A. Detailed Proofs

A.1. Proof of Proposition 1

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* $\prod_{i=1}^{M} \| \amalg_{\mathcal{U}_i}(\boldsymbol{w}_i)\|^2 = \det(\boldsymbol{\mathcal{W}}\boldsymbol{\mathcal{W}}^\top)$.

Proof. Such property has been mentioned in linear algebra textbook, e.g., Chapter 7 in Shafarevich & Remizov (2012). Celis et al. (2018) also gave out a proof by induction[1] in Lemma 3.1. Here we provide our own intuition of such property through the classical Gaussian elimination method.

We first define an orthogonalization operator $\sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_j)$ that takes an input of a vector $\boldsymbol{w}_j \in \mathbb{R}^P$ and outputs another vector that is orthogonal to a given vector $\boldsymbol{w}_i \in \mathbb{R}^P$ by

$$\sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_j) := \boldsymbol{w}_j - \boldsymbol{w}_i\langle \boldsymbol{w}_i, \boldsymbol{w}_j\rangle / \|\boldsymbol{w}_i\|^2 \,. \tag{13}$$

Based on the Eq. 13, we know that $\forall \boldsymbol{w}_i, \boldsymbol{w}_j, \boldsymbol{w}_k \in \mathbb{R}^P$,

$$\sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_j + \boldsymbol{w}_k) = \sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_j) + \sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_k).$$

Besides, we have two properties for the orthogonalization operator that will be used later; we present as lemmas.

Lemma 1 (Change of Projection Base)**.** *Let* $\boldsymbol{w}_i$*,* $\boldsymbol{w}_j$*,* $\boldsymbol{w}_k \in \mathbb{R}^P$*, we have* $\boldsymbol{w}_j \cdot \sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_k)^\top = \sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_j) \cdot \sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_k)^\top$.

Proof. Based the definition of Eq. 13, one can easily write that the left hand side equals to the right hand side. ■

Lemma 2 (Subspace Orthogonalization)**.** *Let* $\boldsymbol{w}_i$*,* $\boldsymbol{w}_j$*,* $\boldsymbol{w}_k \in \mathbb{R}^P$*, we have* $\sqcap_{\sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_j)}(\boldsymbol{w}_k) \cdot \sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_k)^\top = \left\|\amalg_{\mathcal{U}_k}(\boldsymbol{w}_k)\right\|^2$ *where* $\mathcal{U}_k = \text{span}\{\boldsymbol{w}_i, \boldsymbol{w}_j\}$.

Proof. The left-hand side of equation can be written by

$$\begin{aligned}
&\sqcap_{\sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_j)}(\boldsymbol{w}_k) \cdot \sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_k)^\top \\
=&\Big(\boldsymbol{w}_k - \sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_j)\frac{\langle \sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_j), \boldsymbol{w}_k\rangle}{\|\sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_j)\|^2)}\Big) \cdot \Big(\boldsymbol{w}_k - \boldsymbol{w}_i\frac{\langle \boldsymbol{w}_i, \boldsymbol{w}_k\rangle}{\|\boldsymbol{w}_i\|^2}\Big)^\top \\
=& \boldsymbol{w}_k\boldsymbol{w}_k^\top - \frac{\sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_j)\cdot \boldsymbol{w}_k^\top}{\|\sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_j)\|}\langle \frac{\sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_j)}{\|\sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_j)\|}, \boldsymbol{w}_k\rangle \\
&\quad - \frac{\boldsymbol{w}_k \cdot \boldsymbol{w}_i^\top}{\|\boldsymbol{w}_i\|}\langle \frac{\boldsymbol{w}_i^\top}{\|\boldsymbol{w}_i\|}, \boldsymbol{w}_k\rangle \,.
\end{aligned} \tag{14}$$

On the other hand, $\amalg_{\mathcal{U}_k}(\boldsymbol{w}_k)$ represents the orthogonal projection of $\boldsymbol{w}_k$ to the subspace that is spanned by $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$. Since $\left\{\frac{\boldsymbol{w}_i}{\|\boldsymbol{w}_i\|}, \frac{\sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_j)}{\|\sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_j)\|}\right\}$ form a set of orthornormal basis for the subspace $\mathcal{U}_k = \text{span}\{\boldsymbol{w}_i, \boldsymbol{w}_j\}$, according to the definition of $\amalg_{\mathcal{U}_k}(\boldsymbol{w}_k)$ in Section 3.5, we can write $\amalg_{\mathcal{U}_k}(\boldsymbol{w}_k)$ as $\boldsymbol{w}_k$ minus the projection of $\boldsymbol{w}_k$ on the subspace that is spanned by $\left\{\frac{\boldsymbol{w}_i}{\|\boldsymbol{w}_i\|}, \frac{\sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_j)}{\|\sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_j)\|}\right\}$, i.e.,

$$\begin{aligned}
\amalg_{\mathcal{U}_k}(\boldsymbol{w}_k) = \boldsymbol{w}_k \;&-\; \langle \boldsymbol{w}_k, \frac{\boldsymbol{w}_i}{\|\boldsymbol{w}_i\|}\rangle \frac{\boldsymbol{w}_i}{\|\boldsymbol{w}_i\|} \\
&-\; \langle \boldsymbol{w}_k, \frac{\sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_j)}{\|\sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_j)\|}\rangle \frac{\sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_j)}{\|\sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_j)\|}.
\end{aligned} \tag{15}$$

