POCE: Primal Policy Optimization with Conservative Estimation for Multi-constraint Offline Reinforcement Learning

Jiayi Guan[1†] , Li Shen[2†], Ao Zhou[1], Lusong Li[2],Han Hu[3],
Xiaodong He[2], Guang Chen[1‡], Changjun Jiang[1]
[1]Tongji University, [2] JD Explore Academy, [3] Beijing Institute of Technology

A. Proofs and Discussions

A.1. Proof and Discussion the Lemma 4.1

Lemma A.1 *The objective of the multi-constraint offline RL with cumulative and state-wise costs can be formulated as:*

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t r(\hat{s}_t, a_t)\right],$$
$$\text{s.t.}\mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t c_i(\hat{s}_t, a_t)\right] \le \bar{c}_i,\ \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty} D_i(\hat{s}_t, a_t)\right] \le \bar{D}_i. \tag{1}$$

where $r(\hat{s}_t, a_t) \triangleq r(s_t, a_t)$ *and* $c_i(\hat{s}_t, a_t) \triangleq c_i(s_t, a_t)$.

Proof. Before proving this Lemma, let us first review the definition of the objective in multi-constraint offline RL.

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t)\right],$$
$$\text{s.t.}\quad \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t c_i(s_t, a_t)\right] \le \bar{c}_i, \tag{2}$$
$$\forall t \ge 0,\ \mathbb{E}_{s_t\sim p(\cdot|s_t,a_t), a_t\sim\pi(\cdot|s_t)}[c_i(s_t, a_t)] \le \bar{D}_i.$$

Considering that Constraint 2 in the Eq. (2) requires that the state-wise costs $c_i(s, a)$ at any state are below the state-wise cost threshold $\bar{D}_i$, which is difficult to handle in algorithm implementation, we propose ensuring that the maximum state-wise cost in each trajectory is below the cost threshold to guarantee that the state-wise costs at any state are below the state-wise cost threshold. The above proposal can be mathematically expressed as follows:

$$\forall t \ge 0,\ \mathbb{E}_{s_t\sim p, a_t\sim\pi}[c_i(s_t, a_t)] \le \max_{s_t, a_t} c_i(s_t, a_t) \le \bar{D}_i. \tag{3}$$

Accorging to Eq. (3), we rewrite Eq. (2) as follows:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t)\right],$$
$$\text{s.t. } \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t c_i(s_t, a_t)\right] \le \bar{c}_i,\ \max_{s_t, a_t} c_i(s_t, a_t) \le \bar{D}_i. \tag{4}$$

Although Eq. (4) has been simplified, we still find it challenging to implement. To facilitate iterative computations, we consider transforming constraint 2 in Eq. (4) into a structure similar to constraint 1. Inspired by SCPO [4], we convert the state-wise cost into a state-wise cost increment. By introducing a set of maximum state-wise cost $M_{i,t}$ and state-wise cost increment $D_{i,t}$, we transform the maximum value constraint problem of constraint 2 in Eq. (4) into a cumulative state-wise cost increment constraint problem similar to constraint 1. In addition, to facilitate the computation of state-wise cost increments, we augment the maximum state-wise cost into the observed state, expanding the observed state as $\hat{s}_t = (s_t, M_{i,t})$. We define the maximum state-wise cost and state-wise cost increment as follows:

$$D_i(\hat{s}_t, a_t) = \max\{c_i(s_t, a_t) - M_{i,t}, 0\}, \tag{5}$$
$$M_{i,t} = \sum_{k=0}^{k=t-1} D_i(\hat{s}_k, a_k), \tag{6}$$

where $M_{i,t} = M_i(\hat{s}_t, a_t)$ represents the maximum state-wise cost of the i-th cost at step t, and $D_{i,t} = D_i(\hat{s}_t, a_t)$ represents the increment of the state-wise of the i-th cost at step t. Additionally, the initial values of the maximum state-wise cost and the increment of the state-wise cost are defined as $M_{i,0} = 0$ and $D_i(\hat{s}_0, a_0) = c_i(s_0, a_0)$. Combining the maximum state-wise cost and state-wise cost increment, the maximum state-wise cost constraint in Eq. (4) is transformed into a constraint on the cumulative of state-wise cost increments.

