A Algorithms

The single-agent version of ICQ is shown in Algorithm 1. Its multi-agent version counterpart (ICQ-MA) is shown in Algorithm 2.

Algorithm 1: Implicit Constraint Q-Learning in Single-Agent Tasks.

Input: Offline buffer $\mathcal{B}$, target network update rate d.

Initialize critic network $Q^{\pi}(\cdot;\phi)$ and actor network $\pi(\cdot;\theta)$ with random parameters.
Initialize target networks: $\phi' = \phi$, $\theta' = \theta$.
for $t = 1$ **to** T **do**
  Sample trajectories from $\mathcal{B}$.
  Train policy according to $\mathcal{J}_{\pi}(\theta) = \mathbb{E}_{\tau\sim\mathcal{B}}\left[-\frac{1}{Z(\tau)}\log(\pi(a \mid \tau;\theta))\exp\left(\frac{Q^{\pi}(\tau,a)}{\alpha}\right)\right]$.
  Train critic according to
  $\mathcal{J}_{Q}(\phi) = \mathbb{E}_{\tau\sim\mathcal{B}}\left[r + \gamma\frac{1}{Z(\tau')}\exp\left(\frac{Q(\tau',a';\phi')}{\alpha}\right)Q\left(\tau',a';\phi'\right) - Q\left(\tau,a;\phi\right)\right]^{2}$.
  if t mod $d = 0$ **then**
    Update target networks: $\phi' = \phi$, $\theta' = \theta$.
  end
end

Algorithm 2: Implicit Constraint Q-Learning in Multi-Agent Tasks.

Input: Offline buffer $\mathcal{B}$, target network update rate d.

Initialize critic networks $Q^{i}(\cdot;\phi_i)$, actor networks $\pi^{i}(\cdot;\theta_i)$ and Mixer network $M(\cdot;\psi)$ with random parameters.
Initialize target networks: $\phi' = \phi$, $\theta' = \theta$, $\psi' = \psi$.
for $t = 1$ **to** T **do**
  Sample trajectories from $\mathcal{B}$.
  Train individual policy according to
  $\mathcal{J}_{\boldsymbol{\pi}}(\theta) = \sum_i \mathbb{E}_{\tau^i,a^i\sim\mathcal{B}}\left[-\frac{1}{Z^{i}(\tau^{i})}\log(\pi^{i}(a^{i} \mid \tau^{i};\theta_i))\exp\left(\frac{w^{i}(\boldsymbol{\tau})Q^{i}(\tau^{i},a^{i})}{\alpha}\right)\right]$.
  Train critic according to $\mathcal{J}_{Q}(\phi,\psi) =$
  $\mathbb{E}_{\mathcal{B}}\left[\sum_{t\geq 0}(\gamma\lambda)^{t}\left[r_t + \gamma\frac{\exp\left(\frac{1}{\alpha}Q(\boldsymbol{\tau}_{t+1},\boldsymbol{a}_{t+1};\phi',\psi')\right)}{Z(\boldsymbol{\tau}_{t+1};\phi',\psi')}Q(\boldsymbol{\tau}_{t+1},\boldsymbol{a}_{t+1};\phi',\psi') - Q(\boldsymbol{\tau}_t,\boldsymbol{a}_t;\phi,\psi)\right]\right]^{2}$.
  if t mod $d = 0$ **then**
    Update target networks: $\phi' = \phi$, $\theta' = \theta$, $\psi' = \psi$.
  end
end

B Detailed Proof

B.1 Proof of Theorem 1

Theorem 1. *Given a deterministic MDP, the propagation of $\boldsymbol{\epsilon}_\mathbf{b}$ to $\boldsymbol{\epsilon}_\mathbf{s}$ is proportional to $\|P^\pi_{\mathrm{s,u}}\|_\infty$:*

$$\|\boldsymbol{\epsilon}_\mathbf{s}\|_\infty \leq \frac{\gamma \left\|P^\pi_{\mathrm{s,u}}\right\|_\infty}{(1-\gamma)\left(1-\gamma \left\|P^\pi_{\mathrm{s,s}}\right\|_\infty\right)} \|\boldsymbol{\epsilon}_\mathbf{b}\|_\infty . \tag{21}$$

Proof. Based on the Remark 1 in BCQ [16], the exact form of $\epsilon_{\mathrm{MDP}}(\tau, a)$ is:

$$\begin{aligned}\epsilon_{\mathrm{MDP}}(\tau,a) &= Q^\pi_M(\tau,a) - Q^\pi_{\mathcal{B}}(\tau,a) \\ &= \sum_{\tau'}(P_M(\tau' \mid \tau,a) - P_{\mathcal{B}}(\tau' \mid \tau,a))\left(r(\tau,a,\tau') + \gamma \sum_{a'} \pi(a' \mid \tau') Q^\pi_{\mathcal{B}}(\tau',a')\right) \\ &\quad + P_M(\tau' \mid \tau,a)\gamma \sum_{a'} \pi(a' \mid \tau')\epsilon_{\mathrm{MDP}}(\tau',a'),\end{aligned} \tag{22}$$

where $P_{\mathcal{B}} = \frac{\mathcal{N}(\tau,a,\tau')}{\sum_{\tilde{\tau}} \mathcal{N}(\tau,a,\tilde{\tau})}$ and $\mathcal{N}$ is the number of times the tuple (τ, a, τ') is observed in $\mathcal{B}$. If $\sum_{\tilde{\tau}} \mathcal{N}(\tau,a,\tilde{\tau}) = 0$, then $P_{\mathcal{B}}(\tau_{\mathrm{init}} \mid \tau, a) = 1$. Since the considered MDP is deterministic, we have $P_M(\tau' \mid \tau, a) - P_{\mathcal{B}}(\tau' \mid \tau, a) = 0$ for $P^\pi_{\mathrm{s,s}}$ and $P^\pi_{\mathrm{s,u}}$. For notational simplicity, the error generated by $P_M(\tau' \mid \tau, a) - P_{\mathcal{B}}(\tau' \mid \tau, a)$ in $P^\pi_{\mathrm{u,s}}$ and $P^\pi_{\mathrm{u,u}}$ is attributed to $\boldsymbol{\epsilon}_\mathbf{b}$ as they have the same dimension. Then, based on the extrapolation error decomposition assumption, we rewrite Equation 22 in the matrix form:

$$\begin{bmatrix}\boldsymbol{\epsilon}_\mathbf{s} \\ \boldsymbol{\epsilon}_\mathbf{u}\end{bmatrix} = \gamma \begin{bmatrix}P^\pi_{\mathrm{s,s}} & P^\pi_{\mathrm{s,u}} \\ P^\pi_{\mathrm{u,s}} & P^\pi_{\mathrm{u,u}}\end{bmatrix}\begin{bmatrix}\boldsymbol{\epsilon}_\mathbf{s} \\ \boldsymbol{\epsilon}_\mathbf{u}\end{bmatrix} + \begin{bmatrix}\mathbf{0} \\ \boldsymbol{\epsilon}_\mathbf{b}\end{bmatrix}. \tag{23}$$

