A PROOFS

A.1 Proof of Lemma 4.2

We restate Lemma 4.2 as follows.

Lemma A.1. $\overleftarrow{C}_k^\pi(x_t, a_t) \overset{D}{=} C_k^\pi(x_{t-k}, a_{t-k}), \forall k \in \mathbb{N}$.

Proof. With the backward Markov chain characterized by $\overleftarrow{p}^\pi$, the probability for k-step trajectory from any state x_t is

$$\begin{aligned}
&\overleftarrow{p}^\pi(a_t, x_{t-1}, a_{t-1}, \cdots, x_{t-k}, a_{t-k}|x_t) \\
&= \pi(a_t|x_t) \cdot \overleftarrow{p}^\pi(x_{t-1}|x_t) \cdot \pi(a_{t-1}|x_{t-1}) \cdot \cdots\cdot \overleftarrow{p}^\pi(x_{t-k}|x_{t-k+1}) \cdot \pi(a_{t-k}|x_{t-k}) \\
&= \pi(a_t|x_t) \cdot p^\pi(x_t|x_{t-1}) \cdot \frac{\mu^\pi(x_{t-1})}{\mu^\pi(x_t)} \cdot \pi(a_{t-1}|x_{t-1}) \cdot \cdots\cdot p^\pi(x_{t-k+1}|x_{t-k}) \cdot \frac{\mu^\pi(x_{t-k})}{\mu^\pi(x_{t-k+1})} \cdot \pi(a_{t-k}|x_{t-k}) \\
&= \frac{\mu^\pi(x_{t-k})}{\mu^\pi(x_t)} p^\pi(a_{t-k}, x_{t-k+1}, \cdots, a_{t-1}, x_t, a_t|x_{t-k}),
\end{aligned}$$

where in the second equation we utilize the Bayesian's rule. This result shows that the forward probability for the trajectory $(x_{t-k}, a_{t-k}, \cdots, x_t, a_t)$ under policy π is equivalent to the probability of sampling a state x_t from the stationary distribution μ^π and obtaining the trajectory $(x_t, a_t, \cdots, x_{t-k}, a_{t-k})$ via the backwards chain. Given that the cost function is deterministic, thus, the accumulated cost also has the same distribution in the forward view and the backward view. □

A.2 Proof of Lemma 4.3

We restate Lemma 4.3 as follows.

Lemma A.2. *$\overleftarrow{\mathcal{T}}_c^\pi$ is a contraction mapping in the Wasserstein distance and $\overleftarrow{C}^\pi$ is the fixed point, i.e., $\overleftarrow{C}^\pi = \overleftarrow{\mathcal{T}}_c^\pi \overleftarrow{C}^\pi$.*

Proof. We use the same technique from [5]. With the finite-horizon T, we can convert it to an infinite-horizon setting with $\gamma = 1 - 1/T$. We will show that $\overleftarrow{\mathcal{T}}_c^\pi$ is a γ-contraction mapping in the Wasserstein distance.

Given two random variables U, V with cdfs F_U, F_V, the Wasserstein distance d_p is defined as follows.

$$d_p(U, V) \coloneqq d_p(F_U, F_V) = \left(\int_0^1 \left| F_U^{-1}(u) - F_V^{-1}(u) \right|^p du \right)^{1/p}.$$

For any two distributions $\overleftarrow{C}_1(x, a)$ and $\overleftarrow{C}_2(x, a)$, we define the maximal form of the Wasserstein distance as follows.

$$\bar{d}_p(\overleftarrow{C}_1, \overleftarrow{C}_2) = \sup_{x,a} d_p(\overleftarrow{C}_1(x, a), \overleftarrow{C}_2(x, a)).$$

Then,

$$\begin{aligned}
d_p(\overleftarrow{\mathcal{T}}_c^\pi \overleftarrow{C}_1(x, a), \overleftarrow{\mathcal{T}}_c^\pi \overleftarrow{C}_2(x, a)) &= d_p(c(x, a) + \gamma \overleftarrow{C}_1(x', a'), c(x, a) + \gamma \overleftarrow{C}_2(x', a')) \\
&\le \gamma d_p(\overleftarrow{C}_1(x', a'), \overleftarrow{C}_2(x', a')) \\
&\le \gamma \sup_{x,a} d_p(\overleftarrow{C}_1(x, a), \overleftarrow{C}_2(x, a)).
\end{aligned}$$