$$\mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty} D_i(\hat{s}_t, a_t)\right] \le \bar{D}_i \iff \max_{s_t, a_t} c_i(\hat{s}_t, a_t) \le \bar{D}_i. \tag{7}$$

Based on Eq. (7) and (4), we deduce the conclusion stated in Lemma A.1, which defines the objective of multi-constraint

† Equal Contribution;
‡ Corresponding author: guangchen@tongji.edu.cn.

offline RL as follows:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t r(\hat{s}_t, a_t)\right],$$
$$\text{s.t.}\mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t c_i(\hat{s}_t, a_t)\right] \le \bar{c}_i,\ \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty} D_i(\hat{s}_t, a_t)\right] \le \bar{D}_i. \tag{8}$$

Note that for the convenience of comprehension and expression, we denote $\mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty} D_i(\hat{s}_t, a_t)\right] \le \bar{D}_i$ as a state-wise cost constraint.

A.2. Proof and Discussion the Remark 4.2

Remark A.2 *When the state-wise cost satisfies the cost constraint $\mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty} D_i(\hat{s}_t, a_t)\right] \le \bar{D}_i$ and the cost threshold of state-wise cost satisfies the condition $\bar{D}_i \le (1-\gamma)\bar{c}_i$, then the cumulative cost also satisfies the cost constraint $\mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t c(\hat{s}_t, a_t)\right] \le \bar{c}_i$.*

Proof. From the cumulative cost constraint, we have:

$$\begin{aligned}
&\mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t c_i(\hat{s}_t, a_t)\right] = \sum_{t=0}^{\infty}\gamma^t\left[\mathbb{E}_{\hat{s}_t\sim p, a_t\sim\pi} c_i(\hat{s}_t, a_t)\right] \\
&\overset{\{i\}}{\le} \sum_{t=0}^{\infty}\gamma^t\left[\mathbb{E}_{\tau\sim\pi}\sum_{t=0}^{\infty} D_i(\hat{s}_t, a_t)\right] \overset{\{ii\}}{\le} \sum_{t=0}^{\infty}\gamma^t \bar{D}_i \\
&= \bar{D}_i\sum_{t=0}^{\infty}\gamma^t \overset{\{iii\}}{=} \frac{\bar{D}_i}{1-\gamma},
\end{aligned} \tag{9}$$

where $\{i\}$ follows the Eq. (7) and (3) with $\mathbb{E}_{\hat{s}_t\sim p, a_t\sim\pi} c_i(\hat{s}_t, a_t) \le \mathbb{E}_{\tau\sim\pi}\sum_{t=0}^{\infty} D_i(\hat{s}_t, a_t)$, $\{ii\}$ follows the state-wise cost constraints condition, $\{iii\}$ follows the summation formula of a geometric series $\frac{1}{1-\gamma} = \sum_{t=0}^{\infty}\gamma^t$. Assuming the condition $\frac{\bar{D}_i}{1-\gamma} \le \bar{c}_i$ is satisfied, then we have:

$$\bar{c}_i \ge \frac{\bar{D}_i}{1-\gamma} \ge \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t c_i(\hat{s}_t, a_t)\right]. \tag{10}$$

According to Eq. (10), we conclude that when the condition $\bar{D}_i \le (1-\gamma)\bar{c}_i$ is satisfied, the state-wise cost constraint $\mathbb{E}_{\hat{s}_t\sim p(\cdot|\hat{s}_t,a_t), a_t\sim\pi}[D_i(\hat{s}_t, a_t)] \le \bar{D}_i$ encompasses the cumulative cost constraint $\mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t c_i(\hat{s}_t, a_t)\right] \le \bar{c}_i$.