The result indicates that the error is the solution of a linear program with $[0, \boldsymbol{\epsilon}_\mathbf{b}]^T$ as the reward function. Thus, we solve this linear program and arrive at

$$\begin{bmatrix}\boldsymbol{\epsilon}_\mathbf{s} \\ \boldsymbol{\epsilon}_\mathbf{u}\end{bmatrix} = (I - \gamma P^\pi)^{-1}\begin{bmatrix}0 \\ \boldsymbol{\epsilon}_\mathbf{b}\end{bmatrix} = \begin{bmatrix}I - \gamma P^\pi_{\mathrm{s,s}} & -\gamma P^\pi_{\mathrm{s,u}} \\ -\gamma P^\pi_{\mathrm{u,s}} & I - \gamma P^\pi_{\mathrm{u,u}}\end{bmatrix}^{-1}\begin{bmatrix}0 \\ \boldsymbol{\epsilon}_\mathbf{b}\end{bmatrix} = \begin{bmatrix}A & B \\ C & D\end{bmatrix}^{-1}\begin{bmatrix}0 \\ \boldsymbol{\epsilon}_\mathbf{b}\end{bmatrix}. \tag{24}$$

With the block matrix inverse formula, we have

$$\begin{bmatrix}A & B \\ C & D\end{bmatrix}^{-1} = \begin{bmatrix}A^{-1} + A^{-1}B\left(D - CA^{-1}B\right)^{-1}CA^{-1} & -A^{-1}B\left(D - CA^{-1}B\right)^{-1} \\ -\left(D - CA^{-1}B\right)^{-1}CA^{-1} & \left(D - CA^{-1}B\right)^{-1}\end{bmatrix}. \tag{25}$$

Since $\left(D - CA^{-1}B\right)^{-1}$ is just the lower right block of $(I - \gamma P^\pi)^{-1}$, we have

$$\left\|\left(D - CA^{-1}B\right)^{-1}\right\|_\infty \leq \left\|(I - \gamma P^\pi)^{-1}\right\|_\infty \leq \frac{1}{1-\gamma}. \tag{26}$$

Thus, we obtain

$$\begin{aligned}\left\|-A^{-1}B\left(D - CA^{-1}B\right)^{-1}\right\|_\infty &\leq \left\|A^{-1}\right\|_\infty \|-B\|_\infty \left\|\left(D - CA^{-1}B\right)^{-1}\right\|_\infty \\ &\leq \frac{1}{1-\gamma}\left\|A^{-1}\right\|_\infty \|-B\|_\infty \\ &= \frac{1}{1-\gamma}\left\|\left(I - \gamma P^\pi_{\mathrm{s,s}}\right)^{-1}\right\|_\infty \left\|\gamma P^\pi_{\mathrm{s,u}}\right\|_\infty \\ &\leq \frac{\gamma \left\|P^\pi_{\mathrm{s,u}}\right\|_\infty}{(1-\gamma)\left(1-\gamma\left\|P^\pi_{\mathrm{s,s}}\right\|_\infty\right)}.\end{aligned} \tag{27}$$

Plugging the result into Equation 25, we finish our proof at

$$\|\boldsymbol{\epsilon}_\mathbf{s}\|_\infty \leq \left\|-A^{-1}B\left(D - CA^{-1}B\right)^{-1}\right\|_\infty \|\boldsymbol{\epsilon}_\mathbf{b}\|_\infty \leq \frac{\gamma \left\|P^\pi_{\mathrm{s,u}}\right\|_\infty}{(1-\gamma)\left(1-\gamma\left\|P^\pi_{\mathrm{s,s}}\right\|_\infty\right)} \|\boldsymbol{\epsilon}_\mathbf{b}\|_\infty . \tag{28}$$

□

B.2 Proof of Theorem 2

The proof of our Theorem 2 is based on the Theorem 3 in [46]. The main difference is that we consider a behavior policy to regularize the softmax operation. All the actions considered in the analysis are batch-constrained, thus $\mu(a \mid \tau) > 0, \forall \tau, a$ in the proof.

Lemma 1. *By assuming $f_\alpha^T(Q(\tau,))Q(\tau,)$ as target value of the Implicit Constraint Q-learning operator, we have $\forall Q, \quad \max_{a\sim\mathcal{B}} Q(\tau,a) - f_\alpha^T(Q(\tau,))Q(\tau,) \leq (|A_\tau|-1)\max\{\frac{1}{(\frac{1}{\alpha}+1)C+1}, \frac{2Q_{\max}}{1+C\exp(\frac{1}{\alpha})}\}$, where $Q_{\max} = \frac{R_{\max}}{1-\gamma}$ represents the maximum Q-value in Q-iteration with $\mathcal{T}_{\mathrm{ICQ}}$.*

Proof. The target value operation of Implicit Constraint Q-learning is defined as:

$$f_\alpha(\tau \mid \mu) = \frac{\left[\mu_1 \exp(\frac{1}{\alpha}\tau_1), \mu_2 \exp(\frac{1}{\alpha}\tau_2), ..., \mu_{|A_\tau|} \exp(\frac{1}{\alpha}\tau_{|A_\tau|})\right]^T}{\sum_{i=1}^{|A_\tau|} \mu_i \exp(\frac{1}{\alpha}\tau_i)}, \tag{29}$$

We first sort the sequence $\{Q(\tau, a_i)\}$ such that $Q(\tau, a_{[1]}) \geq \cdots \geq Q(\tau, a_{[|A_\tau|]})$. Then, $\forall Q$ and $\forall \tau$, we have that the distance between optimal Q-value and Implicit Constraint Q-value is:

$$\begin{aligned}
&\max_{a\sim\mathcal{B}} Q(\tau,a) - f_\alpha^T(Q(\tau,\cdot) \mid \mu(\cdot|\tau))Q(\tau,) \\
&= Q(\tau, a_{[1]}) - \frac{\sum_{i=1}^{|A_\tau|} \mu(a_{[i]} \mid \tau) \exp\left[\frac{1}{\alpha} Q\left(\tau, a_{[i]}\right)\right] Q\left(\tau, a_{[i]}\right)}{\sum_{i=1}^{|A_\tau|} \mu(a_{[i]} \mid \tau) \exp\left[\frac{1}{\alpha} Q\left(\tau, a_{[i]}\right)\right]} \\
&= \frac{\sum_{i=1}^{|A_\tau|} \mu(a_{[i]} \mid \tau) \exp\left[\frac{1}{\alpha} Q\left(\tau, a_{[i]}\right)\right] \left[Q\left(\tau, a_{[1]}\right) - Q\left(\tau, a_{[i]}\right)\right]}{\sum_{i=1}^{|A_\tau|} \mu(a_{[i]} \mid \tau) \exp\left[\frac{1}{\alpha} Q\left(\tau, a_{[i]}\right)\right]}.
\end{aligned} \tag{30}$$