Thus,

$$\begin{aligned}
\bar{d}_p(\overleftarrow{\mathcal{T}}_c^\pi \overleftarrow{C}_1, \overleftarrow{\mathcal{T}}_c^\pi \overleftarrow{C}_2) &= \sup_{x,a} d_p(\overleftarrow{\mathcal{T}}_c^\pi \overleftarrow{C}_1(x, a), \overleftarrow{\mathcal{T}}_c^\pi \overleftarrow{C}_2(x, a)) \\
&\le \gamma \sup_{x,a} d_p(\overleftarrow{C}_1(x, a), \overleftarrow{C}_2(x, a)) \\
&= \gamma \bar{d}_p(\overleftarrow{C}_1, \overleftarrow{C}_2).
\end{aligned}$$

Hence, the operator $\overleftarrow{\mathcal{T}}_c^\pi$ is a contraction mapping, and $\overleftarrow{C}^\pi$ is the unique fixed point by inspection. □

A.3 Proof of Lemma 5.1

Lemma 5.1 is restated as follows.

Lemma A.3. *For any trajectory $\{(x_i, a_i)\}_{i=0}^{T}$ generated by $\forall \pi \in \Pi_{nsta}$, we have*

$$F^{-1}_{C^\pi(x_0,a_0)}(\tau) \le F^{-1}_{\overleftarrow{C}^\pi(x_{t-1},a_{t-1})}(\nu) + F^{-1}_{C^\pi(x_t,a_t)}(1-\nu+\tau),$$

where $t \in [1, \cdots, T]$, $\nu \in [0,1]$ is the quantile level of random variable $\overleftarrow{C}^\pi(x_{t-1}, a_{t-1})$ at realization $\sum_{t'=0}^{t'=t-1} c(x_{t'}, a_{t'})$, and $\tau \in [0, \nu]$.

Proof. We firstly consider the general case for any two random variables X and Y with unknown dependency structure. From the theory of copulas, there exists a copula $C_{XY}(\nu, w) = Pr(X \le F_X^{-1}(\nu), Y \le F_Y^{-1}(w))$, where $\nu, w \in [0,1]$ are quantiles.

With the Frechet-Hoeffding inequality, we have that

$$\max(0, 1-(1-\nu)-(1-w)) \le C_{XY}(\nu, w) \le \min(\nu, w).$$

Let $\tau = 1-(1-\nu)-(1-w)$ and fix τ. Then, $v = 1-\nu+\tau$ with $\nu \in [\tau, 1]$. Apply the left-hand side of the Frechet-Hoeffding inequality again:

$$\begin{aligned}\max(0,\tau) &\le C_{XY}(\nu, 1-\nu+\tau)\\ &= Pr(X \le F_X^{-1}(\nu), Y \le F_Y^{-1}(1-\nu+\tau))\end{aligned}$$

That is, the probability that both X and Y are below the value at quantile ν and $1-\nu+\tau$ respectively is at least τ. In other words, we have that

$$F^{-1}_{X+Y}(\tau) \le F_X^{-1}(\nu) + F_Y^{-1}(1-\nu+\tau). \tag{5}$$

Hence, for the cost return random variable $C^\pi(x_0, a_0)$, we can decompose it into two random variables $C^\pi(x_0,a_0) = \sum_{t'=0}^{t-1} c(x_{t'}, a_{t'}) + \sum_{t'=t}^{T} c(x_{t'}, a_{t'})$ for any $t \in \mathcal{T}$, with $x_{t+1} \sim p(\cdot|x_t, a_t)$, $a_t \sim \pi(\cdot|x_t)$. As stated in Sec. 4.2, the first part $\sum_{t'=0}^{t-1} c(x_{t'}, a_{t'})$ in right-hand-side can be described by random variable $\overleftarrow{C}^\pi(x_{t-1}, a_{t-1})$, while the second part $\sum_{t'=t}^{T} c(x_{t'}, a_{t'})$ can be described by random variable $C^\pi(x_t, a_t)$. Apply inequality (5), we have that

$$F^{-1}_{C^\pi(x_0,a_0)}(\tau) \le F^{-1}_{\overleftarrow{C}^\pi(x_{t-1},a_{t-1})}(\nu) + F^{-1}_{C^\pi(x_t,a_t)}(1-\nu+\tau), \forall x_t, a_t,$$

where $\nu \in [0,1], \tau \in [0, \nu]$. □

A.4 Proof of Theorem

The formal theorem and all assumptions are given as follows.

