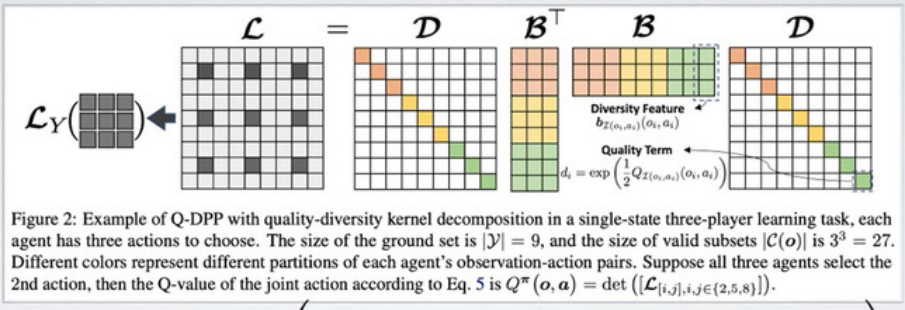


Figure 2: Example of Q-DPP with quality-diversity kernel decomposition in a single-state three-player learning task, each agent has three actions to choose. The size of the ground set is $|\mathcal{Y}| = 9$, and the size of valid subsets $|\mathcal{C}(\boldsymbol{o})|$ is $3^3 = 27$. Different colors 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 $Q^{\boldsymbol{\pi}}(\boldsymbol{o}, \boldsymbol{a}) = \det\left([\mathcal{L}_{[i,j],i,j\in\{2,5,8\}}]\right)$.

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

$$= \log\left(\mathrm{tr}\left(\mathscr{D}_Y^\top\mathscr{D}_Y\right)\det\left(\mathscr{B}_Y^\top\mathscr{B}_Y\right)\right) \quad \text{quality-diversity decomposition}$$

$$= \sum_{i=1}^{N} Q_{\mathcal{I}(o_i^t,a_i)}(o_i,a_i) + \log\det\left(\mathscr{B}_Y^\top\mathscr{B}_Y\right)$$

quality | diversity

Sampling from Q-DPP through Orthogonal Projection

- **Q-DPP Sampler: each agent's decision making is equivalent to sampling from the Q-DPP.**
  - Kernel $\mathscr{L}$ can be learned by Q-learning, but sampling from Q-DPP is challenging since $|\mathscr{C}(\boldsymbol{o})| = |\mathscr{A}|^N$.

$$\tilde{\mathbb{P}}_{\mathscr{L}}\big(\boldsymbol{Y} = Y \,|\, \boldsymbol{Y} \in \mathscr{C}(\boldsymbol{o})\big) := \frac{\det\left(\boldsymbol{\mathscr{W}}_Y \boldsymbol{\mathscr{W}}_Y^\top\right)}{\sum_{Y' \in \mathscr{C}(o)} \det\left(\boldsymbol{\mathscr{W}}_{Y'} \boldsymbol{\mathscr{W}}_{Y'}^\top\right)}$$

  - Best DPP sampler needs $\mathcal{O}\big(|\mathscr{A}|^{NP}\big)$, we instead propose a biased sampler and then control the bias.
    - Sample each partition sequentially, then the size of ground set is $|\mathscr{C}(\boldsymbol{o})| = |\mathscr{A}|^N \rightarrow |\mathscr{A}|N$.
    - The validity comes from the fact that Gram-Schmidt process preserve the volume.

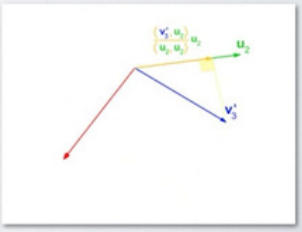
$$\det(\boldsymbol{\mathscr{W}}_Y \boldsymbol{\mathscr{W}}_Y^\top) = \prod_{i=1}^{M} \left\| \mathrm{LI}_{\mathscr{U}_i}(\boldsymbol{w}_i) \right\|^2, \; \mathscr{U}_i = \mathrm{span}\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{i-1}\}$$

    - Sampling is also made easy if the items in partitions are orthogonalised.

$$\mathbb{P}(Y) \propto \prod_{i \in Y} \|\boldsymbol{w}_i\|^2 = \det(\boldsymbol{\mathscr{W}}_Y \boldsymbol{\mathscr{W}}_Y^\top) \propto \det\left(\boldsymbol{\mathscr{L}}_Y\right)$$

this is what we want!

    - **Remaining issue:** how to control of the bias in the denominator.



