

## A PROOF OF THEOREM 1

**Theorem 1.** Let  $\pi_i^*$  be the policy obtained by optimizing Eq. (2). Then, we have that  $J(\pi_i^*) - J(\hat{\pi}_{\beta_i}) \geq \frac{\alpha}{1-\gamma} \mathbb{E}_{o_i \sim d^{\pi_i^*}(o_i)} [D(\pi_i^*, \hat{\pi}_{\beta_i})(o_i)] + \frac{\tau}{1-\tau} \mathbb{E}_{o_i \sim d^{\pi_i^*}(o_i)} [(\pi_i^*(o_i) - \hat{a}_i)^2] - \frac{\tau}{1-\tau} \mathbb{E}_{o_i \sim d^{\hat{\pi}_{\beta_i}}(o_i), a_i \sim \hat{\pi}_{\beta_i}(a_i|o_i)} [(a_i - \hat{a}_i)^2]$ , where  $D(\pi_i, \hat{\pi}_{\beta_i})(o_i) = \frac{1}{\hat{\pi}_{\beta_i}(\pi_i(o_i)|o_i)} - 1$ , and  $d^{\pi_i}(o_i)$  is the marginal discounted distribution of observations of policy  $\pi_i$ .

*Proof.* For OMAR, we have the following iterative update for agent  $i$ :

$$\begin{aligned} \hat{Q}_i^{k+1} \leftarrow \arg \min_{Q_i} \alpha \mathbb{E}_{o_i \sim \mathcal{D}_i} \left[ \mathbb{E}_{a_i \sim \hat{\pi}_i(a_i|o_i)} [Q_i(o_i, a_i)] - \mathbb{E}_{a_i \sim \hat{\pi}_{\beta_i}(a_i|o_i)} [Q_i(o_i, a_i)] \right] \\ + \frac{1}{2} \mathbb{E}_{o_i, a_i, o_i' \sim \mathcal{D}} \left[ \left( Q_i(o_i, a_i) - \hat{B}^{\pi_i} \hat{Q}_i^k(o_i, a_i) \right)^2 \right], \end{aligned} \quad (4)$$

where  $\hat{\pi}_i(a_i|o_i) = 1$  if and only if  $a_i = \pi_i(o_i)$ .

Let  $\hat{Q}_i^{k+1}$  be the fixed point of solving Equation (4) by setting the derivative of Eq. (4) with respect to  $Q_i$  to be 0, then we have that

$$\hat{Q}_i^{k+1}(o_i, a_i) = \hat{B}^{\pi_i} \hat{Q}_i^k(o_i, a_i) - \alpha \left( \frac{I_{a_i=\pi_i(o_i)}}{\hat{\pi}_{\beta_i}(a_i|o_i)} - 1 \right), \quad (5)$$

where  $I$  is the indicator function.

Denote  $D(\pi_i, \hat{\pi}_{\beta_i})(o_i) = \frac{1}{\hat{\pi}_{\beta_i}(\pi_i(o_i)|o_i)} - 1$ , and we obtain the difference between the value function  $\hat{V}_i(o_i)$  and the original value function as:

$$\hat{V}_i(o_i) = V_i(o_i) - \alpha D(\pi_i, \hat{\pi}_{\beta_i})(o_i), \quad (6)$$

Then, the policy that minimizes the loss function defined in Eq. (2) is equivalently obtained by maximizing

$$(1 - \tau) \left( J(\pi_i) - \alpha \frac{1}{1-\gamma} \mathbb{E}_{o_i \sim d_{M_i}^{\pi_i}(o_i)} [D(\pi_i, \hat{\pi}_{\beta_i})(o_i)] \right) - \tau \mathbb{E}_{o_i \sim d_{M_i}^{\pi_i}(o_i)} [(\pi_i(o_i) - \hat{a}_i)^2]. \quad (7)$$