Let $\delta_i(\tau) = Q\left(\tau, a_{[1]}\right) - Q\left(\tau, a_{[i]}\right)$. The distance in the Equation 30 can be rewritten as:

$$\begin{aligned}
&\frac{\sum_{i=1}^{|A_\tau|} \mu(a_{[i]} \mid \tau) \exp\left[\frac{1}{\alpha} Q\left(\tau, a_{[i]}\right)\right] \left[Q\left(\tau, a_{[1]}\right) - Q\left(\tau, a_{[i]}\right)\right]}{\sum_{i=1}^{|A_\tau|} \mu(a_{[i]} \mid \tau) \exp\left[\frac{1}{\alpha} Q\left(\tau, a_{[i]}\right)\right]} \\
&= \frac{\sum_{i=1}^{|A_\tau|} \mu(a_{[i]} \mid \tau) \exp\left[-\frac{1}{\alpha}\delta_i(\tau)\right] \delta_i(\tau)}{\sum_{i=1}^{|A_\tau|} \mu(a_{[i]} \mid \tau) \exp\left[-\frac{1}{\alpha}\delta_i(\tau)\right]} \\
&= \frac{\sum_{i=2}^{|A_\tau|} \mu(a_{[i]} \mid \tau) \exp\left[-\frac{1}{\alpha}\delta_i(\tau)\right] \delta_i(\tau)}{\mu(a_{[1]} \mid \tau) + \sum_{i=2}^{|A_\tau|} \mu(a_{[i]} \mid \tau) \exp\left[-\frac{1}{\alpha}\delta_i(\tau)\right]}
\end{aligned} \tag{31}$$

First note that for any two non-negative sequences $\{x_i\}$ and $\{y_i\}$,

$$\frac{\sum_i x_i}{1 + \sum_i y_i} \leq \sum_i \frac{x_i}{1 + y_i}. \tag{32}$$

We have the following conclusion by applying the Equation 32 to Equation 31:

$$\begin{aligned}
\frac{\sum_{i=2}^{|A_\tau|} \mu(a_{[i]} \mid \tau) \exp\left[-\frac{1}{\alpha}\delta_i(\tau)\right] \delta_i(\tau)}{\mu(a_{[1]} \mid \tau) + \sum_{i=2}^{|A_\tau|} \mu(a_{[i]} \mid \tau) \exp\left[-\frac{1}{\alpha}\delta_i(\tau)\right]} &\leq \sum_{i=2}^{|A_\tau|} \frac{\mu(a_{[i]} \mid \tau) \exp\left[-\frac{1}{\alpha}\delta_i(\tau)\right] \delta_i(\tau)}{\mu(a_{[1]} \mid \tau) + \mu(a_{[i]} \mid \tau) \exp\left[-\frac{1}{\alpha}\delta_i(\tau)\right]} \\
&= \sum_{i=2}^{|A_\tau|} \frac{\mu(a_{[i]} \mid \tau)\delta_i(\tau)}{\mu(a_{[i]} \mid \tau) + \mu(a_{[1]} \mid \tau) \exp\left[\frac{1}{\alpha}\delta_i(\tau)\right]} \\
&= \sum_{i=2}^{|A_\tau|} \frac{\delta_i(\tau)}{1 + \frac{\mu(a_{[1]}|\tau)}{\mu(a_{[i]}|\tau)} \exp\left[\frac{1}{\alpha}\delta_i(\tau)\right]} \\
&\leq \sum_{i=2}^{|A_\tau|} \frac{\delta_i(\tau)}{1 + C \exp\left[\frac{1}{\alpha}\delta_i(\tau)\right]},
\end{aligned} \tag{33}$$

where $C = \inf_{\tau \in S} \inf_{2 \leq i \leq |A_\tau|} \frac{\mu(a_{[1]}|\tau)}{\mu(a_{[i]}|\tau)}$.
If $\delta_i(\tau) > 1$, we have

$$\frac{\delta_i(\tau)}{1 + C \exp\left[\frac{1}{\alpha}\delta_i(\tau)\right]} \leq \frac{\delta_i(\tau)}{1 + C \exp\left(\frac{1}{\alpha}\right)} \leq \frac{2Q_{\max}}{1 + C \exp(\frac{1}{\alpha})}. \tag{34}$$

else $0 \leq \delta_i(\tau) \leq 1$:

$$\frac{\delta_i(\tau)}{1 + C \exp\left[\frac{1}{\alpha}\delta_i(\tau)\right]} = \frac{1}{\frac{1+C}{\delta_i(\tau)} + \frac{1}{\alpha}C + 0.5\frac{1}{\alpha^2}\delta_i(\tau)C + \cdots} \leq \frac{1}{(\frac{1}{\alpha} + 1)C + 1}. \tag{35}$$

By combining these two cases with Equation 33, we complete the proof. □

Theorem 2. *Let $\mathcal{T}^k_{\mathrm{ICQ}} Q_0$ denote that the operator $\mathcal{T}_{\mathrm{ICQ}}$ are iteratively applied over an initial state-action value function Q_0 for k times. Then, we have $\forall(\tau, a)$, $\limsup_{k\to\infty} \mathcal{T}^k_{\mathrm{ICQ}} Q_0(\tau, a) \leq Q^*(\tau, a)$,*

$$\liminf_{k\to\infty} \mathcal{T}^k_{\mathrm{ICQ}} Q_0(\tau, a) \geq Q^*(\tau, a) - \frac{\gamma(|A| - 1)}{(1 - \gamma)} \max\left\{\frac{1}{(\frac{1}{\alpha} + 1)C + 1}, \frac{2Q_{\max}}{1 + C \exp(\frac{1}{\alpha})}\right\}, \tag{36}$$

where $|A|$ is the action space, $|A_\tau|$ is the action space for state τ, $C \triangleq \inf_{\tau \in S} \inf_{2 \leq i \leq |A_\tau|} \frac{\mu(a_{[1]}|\tau)}{\mu(a_{[i]}|\tau)}$ and $\mu(a_{[1]} \mid \tau)$ denotes the probability of choosing the expert action according to behavioral policy μ. Moreover, the upper bound of $\mathcal{T}^k_{\mathrm{BCQ}} Q_0 - \mathcal{T}^k_{\mathrm{ICQ}} Q_0$ decays exponentially fast in terms of α.