Theorem A.4. *We firstly make the following assumptions.*

- *The state space $\mathcal{S}$ and action space $\mathcal{A}$ are discrete and finite;*
- *Each policy in Π_{nsta} is log-Lipschitz in τ with coefficient L;*
- *Let set $\mathcal{V} = \{v_i\}_{i=1}^{V}$ contain all possible values of the long-term cost of any full trajectory under any policy $\pi \in \Pi_{nsta}$, with each v_i ranked, i.e., $v_i > v_j, \forall i, j \in \{1, \cdots, V\}, i < j$. Let set $\mathcal{V}_i = \left\{\{(s_t, a_t)\}_{t=0}^{T} | \sum_{t=0}^{T} c(s_t, a_t) = v_i\right\}$ contain all trajectories that have the long-term cost v_i. Assume $|\mathcal{V}_i| \le M, \forall i \in \{1, \cdots, V\}$;*
- *$\frac{\pi_{k+1}(a|x)}{\pi_k(a|x)} \in [1-\delta, 1+\delta], \pi_k(a|x) > 0, \forall x, a$ with $\delta \in [0,1)$, and $L = L_\delta \delta$ for some $L_\delta \in \mathbb{R}_{>0}$.*

Then, if $g_c = Uniform([\xi, 1.0])$, $\mathbb{E}_{a_0 \sim \pi_{k+1}(\cdot|x_0)} \Phi_{g_c}\left[C^{\pi_{k+1}}(x_0, a_0)\right] \le d + D(\delta, \mathcal{V}, M, \xi, T)$.

Proof. Let us firstly focus on one possible value of the long-term cost, say v_i. The set of trajectories that has v_i cost is $\mathcal{V}_i$, which has the maximal cardinality M.

Let the trajectory that generates v_i be $\{(s_t, a_t)\}_{t=0}^{T}$. Assume for any policy π_k and π_{k+1}, their trajectories to generate v_i are $\{(s_t, \tau_t^k, a_t)\}_{t=0}^{T}$ and $\{(s_t, \tau_t^{k+1}, a_t)\}_{t=0}^{T}$. Note that τ_t^k may not equal τ_t^{k+1} as the policy is different.

According to the log-Lipschitz of π_{k+1}, we have that

$$\left|\log \pi_{k+1}(a_t|s_t, \tau_t^{k+1}) - \log \pi_{k+1}(a_t|s_t, \tau_t^k)\right| \le L \cdot |\tau_t^{k+1} - \tau_t^k| \le L,$$

where the second inequality uses the fact that the quantile levels are in $[0,1]$.

For policy π_k and π_{k+1}, let the probability for trajectories $\{(s_t, \tau_t^k, a_t)\}_{t=0}^T$ and $\{(s_t, \tau_t^{k+1}, a_t)\}_{t=0}^T$ be p_k and p_{k+1}. Then, we have that