A.3. Proof and Discussion the Proposition 4.3

Proposition A.3 *Within the scope of behavioral policies π_β, the conditional Bellman operator $\mathcal{T}_{\mathcal{CB}}$ is a γ-contractive operator under the $\mathcal{L}_{+\infty}$ norm, and any initial Q-value can converge to a unique fixed point through $\mathcal{T}_{CB}$.*

Proof. Before proving this proposition, let's first review the definition of the conditional Bellman operator.

$$\mathcal{T}_{\mathcal{CB}}Q(\hat{s}, a) = \begin{cases} \chi(\hat{s}, a) + \gamma\mathbb{E}_{\hat{s}'}\max_{a'} Q(\hat{s}', a'), & (a \in \pi_\beta(a|\hat{s})) \\ \max_{\hat{a}\sim\pi_\beta(\cdot|\hat{s})} Q(\hat{s}, \hat{a}), & (a \notin \pi_\beta(a|\hat{s})) \end{cases}, \tag{11}$$

Let Q_1 and Q_2 be two arbitrary Q-values obtained through iterative computation using the conditional Bellman equation. To analyze the iterative properties of the conditional Bellman operator, we categorize the discussion based on whether action a belongs to the behavior policy π_β.

(**Case 1** $a \in \pi_\beta(a|\hat{s})$) In this case, based on the conditional Bellman equation in Eq. (11), we can obtain:

$$\begin{aligned}
&\|\mathcal{T}_{\mathcal{CB}}Q_1(\hat{s}, a) - \mathcal{T}_{\mathcal{CB}}Q_2(\hat{s}, a)\|_{+\infty} \\
&= \max_{\hat{s},a}\Bigg|\left[\chi(\hat{s}, a) + \gamma\mathbb{E}_{\hat{s}'}\max_{a'} Q_1(\hat{s}', a')\right] - \\
&\qquad\left[\chi(\hat{s}, a) + \gamma\mathbb{E}_{\hat{s}'}\max_{a'} Q_2(\hat{s}', a')\right]\Bigg| \\
&= \max_{\hat{s},a}\left|\gamma\mathbb{E}_{\hat{s}'}\max_{a'} Q_1(\hat{s}', a') - \gamma\mathbb{E}_{\hat{s}'}\max_{a'} Q_2(\hat{s}', a')\right| \\
&= \gamma\max_{\hat{s},a}\left|\mathbb{E}_{\hat{s}'}\left[\max_{a'} Q_1(\hat{s}', a') - \max_{a'} Q_2(\hat{s}', a')\right]\right| \\
&\overset{\{i\}}{\le} \gamma\max_{\hat{s},a}\mathbb{E}_{\hat{s}'}\left|\max_{a'} Q_1(\hat{s}', a') - \max_{a'} Q_2(\hat{s}', a')\right| \\
&\le \gamma\max_{\hat{s},a}\left\|Q_1 - Q_2\right\|_{+\infty} \\
&= \gamma\left\|Q_1 - Q_2\right\|_{+\infty},
\end{aligned} \tag{12}$$

where $\{i\}$ follow the Jensen's inequality $\mathbb{E}(f(x)) \ge f(\mathbb{E}(x))$.

(**Case 2** $a \notin \pi_\beta(a|\hat{s})$) In this case, based on the conditional Bellman equation in Eq. (11), we can obtain:

$$\begin{aligned}
&\|\mathcal{T}_{\mathcal{CB}}Q_1(\hat{s}, a) - \mathcal{T}_{\mathcal{CB}}Q_2(\hat{s}, a)\|_{+\infty} \\
&= \max_{\hat{s},a}\left|\max_{\hat{a}} Q_1(\hat{s}, \hat{a}) - \max_{\hat{a}} Q_2(\hat{s}, \hat{a})\right| \\
&= \max_{\hat{s},a}\Bigg|\max_{\hat{a}}\left[\chi(\hat{s}, \hat{a}) + \gamma\mathbb{E}_{\hat{s}'}\max_{\hat{a}'} Q_1(\hat{s}', \hat{a}')\right] - \\
&\qquad\max_{\hat{a}}\left[\chi(\hat{s}, \hat{a}) + \gamma\mathbb{E}_{\hat{s}'}\max_{\hat{a}'} Q_2(\hat{s}', \hat{a}')\right]\Bigg| \\
&= \max_{\hat{s},a}\left|\max_{\hat{a}}\left[\gamma\mathbb{E}_{\hat{s}'}\max_{\hat{a}'} Q_1(\hat{s}', \hat{a}') - \gamma\mathbb{E}_{\hat{s}'}\max_{\hat{a}'} Q_2(\hat{s}', \hat{a}')\right]\right| \\
&= \max_{\hat{s},a}\left|\gamma\mathbb{E}_{\hat{s}'}\max_{\hat{a}'} Q_1(\hat{s}', \hat{a}') - \gamma\mathbb{E}_{\hat{s}'}\max_{\hat{a}'} Q_2(\hat{s}', \hat{a}')\right| \\
&\le \gamma\max_{\hat{s},a}\mathbb{E}_{\hat{s}'}\left|\max_{\hat{a}'} Q_1(\hat{s}', \hat{a}') - \max_{\hat{a}'} Q_2(\hat{s}', \hat{a}')\right| \\
&\le \gamma\max_{\hat{s},a}\left\|Q_1 - Q_2\right\|_{+\infty} \\
&= \gamma\left\|Q_1 - Q_2\right\|_{+\infty}.
\end{aligned} \tag{13}$$