Proof. We first conjecture that

$$\mathcal{T}^k_{\mathrm{BCQ}} Q_0(\tau, a) - \mathcal{T}^k_{\mathrm{ICQ}} Q_0(\tau, a) \leq \sum_{j=1}^{k} \gamma^j \zeta, \tag{37}$$

where $\zeta = \sup_Q \max_\tau \left[\max_{a \sim \mathcal{B}} Q(\tau, a) - f^T_\alpha\left(Q(\tau,)\right) Q(\tau,)\right]$ denotes the supremum of the difference between the BCQ and ICQ operators, over all Q-functions that occur during Q-iteration, and state τ. Equation 37 is proven using induction as follows:

- When $i = 1$, we start from the definitions for $\mathcal{T}_{\mathrm{BCQ}}$ and $\mathcal{T}_{\mathrm{ICQ}}$, and proceed as

$$\begin{aligned}
\mathcal{T}_{\mathrm{BCQ}} Q_0(\tau, a) - \mathcal{T}_{\mathrm{ICQ}} Q_0(\tau, a) &= \gamma \sum_{\tau'} P(\tau' \mid \tau, a) \left[\max_{a' \sim \mathcal{B}} Q_0(\tau', a') - f^T_\alpha\left(Q_0(\tau',)\right) Q_0(\tau',)\right] \\
&\leq \gamma \sum_{\tau'} P(\tau' \mid \tau, a)\zeta = \gamma\zeta.
\end{aligned} \tag{38}$$

- Suppose the conjecture holds when $i = l$, i.e., $\mathcal{T}^l_{\mathrm{BCQ}} Q_0(\tau, a) - \mathcal{T}^l_{\mathrm{ICQ}} Q_0(\tau, a) \leq \sum_{j=1}^{l} \gamma^j \zeta$, then

$$\begin{aligned}
\mathcal{T}^{l+1}_{\mathrm{BCQ}} Q_0(\tau, a) - \mathcal{T}^{l+1}_{\mathrm{ICQ}} Q_0(\tau, a) &= \mathcal{T}_{\mathrm{BCQ}} \mathcal{T}^l_{\mathrm{BCQ}} Q_0(\tau, a) - \mathcal{T}^{l+1}_{\mathrm{ICQ}} Q_0(\tau, a) \\
&\leq \mathcal{T}_{\mathrm{BCQ}} \left[\mathcal{T}^l_{\mathrm{ICQ}} Q_0(\tau, a) + \sum_{j=1}^{l} \gamma^j \zeta\right] - \mathcal{T}^{l+1}_{\mathrm{ICQ}} Q_0(\tau, a) \\
&= \sum_{j=1}^{l} \gamma^{j+1} \zeta + \left(\mathcal{T}_{\mathrm{BCQ}} - \mathcal{T}_{\mathrm{ICQ}}\right) \mathcal{T}^l_{\mathrm{ICQ}} Q_0(\tau, a) \\
&\leq \sum_{j=1}^{l} \gamma^{j+1} \zeta + \gamma\zeta = \sum_{j=1}^{l+1} \gamma^j \zeta.
\end{aligned} \tag{39}$$

By using the fact that $\lim_{k\to\infty}\mathcal{T}^k_{\mathrm{BCQ}}Q_0(\tau,a)$ and applying Lemma 1 to bound ζ, we have $\forall(\tau,a)$, $\limsup_{k\to\infty}\mathcal{T}^k_{\mathrm{ICQ}}Q_0(\tau,a) \le Q^*(\tau,a)$ and $\liminf_{k\to\infty}\mathcal{T}^k_{\mathrm{ICQ}}Q_0(\tau,a) \ge Q^*(\tau,a) - \frac{\gamma(|A|-1)}{(1-\gamma)}\max\{\frac{1}{(\frac{1}{\alpha}+1)C+1}, \frac{2Q_{\max}}{1+C\exp(\frac{1}{\alpha})}\}$. Based on the Equation 33, we can bound Equation 37 as:

$$\mathcal{T}^k_{\mathrm{BCQ}}Q_0(\tau,a) - \mathcal{T}^k_{\mathrm{ICQ}}Q_0(\tau,a) \le \frac{\gamma(1-\gamma^k)}{1-\gamma}\sum_{i=2}^{|A|}\frac{\delta_i(\tau)}{1+C\exp\left[\frac{1}{\alpha}\delta_i(\tau)\right]}. \tag{40}$$

From the definition of $\delta_i(\tau)$, we have $\delta_{|A_\tau|}(\tau) \ge \delta_{|A_\tau|-1}(\tau) \ge \cdots \ge \delta_2(\tau) \ge 0$. Furthermore, there must exist an index $i^* \le |A_\tau|$ such that $\delta_i > 0, \forall i^* \le i \le |A_\tau|$. Therefore, we can proceed from Equation 40 as

$$\begin{aligned}
\frac{\gamma(1-\gamma^k)}{1-\gamma}\sum_{i=2}^{|A|}\frac{\delta_i(\tau)}{1+C\exp\left[\frac{1}{\alpha}\delta_i(\tau)\right]} &= \frac{\gamma(1-\gamma^k)}{1-\gamma}\sum_{i=i^*}^{|A|}\frac{\delta_i(\tau)}{1+C\exp\left[\frac{1}{\alpha}\delta_i(\tau)\right]} \\
&\le \frac{\gamma(1-\gamma^k)}{1-\gamma}\sum_{i=i^*}^{|A|}\frac{\delta_i(\tau)}{C\exp\left[\frac{1}{\alpha}\delta_i(\tau)\right]} \le \frac{\gamma(1-\gamma^k)}{1-\gamma}\sum_{i=i^*}^{|A|}\frac{\delta_i(\tau)}{C\exp\left[\frac{1}{\alpha}\delta_{i^*}(\tau)\right]} \\
&= \frac{\gamma(1-\gamma^k)}{1-\gamma}\exp\left[-\frac{1}{\alpha}\delta_{i^*}(\tau)\right]\sum_{i=i^*}^{|A|}\frac{\delta_i(\tau)}{C},
\end{aligned} \tag{41}$$

which implies an exponential convergence rate in terms of α. □

B.3 Proof of Remark 3.2

We analyze the MMDP experimental result in Section 3.2 from the perspective of the concentrability coefficient $C(\Pi)$, which illustrates the degree to which states and actions are out of distribution. In the MMDP case, we theoretically prove $C(\Pi^i)$ satisfies: $C(\Pi^1) < C(\Pi^2) < \cdots < C(\Pi^n)$, where Π^i denotes the set of joint policies including i agents. As illustrated in the above conclusion, the increase in the number of agents makes the distribution shift issue more severe in the MMDP case.