Under the orthonormal property of $\frac{\boldsymbol{w}_i}{\|\boldsymbol{w}_i\|} \cdot \frac{\sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_j)}{\|\sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_j)\|}^\top = 0$ and $\left\|\frac{\boldsymbol{w}_i}{\|\boldsymbol{w}_i\|}\right\|^2 = \left\|\frac{\sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_j)}{\|\sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_j)\|}\right\|^2 = 1$, finally, squaring the Eq. 15 from both sides leads us to the Eq. 14, i.e., $\left\|\amalg_{\mathcal{U}_k}(\boldsymbol{w}_k)\right\|^2 = \sqcap_{\sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_j)}(\boldsymbol{w}_k) \cdot \sqcap_{\boldsymbol{w}_i}(\boldsymbol{w}_k)^\top$. ■

Assuming $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_M\}$ being the rows of $\boldsymbol{\mathcal{W}}$, then applying the Gram-Schmidt orthogonalization process gives

$$\text{Gram-Schmidt}\Big(\{\boldsymbol{w}_i\}_{i=1}^M\Big) = \Big\{\amalg_{\mathcal{U}_i}(\boldsymbol{w}_i)\Big\}_{i=1}^M$$

where $\mathcal{U}_i = \text{span}\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{i-1}\}$. Note that we don't consider normalizing each $\amalg_{\mathcal{U}_i}(\boldsymbol{w}_i)$ in this work.

In fact, the effect on the Gram matrix determinant $\det(\boldsymbol{\mathcal{W}}\boldsymbol{\mathcal{W}}^\top)$ of applying the Gram-Schmidt process on the rows of $\boldsymbol{\mathcal{W}}$ is equivalent to applying Gaussian elimination (Noble et al., 1988) to transform the Gram matrix to be upper triangular. Since adding a row/column of a matrix multiplied by a scalar to another row/column of that matrix will not change the determinant value of the original matrix (Noble et al., 1988), Gaussian elimination, so as the Gram-Schmidt process, preserves the determinant.

To illustrate the above equivalence, we demonstrate the Gaussian elimination process step-by-step on the case of $M = 3$, the determinant of such a Gram matrix is

$$\det\left(\boldsymbol{\mathcal{W}}\boldsymbol{\mathcal{W}}^\top\right) = \det\begin{pmatrix} \boldsymbol{w}_1\boldsymbol{w}_1^\top & \boldsymbol{w}_1\boldsymbol{w}_2^\top & \boldsymbol{w}_1\boldsymbol{w}_3^\top \\ \boldsymbol{w}_2\boldsymbol{w}_1^\top & \boldsymbol{w}_2\boldsymbol{w}_2^\top & \boldsymbol{w}_2\boldsymbol{w}_3^\top \\ \boldsymbol{w}_3\boldsymbol{w}_1^\top & \boldsymbol{w}_3\boldsymbol{w}_2^\top & \boldsymbol{w}_3\boldsymbol{w}_3^\top \end{pmatrix}. \tag{16}$$

To apply Gaussian elimination to turn the Gram matrix to be upper triangular, first, we multiply the 1-st row by $-\frac{\boldsymbol{w}_2\boldsymbol{w}_1^\top}{\boldsymbol{w}_1\boldsymbol{w}_1^\top}$ and then add the result to the 2-nd row; without affecting the determinant, we have the 2-nd row transformed into

$$\begin{aligned}
&\Big[0, \boldsymbol{w}_2\boldsymbol{w}_2^\top - \frac{\boldsymbol{w}_2\boldsymbol{w}_1^\top}{\boldsymbol{w}_1\boldsymbol{w}_1^\top}\boldsymbol{w}_1\boldsymbol{w}_2^\top, \boldsymbol{w}_2\boldsymbol{w}_3^\top - \frac{\boldsymbol{w}_2\boldsymbol{w}_1^\top}{\boldsymbol{w}_1\boldsymbol{w}_1^\top}\boldsymbol{w}_1\boldsymbol{w}_3^\top\Big] \\
=&\Big[0, \boldsymbol{w}_2 \cdot \sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_2)^\top, \boldsymbol{w}_3 \cdot \sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_2)^\top\Big] \\
=&\Big[0, \sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_2) \cdot \sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_2)^\top, \sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_3) \cdot \sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_2)^\top\Big]. \quad (\textit{Lemma 1})
\end{aligned} \tag{17}$$

[1] We believe their proof is a special case, as interchanging the order of rows can actually change the determinant value, i.e., $\det(WW^\top) \neq \begin{bmatrix} w_k \\ W' \end{bmatrix}\begin{bmatrix} w_k^\top & W'^\top \end{bmatrix}$ where the row vectors are denoted as $W = \{w_1, \ldots, w_k\}$ and $W' = \{w_1, \ldots, w_{k-1}\}$.