$$\begin{aligned}
\log \frac{p_{k+1}}{p_k} &= \log \frac{\pi_{k+1}(a_0|s_0,1)p(s_1|s_0,a_0)\pi_{k+1}(a_1|s_1,\tau_1^{k+1})p(s_2|s_1,a_1)\cdots\pi_{k+1}(a_T|s_T,\tau_T^{k+1})}{\pi_k(a_0|s_0,1)p(s_1|s_0,a_0)\pi_k(a_1|s_1,\tau_1^k)p(s_2|s_1,a_1)\cdots\pi_k(a_T|s_T,\tau_T^k)} \\
&= \log \frac{\pi_{k+1}(a_0|s_0,1)}{\pi_k(a_0|s_0,1)} + \log \frac{\pi_{k+1}(a_1|s_1,\tau_1^{k+1})}{\pi_k(a_1|s_1,\tau_1^k)} + \cdots + \log \frac{\pi_{k+1}(a_T|s_T,\tau_T^{k+1})}{\pi_k(a_T|s_T,\tau_T^k)} \\
&= \log \frac{\pi_{k+1}(a_0|s_0,1)}{\pi_k(a_0|s_0,1)} + \log \frac{\pi_{k+1}(a_1|s_1,\tau_1^{k+1})}{\pi_{k+1}(a_1|s_1,\tau_1^k)} + \log \frac{\pi_{k+1}(a_1|s_1,\tau_1^k)}{\pi_k(a_1|s_1,\tau_1^k)} + \cdots \\
&+ \log \frac{\pi_{k+1}(a_T|s_T,\tau_T^{k+1})}{\pi_{k+1}(a_T|s_T,\tau_T^k)} + \log \frac{\pi_{k+1}(a_T|s_T,\tau_T^k)}{\pi_k(a_T|s_T,\tau_T^k)} \\
&\le T(L + \log(1+\delta)).
\end{aligned}$$

Similar arguments can be applied to the reverse side. Thus, we have that $e^{T(-L+\log(1-\delta))} \le \frac{p_{k+1}}{p_k} \le e^{T(L+\log(1+\delta))}$. Note that with $\pi_k(a|x) > 0, \forall x, a$, p_k is also positive for any trajectory. Hence, when $g_c = \text{Uniform}([\xi, 1.0])$, the maximal difference in $\mathbb{E}_{a_0\sim\pi_k(\cdot|x_0)}\Phi_{g_c}\left[C^{\pi_k}(x_0,a_0)\right]$ will be the solution value to the following problem:

$$\begin{aligned}
\max_{\boldsymbol{y},\boldsymbol{x}} & \sum_{i=1}^{V}\sum_{j=1}^{M} v_i \cdot y_{i,j} \cdot \mathbf{1}\{j \le |\mathcal{V}_i|\} \cdot \mathbf{1}\Big\{\sum_{i'=1}^{i-1}\sum_{j'=1}^{M} y_{i',j'} + \sum_{j''=1}^{j-1} y_{i,j''} \le 1-\xi\Big\} - \\
& \sum_{i=1}^{V}\sum_{j=1}^{M} v_i \cdot x_{i,j} \cdot \mathbf{1}\{j \le |\mathcal{V}_i|\} \cdot \mathbf{1}\Big\{\sum_{i'=1}^{i-1}\sum_{j'=1}^{M} x_{i',j'} + \sum_{j''=1}^{j-1} x_{i,j''} \le 1-\xi\Big\} \\
\text{s.t.} & \sum_{i=1}^{V}\sum_{j=1}^{M} y_{i,j} \cdot \mathbf{1}\{j \le |\mathcal{V}_i|\} = 1 \\
& \sum_{i=1}^{V}\sum_{j=1}^{M} x_{i,j} \cdot \mathbf{1}\{j \le |\mathcal{V}_i|\} = 1 \\
& e^{T(-L+\log(1-\delta))} \le \frac{y_{i,j}}{x_{i,j}} \le e^{T(L+\log(1+\delta))}, x_{i,j} > 0, \forall i \in \{1,\cdots,V\}, j \le |\mathcal{V}_i|.
\end{aligned} \tag{6}$$

The problem has no closed-form solution, and we denote its solution as $D(\delta, \mathcal{V}, M, \xi, T)$ to emphasize the dependence. Moreover, such difference is tight as $D(0, \mathcal{V}, M, \xi, T) = 0$ if we assume $L = \Theta(\delta)$, for example, L is linear with δ. Note that the first term in the objective is just $\mathbb{E}_{a_0\sim\pi_{k+1}(\cdot|x_0)}\Phi_{g_c}\left[C^{\pi_{k+1}}(x_0,a_0)\right]$, and the second term is $\mathbb{E}_{a_0\sim\pi_k(\cdot|x_0)}\Phi_{g_c}\left[C^{\pi_k}(x_0,a_0)\right]$. From lemma 5 and the inequality in DCPI we know that

$$\mathbb{E}_{a_0\sim\pi_{k+1}(\cdot|x_0)} F^{-1}_{C^{\pi_k}(x_0,a_0)}(\tau) \le \frac{d}{g_c(\tau)H(g_c)}, \forall \tau.$$