Remark 1. *Let $\varrho(s)$ denote the marginal distribution over S, ρ_0 indicate the initial state distribution, and Π^i represent the set of joint policies including i agents. Assume there exist coefficients $c(k)$ satisfying $\rho_0 P^{\boldsymbol{\pi}_1}P^{\boldsymbol{\pi}_2}...P^{\boldsymbol{\pi}_k}(s) \le c(k)\varrho(s)$. We define the concentrability coefficient $C(\Pi) \triangleq (1-\gamma)^2\sum_{k=1}^{\infty}k\gamma^{k-1}c(k)$, which illustrates the degree to which states and actions are out of distribution. Due to the limited datasets, the number of seen state-action pairs m is fixed. Then, $C(\Pi^i)$ is monotonically increasing with the number of agents*

$$C(\Pi^1) < C(\Pi^2) < \cdots < C(\Pi^n) \tag{42}$$

Proof. We first note that $c(k) \ge \frac{\rho_0 P^{\boldsymbol{\pi}_1}P^{\boldsymbol{\pi}_2}...P^{\boldsymbol{\pi}_k}(s)}{\varrho(s)}$ and $c(k)$ determines the value of $C(\Pi^i)$. To compare $C(\Pi^i)$, we just need to compare $c(k)$ at iteration k. For clarity of analysis, we assume each state-action pair is visited only once, and individual policy is random $\pi^i(A^{(i)}|s) = \frac{1}{2}$. In the MMDP case, the transition matrix $P^{\boldsymbol{\pi}}$ is stable for the number of agents:

$$P^{\pi^1_k} = P^{\boldsymbol{\pi}_k} = P^{\pi^1_k\pi^2_k...\pi^n_k} = \begin{bmatrix} 1 & 0 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}. \tag{43}$$

For this reason, $\rho_0 P^{\boldsymbol{\pi}_1}P^{\boldsymbol{\pi}_2}...P^{\boldsymbol{\pi}_k}(s)$ does not change with the number of agents. As $\varrho(s) = \sum_a \varrho(s,a) = \sum_a \frac{\sum_{s,\boldsymbol{a}\in\mathcal{D}}\mathbf{1}[s=s,\boldsymbol{a}=\boldsymbol{a}]}{\sum_{s',\boldsymbol{a}'\in\mathcal{D}}\mathbf{1}[s=s',\boldsymbol{a}=\boldsymbol{a}']}$, we can calculate $\varrho(s)$ by counting state-action pairs in $\mathcal{D}$ as follows

$$\varrho(s) = \frac{m}{2^{n+1}}. \tag{44}$$

The gradient of $\varrho(s)$ is:

$$\varrho(s)' = \left(\frac{m}{2^{n+1}}\right)' = \frac{-m\cdot 2^{n+1}\ln 2}{(2^{n+1})^2} < 0. \tag{45}$$

Therefore, $c(k)$ is monotonically increasing with the number of agents and $C(\Pi^1) < C(\Pi^2) < \cdots < C(\Pi^n)$. □

B.4 Proof of Remark 2

Remark 2. *For the optimization problem*

$$\pi_{k+1} = \arg\max_{\pi} \mathbb{E}_{a\sim\pi(\cdot|\tau)}[Q^{\pi_k}(\tau, a)] \qquad \text{s.t.} \quad D_{\mathrm{KL}}(\pi\|\mu)[\tau] \leq \epsilon, \quad \sum_a \pi(a|\tau) = 1, \tag{46}$$

the optimal policy is $\pi^*_{k+1}(a \mid \tau) = \frac{\mu(a|\tau)\exp\left(\frac{1}{\alpha}Q^{\pi_k}(\tau,a)\right)}{\sum_{\tilde{a}}\mu(\tilde{a}|\tau)\exp\left(\frac{1}{\alpha}Q^{\pi_k}(\tau,\tilde{a})\right)}$.

Proof. First, note the objective is a linear function of the decision variables π. All constraints are convex functions. Thus Equation 46 is a convex optimization problem. The Lagrangian equation is

$$\mathcal{L}(\pi, \alpha) = \mathbb{E}_{a\sim\pi}[Q^{\pi_k}(\tau, a)] + \alpha\left(\epsilon - D_{\mathrm{KL}}\left(\pi\|\mu\right)[\tau]\right) + \lambda\left(1 - \sum_a \pi(a \mid \tau)\right), \tag{47}$$

where $\alpha > 0$ denotes the Lagrangian coefficient. Differentiate π to get the following formula

$$\frac{\partial\mathcal{L}}{\partial\pi} = Q^{\pi_k}(\tau, a) - \alpha\left(1 + \log\left(\frac{\pi(a \mid \tau)}{\mu(a \mid \tau)}\right)\right) - \lambda. \tag{48}$$

Setting $\frac{\partial\mathcal{L}}{\partial\pi}$ to zero, then:

$$\begin{aligned}
&Q^{\pi_k}(\tau, a) - \alpha\left(1 + \log\left(\frac{\pi(a \mid \tau)}{\mu(a \mid \tau)}\right)\right) - \lambda = 0 \\
&Q^{\pi_k}(\tau, a) = \alpha\left(1 + \log\left(\frac{\pi(a \mid \tau)}{\mu(a \mid \tau)}\right)\right) + \lambda \\
&\frac{Q^{\pi_k}(\tau, a)}{\alpha} - 1 - \frac{\lambda}{\alpha} = \log\left(\frac{\pi(a \mid \tau)}{\mu(a \mid \tau)}\right) \\
&\frac{\pi(a \mid \tau)}{\mu(a \mid \tau)} = \exp\left(\frac{Q^{\pi_k}(\tau, a)}{\alpha} - 1 - \frac{\lambda}{\alpha}\right) \\
&\pi(a \mid \tau) = \mu(a \mid \tau)\exp\left(\frac{Q^{\pi_k}(\tau, a)}{\alpha} - 1 - \frac{\lambda}{\alpha}\right)
\end{aligned} \tag{49}$$

Due to the second constraint in Equation 46, the policy is a probability distribution. Therefore, we adopt Z to normalize the result by moving the constant $\mu(a \mid \tau)\exp(-1 - \frac{\lambda}{\alpha})$ to Z:

$$\pi^*_{k+1}(a \mid \tau) = \frac{1}{Z(\tau)}\mu(a \mid \tau)\exp\left(\frac{Q^{\pi_k}(\tau, a)}{\alpha}\right), \tag{50}$$

where $Z(\tau) = \sum_{\tilde{a}}\mu(\tilde{a} \mid \tau)\exp\left(\frac{1}{\alpha}Q^{\pi_k}(\tau, \tilde{a})\right)$ is the normalizing partition function. □