Similarly, we can apply the same process on the 3-rd row, which can be written as

$$\begin{aligned}&\left[0, \boldsymbol{w}_3\boldsymbol{w}_2^\top - \frac{\boldsymbol{w}_2\boldsymbol{w}_1^\top}{\boldsymbol{w}_1\boldsymbol{w}_1^\top}\boldsymbol{w}_1\boldsymbol{w}_2^\top, \boldsymbol{w}_3\boldsymbol{w}_3^\top - \frac{\boldsymbol{w}_2\boldsymbol{w}_1^\top}{\boldsymbol{w}_1\boldsymbol{w}_1^\top}\boldsymbol{w}_1\boldsymbol{w}_3^\top\right]\\ =&\left[0, \sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_2)\cdot\sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_3)^\top, \sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_3)\cdot\sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_3)^\top\right].\end{aligned} \tag{18}$$

To make $\boldsymbol{\mathcal{W}}\boldsymbol{\mathcal{W}}^\top$ upper triangular, we need to make the 2-nd element in the 3-rd row be zero. To achieve that, we multiply $-\frac{\sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_2)\cdot\sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_3)^\top}{\sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_2)\cdot\sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_2)^\top}$ to Eq. 17 and add the multiplication to Eq. 18, and the 3-rd row can be further transformed into

$$\begin{aligned}&\Big[0, 0, \sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_3)\cdot\sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_3)^\top - \\ &\qquad\frac{\sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_2)\cdot\sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_3)^\top}{\sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_2)\cdot\sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_2)^\top}\sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_3)\cdot\sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_2)^\top\Big]\\ =&\left[0, 0, \sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_3)\cdot\sqcap_{\sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_2)}\big(\sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_3)\big)^\top\right]\\ =&\left[0, 0, \sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_3)\cdot\sqcap_{\sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_2)}\Big(\boldsymbol{w}_3 - \boldsymbol{w}_1\frac{\langle\boldsymbol{w}_1,\boldsymbol{w}_3\rangle}{\|\boldsymbol{w}_1\|^2}\Big)^\top\right]\\ =&\Big[0, 0, \sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_3)\cdot\Big(\sqcap_{\sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_2)}(\boldsymbol{w}_3)\\ &\qquad-\sqcap_{\sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_2)}\big(\boldsymbol{w}_1\frac{\langle\boldsymbol{w}_1,\boldsymbol{w}_3\rangle}{\|\boldsymbol{w}_1\|^2}\big)\Big)^\top\Big]\\ =&\left[0, 0, \sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_3)\cdot\sqcap_{\sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_2)}(\boldsymbol{w}_3)^\top\right]\\ =&\left[0, 0, \left\|\amalg_{\mathcal{U}_3}(\boldsymbol{w}_3)\right\|^2\right]. \qquad (\textit{Lemma 2})\end{aligned} \tag{19}$$

In the fourth equation of Eq.19, we use the property that $\sqcap_{\boldsymbol{w}_1}(\cdot)\cdot\sqcap_{\sqcap_{\boldsymbol{w}_1}(\cdot)}(\boldsymbol{w}_1)^\top = 0$, i.e., the inner product between a vector and its own orthogonalization equals to zero.

Given the Gran matrix is now upper triangular, by putting Eq. 17 and Eq. 19 into Eq. 16, and define $\mathcal{U}_1 = \emptyset, \mathcal{U}_2 = \{\boldsymbol{w}_1\}, \mathcal{U}_3 = \{\boldsymbol{w}_1, \boldsymbol{w}_2\}$, we can write the determinant to be

$$\begin{aligned}&\det\left(\boldsymbol{\mathcal{W}}\boldsymbol{\mathcal{W}}^\top\right)\\ =&\det\begin{pmatrix}\boldsymbol{w}_1\boldsymbol{w}_1^\top & \boldsymbol{w}_1\boldsymbol{w}_2^\top & \boldsymbol{w}_1\boldsymbol{w}_3^\top\\ 0 & \left\|\sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_2)\right\|^2 & \sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_3)\cdot\sqcap_{\boldsymbol{w}_1}(\boldsymbol{w}_2)^\top\\ 0 & 0 & \left\|\amalg_{\mathcal{U}_3}(\boldsymbol{w}_3)\right\|^2\end{pmatrix}\\ =&\prod_{i=1}^3\left\|\amalg_{\mathcal{U}_i}(\boldsymbol{w}_i)\right\|^2.\end{aligned}$$

When $M \geq 3$, the consequence of eliminating all j-th elements ($j < i$) in the i-th row of the Gram matrix $\boldsymbol{\mathcal{W}}\boldsymbol{\mathcal{W}}^\top_{(i,j)}$ by Gaussian elimination is equivalent to the i-th step of the Gran-Schmidt process applied on the vector set $\{\boldsymbol{w}_i\}_{i=1}^M$, in other words, the (i,i)-th element of the Gram matrix after Gaussian elimination is essentially the squared norm of $\amalg_{\mathcal{U}_i}(\boldsymbol{w}_i)$. Finally, since the determinant of an upper-triangular matrix is simply the multiplication of its diagonal elements, we have $\prod_{i=1}^M\left\|\amalg_{\mathcal{U}_i}(\boldsymbol{w}_i)\right\|^2$. ■

A.2. Proof of Theorem 1

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. This result can be regarded as a special case of Theorem 3.2 in Celis et al. (2018) when the number of sample from each partition in P-DPP is set to one (please find *Appendix A.3* for the differences between P-DPP and Q-DPP).

Sine our sampling algorithm generates samples with the probability in proportional to the determinant value $\det(\boldsymbol{\mathcal{L}}_Y)$, which is also the nominator in Eq. 4, it is then necessary to bound the denominator of the probability of samples from our proposed sampler so that the error to the exact denominator defined in Eq. 4 can be controlled. We start from the Lemma that is going to be used.