Gram-Schmidt Process [wiki]

Controlling the Bias of the Q-DPP Sampler

- Q-DPP sampler, though is linear-time, is biased in that samples have larger probability.

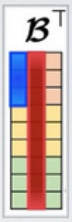
DEF Orthogonalizing-Sampler $(\mathcal{Y}, \boldsymbol{\mathcal{D}}, \boldsymbol{\mathcal{B}}, \boldsymbol{o})$:
Init: $\boldsymbol{b}_j \leftarrow \boldsymbol{\mathcal{B}}_{[:,j]}, Y \leftarrow \emptyset, B \leftarrow \emptyset, J \leftarrow \emptyset$.
for each partition $\mathcal{Y}_i$ **do**
  Define $\forall (o,a) \in \mathcal{Y}_i(o_i)$
  $q(o,a) := \left\|\boldsymbol{b}_{\mathcal{J}(o,a)}\right\|^2 \exp\left(\boldsymbol{\mathcal{D}}_{\mathcal{J}(o,a),\mathcal{J}(o,a)}\right)$.
  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 $\boldsymbol{b}_j = \Pi_{\text{span}\{B\}}(\boldsymbol{b}_j), \forall j \in \{1, ..., M\} - J$
end for
Return: Y.

- Q-DPP defines the denominator $\sum_{Y\in\mathcal{C}(o)} \det\left(\boldsymbol{\mathcal{W}}_Y \boldsymbol{\mathcal{W}}_Y^\top\right)$.
- Inspired by P-DPP [Celis 2018], we can control the level of violation by bounding the singular value of each partition, such that, $\mathbb{P}(Y) \leq 1/\delta^N \cdot \tilde{\mathbb{P}}_{\mathscr{L}}(\boldsymbol{Y} = Y)$.

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., $\min_{i\in\{1,\ldots,N\}} \hat{\sigma}_{i,j}^2/\delta \geq \sigma_j^2$ holds.*



- Total complexity $\mathcal{O}(N^2\,|\,\mathscr{A}\,|\,P)$ with input $\mathcal{O}(N|\,\mathscr{A}\,|\,P)$.
- To maintain such assumption, we adopt an additional loss function of $\max(0,\ \sigma_i^2 - \hat{\sigma}_{i,i}^2/\delta)$ on top of the Q-update.

Learning the Q-DPP Kernel through CTDE Method

- Note that $Q^{\pi}(\boldsymbol{o}, \boldsymbol{a})$ is still learned via normal Q-learning update:

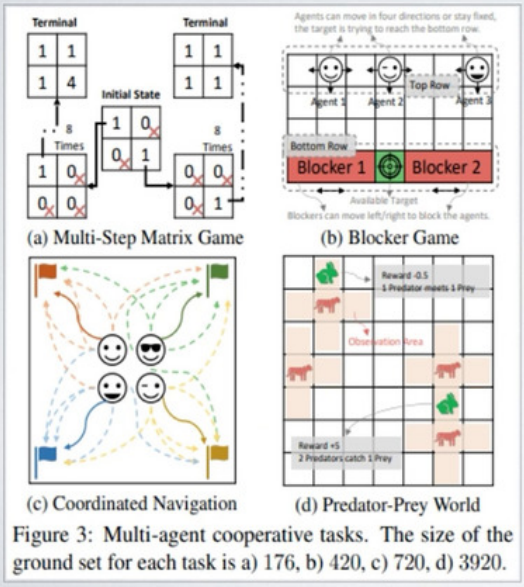
DEF Determinantal-Q-Learning ($\theta = [\theta_{\mathcal{D}}, \theta_{\mathcal{B}}], \mathcal{Y}$):
Init: $\theta^{-} \leftarrow \theta$, $D \leftarrow \emptyset$.
for each time-step **do**
  Collect observations $\boldsymbol{o} = [o_1, \ldots, o_N]$ for all agents.
  $\boldsymbol{a} =$ **Orthogonalizing-Sampler**$(\mathcal{Y}, \theta_{\mathcal{D}}, \theta_{\mathcal{B}}, \boldsymbol{o})$.
  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*
  $a_i^* = \arg\max_{a_i \in \mathcal{A}_i} \left[\left\| \theta^{-}_{\boldsymbol{b}_{\mathcal{J}(o'_i, a_i)}} \right\|^2 \cdot \exp\left(\theta^{-}_{\boldsymbol{\mathcal{D}}_{\mathcal{J}(o'_i, a_i), \mathcal{J}(o'_i, a_i)}}\right) \right].$
  decentralized execution
  // centralized training
  Update θ by minimizing $\mathcal{L}(\theta)$ defined in Eq. 2.
  Update target $\theta^{-} = \theta$ periodically.
end for
Return: $\theta_{\mathcal{D}}, \theta_{\mathcal{B}}$.