B.5 Proof of Theorem 3

Theorem 3. *Assuming the joint action-value function is linearly decomposed, we can decompose the multi-agent joint-policy under implicit constraint as follows*

$$\boldsymbol{\pi} = \arg\max_{\pi^1,\ldots,\pi^n} \sum_i \mathbb{E}_{\tau^i, a^i\sim\mathcal{B}}\left[\frac{1}{Z^i(\tau^i)}\log(\pi^i(a^i \mid \tau^i))\exp\left(\frac{w^i(\boldsymbol{\tau})Q^i(\tau^i, a^i)}{\alpha}\right)\right], \tag{51}$$

where $Z^i(\tau^i) = \sum_{\tilde{a}^i}\mu^i(\tilde{a}^i \mid \tau^i)\exp\left(\frac{1}{\alpha}w^i(\boldsymbol{\tau})Q^i(\tau^i, \tilde{a}^i)\right)$ *is the normalizing partition function.*

Proof. Let $\mathcal{J}_{\boldsymbol{\pi}}$ denote the joint-policy loss. According to the assumption, $\mathcal{J}_{\boldsymbol{\pi}}$ is written:

$$\begin{aligned}
\mathcal{J}_{\boldsymbol{\pi}} &= \mathbb{E}_{\boldsymbol{\tau},\boldsymbol{a}\sim\mathcal{B}}\left[-\frac{1}{Z(\boldsymbol{\tau})}\log(\boldsymbol{\pi}(\boldsymbol{a}|\boldsymbol{\tau}))\exp\left(\frac{1}{\alpha}Q^{\boldsymbol{\pi}}(\boldsymbol{\tau}, \boldsymbol{a})\right)\right] \\
&= \mathbb{E}_{\boldsymbol{\tau},a^1,\ldots,a^n\sim\mathcal{B}}\left[-\frac{1}{Z(\boldsymbol{\tau})}\left(\sum_i \log(\pi^i(a^i \mid \tau^i))\right)\exp\left(\frac{1}{\alpha}\left(\sum_i w^i(\boldsymbol{\tau})Q^i(\tau^i, a^i) + b(\boldsymbol{\tau})\right)\right)\right].
\end{aligned} \tag{52}$$

The loss function $\mathcal{J}_{\boldsymbol{\pi}}$ is equivalent to the following form by relocating the sum operator:

$$\begin{aligned}
\mathcal{J}_{\boldsymbol{\pi}} &= \sum_i \mathbb{E}_{\boldsymbol{\tau},a^1,\ldots,a^n\sim\mathcal{B}}\left[-\frac{1}{Z(\boldsymbol{\tau})}\log(\pi^i(a^i\mid\tau^i))\exp\left(\frac{\sum_i w^i(\boldsymbol{\tau})Q^i(\tau^i,a^i)+b(\boldsymbol{\tau})}{\alpha}\right)\right] \\
&= \sum_i \mathbb{E}_{\boldsymbol{\tau},a^1,\ldots,a^n\sim\mathcal{B}}[-\frac{1}{Z(\boldsymbol{\tau})}\log(\pi^i(a^i\mid\tau^i))\exp\left(\frac{w^i(\boldsymbol{\tau})Q^i(\tau^i,a^i)}{\alpha}\right) \\
&\qquad\qquad \exp\left(\frac{\sum_{j\neq i} w^j(\boldsymbol{\tau})Q^j(\tau^j,a^j)+b(\boldsymbol{\tau})}{\alpha}\right)] \\
&= \sum_i \mathbb{E}_{\boldsymbol{\tau},a^i\sim\mathcal{B}}\mathbb{E}_{a^{j\neq i}\sim\mathcal{B}}[-\frac{1}{Z(\boldsymbol{\tau})}\log(\pi^i(a^i\mid\tau^i))\exp\left(\frac{w^i(\boldsymbol{\tau})Q^i(\tau^i,a^i)}{\alpha}\right) \\
&\qquad\qquad \exp\left(\frac{\sum_{j\neq i} w^j(\boldsymbol{\tau})Q^j(\tau^j,a^j)+b(\boldsymbol{\tau})}{\alpha}\right)] \\
&= \sum_i \mathbb{E}_{\boldsymbol{\tau},a^i\sim\mathcal{B}}\left[-\frac{1}{Z^i(\tau^i)}\log(\pi^i(a^i\mid\tau^i))\exp\left(\frac{w^i(\boldsymbol{\tau})Q^i(\tau^i,a^i)}{\alpha}\right)\right],
\end{aligned} \tag{53}$$

$$\begin{aligned}
Z^i(\tau^i) &= \frac{\sum_{\tilde{a}^i}\sum_{\tilde{a}^{j\neq i}}\boldsymbol{\mu}(\bar{\boldsymbol{a}}\mid\boldsymbol{\tau})\exp\left(\frac{1}{\alpha}w^i(\boldsymbol{\tau})Q^i(\tau^i,\tilde{a}^i)\right)\exp\left(\frac{1}{\alpha}(\sum_{j\neq i}w^j(\boldsymbol{\tau})Q^j(\tau^j,\tilde{a}^j)+b(\boldsymbol{\tau}))\right)}{\mathbb{E}_{\tilde{a}^{j\neq i}\sim\mathcal{B}}\left[\exp\left(\frac{1}{\alpha}\left(\sum_{j\neq i}w^j(\boldsymbol{\tau})Q^j(\tau^j,\tilde{a}^j)+b(\boldsymbol{\tau})\right)\right)\right]} \\
&= \frac{\sum_{\tilde{a}^i}\sum_{\tilde{a}^{j\neq i}}\mu^i(\tilde{a}^i\mid\tau^i)\mu^{j\neq i}(\tilde{a}^j\mid\tau^j)\exp\left(\frac{1}{\alpha}w^i(\boldsymbol{\tau})Q^i(\tau^i,\tilde{a}^i)\right)}{\sum_{\tilde{a}^{j\neq i}}\mu^{j\neq i}(\tilde{a}^j\mid\tau^j)\exp\left(\frac{1}{\alpha}\left(\sum_{j\neq i}w^j(\boldsymbol{\tau})Q^j(\tau^j,\tilde{a}^j)+b(\boldsymbol{\tau})\right)\right)}\cdot \\
&\qquad\qquad \exp\left(\frac{1}{\alpha}\left(\sum_{j\neq i}w^j(\boldsymbol{\tau})Q^j(\tau^j,\tilde{a}^j)+b(\boldsymbol{\tau})\right)\right) \\
&= \sum_{\tilde{a}^i}\mu^i(\tilde{a}^i\mid\tau^i)\exp\left(\frac{1}{\alpha}w^i(\boldsymbol{\tau})Q^i(\tau^i,\tilde{a}^i)\right).
\end{aligned} \tag{54}$$