Lemma 3 (Eckart-Young-Mirsky Theorem)**.** *For a real matrix $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{M\times P}$ with $M \geq P$, suppose that $\boldsymbol{\mathcal{W}} = \boldsymbol{U}\Sigma\boldsymbol{V}^\top$ is the singular value decomposition (SVD) of $\boldsymbol{\mathcal{W}}$, then the best rank k approximation to $\boldsymbol{\mathcal{W}}$ under the Frobenius norm $\|\cdot\|_F$ described as*

$$\min_{\boldsymbol{\mathcal{W}}':\mathrm{rank}(\boldsymbol{\mathcal{W}}')=k}\left\|\boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{W}}'\right\|_F^2$$

is given by $\boldsymbol{\mathcal{W}}' = \boldsymbol{\mathcal{W}}^k = \sum_{i=1}^k \sigma_i\boldsymbol{u}_i\boldsymbol{v}_i^\top$ where $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ denote the i-th column of $\boldsymbol{U}$ and $\boldsymbol{V}$ respectively, and,

$$\left\|\boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{W}}^k\right\|_F^2 = \left\|\sum_{i=k+1}^P \sigma_i\boldsymbol{u}_i\boldsymbol{v}_i^\top\right\|_F^2 = \sum_{i=k+1}^P \sigma_i^2.$$

Note that the singular values σ_i in Σ is ranked by size by the SVD procedures such that $\sigma_1 \geq \ldots \geq \sigma_P$. ■

Lemma 4 (Lemma 3.1 in (Deshpande et al., 2006))**.** *For a matrix $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{M\times P}$ with $M \geq P \geq N$, assume $\{\sigma_i\}_{i=1}^P$ are the singular values of $\boldsymbol{\mathcal{W}}$ and $\boldsymbol{\mathcal{W}}_Y$ is the submatrix of $\boldsymbol{\mathcal{W}}$ with rows indexed by the elements in Y, then we have*

$$\sum_{|Y|=N}\det\left(\boldsymbol{\mathcal{W}}_Y\boldsymbol{\mathcal{W}}_Y^\top\right) = \sum_{k_1<\cdots<k_N}\sigma_{k_1}^2\cdots\sigma_{k_N}^2.$$ ■

To stay consistency on notations, we use N for number of agents, M for the size of ground set of Q-DPP, P is the dimension of diverse feature vectors, we assume $M \geq P \geq N$. Let $\boldsymbol{Y}$ be the random variable representing the output of our proposed sampler in Algorithm 1. Since the algorithm visit each partition in Q-DPP sequentially, a sample $\tilde{Y} = \left\{(o_1^1, a_1^1), \ldots, (o_N^{|\mathcal{O}|}, a_N^{|\mathcal{A}|})\right\}$ is therefore an ordered set. Note that the algorithm is agnostic to the partition number (i.e. the agent identity), without losing generality,

we denote the first partition chosen as $\mathcal{Y}_1$. We further denote $\tilde{Y}_i, i \in \{1, \dots, N\}$ as the i-th observation-state pair in $\tilde{Y}$, and $\mathcal{I}(\tilde{Y}_i) \in \{1, \dots, N\}$ denotes the partition number where i-th pair is sampled.

According to the Algorithm 1, at first step, we choose $\mathcal{Y}_1$, and based on the corresponding observation o_1, we then locate the valid subsets $\forall (o, a) \in \mathcal{Y}_i(o_i)$, and finally sample one observation-action pair from the valid set $\mathcal{Y}_i(o_i)$ with probability proportional to the norm of the vector defined in the Line $4-5$ in Algorithm 1, that is,

$$\mathbb{P}(\tilde{Y}_i) \propto \left\|\boldsymbol{w}_{\mathcal{J}(o,a)}\right\|^2 = \left\|\boldsymbol{b}_{\mathcal{J}(o,a)}\right\|^2 \exp\left(\boldsymbol{\mathcal{D}}_{\mathcal{J}(o,a),\mathcal{J}(o,a)}\right). \tag{20}$$

After $\tilde{Y}_i$ is sampled, the algorithm then moves to the next partition and repeat the same process until all N partitions are covered.

The specialty of this sampler is that before sampling at each partition $i \in \{1, \dots, N\}$, the Gram-Schmidt process will be applied to ensure all the rows in the i-th partition of $\boldsymbol{\mathcal{W}}$ to be orthogonal to all previous sampled pairs

$$\boldsymbol{b}_j^i = \mathrm{II}_{\mathrm{span}\{B^i\}}\left(\boldsymbol{b}_j^{i-1}\right), \forall j \in \{1, ..., M\} - J.$$

where $B^i = \{\boldsymbol{b}^t_{\mathcal{J}(o^t,a^t)}\}_{t=1}^{i-1}, J = \{\mathcal{J}(o^t, a^t)\}_{t=1}^{i-1}$. Note that since $\boldsymbol{\mathcal{D}}$ only contributes a scalar to $\boldsymbol{w}_j$, and $\boldsymbol{b}_j$ is a P-dimensional vector same as $\boldsymbol{w}_j$, in practice, the Gram-Schmidt orthorgonalization needs only conducting on $\boldsymbol{b}_j$ in order to make rows of $\boldsymbol{\mathcal{W}}$ mutually orthogonal.