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

- Q-DPP is a CTDE method. During training, the joint Q-function is updated. During execution, only each agent's own quality and diversity terms are needed.
- The diversity guarantee ensures the Q-DPP can meet the decentralizability assumption after training.
- The orthogonalizing sampling procedures help agents explore different state-space during training thus boosting the efficiency of exploration.
- Note the Q-DPP can only deal with discrete state-action case. The deep Q-DPP is on its way.

Q-DPP Performance

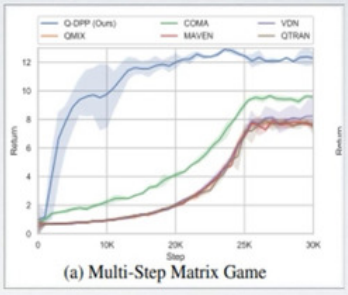
- Performance: the new SOTA algorithm with no neural network needed!



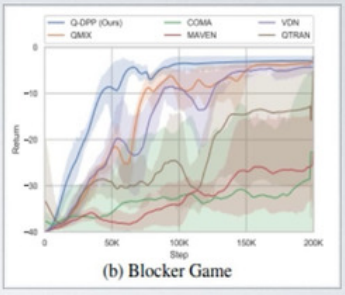
(a) Multi-Step Matrix Game (b) Blocker Game

(c) Coordinated Navigation (d) Predator-Prey World

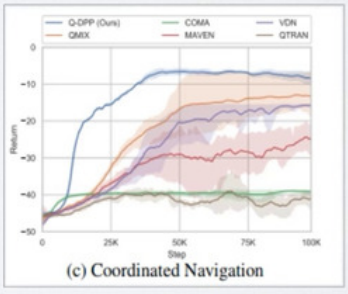
Figure 3: Multi-agent cooperative tasks. The size of the ground set for each task is a) 176, b) 420, c) 720, d) 3920.



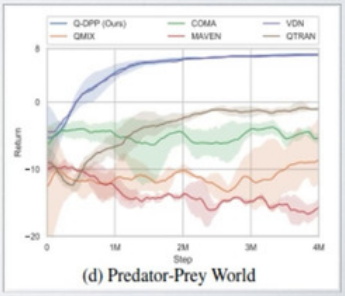
(a) Multi-Step Matrix Game



(b) Blocker Game



(c) Coordinated Navigation

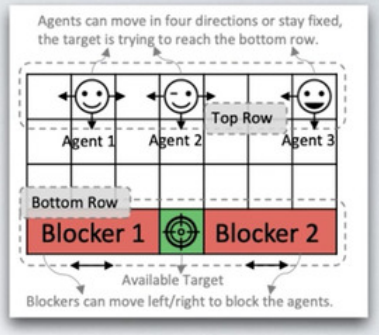


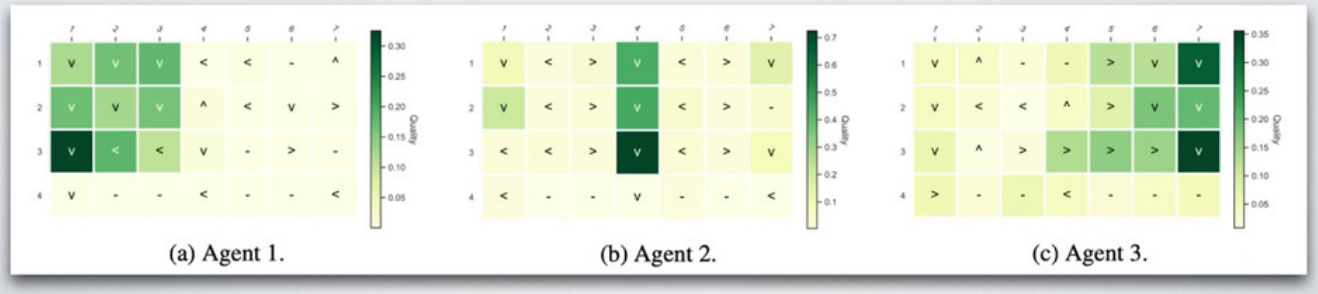
(d) Predator-Prey World

Some Insights: What is the Learned Diverse Behavioural Model

- **The acquired diverse behaviours on Blocker Game.**
  - The optimal action of one agent does not depend on the others
  - The behaviours acquired by different agents are orthogonal.
  - Q-DPP meets the decentralisability assumption naturally.