□

C Additional Results

C.1 Additional Ablation Results in StarCraft II

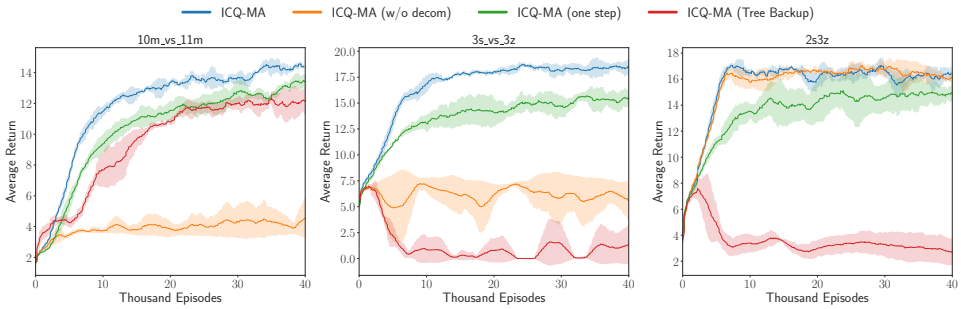


Figure 6: Module ablation study in additional StarCraft II environments.

C.2 Additional Results in MMDP

Due to the space limits, we put the complete results in MMDP in Figure 7. BCQ gradually diverges as the number of agents increases, while ICQ has accurate estimates.

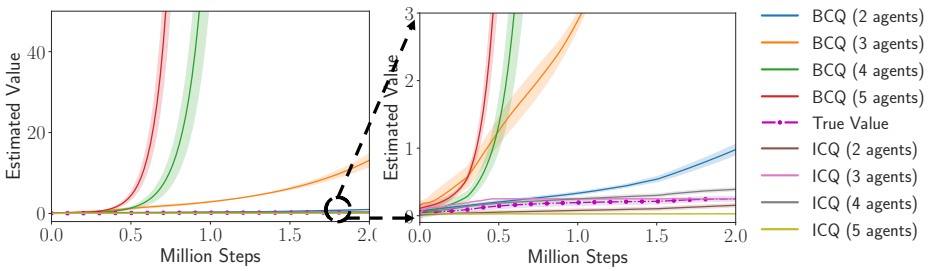
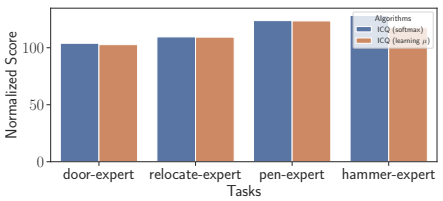
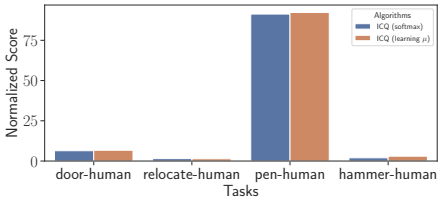


Figure 7: Additional results in MMDP.

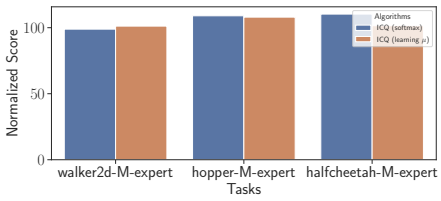
C.3 Additional Results in D4RL



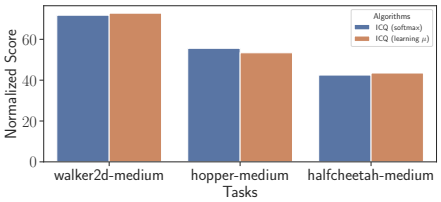
(a) Adroit-expert.



(b) Adroit-human.



(c) Mujoco-medium-expert.



(d) Mujoco-medium.

Figure 8: The performance on D4RL tasks with different implementation of ICQ.

C.4 Ablation Study

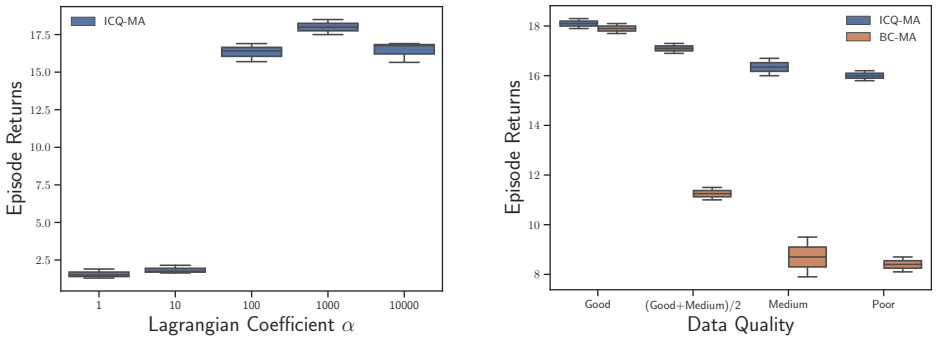


Figure 9: Ablation study on MMM map.

D Experimental Details

D.1 Implementation details of ICQ

We provide the two implementation options of our methods regards whether learning μ to calculate ρ.

Learning an auxiliary behavior model $\hat{\mu}$. We first consider to learn the behavior policy $\hat{\mu}$ using conditional variational auto-encoder as BCQ. Next, we will sample actions n times ($n = 100$ in our experiment) from $\hat{\mu}$ to calculate $Z(\tau)$ on each value update:

$$\rho(\tau, a) = \frac{\exp(\frac{Q(\tau,a)}{\alpha})}{Z(\tau)} \approx \frac{\exp(\frac{Q(\tau,a)}{\alpha})}{\mathbb{E}_{\tilde{a}\sim\hat{\mu}} \exp(\frac{Q(\tau,\tilde{a})}{\alpha})}. \tag{55}$$

If $\hat{\mu} \approx \mu$, this method is favored as it provides an accurate approximation. However, since it may introduce unseen pairs sampled from the learned behavior model, it is against the principle of our analysis. Nevertheless, we believe it is still a better choice compared with BCQ. If there is any unseen pair $\tau, \tilde{a}$ with large extrapolation error sampled from $\hat{\mu}$, e.g, $Q_{\mathcal{B}}(\tau, \tilde{a}) \gg Q_M(\tau, \tilde{a})$, we will have $\hat{\rho}(\tau, a) < \rho(\tau, a)$, which means the unsafe estimation is truncated and the resulting target Q-value tends to be conservative.