Based on the above sampling process and each time-step i, we can write the probability of getting a sample $\tilde{Y}$ by

$$\begin{aligned}
&\mathbb{P}(\boldsymbol{Y} = \tilde{Y}) \\
&= \mathbb{P}\big(\tilde{Y}_1\big) \prod_{i=2}^{N} \mathbb{P}\big(\tilde{Y}_i \big| \tilde{Y}_1, \dots, \tilde{Y}_{i-1}\big) \\
&= \prod_{i=1}^{N} \frac{\left\| \mathrm{II}_{\mathrm{span}\{B^i\}}\left(\boldsymbol{w}_{\mathcal{I}(\tilde{Y}_i)}\right)\right\|^2}{\sum_{(o,a)\in\mathcal{Y}_{\mathcal{I}(\tilde{Y}_i)}} \left\| \mathrm{II}_{\mathrm{span}\{B^i\}}\left(\boldsymbol{w}_{\mathcal{I}(o,a)}\right)\right\|^2} \\
&= \frac{\prod_{i=1}^{N}\left(\left\| \mathrm{II}_{\mathrm{span}\{B^i\}}\left(\boldsymbol{w}_{\mathcal{I}(\tilde{Y}_i)}\right)\right\|^2\right)}{\prod_{i=1}^{N}\left(\sum_{(o,a)\in\mathcal{Y}_{\mathcal{I}(\tilde{Y}_i)}} \left\| \mathrm{II}_{\mathrm{span}\{B^i\}}\left(\boldsymbol{w}_{\mathcal{I}(o,a)}\right)\right\|^2\right)} \\
&= \frac{\det\left(\boldsymbol{\mathcal{W}}_{\tilde{Y}}\boldsymbol{\mathcal{W}}_{\tilde{Y}}^\top\right)}{\prod_{i=1}^{N}\left(\sum_{(o,a)\in\mathcal{Y}_{\mathcal{I}(\tilde{Y}_i)}} \left\| \mathrm{II}_{\mathrm{span}\{B^i\}}\left(\boldsymbol{w}_{\mathcal{I}(o,a)}\right)\right\|^2\right)},
\end{aligned} \tag{21}$$

where the 4-th equation in Eq. 21 is valid because of Proposition 1.

For each term in the denominator, according to the definition of the operator $\mathrm{II}_{\mathrm{span}\{B^i\}}$, we can rewrite into

$$\sum_{(o,a)\in\mathcal{Y}_{\mathcal{I}(\tilde{Y}_i)}} \left\| \mathrm{II}_{\mathrm{span}\{B^i\}}\left(\boldsymbol{w}_{\mathcal{I}(o,a)}\right)\right\|^2 = \left\|\boldsymbol{\mathcal{W}}_{\mathcal{I}(\tilde{Y}_i)} - \boldsymbol{\mathcal{W}}'_{\mathcal{I}(\tilde{Y}_i)}\right\|_F^2$$

where the rows of $\boldsymbol{\mathcal{W}}_{\mathcal{I}(\tilde{Y}_i)}$ are $\{\boldsymbol{w}_{\mathcal{I}(o,a)}\}_{(o,a)\in\mathcal{Y}_{\mathcal{I}(\tilde{Y}_i)}}$ which are essentially the submatrix of $\boldsymbol{\mathcal{W}}$ that corresponds to partition $\mathcal{I}(\tilde{Y}_i)$, and the rows of $\boldsymbol{\mathcal{W}}'_{\mathcal{I}(\tilde{Y}_i)}$ are the orthogonal projections of $\{\boldsymbol{w}_{\mathcal{I}(o,a)}\}_{(o,a)\in\mathcal{Y}_\mathcal{I}}$ onto $\mathrm{span}\{B^i\}$, and we know $\mathrm{rank}\left(\boldsymbol{\mathcal{W}}'_{\mathcal{I}(\tilde{Y}_i)}\right) = |B^i| = i-1$. According to Lemma 3, with $\hat{\sigma}_{\mathcal{I}(\tilde{Y}_i),k}$ being the k-th singular value of $\boldsymbol{\mathcal{W}}_{\mathcal{I}(\tilde{Y}_i)}$, we know that

$$\left\|\boldsymbol{\mathcal{W}}_{\mathcal{I}(\tilde{Y}_i)} - \boldsymbol{\mathcal{W}}'_{\mathcal{I}(\tilde{Y}_i)}\right\|_F^2 \geq \sum_{k=i}^{P} \hat{\sigma}^2_{\mathcal{I}(\tilde{Y}_i),k} \ . \tag{22}$$

Therefore, we have the denominator of Eq. 21 as:

$$\begin{aligned}
&\prod_{i=1}^{N}\Big(\sum_{(o,a)\in\mathcal{Y}_{\mathcal{I}(\tilde{Y}_i)}} \left\| \mathrm{II}_{\mathrm{span}\{B^i\}}\left(\boldsymbol{w}_{\mathcal{I}(o,a)}\right)\right\|^2\Big) \\
&\geq \prod_{i=1}^{N}\sum_{k=i}^{P} \hat{\sigma}^2_{\mathcal{I}(\tilde{Y}_i),k} \ \geq \ \prod_{i=1}^{N}\sum_{k=i}^{P} \delta \cdot \sigma_k^2 \qquad (\textit{Assumption 1}) \\
&\geq \delta^N \cdot \sum_{k_1<\cdots<k_N} \sigma^2_{k_1}\cdots\sigma^2_{k_N} \\
&= \delta^N \cdot \sum_{Y\subseteq\mathcal{Y}:|Y|=N} \det\left(\boldsymbol{\mathcal{W}}_Y\boldsymbol{\mathcal{W}}_Y^\top\right) \qquad (\textit{Lemma 4}) \\
&\geq \delta^N \cdot \sum_{Y\in\mathcal{C}(\boldsymbol{o})} \det\left(\boldsymbol{\mathcal{W}}_Y\boldsymbol{\mathcal{W}}_Y^\top\right)
\end{aligned} \tag{23}$$