Based on the analysis of the aforementioned **Case 1** and **Case 2**, we conclude that within the range supported by the

behavioral policy π_β, the conditional Bellman operator $\mathcal{T}_{\mathcal{CB}}$ is a γ-contraction operator under the $\mathcal{L}_{+\infty}$ norm, and any initial Q-value can converge to a unique fixed point through $\mathcal{T}_{\mathcal{CB}}$.

A.4. Proof and Discussion the Proposition 4.4

Proposition A.4 *As the unique fixed point of the conditional Bellman operator, $Q_{\mathcal{T}_{CB}}$ is bounded within the range of behavioral policies π_β, with $Q_{\mathcal{T}_{CB}} \in [Q_{\pi_\beta}, Q_{\pi^*_\beta}]$. Here, Q_{π_β} represents the Q-value of the behavioral policy, and $Q_{\pi^*_\beta}$ represents the Q-value of the optimal policy.*

Proof. Before proving this proposition, we first define $Q_{\pi_\beta}(\hat{s}, a) = \chi(\hat{s}, a) + \gamma \mathbb{E}_{\hat{s}'} \mathbb{E}_{a' \sim \pi_\beta}(\hat{s}', a')$ and $Q_{\pi^*_\beta}(\hat{s}, a) = \chi(\hat{s}, a) + \gamma \mathbb{E}_{\hat{s}'} \max_{a' \sim \pi_\beta}(\hat{s}', a')$, Separately. Subsequently, we continue to employ a categorization approach to discuss the range of Q-value $Q_{\mathcal{T}_{\mathcal{CB}}}$ obtained through conditional Bellman iteration.

(**Case 1** $a \in \pi_\beta(a|\hat{s})$)in this case, based on the conditional Bellman equation in Eq. (11) we obtain:

$$Q_{\mathcal{T}_{\mathcal{CB}}}(\hat{s}, a) = \chi(\hat{s}, a) + \gamma \mathbb{E}_{\hat{s}'} \max_{a' \sim \pi_\beta(\cdot|\hat{s}')} Q(\hat{s}', a') = Q_{\pi^*_\beta} \quad . \tag{14}$$

Additionally, we have:

$$\begin{aligned} Q_{\mathcal{T}_{\mathcal{CB}}}(\hat{s}, a) &= \chi(\hat{s}, a) + \gamma \mathbb{E}_{\hat{s}'} \max_{a' \sim \pi_\beta(\cdot|\hat{s}')} Q(\hat{s}', a') \\ &\geq \chi(\hat{s}, a) + \gamma \mathbb{E}_{\hat{s}'} \mathbb{E}_{a' \sim \pi_\beta} Q(\hat{s}', a') \\ &= Q_{\pi_\beta} \end{aligned} \quad . \tag{15}$$

(**Case 2** $a \notin \pi_\beta(a|\hat{s})$) in this case, based on the conditional Bellman equation in Eq.(11) we obtain:

$$\begin{aligned} Q_{\mathcal{T}_{\mathcal{CB}}}(\hat{s}, a) &= \max_{\hat{a} \sim \pi_\beta(\cdot|s)} Q(\hat{s}, \hat{a}) \\ &= \max_{\hat{a} \sim \pi_\beta(\cdot|\hat{s})} \left[\chi(\hat{s}, \hat{a}) + \gamma \mathbb{E}_{\hat{s}'} \max_{a' \sim \pi_\beta(\cdot|\hat{s}')} Q(\hat{s}', a') \right] \\ &= \chi(\hat{s}, \hat{a}) + \gamma \max_{\hat{a} \sim \pi_\beta(\cdot|\hat{s})} \mathbb{E}_{\hat{s}'} \max_{a' \sim \pi_\beta(\cdot|\hat{s}')} Q(\hat{s}', a') \\ &= \chi(\hat{s}, \hat{a}) + \gamma \mathbb{E}_{\hat{s}'} \max_{a' \sim \pi_\beta(\cdot|\hat{s}')} Q(\hat{s}', a') \\ &= Q_{\pi^*_\beta}(\hat{s}, \hat{a}) \end{aligned} \quad . \tag{16}$$

In addition, we have:

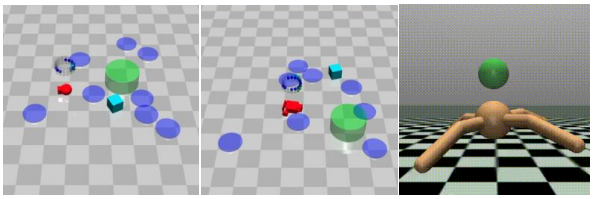
$$\begin{aligned} Q_{\mathcal{T}_{\mathcal{CB}}}(\hat{s}, a) &= \max_{\hat{a} \sim \pi_\beta(\cdot|\hat{s})} Q(\hat{s}, \hat{a}) \\ &= \max_{\hat{a} \sim \pi_\beta(\cdot|\hat{s})} \left[\chi(\hat{s}, \hat{a}) + \gamma \mathbb{E}_{\hat{s}'} \max_{a' \sim \pi_\beta(\cdot|\hat{s}')} Q(\hat{s}', a') \right] \\ &= \chi(\hat{s}, \hat{a}) + \gamma \max_{\hat{a} \sim \pi_\beta(\cdot|\hat{s})} \mathbb{E}_{\hat{s}'} \max_{a' \sim \pi_\beta(\cdot|\hat{s}')} Q(\hat{s}', a') \\ &\geq \chi(\hat{s}, \hat{a}) + \gamma \mathbb{E}_{\hat{s}'} \mathbb{E}_{a' \sim \pi_\beta(\cdot|\hat{s}')} Q(\hat{s}', a') \\ &= Q_{\pi_\beta}(\hat{s}, \hat{a}) \end{aligned} \tag{17}$$

Based on the analysis of **Case 1** and **Case 2** mentioned above, we conclude that the Q-values $Q_{\mathcal{T}_{\mathcal{CB}}}$ obtained through the iteration of the conditional Bellman equation converge to $Q_{\mathcal{T}_{\mathcal{CB}}} \in [Q_{\pi_\beta}, Q_{\pi^*_\beta}]$ within the range supported by the behavioral policy π_β.

B. Experimental Details

B.1. Task and Dataset

Task. To evaluate the performance of POCE in various tasks across different domains, we select widely adopted tasks [2, 3] including *PointGoal*, *CarGoal*, and *AntVelocity* as the experimental tasks for this work. The schematic diagrams of the three tasks are presented in Fig. 1.



(*a*) (*b*) (*c*)

Figure 1. The three experimental tasks employed in this work. Where (a) and (b) illustrate the *PointGoal* and *CarGoal* tasks belonging to the Safety-Gym domain, and (c) portrays the *AntVelocity* task from the Mujoco domain. Additionally, the maximum number of steps for these three tasks is set to 500 steps.

The *PointGoal* and *CarGoal* tasks require the agent to resist the interference of non-hazardous obstacles in the environment and navigate around dangerous obstacles to reach the target location. The agent receives rewards for approaching or reaching the target location, and the environment resets when the agent reaches the target location or reaches the maximum simulation steps set. Additionally, when the agent enters a hazardous obstacle area, the environment provides a non-zero cost feedback, and the agent is able to pass through the hazardous obstacles. Note that the existing *PointGoal* and *CarGoal* tasks have a fixed constant cost, where the agent incurs a cost $c = 1$ whenever it enters the hazardous obstacle area. This setting is not consistent with real-world applications, where the costs incurred by the agent should differ between the edge and center regions of the hazardous obstacle area. Inspired by this, we set the cost of the agent at the center of a hazardous obstacle to be 1, and the cost at the edge of the danger zone to be zero, with a linear increase in cost as the distance decreases. Therefore, the cost function in the *PointGoal* and *CarGoal* tasks be formulated as follows:

$$c = \begin{cases} 0, & (d > \bar{d}) \\ \dfrac{\bar{d} - d}{\bar{d}}, & (d \leq \bar{d}) \end{cases}, \tag{18}$$

where d is the distance between the agent and the center of the hazardous obstacle, and $\bar{d}$ is the maximum distance from the center of the hazardous obstacle to its edge. By applying the aforementioned modifications to the cost function, the range of the cost values is transformed from the original $c \in \{0, 1\}$ to $c \in [0, 1]$.

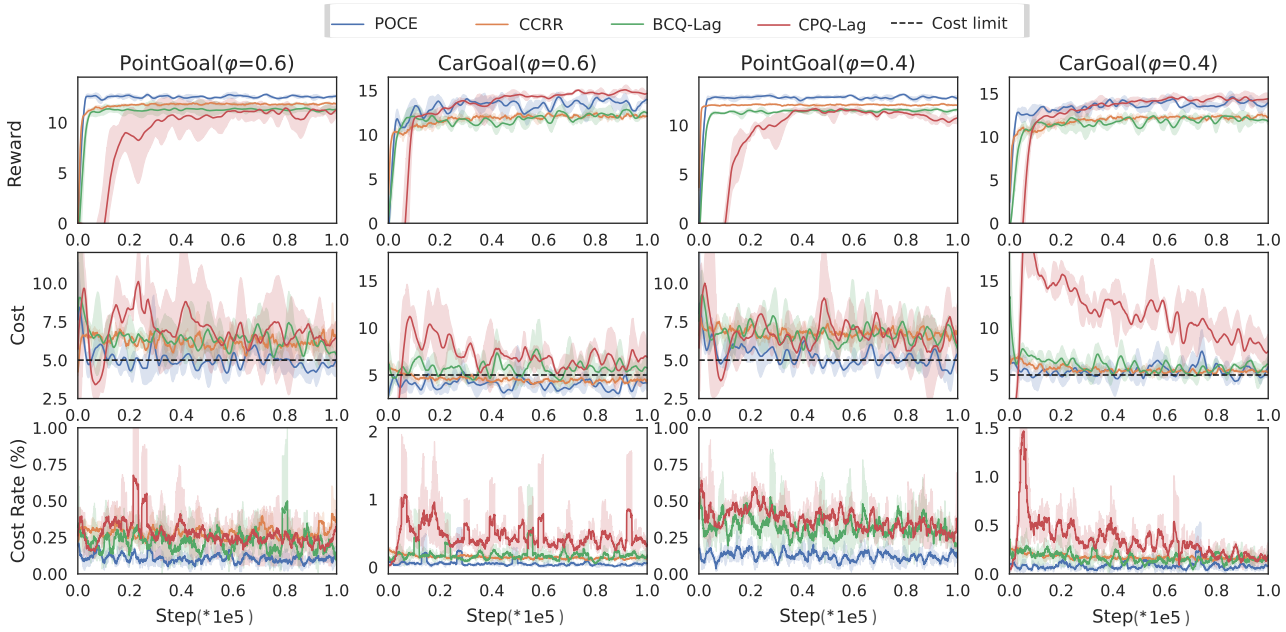


Figure 2. The reward, cumulative cost, and state-wise cost violation rate curves for the tasks. The shaded curves displayed represent the mean and variance of online testing during the training process with three different random seeds. The safety factor for this experiment is $\varphi = 0.6$ and $\varphi = 0.4$, with a cumulative cost threshold set at $\bar{c}_i = 5$. and a state-wise cost threshold set to $\bar{D}_i = 0.8$.