Approximate with softmax operation over a mini-batch. We have the following measure to approximately calculate ρ without μ, which reduces the computational complexity:

$$\rho(\tau, a) = \frac{\exp(\frac{Q(\tau,a)}{\alpha})}{Z(\tau)} \approx \frac{\exp(\frac{Q(\tau,a)}{\alpha})}{\sum_{(\tau',a')\sim\text{mini-batch}} \exp(\frac{Q(\tau',a')}{\alpha})}, \tag{56}$$

where $Z^i(\tau^i)$ is approximated by softmax operation over mini-batch samples. The benefit of the softmax operation is that it does not include any unseen pairs, which is consistent with our theoretical analysis. However, the price is that the softmax operation ignores the difference of states over a mini-batch, which introduces an additional bias. However, we find it does not harm the performance a lot in practice. There are also some previous works using softmax to deal with the partition function, such as AWAC [29]) and VMPO [45], which has been confirmed to promote performance improvement.

Considering the concise form of the softmax operation, we prefer the the second version in the multi-agent tasks. We conduct ablation studies of these two measures on D4RL to demonstrate their superior performance (see Figure 8).

D.2 Baselines Details

BCQ-MA is trained by minimizing the following loss:

$$\mathcal{J}_Q^{\mathrm{BCQ}}(\phi, \psi) = \mathbb{E}_{\boldsymbol{\tau}\sim\mathcal{B}, \boldsymbol{a}\sim\boldsymbol{\mu}} \left[\left(r(\boldsymbol{\tau}, \boldsymbol{a}) + \gamma \max_{\tilde{\boldsymbol{a}}^{[j]}} Q^{\boldsymbol{\pi}}(\boldsymbol{\tau}', \tilde{\boldsymbol{a}}^{[j]}; \phi', \psi') - Q^{\boldsymbol{\pi}}(\boldsymbol{\tau}, \boldsymbol{a}; \phi, \psi) \right)^2 \right], \tag{57}$$
$$\tilde{\boldsymbol{a}}^{[j]} = \boldsymbol{a}^{[j]} + \xi(\boldsymbol{\tau}, \boldsymbol{a}^{[j]})$$

where $Q^{\boldsymbol{\pi}}(\boldsymbol{\tau}, \boldsymbol{a}) = w^i(\boldsymbol{\tau}) Q^i(\tau^i, a^i) + b(\boldsymbol{\tau})$ and $\xi(\boldsymbol{\tau}, \boldsymbol{a}^{[j]})$ denotes the perturbation model, which is decomposed as $\xi^i(\tau^i, a^{i,[j]})$. If $\frac{a^{i,[j]}\sim G^i(\tau^i;\psi^i)}{\max\{a^{i,[j]}\sim G^i(\tau^i;\psi^i)\}_{j=1}^m} \leq \zeta$ in agent i, $a^{i,[j]}$ is considered an unfamiliar action and $\xi^i(\tau^i, a^{i,[j]})$ will mask $a^{i,[j]}$ in maximizing Q^i-value operation.

CQL-MA is trained by minimizing the following loss:

$$\begin{aligned}
\mathcal{J}_Q^{\mathrm{CQL}}(\phi, \psi) = \alpha^{\mathrm{CQL}} \mathbb{E}_{\boldsymbol{\tau}\sim\mathcal{B}} \left[\sum_i \log \sum_{a^i} \exp(w^i(\boldsymbol{\tau}) Q^i(\tau^i, a^i) + b(\boldsymbol{\tau})) - \mathbb{E}_{\boldsymbol{a}\sim\boldsymbol{\mu}(\boldsymbol{a}|\boldsymbol{\tau})}[Q^{\boldsymbol{\pi}}(\boldsymbol{\tau}, \boldsymbol{a})] \right] \\
+ \frac{1}{2} \mathbb{E}_{\boldsymbol{\tau}\sim\mathcal{B}, \boldsymbol{a}\sim\boldsymbol{\mu}(\boldsymbol{a}|\boldsymbol{\tau})} \left[\left(y^{\mathrm{CQL}} - Q^{\boldsymbol{\pi}}(\boldsymbol{\tau}, \boldsymbol{a}) \right)^2 \right] \\
\mathcal{J}_{\boldsymbol{\pi}}^{\mathrm{CQL}}(\theta) = \sum_i \mathbb{E}_{\boldsymbol{\tau}^i, a^i\sim\mathcal{B}} \left[-\log(\pi^i(a^i \mid \tau^i; \theta_i)) Q^i(\tau^i, a^i) \right],
\end{aligned} \tag{58}$$

where we adopt the decomposed policy gradient to train $\boldsymbol{\pi}$, and y^{CQL} is calculated based on n-step off-policy estimation (e.g., Tree Backup algorithm). Besides, $w^i(\boldsymbol{\tau}) = w^i(\boldsymbol{\tau}; \psi)$, $b(\boldsymbol{\tau}) = b(\boldsymbol{\tau}; \psi)$ and $Q^{\boldsymbol{\pi}}(\boldsymbol{\tau}, \boldsymbol{a}) = Q^{\boldsymbol{\pi}}(\boldsymbol{\tau}, \boldsymbol{a}; \phi, \psi)$.

BC-MA only optimize $\boldsymbol{\pi}$ by minimizing the following loss:

$$\mathcal{J}_{\boldsymbol{\pi}}^{\mathrm{BC}}(\theta) = \sum_i \mathbb{E}_{\tau^i, a^i \sim \mathcal{B}}[-\log(\pi^i(a^i \mid \tau^i; \theta_i))]. \tag{59}$$

E Multi-Agent Offline Dataset Based on StarCraft II

We divide maps in StarCraft II into three classifications based on difficulty (see Table 2). We divide behavior policies into three classifications based on the episode returns (see Table 3).

Table 2: Classification of maps in the dataset.

Difficulties	Maps
Easy	MMM, 2s_vs_3z, 3s_vs_3z, 3s5z, 2s3z, so_many_baneling
Hard	10m_vs_11m, 2c_vs_64zg
Super Hard	MMM2, 27m_vs_30m

Table 3: Classification of behavior policies in the dataset.

Level	Episode Returns
Good	$15 \sim 20$
Medium	$10 \sim 15$
Poor	$0 \sim 10$

E.1 Hyper-parameters

Hyper-parameters in multi-agent tasks are respectively presented in Table 4. Please refer to our official code for the hyper-parameter in single-agent tasks.

Table 4: Multi-agent hyper-parameters sheet

Hyper-parameter	Value
Shared	
Policy network learning rate	5×10^{-4}
Value network learning rate	10^{-4}
Optimizer	Adam
Discount factor γ	0.99
Parameters update rate d	600
Gradient clipping	20
Mixer network dimension	32
RNN hidden dimension	64
Activation function	ReLU
Batch size	16
Replay buffer size	1.2×10^4
Others	
Lagrangian coefficient α	1000 or 100
λ	0.8
α^{CQL}	2.0
ζ	0.3