Taking Eq. 23 into Eq. 21, we can obtain that

$$\mathbb{P}(\boldsymbol{Y} = \tilde{Y}) \leq \frac{\delta^N \cdot \det\left(\boldsymbol{\mathcal{W}}_{\tilde{Y}}\boldsymbol{\mathcal{W}}_{\tilde{Y}}^\top\right)}{\sum_{Y\in\mathcal{C}(\boldsymbol{o})}\det\left(\boldsymbol{\mathcal{W}}_Y\boldsymbol{\mathcal{W}}_Y^\top\right)} = 1/\delta^N \cdot \tilde{\mathbb{P}}(\boldsymbol{Y} = \tilde{Y})$$

where $\mathbb{P}(\boldsymbol{Y} = \tilde{Y})$ is the probability of obtaining the sample $\tilde{Y}$ from our proposed sampler and $\tilde{\mathbb{P}}(\boldsymbol{Y} = \tilde{Y})$ is the probability of getting that sample $\tilde{Y}$ under Eq. 4. ■

A.3. Difference between Q-DPP and P-DPP

The design of Q-DPP and its samply-by-projection sampling process is inspired by and based on P-DPP (Celis et al., 2018). However, we would like to highlight the multiple differences in that **1)** P-DPP is designed for modeling the fairness for data summarization whereas Q-DPP serves as a function approximator for the joint Q-function in the context of multi-agent learning; **2)** though we analyze Eq. 21 based on $\boldsymbol{\mathcal{W}}$, the actual orthorgonalziation step of our sampler only needs performing on the vectors of $\boldsymbol{b}_j$ rather than the entire matrix $\boldsymbol{\mathcal{W}}$ due to our unique quality-diversity decomposition on the joint Q-function in Eq. 6; **3)** the set of elements in each partition $\mathcal{Y}_i(o_i)$ of Q-DPP change with the observation at each time-step, while the partitions stay fixed in the case of P-DPP; **4)** the parameters of $\boldsymbol{\mathcal{W}}$ are learned through a trail-and-error multi-agent reinforcement learning process compared to the cases in P-DPP where the kernel is given by hand-crafted features (e.g. SIFT features on images); **5)** we implement the constraint in Assumption 1 via a penalty term during the CTDE learning process, while P-DPP does not consider meeting such assumption through optimization.

A.4. Time Complexity of Algorithm 1

Let's analyze the time complexity of the proposed Q-DPP sampler in steps $1-10$ of Algorithm 1. Given the observation o, and the input matrices $\boldsymbol{\mathcal{D}}, \boldsymbol{\mathcal{B}}$ (whose sizes are $M \times M$, $P \times M$, with $M = |A| \times N$ being the size of all N agents' allowed actions under o and P being the diversity feature dimension), the sampler samples one action for each agent sequentially, so the outer loop of step 3 is $\mathcal{O}(N)$. Within the partition of each agent, step 4 is $\mathcal{O}(P)$, step 5 is $\mathcal{O}(P|A|)$, step 6 is $\mathcal{O}(1)$, so the complexity so far is $\mathcal{O}(NP|A|)$. Computing step 8 for ALL partitions is of $\mathcal{O}(N^2P|A|)^\dagger$. The overall complexity is $\mathcal{O}(N^2P|A|) = \mathcal{O}(NMP)$, since the input is $\mathcal{O}(MP)$ and the agent number N is a constant, our sampler has linear-time complexity with respect to the input, also linear-time with respect to the number of agents. Such argument is in line with the project-and-sample sampler in Celis et al. (2018).

†: In the Gram-Schmidt process, orthogonalizing a vector to another takes $\mathcal{O}(P)$. Considering all valid actions for each agent takes $\mathcal{O}(P|A|)$. Note that while looping over different partitions, the remaining unsampled partitions do not need repeatedly orthogonalizing to all the previous samples, in fact, they only need orthogonalizing to the LATEST sample. In the example of Fig 2, after agent 2 selects action 5, agent 3's three actions only need orthogonalizing to action 5 but not action 2 because it has been performed when the partition of agent 1 was visited. So the total number of orthogonalization is $(N-1)N/2$ across all partitions, leading to $\mathcal{O}(N^2P|A|)$ time for step 8.

B. Experimental Parameter Settings

The hyper-parameters settings for Q-DPP are given in Table 1. For all experiments we update the target networks after every 100 episodes. All activation functions in hidden layers are ReLU. The optimization is conducted using RMSprop with a learning rate of 5×10^{-4} and $\alpha = 0.99$ with no weight decay or momentum.

If not particularly indicated, all the baselines use common settings as listed in Section B. IQL, VDN, QMIX, MAVEN and QTRAN use common individual action-value networks as those used by Q-DPP; each consists of two 32-width hidden layers. The specialized parameter settings for each algorithm are provided in Table 2:

Table 1: Q-DPP Hyper-parameter Settings.