Then, we have that

$$\begin{aligned}
\mathbb{E}_{a_0\sim\pi_{k+1}(\cdot|x_0)}\Phi_{g_c}\left[C^{\pi_k}(x_0,a_0)\right] &= \mathbb{E}_{a_0\sim\pi_k(\cdot|x_0)}\Phi_{g_c}\Big[\frac{\pi_{k+1}(a_0|x_0)}{\pi_k(a_0|x_0)} \cdot C^{\pi_k}(x_0,a_0)\Big] \\
&\ge (1-\delta)\mathbb{E}_{a_0\sim\pi_k(\cdot|x_0)}\Phi_{g_c}\left[C^{\pi_k}(x_0,a_0)\right],
\end{aligned}$$

and $\mathbb{E}_{a_0\sim\pi_k(\cdot|x_0)}\Phi_{g_c}\left[C^{\pi_k}(x_0,a_0)\right] \le \frac{d}{1-\delta}$. Thus, we have that $\mathbb{E}_{a_0\sim\pi_{k+1}(\cdot|x_0)}\Phi_{g_c}\left[C^{\pi_{k+1}}(x_0,a_0)\right] \le \frac{d}{1-\delta} + D(\delta, \mathcal{V}, M, \xi, T)$.

□

B ENVIRONMENTS AND EXPERIMENTS

All experiments are done with NVIDIA GeForce RTX 2080 Ti GPU. 4 random seeds are used in safety gym benchmarks and the two hard constrained environments.

B.1 High Income High Risk Environment

In CVaR constrained HIHR, the CVaR level is 0.0, 0.5, 0.9, respectively. We solve HIHR with brute-force (up to 0.001 approximation error) by enumerating α. Note that for the stationary policy class Π_{sta}, its limitation lies in that it can only set one α for state s_1 and sample actions from the same bernoulli distribution every time visiting s_1 along the trajectory. In contrast, each policy in class Π_{nsta} could set different α every time visiting s_1, as it has different quantile levels every time visiting s_1.

More specifically, we ignore the edge from s_2 to s_0, and enumerate all possible long-term cost in HIHR: 8, 11, 14, 17, 21, 25. For the stationary policy class, it can only have one α value. Hence, the probability for each long-term cost is $\alpha^k \cdot (1-\alpha)$, with $k \in \{0, 1, 2, 3, 4, 5\}$. We enumerate α with interval 0.0001 for efficiency. For the quantile-level-driven policy class, it can set different α values every time visiting s_1, as agent always has different quantile levels for each visitation at s_1 along any trajectory. Hence, the probability for each long-term cost is $1-\alpha_1$, $(1-\alpha_2)\cdot\alpha_1$, $(1-\alpha_3)\cdot\alpha_1\cdot\alpha_2, \cdots$. We enumerate each $\alpha_1, \alpha_2, \cdots$ with interval 0.1 for efficiency. After that, we select the assignment that satisfies the constraint and maximizes the expected reward as the optimal solution. The results of the expected reward and solved α values are shown in Table 3 and 4, respectively.

Risk Measure	Π_{sta}	Π_{nsta}
$\xi = 0.0$	15.00	15.00
$\xi = 0.5$	12.07	14.40
$\xi = 0.9$	9.40	13.93

Table 3: Maximal expected reward of the feasible policy under different ξ in HIHR.

Risk Measure	Π_{sta}	Π_{nsta}
$\xi = 0.0$	0.73	0.80, 0.90, 0.60, 0.60, 0.10
$\xi = 0.5$	0.58	0.50, 0.90, 0.90, 0.90, 0.60
$\xi = 0.9$	0.32	0.90, 0.90, 0.10, 0.90, 0.90

Table 4: The solved α values in HIHR, with 2 decimal points reserved.