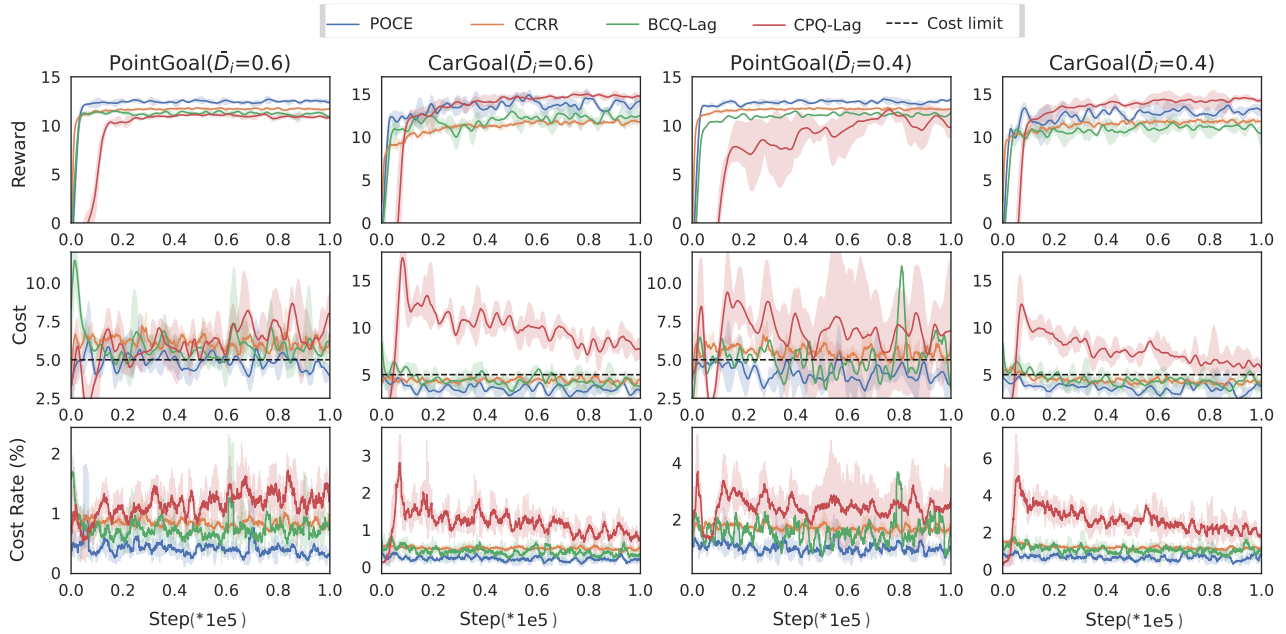


Figure 3. The reward, cumulative cost, and state-wise cost violation rate curves for the tasks. The shaded curves displayed represent the mean and variance of online testing during the training process with three different random seeds. The safety factor for this experiment is $\varphi = 0.8$, with the cumulative cost threshold set at $\bar{c}_i = 5$, and the state-wise threshold set at $\bar{D}_i = 0.6$ and $\bar{D}_i = 0.4$.

The *AntVelocity* task requires the agent to walk or run within a specified velocity range. The higher the velocity of the ant agent, the higher the reward obtained. However, when the agent's velocity exceeds a predefined threshold, the environment provides non-zero cost feedback. Similar to the problems of the *PointGoal* and *CarGoal* tasks, the

cost of the existing *AntVelocity* task is also not a continuous real number within the range $[0, 1]$. Therefore, we redefine the cost function as follows:

$$c = \begin{cases} 0, & (v < \bar{v}) \\ \dfrac{v - \bar{v}}{v_{\max} - \bar{v}}, & (v \geq \bar{v}) \end{cases}, \tag{19}$$

where $v_{\max}$ is the maximum velocity of the ant in the environment, and $\bar{v}$ is the speed limit set for the current task.

Dataset. To validate the performance of the algorithm under samples with different behaviors, we collected sample data of various behaviors. Additionally, to quantify the sample data of different behaviors, we are inspired by VOCE [1] and introduced a safety factor φ to measure the sample data of different behaviors. The safety factor φ is the proportion of safe episodes among the total number of sampled episodes.

Table 1. The hyper-parameters of the POCE algorithm model. Where s and a denote the dimensions of the state and action, respectively.