Therefore, we obtain that

$$\begin{aligned} (1 - \tau) \left( J(\pi_i^*) - \alpha \frac{1}{1-\gamma} \mathbb{E}_{o_i \sim d_{M_i}^{\pi_i^*}(o_i)} [D(\pi_i^*, \hat{\pi}_{\beta_i})(o_i)] \right) - \tau \mathbb{E}_{o_i \sim d_{M_i}^{\pi_i^*}(o_i)} [(\pi_i^*(o_i) - \hat{a}_i)^2] \\ \geq (1 - \tau) J(\hat{\pi}_{\beta_i}) - \tau \mathbb{E}_{o_i \sim d_{M_i}^{\hat{\pi}_{\beta_i}}(o_i), a_i \sim \hat{\pi}_{\beta_i}(a_i|o_i)} [(a_i - \hat{a}_i)^2]. \end{aligned} \quad (8)$$

Then, from Eq. (8) we obtain the result.  $\square$

## B MORE DETAILS OF THE EXPERIMENTS

### B.1 EXPERIMENTAL SETUP

**Tasks.** We adopt the open-source implementations for multi-agent particle environments<sup>2</sup> from (Lowe et al., 2017) and Multi-Agent MuJoCo<sup>3</sup> from (Peng et al., 2020). Figure 5 illustrates the tasks.

<sup>2</sup><https://github.com/openai/multiagent-particle-envs>

<sup>3</sup>[https://github.com/schroederdewitt/multiagent\\_mujoco](https://github.com/schroederdewitt/multiagent_mujoco)

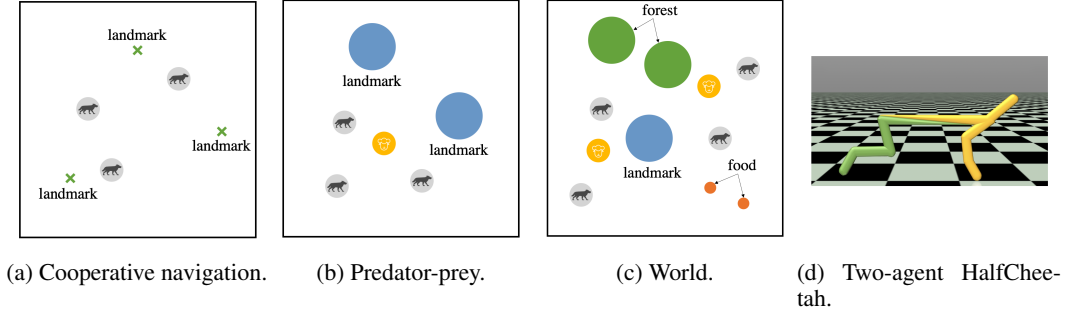


Figure 5: Multi-agent particle environments and Multi-Agent HalfCheetah.

**Baselines.** All baseline methods are implemented based on an open-source implementation<sup>4</sup> from (Iqbal & Sha, 2019), where we implement MA-TD3+BC<sup>5</sup>, MA-CQL<sup>6</sup> and MA-ICQ<sup>7</sup> based on authors’ open-source implementations with fine-tuned hyperparameters. For MA-CQL, we tune a best critic regularization coefficient from  $\{0.1, 0.5, 1.0, 5.0\}$  following (Kumar et al., 2020) for each task. Specifically, we use the discount factor  $\gamma$  of 0.99. We sample a minibatch of 1024 samples from the dataset for updating each agent’s actor and critic using the Adam (Kingma & Ba, 2014) optimizer with the learning rate to be 0.01. The target networks for the actor and critic are soft updated with the update rate to be 0.01. Both the actor and critic networks are feedforward networks consisting of two hidden layers with 64 neurons per layer using ReLU activation. For OMAR, the only hyperparameter that requires tuning is the regularization coefficient  $\lambda$ , where we use a smaller value for datasets with more diverse data distribution in random and medium-replay with a value of 0.5, while we use a larger value for datasets with more narrow data distribution in medium and expert with values of 0.7 and 0.9 respectively. As OMAR is insensitive to the hyperparameters of the sampling mechanism, we set them to a fixed set of values for all types of datasets in all tasks, where the number of iteration is 3, the number of samples is 10, the mean is 0.0, and the standard deviation is 2.0. The code will be released upon publication of the paper.