B.2 Two-way Selection Environment

B.2.1 RAC Details. In the two-way selection environment, we use linear function approximators for both policy and (forward and backward) distributional critics. Formally, we use the following policy in experiments:

$$policy(s, \tau) = w_1^T * s + w_2 * \tau,$$

where $w_1 \in \mathbb{R}^{6\times 2}$, $w_2 \in \mathbb{R}^2$. A softmax layer is followed behind $policy$ to generate the action-selecting distribution, with temperature 0.04. The reward critic, cost distributional critic and IBDC use the following linear function approximator:

$$critic(s, a, \tau) = w_1 * \tau + w_2^T * s + w_3^T * a,$$

where $w_1 \in \mathbb{R}$, $w_2 \in \mathbb{R}^6$, $w_3 \in \mathbb{R}^2$. Such linear function approximator is in fact class $\mathcal{F}_{linear}$ introduced in Sec. 4.2 for IBDC, and we make it valid for outputting the quantile levels by adding a sigmoid layer after the output with weight 0.2.

Thresholds for $\xi \in \{0.0, 0.9\}$ are both 40. The learning rate for policy and critics are all 0.01. κ in the Huber quantile regression loss is 5 for all critics. L_2 regularizations are also added for policy and critics to avoid overlarge weights.

B.3 Constrained Four Room Environment

B.3.1 Environment Details. The constrained four room environment (CFR) is an extension of the classical four room environment [13], as shown in Fig. 7. CFR consists of 11*11 grids, and each grid may have reward sign, cost sign, wall or empty. The starting grid is grid S, the goal grid is G, and grid V is labeled for visualization. The goal grid is absorbing, i.e., agent cannot get away from G once he visited there. The total horizon is 100. At each step, agent can move to one of its neighboring grids with direction *up*, *right*, *down* or *left*. The move is successful only if the next grid is within CFR and is not a wall (i.e., black grid). Agent also gets reward -1 when unsuccessful move is made. When the next grid has a solid circle sign, a solid square sign or a solid triangle sign, agent gets reward 10, 5, 1, respectively. When the next grid has a hollow stop sign, agent gets cost 1. The rewarding signs will disappear once being visited, while the stop signs will stay the same. Finally,

the negative Manhattan distance from agent's location to grid G also counts as the reward at step 100. When agent locates at grid G at step 100, he also get bonus reward 20.

In CFR, each state has 4 dimensions, and each dimension is a 11*11 matrices. The first dimension contains the unvisited rewarding signs; the second dimension contains the locations of the cost signs; the third dimension contains the locations of the walls; the final dimension contains the location of the agent. Action space of each grid is $\{up, right, down, left\}$.

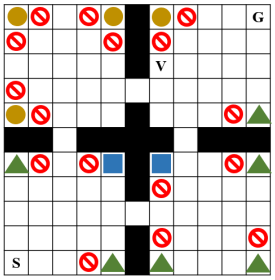

Figure 7: Illustration of the CFR environment.

B.3.2 RAC Details. In CFR, the reward critic has 2 convolutional layers and 3 linear layers with leaky relu activation function. The policy has 2 convolutional layers and 3 linear layers with leaky relu activation function, with an extra 2-layer network for the inputted quantile level. A softmax layer is added to the output of policy to generate the discrete action distribution with dimension 4. The forward cost critic has a *cos* function with 8 scales of π to generate embeddings for the sampled τ. All three networks in IBDC class $\mathcal{F}_{nn}$ use 3 linear layers with relu activation function. κ is set as 10. The clip threshold is 0.2. The learning rates for policy and critics are 0.0001 and 0.0003, respectively. The policy learning rate decays linearly as training proceeds. The Lagrangian multiplier is initialized as 0 and has learning rate 0.01. The maximal Lagrangian multiplier is 100. The target networks of forward cost critic and IBDC are updated every 2 epoch with ratio 0.1, and the policy network changes with ratio 0.4 for each update.

B.4 Safety Gym Benchmarks

B.4.1 Environment Details. We also test RAC on two safety Gym benchmarks [1]: PointGoal1 and CarGoal1. The goal of PointGoal1 is to push the cubic box into the green cylinder, while avoiding the blue unsafe regions, as shown in Fig. 8. The state space of PointGoal1 has dimension 60, and the action space has dimension 2. The goal of CarGoal1 is to control a robot with two independently-driven parallel wheels and a free-rolling rear wheel to push the cubic box and avoid unsafe regions, as shown in Fig. 9. The state space of CarGoal1 has 72 dimensions, and the action space has 2 dimensions. Both benchmark has maximal 1000 steps, and the threshold remains $d = 25$ for each $\xi \in \{0.1, 0.5, 0.9\}$.