Sort	Hyper-parameters	Setting
Policy	Number of neurons	$(s+a)\times256\times256\times256\times a$
	Activation function	ReLu
	Number of networks	2
	Learning rate	5.00e-05
	Optimizer	Adam
Q-value(r)	Number of neurons	$(s+a)\times256\times256\times256\times1$
	Activation function	ReLu
	Number of networks	2
	Learning rate	1.00e-04
	Optimizer	Adam
Q-value(c_i)	Number of neurons	$(s+a)\times256\times256\times256\times1$
	Activation function	ReLu
	Number of networks	2
	Learning rate	1.00e-04
	Optimizer	Adam
Q-value(D_i)	Number of neurons	$(s+a)\times256\times256\times256\times1$
	Activation function	ReLu
	Number of networks	2
	Learning rate	1.00e-04
	Optimizer	Adam
CVAE	Number of neurons(e)	$(s+a)\times750\times750\times(2a+2a)$
	Number of neurons(d)	$(s+2a)\times750\times750\times a$
	Activation function	ReLU
	Number of networks	1
	Learning rate	1.00e-03
	Optimizer	Adam
Others	Batch size	256
	Discount factor γ	0.99
	Balances factors for OOD λ	0.995
	Network update factors τ	0.005

B.2. Experimental Results

For the convenience of examining the trend in various performance metrics during the algorithm training process, we record the curves of rewards, cumulative cost, and state-wise costs in both the POCE and baseline algorithm training. These experimental results, supplementary to the comparative experiments in the manuscript, serve to provide a clearer understanding of the performance of the POCE and baseline algorithms during the training process.

Performance on various tasks and behavioral samples. Fig. 2 illustrates the testing curves of the POCE and baseline algorithms for *PointGoal* and *CarGoal* tasks under various safety factors. The results from the graph indicate that the curve of the POCE shows a stable variation and the cumulative costs of the sample under different safety factors all fall within the cost threshold range. Furthermore, the violation rate of state-wise costs for the POCE is consistently lower than that of other baseline algorithms. The above results demonstrate that the POCE algorithm ensures that cumulative costs meet constraints for samples under different safety factors, and it also surpasses other baseline algorithms in terms of adhering to state-wise constraints.

The parameters safety factors φ and state-wise cost thresholds $\bar{D}_i$. Fig. 3 illustrates the testing curves of the POCE and baseline algorithms for the *PointGoal* and *CarGoal* tasks under different state-wise cost thresholds $\bar{D}_i$. The results from the graph indicate that the POCE algorithm, under various state-wise cost thresholds, consistently satisfies cumulative cost constraints while providing competitive reward returns. Additionally, we observed that as the state-wise cost threshold is set lower, the violation rate of state-wise costs increases, resulting in lower cumulative costs and rewards.

B.3. Experimental Setting

This experiment is conducted on a server equipped with an RTX 3090 for both training and testing. Additionally, the provided code is compatible with training and testing on both GPU and CPU. We provide a detailed explanation of the experimental environment and dataset in Section B.1. Table 1 displays the parameters of the neural network model utilized in our POCE algorithm. Additionally, detailed configuration information for the testing environment is provided in the code appendix. You can refer to the README file in the appendix code for instructions on installing and configuring the training and testing environment for the POCE model.

References

[1] Jiayi Guan, Guang Chen, Jiaming Ji, Long Yang, Ao Zhou, Zhijun Li, and changjun jiang. VOCE: Variational optimization with conservative estimation for offline safe reinforce-

ment learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 5

[2] Jiaming Ji, Jiayi Zhou, Borong Zhang, Juntao Dai, Xuehai Pan, Ruiyang Sun, Weidong Huang, Yiran Geng, Mickel Liu, and Yaodong Yang. Omnisafe: An infrastructure for accelerating safe reinforcement learning research. *arXiv preprint arXiv:2305.09304*, 2023. 3

[3] Zuxin Liu, Zijian Guo, Haohong Lin, Yihang Yao, Jiacheng Zhu, Zhepeng Cen, Hanjiang Hu, Wenhao Yu, Tingnan Zhang, Jie Tan, et al. Datasets and benchmarks for offline safe reinforcement learning. *arXiv preprint arXiv:2306.09303*, 2023. 3

[4] Weiye Zhao, Tairan He, Rui Chen, Tianhao Wei, and Changliu Liu. State-wise safe reinforcement learning: A survey. *arXiv preprint arXiv:2302.03122*, 2023. 1