$$\arg\max_{\boldsymbol{a}} Q^{\pi}(\boldsymbol{s}, \boldsymbol{a}) = \begin{bmatrix} \arg\max_{a_1} Q_1\left(s_1, a_1\right) \\ \vdots \\ \arg\max_{a_N} Q_N\left(s_N, a_N\right) \end{bmatrix}$$





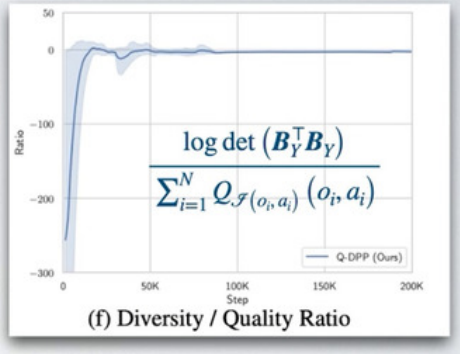
(a) Agent 1. (b) Agent 2. (c) Agent 3.

Some Insights: Quality vs Diversity

- The importance of diversity decrease with training.
- Q-DPP generalizes VDN, QMIX, QTRAN.

$$Q^{\pi}(\boldsymbol{o}, \boldsymbol{a}^*) = \sum_{i=1}^{N} Q_{\mathcal{J}(o_i', a_i^*)}\left(o_i, a_i^*\right) + \cancel{\log\det\left(\boldsymbol{B}_Y^{\top}\boldsymbol{B}_Y = 1\right)}$$

- Recovers VDN: $Q^{\pi}(\boldsymbol{o}, \boldsymbol{a}^*) = \sum_{i=1}^{N} Q_{\mathcal{J}(o_i, a_i^*)}\left(o_i, a_i^*\right)$
- Recovers QMIX: $\dfrac{\partial Q^{\pi}(\boldsymbol{o}, \boldsymbol{a}^*)}{\partial Q_{\mathcal{J}(o_i, a_i^*)}\left(o_i, a_i^*\right)} = 1 \geq 0, \quad \forall \mathcal{J}\left(o_i, a_i^*\right) \in \mathcal{N}$
- Recovers QTRAN: $\sum_{i=1}^{N} Q_i\left(o_i, a_i\right) - Q^{\pi}(\boldsymbol{o}, \boldsymbol{a}) + V(\boldsymbol{o}) = \begin{cases} 0 & \boldsymbol{a} = \boldsymbol{a}^* \\ \geq 0 & \boldsymbol{a} \neq \boldsymbol{a}^* \end{cases}$

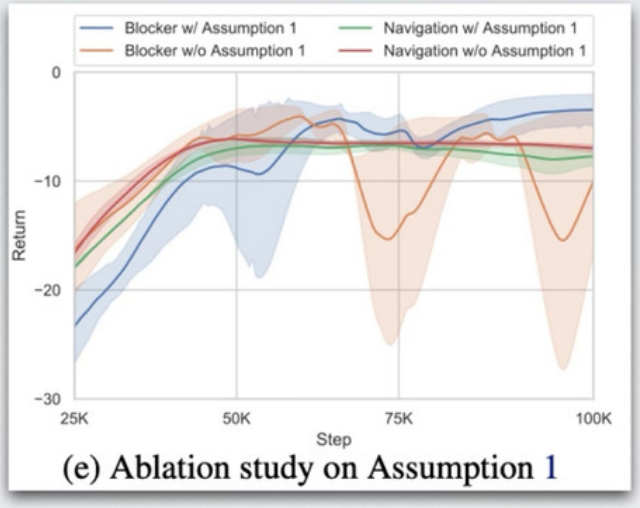


(f) Diversity / Quality Ratio

- Diversity is defined from the perspective of maximising the reward. Agents are still allowed to take the same actions in a task.