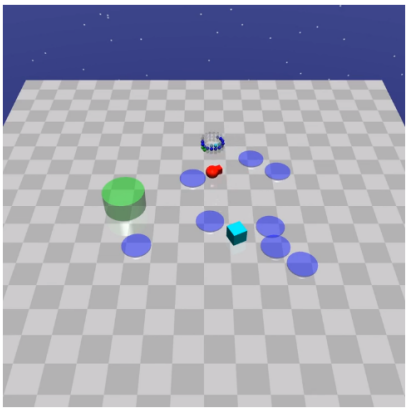

Figure 8: Illustration of the PointGoal1 environment.

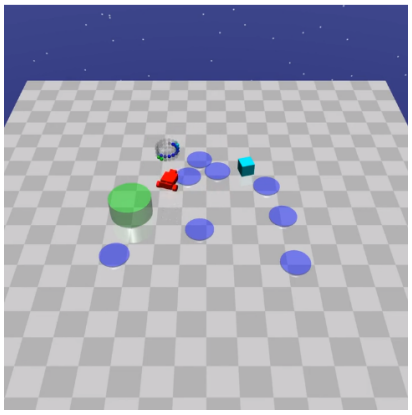

Figure 9: Illustration of the CarGoal1 environment.

B.4.2 RAC Details. In PointGoal1, the reward critic is implemented with 4 linear layers with elu activation function. The action distribution is a normal distribution, whose mean is the output of the policy network, and its std is decayed stepwise. The policy network also has 4 linear layers with elu activation function, with an extra 2-layer network for inputted quantile level. The forward cost critic follows the network architecture of IQN in [12], where a *cos* function with 8 scales of π is used for generating embeddings for the sampled τ. All three networks in IBDC class $\mathcal{F}_{nn}$ use 3 linear layers with elu activation function. κ in the Huber quantile regression loss is 1.0. The updated is conducted every 200 steps. The clip threshold is 0.01. The learning rates for policy and critics are 0.00005 and 0.0005, respectively. The Lagrangian multiplier is initialized as 0 and has learning rate 0.05. The maximal Lagrangian multiplier is 100. The target networks of forward cost critic and IBDC are updated every epoch with ratio 0.2, and the policy network changes with ratio 0.01 for each update.

In CarGoal1, the network structures are the same as those in PointGoal1. The clip threshold is 0.01. The learning rates for policy and critics are 0.00002 and 0.0001, respectively. The Lagrangian multiplier is initialized as 0 and has learning rate 0.01. The maximal Lagrangian multiplier is 100. The target networks of forward cost critic and IBDC are updated every epoch with ratio 0.2, and the policy network changes with ratio 0.1 for each update.

B.5 Hard Constrained Environments

B.5.1 Environment Details. UAV Maneuvering (UAVM) [14, 31] has 6 Degrees of Freedom (DoF), with state dimension 13 and action dimension 4. It controls a Unmanned Aerial Vehicle (UAV) to be close to the original point while avoiding the unsafety regions, as shown in Fig. 10. Each state in UAVM can be represented by vector $s = [p, v, q, w] \in \mathbb{R}^{13}$, where $p \in \mathbb{R}^3$ denotes the position, $v \in \mathbb{R}^3$ denotes the velocity. $q \in \mathbb{R}^4$ denotes the unit quaternion for attitude and $w \in \mathbb{R}^3$ denotes the angular velocity with respect to the inertial frame. The action is a 4-d vector, denoting the four rotating propellers of the quadrotor. The initial state of UAVM follows the setting of [31], which has position $(-8, -6, 9)$, velocity and angular velocity $\mathbf{0}$, and unit quaternion $(1, 0, 0, 0)$. The unsafe region is $\{(x, y, z)|(x+4.5)^2 + (y+4)^2 \leq 1, -2 \leq z \leq 5\}$. The threshold $d = 0$.