<sup>4</sup><https://github.com/shariqiqbal2810/maddpg-pytorch>

<sup>5</sup>[https://github.com/sfujim/TD3\\_BC](https://github.com/sfujim/TD3_BC)

<sup>6</sup><https://github.com/aviralkumar2907/CQL>

<sup>7</sup><https://github.com/YiqinYang/ICQ>

## B.2 LEARNING CURVES

Figure 6 demonstrates the learning curves of MA-ICQ, MA-TD3+BC, MA-CQL and OMAR in different types of datasets in multi-agent particle environments, where the solid line and shaded region represent mean and standard deviation, respectively.

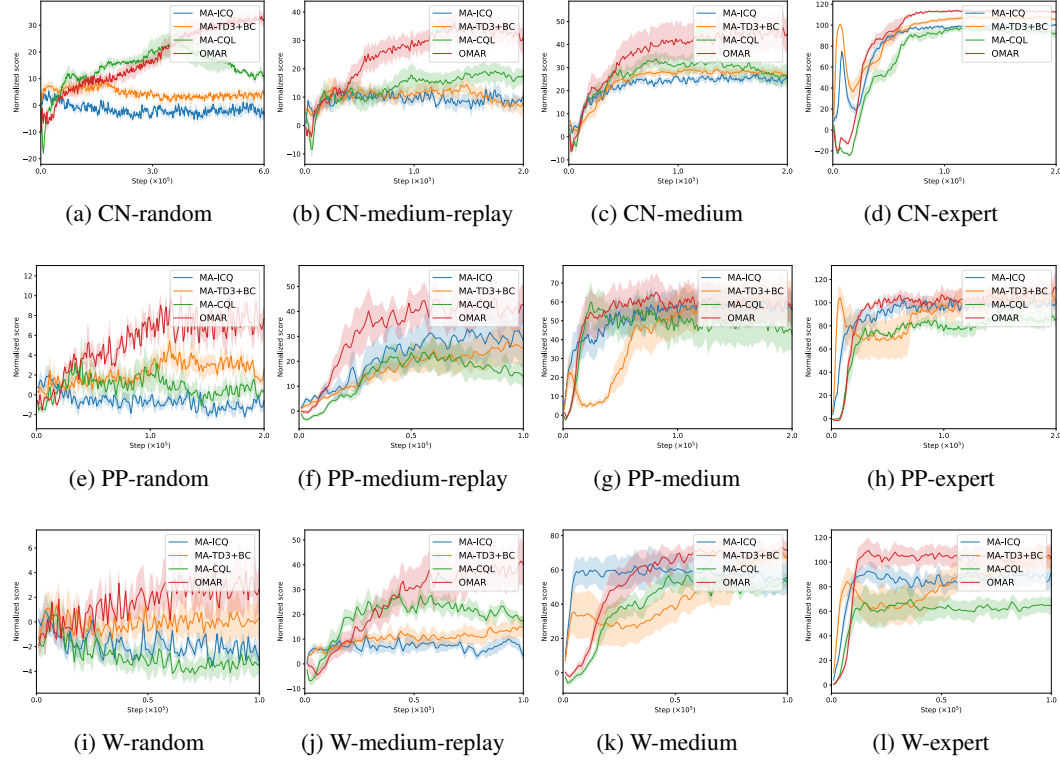


Figure 6: Learning curves of MA-ICQ, MA-TD3+BC, MA-CQL, and OMAR in multi-agent particle environments (CN, PP, and W is abbreviated for cooperative navigation, predator-prey, and world respectively).