

## 432 A Algorithms

433 Algorithm 1 demonstrates the D-ATACOM, and Algorithm 2 shows the adapted SAC update used  
 434 in line 9.

---

### Algorithm 1 D-ATACOM with constraint learning

---

**Initialize:** FVF network parameters  $\phi$ , number of steps  $N$ , threshold  $\delta$ , cost budget  $\bar{C}$ .

- 1: **for**  $1 \dots N$  **do**
  - 2:   Construct  $\text{CVaR}_\alpha^F(s_t)$  using  $\mu_\theta^F(s_t)$ ,  $\Sigma_\theta^F(s_t)$  from Equation (7).
  - 3:   Draw action in safe action space  $u_t$  and obtain the actual action  $a_t$  from Equation (3).
  - 4:   Observe  $s_{t+1}, r_t, k_t$  from the environment.
  - 5:   Save replay buffer  $(s_t, a_t, r_t, k_t, s_{t+1}) \rightarrow \mathcal{D}$  and  $(s_t, k_t, s_{t+1}) \rightarrow \mathcal{D}_f$  if  $k_t > 0$ .
  - 6:   If the episode terminates, update  $\delta$  using Equation (8).
  - 7:   Sample a batch of transitions  $(s, a, r, k, s')$  from  $\mathcal{D} \cup \mathcal{D}_f$ .
  - 8:   Update  $\phi$  using Equation (5),  $\phi \leftarrow \phi - \alpha_\phi \nabla_\phi \mathcal{L}_F$
  - 9:   Update value function and policy  $\pi$  with RL, the update with SAC is shown in Algorithm 2.
  - 10: **end for**
- 

---

### Algorithm 2 SAC implementation for D-ATACOM

---

**Initialize:** Batch of transitions  $\mathcal{B} = (s, a, r, k, s')$ , policy parameters  $\theta$  and Q-function parameters  $\psi_1, \psi_2$ .

- 1: Draw next action in safe action space  $u'$  and obtain the actual next action  $a'$  and  $B'_u$  from Equation (3).
  - 2: Compute FVF adjusted log probability  $\log p' = \log \pi_\theta(a'|s') - \log |B'_u|$
  - 3: Update Q-functions with the TD loss  $\mathcal{L}_\psi = \frac{1}{|\mathcal{B}|} (Q_\psi(s, a) - r - \gamma(Q_\psi(s', a') + \alpha \log p'))^2$ .
  - 4: Draw actions  $u_\theta$  and obtain  $a_\theta$  that are differentiable w.r.t.  $\theta$ , obtain  $B_u$  from Equation (3).
  - 5: Compute FVF adjusted log probability,  $\log p_\theta = \log \pi_\theta(a|s) - \log |B_u|$
  - 6: Update policy with the gradient  $-\nabla_\theta \frac{1}{|\mathcal{B}|} (Q_\psi(s, a_\theta) + \alpha \log p_\theta)$
-

## B Experiment Environments

In this Section, we provide the full description of the environments used for the experiments. In all environments, the cost value is a continuous variable. A value greater than zero indicates how much the constraints are violated.

### B.1 Cartpole

The cartpole environment, depicted in Figure 7a is a classic control problem with the goal of moving the pole tip to a desired position (green point) by controlling a cart. The pole is one unit in length and is initialized in an upright position on the cart. The cart can move on a rail 10 units long. The cart is initialized on the left side of the rail, and the goal is to move the cart towards the goal position of the pole tip on the right rail's side while keeping the pole upright.

The state space of the environment is  $s = [x, \sin \theta, \cos \theta, \dot{x}, \dot{\theta}]^T$  where  $x$  is the position of the cart,  $\dot{x}$  is the velocity of the cart,  $\theta$  is the angle of the pole with the vertical axis, and  $\dot{\theta}$  is the angular velocity of the pole. The action space is  $a \in [-1, 1]$  where the action is the force applied to the cart.

The reward function given a goal position  $x_G$  and pole tip position  $x_T$  is defined as  $r(s) = \text{clip}(1 - \frac{\|x_G - x_T\|}{4}, 0, 1)$ . The constraint function prevents the pole from deviating more than  $\pi$  from the vertical axis. Thus we define the cost function as  $c(s) = \max(\frac{\theta}{0.5\pi} - 1, 0)$

### B.2 Navigation

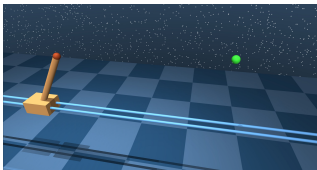
The Navigation task consists of two robots, one differential-driven TIAGo++ (white) that moves in a room while avoiding the Fetch robot (blue), as shown in Figure 7b. The Fetch robot constantly moves its robotic arm in a periodic motion, such that the end-effector draws a lemniscate into the air in front of the robot. Additionally, the Fetch robot constantly moves to a randomly assigned target position using a hand-crafted policy that ignores the TIAGo. The agent controls the TIAGo robot to reach the target position while avoiding the Fetch robot, which serves as a dynamic obstacle.