Budgeted Load Balancing (BLB) environment generates the load balancing environment proposed by [17]. In BLB, the jobs come with Poisson distribution ($\lambda = 100$), with sizes drawing from an uniform distribution $(100 - 1000)$. BLB has 10 servers with different processing rates, ranging linearly from 0.15 to 1.05. The state in BLB is a 12-d vector, where the first 10 dimensions represent the normalized job size in the queue of each server, the 11-th dimension represents the normalized job size of the current incoming job, and the final dimension represents the normalized working time. The action in BLB is choosing one server among $\{1, \cdots, 10\}$ to process the incoming job. The reward is the number of uncompleted jobs times the elapsed time since the last action. Apart from maximizing reward return, BLB also considers the budget limit for processing the jobs. Each server generates a computational expense when it begins to process a new job, which equals the service rate of the server times the elapsed time of the new job. BLB requires the computational expense of all servers during the episode to be no larger than the budget. The threshold $d = 90000$.

B.5.2 RAC Details. In UAVM, the reward critic is implemented with 4 linear layers with elu activation function. The action distribution is a normal distribution, whose mean is the output of the policy network, and its std is decayed stepwise. The policy network also has 4 linear layers with elu activation function, with an extra 2-layer network for inputted quantile level. The forward cost critic follows the network architecture of IQN in [12], where a *cos* function with 8 scales of π is used for generating embeddings for the sampled τ. All three networks in IBDC class $\mathcal{F}_{nn}$ use 3 linear layers with elu activation function. κ in the Huber quantile regression loss is 1.0. The clip threshold is 0.2. The learning rates for policy and critics are 0.00004 and 0.00004, respectively. The Lagrangian multiplier is initialized as 0 and has learning rate 0.02. The maximal Lagrangian multiplier is 100. The target networks of forward cost critic and IBDC are updated every 2 epoch with ratio 0.1 and 0.05, respectively, and the policy network changes with ratio 0.2 for each update. In UAVM, the three hyper-parameters of Saute is $n_1 = -20, n_2 = -40, n_3 = -80$.

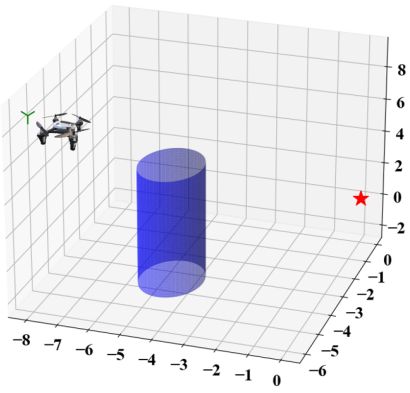

Figure 10: Illustration of the UAVM environment.

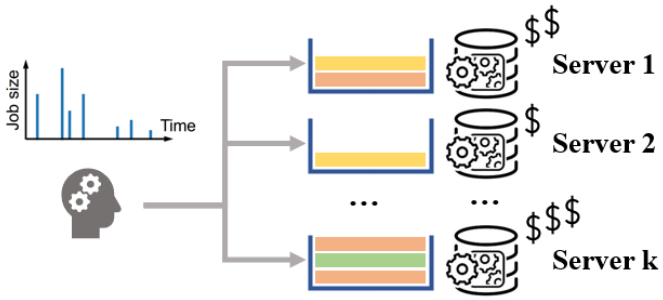

Figure 11: Illustration of the BLB environment.

In BLB, the reward critic has 4 linear layers with elu activation function. The policy has 3 linear layers with leaky relu activation function, with an extra 2-layer network for the inputted quantile level. A softmax layer is added to the output of policy to generate the action distribution. The forward cost critic has a *cos* function with 64 scales of π to generate embeddings for the sampled τ. All three networks in IBDC class $\mathcal{F}_{nn}$ use 3 linear layers with elu activation function. κ is set as 0.1. The clip threshold is 0.2. The learning rates for policy and critics are 0.00001 and 0.00001, respectively. The Lagrangian multiplier is initialized as 0 and has learning rate 0.05. The maximal Lagrangian multiplier is 100. The target networks of forward cost critic and IBDC are updated every 2 epoch with ratio 0.25, and the policy network changes with ratio 0.1 for each update. In BLB, the three hyper-parameters of Saute is $n_1 = -2.5, n_2 = -5.0, n_3 = -10.0$.