COMMON SETTINGS	VALUE	DESCRIPTION
LEARNING RATE	0.0005	OPTIMIZER LEARNING RATE.
BATCH SIZE	32	NUMBER OF EPISODES TO USE FOR EACH UPDATE.
GAMMA	0.99	LONG TERM DISCOUNT FACTOR.
HIDDEN DIMENSION	64	SIZE OF HIDDEN STATES.
NUMBER OF HIDDEN LAYERS	3	NUMBER OF HIDDEN LAYERS.
TARGET UPDATE INTERVAL	100	INTERVAL OF UPDATING THE TARGET NETWORK.
MULTI-STEP MATRIX GAME		
STEP	40K	MAXIMUM TIME STEPS.
FEATURE MATRIX SIZE	176×32	NUMBER OF OBSERVATION-ACTION PAIR TIMES EMBEDDING SIZE.
INDIVIDUAL POLICY TYPE	RNN	RECURRENT DQN.
EPSILON DECAY SCHEME	LINEAR DECAY FROM 1 TO 0.05 IN 30K STEPS.	
COORDINATED NAVIGATION		
STEP	100K	MAXIMUM TIME STEPS.
FEATURE MATRIX SIZE	720×32	NUMBER OF OBSERVATION-ACTION PAIR TIMES EMBEDDING SIZE.
INDIVIDUAL POLICY TYPE	RNN	RECURRENT DQN.
EPSILON DECAY SCHEME	LINEAR DECAY FROM 1 TO 0.1 IN 10K STEPS.	
BLOCKER GAME		
STEP	200K	MAXIMUM TIME STEPS.
FEATURE MATRIX SIZE	420×32	NUMBER OF OBSERVATION-ACTION PAIR TIMES EMBEDDING SIZE.
INDIVIDUAL POLICY TYPE	RNN	RECURRENT DQN.
EPSILON DECAY SCHEME	LINEAR DECAY FROM 1 TO 0.01 IN 100K STEPS.	
PREDATOR-PREY WORLD (FOUR PREDATORS, TWO PREYS)		
STEP	4M	MAXIMUM TIME STEPS.
FEATURE MATRIX SIZE	3920×32	NUMBER OF OBSERVATION-ACTION PAIR TIMES EMBEDDING SIZE.
INDIVIDUAL POLICY TYPE	FEEDFORWARD	FEEDFORWARD DQN.
EPSILON DECAY SCHEME	LINEAR DECAY FROM 1 TO 0.1 IN 300K STEPS.	

Table 2: Hyper-parameter Settings for Baseline Algorithms.

SETTINGS	VALUE	DESCRIPTION
QMIX		
MONOTONE NETWORK LAYER	2	LAYER NUMBER OF MONOTONE NETWORK.
MONOTONE NETWORK SIZE	64	HIDDEN LAYER SIZE OF MONOTONE NETWORK.
QTRAN		
JOINT ACTION-VALUE NETWORK LAYER	2	LAYER NUMBER OF JOINT ACTION-VALUE NETWORK.
JOINT ACTION-VALUE NETWORK SIZE	64	HIDDEN LAYER SIZE OF JOINT ACTION-VALUE NETWORK.
MAVEN		
z	2	NOISE DIMENSION.
λ_{MI}	0.001	WEIGHT OF MI OBJECTIVE.
λ_{QL}	1	WEIGHT OF QL OBJECTIVE.
ENTROPY REGULARIZATION	0.001	FEEDFORWARD DQN.
DISCRIMINATOR LAYER	1	NUMBER OF DISCRIMINATOR NETWORK LAYER.
DISCRIMINATOR SIZE	32	HIDDEN LAYER SIZE OF DISCRIMINATOR NETWORK.

C. Solution for Continuous States: Deep Q-DPP

Although our proposed Q-DPP serves as a new type of function approximator for the value function in multi-agent reinforcement learning, deep neural networks can also be seamlessly applied on Q-DPP. Specifically, one can adopt deep networks to respectively represent the quality and diversity terms in the kernels of Q-DPP to tackle continuous state-action space, and we name such approach Deep Q-DPP. In Fig. 2, one can think of Deep Q-DPP as modeling $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{B}}$ by neural networks rather than look-up tables. An analogy of Deep Q-DPP to Q-DPP would be Deep Q-learning (Mnih et al., 2015) to Q-learning (Watkins & Dayan, 1992). As the main motivation of introducing Q-DPP is to eliminate structural constraints and bespoke neural architecture designs in solving multi-agent cooperative tasks, we omit the study of Deep Q-DPP in the main body of this paper. Here we demonstrate a proof of concept for Deep Q-DPP and its effectiveness on StarCraft II micro-management tasks (Samvelyan et al., 2019b) as an initiative. However, we do believe a full treatment needs substantial future work.

C.1. Neural Architectures for Deep Q-DPP.