Some Insights: the Effect of Assumption 1

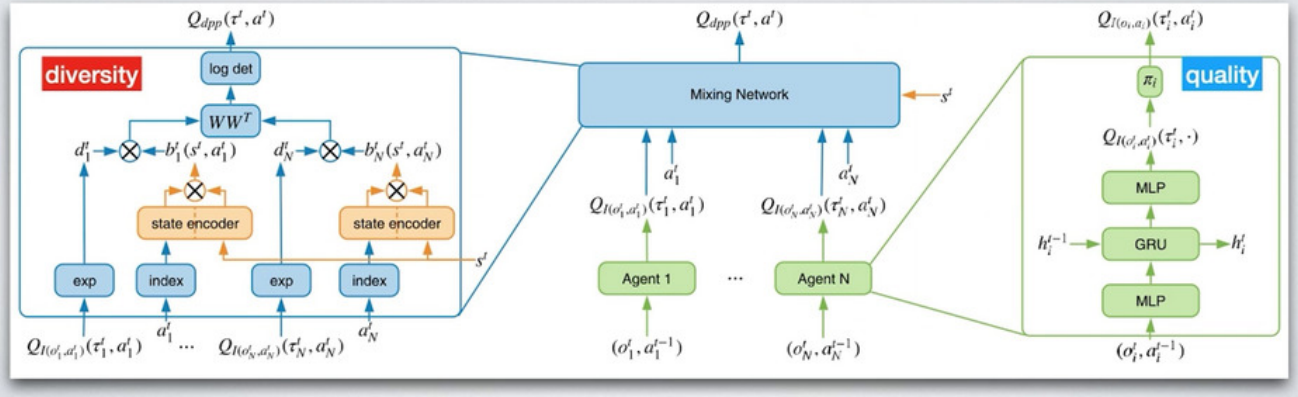
- Assumption 1 reduces the variance of the rewards. It is because it mitigates the over-estimation of the samples from the biased Q-DPP sampler, which helps agents avoid getting over-confident, as a result, trapped in the bad local optima.



(e) Ablation study on Assumption 1

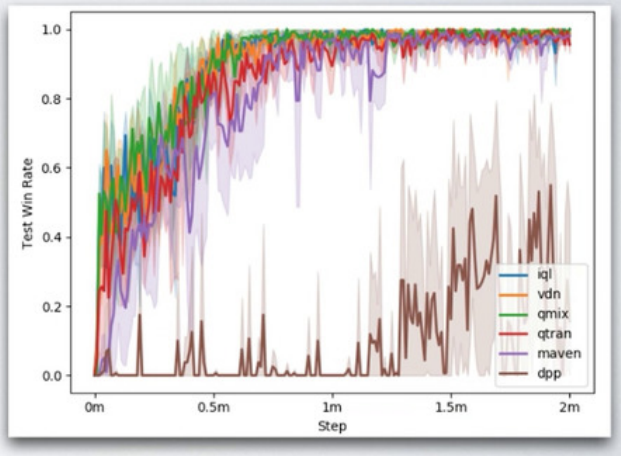
Q-DPP on Continuous State Space

- **Proof of concept on the continuous-state tasks.**
  - Both the quality term and the diversity term can be modelled by a neural network.
  - The quality term that models $Q_i(s, a_i)$ is to the most right.
  - The diversity term is modelled by a directional network and a norm network separately.



Q-DPP on StarCraft II Micro-management

- **Proof of concept on the continuous-state tasks.**
  - How to model continuous set via DPPs are still under development.
  - We believe a full treatment of the idea of **Depp Q-DPP** needs substantial future work.



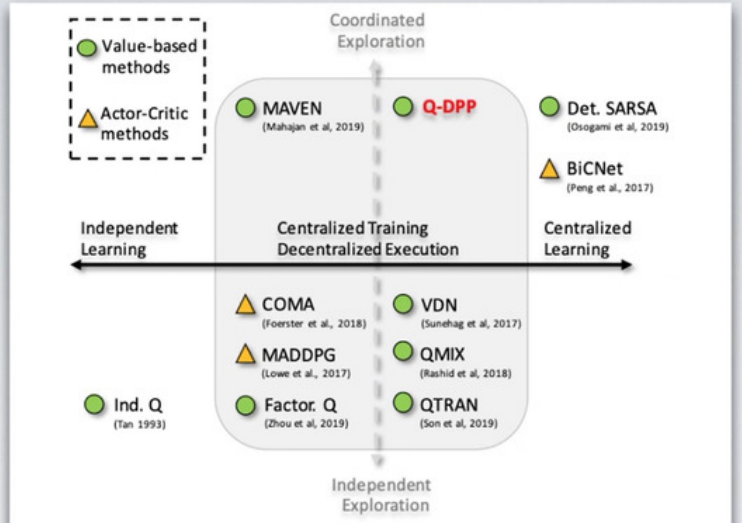


Figure 1: Spectrum of MARL methods on cooperative tasks.

Thank you!