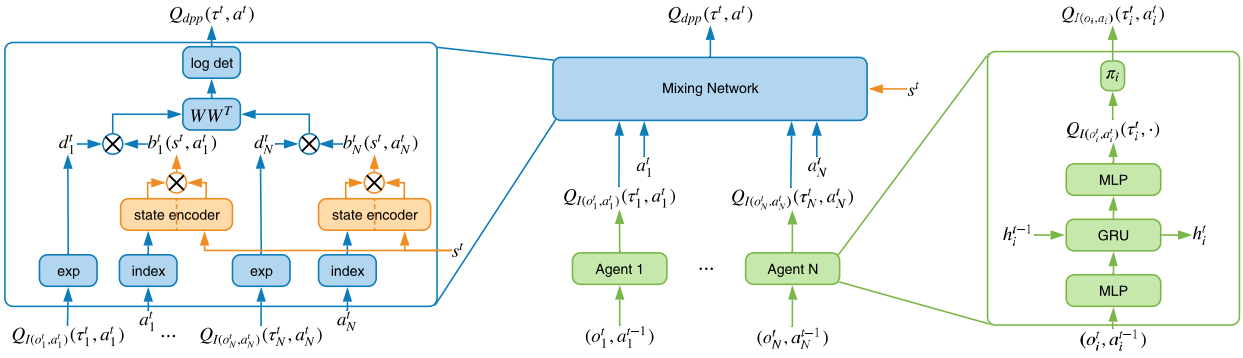
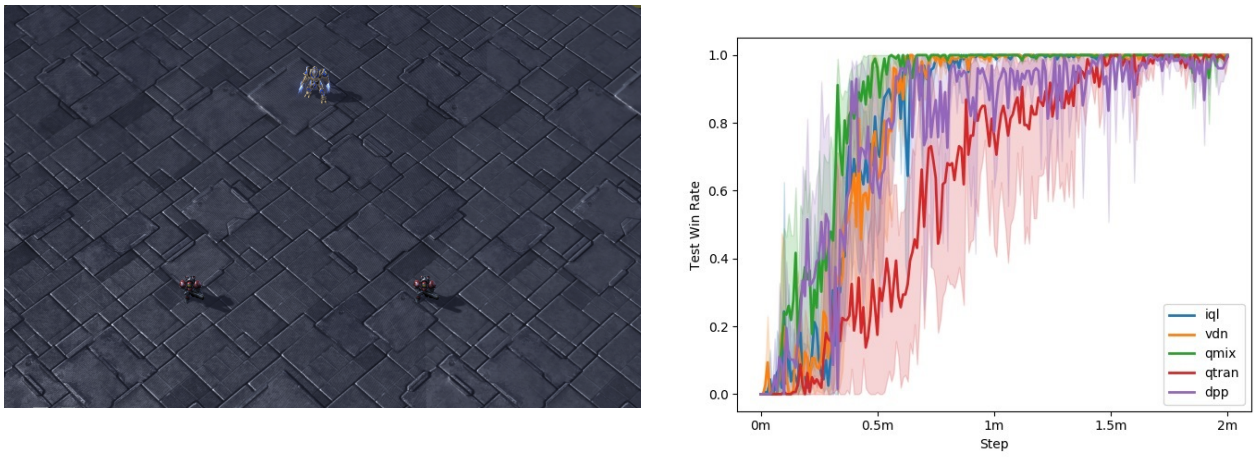


Figure 6: Neural Architecture of Deep Q-DPP. The middle part of the diagram shows the overall architecture of Q-DPP, which consists of each agent's individual Q-networks and a centralized mixing network. Details of the mixing network are presented in the left. We compute the quality term, d_i, by applying the exponential operator on the individual Q-value, and compute the diversity feature term, $\boldsymbol{b}_i$, by index the corresponding vector in $\boldsymbol{\mathcal{B}}$ through the global state s and each action a_i.

A critical advantage of Deep Q-DPP is that it can deal with continuous states/observations. When the input state s is continuous, we first index the raw diversity feature $\boldsymbol{b}_i'$ based on the embedding of discrete action a_i. To integrate the information of the continuous state, we use two multi-layer feed-forward neural networks f_d and f_n, which encodes the direction and norm of the diversity feature separately. f_d outputs a feature vector with same shape as $\boldsymbol{b}_i'$ indicating the **direction**, and f_n outputs a real value for computing the **norm**. In practice, we find modeling the direction and norm of the diversity features separately by two neural networks helps stabilize training, and the diversity feature vector is computed as $\boldsymbol{b_i} = f_d(\boldsymbol{b}_i', s) \times \sigma(f_n(\boldsymbol{b}_i', s))$. Finally, the centralized Q-value can then be computed from d_i and $\boldsymbol{b_i}$ following Eq. 6.

C.2. Experiments on StarCraft II Micro-Management



(a) Scenario Screenshot

(b) 2m_vs_1z

Figure 7: StarCraft II micro-management on the scenario of *2 Marines vs. 1 Zealot* and its performance.

We study one of the simplest continuous state-action micro-management games in StarCraft II in SMAC (Samvelyan et al., 2019b), i.e., **2m_vs_1z**, the screenshots of scenarios are given in Fig. 7(a). In the 2m_vs_1z map, we control a team of 2 Marines to fight with 1 enemy Zergling. In this task, it requires the Marine units to take advantage of their larger firing range to defeat over Zerglings which can only attack local enemies. The agents can observe a **continuous** feature vector including the information of health, positions and weapon cooldown of other agents. In terms of reward design, we keep the default setting. All agents receive a large final reward for winning a battle, at the meantime, they also receive immediate rewards that are proportional to the difference of total damages between the two teams in every time-step. We compare Q-DPP with aforementioned baseline models, i.e., COMA, VDN, QMIX, MAVEN, and QTRAN, and plot the results in Fig. 7(b). The results show that Q-DPP can perform as good as the state-of-the-art model, QMIX, even when the state feature is continuous. However, the performance is not stable and presents high variance. We believe full treatments need substantial future work.