The state space consists of the cartesian position and velocity of the two robots, the target position of the TIAGo, the previous action, and the cartesian position and velocity of Fetch's end-effector. The action space is the linear velocity in the x-direction and angular velocity around the z-axis of the TIAGo robot. These are converted into the left and right wheel velocities.

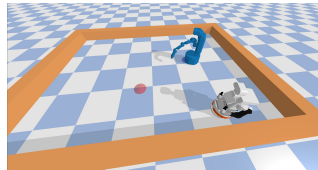
Given the distance to the goal  $d_G$ , the current orientation  $\theta$  and the goal orientation  $\theta_G$  the reward is defined as:

$$r(s) = -\|d_G\| - \text{sigmoid}(30(\|d_G\| - 0.2)) \frac{\theta_G - \theta}{\pi} - 0.1\|a\|$$

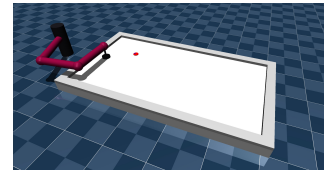
The constraint is the smallest 2d cartesian distance between the TIAGo base and every joint of the Fetch Robot. Additionally, the constraint also prevents the TIAGo from hitting the surrounding walls. Given the TIAGos' position  $p_T$  and cartesian position of the  $i$ th Fetch joint  $p_F^i$  the Fetch cost is  $c_F(s) = \max_i(-(\|p_T - p_F^i\| - \omega))$  where  $\omega$  is a constant that accounts for the width of the robots. The wall cost is defined as  $c_W(s) = \max_i(-(d_{\text{wall}}^i - \omega))$  where  $d_{\text{wall}}^i$  is the distance to the  $i$ th wall. The step cost is  $c(s) = \max(c_F(s), c_W(s))$ .



(a) Cartpole Environment



(b) Navigation Environment



(c) Air Hockey Environment

Figure 7: The three Environments used for evaluation of all algorithms

### 470 B.3 Planar Air Hockey

471 In the Planar Air Hockey environment, the agent controls a 3-DoF robot arm with a mallet attached  
 472 to the end-effector. The goal is to hit a puck into the opponent's goal, located on the opposite side of  
 473 the table, as shown in Figure 7c. The episode terminates when the puck enters the goal or hits one  
 474 of the table's walls.

475 The state space consists of the robots' joint positions, velocities, puck position, and velocity. The  
 476 action space is the acceleration setpoint for each robot joint.

477 The reward for non-absorbing states is the change of distance between the puck and the goal. In  
 478 absorbing states the reward depends on the distance of the puck to the goal. Given the puck position  
 479  $[x^t, y^t]^T$  at timestep  $t$  and the distance between puck and goal as  $d^t$ , we define the reward as:

$$r(s_t) = \begin{cases} 50(d^{t-1} - d^t) & \text{if not absorbing} \\ \rho(1.5 - 5 \cdot \text{clip}(|y^t|, 0, 0.1)) & \text{if puck in goal} \\ \rho(1 - 2 \cdot \text{clip}(|y^t| - 0.1, 0, 0.35)) & \text{if puck on backboard next to goal} \\ \rho(0.3 - 0.3 \cdot \text{clip}(1.43 - |x^t|, 0, 1)) & \text{if puck on sidebars} \\ 0 & \text{otherwise} \end{cases}$$

480 where  $\rho$  is a constant that scales the reward. The constraint prevents the mallet from touching  
 481 the sides of the table and the robot from violating its joint position and joint velocity limits. The  
 482 mallet cost is defined as  $c_M(s) = \max_i(-d_W^i + \omega)$  where  $d_W^i$  is the distance to the  $i$ th wall and  
 483  $\omega$  is a constant that accounts for the width of the mallet. Given the joint positions  $q_i$  and the joint  
 484 velocities  $\dot{q}_i$  the position cost is  $c_P(s) = \max_i(q_i - q_{u,i}, -q_i + q_{l,i})$  and the velocity cost is  $c_V(s) =$   
 485  $\max(\{\dot{q}_i - \dot{q}_{u,i}, -\dot{q}_i + \dot{q}_{l,i}\})$ . The total cost is  $c(s) = \max(c_P(s), c_V(s), c_M(s), 0)$

## C Implicit Quantile Network

IQN is a parametric model representing the quantile function of the distribution, which takes a quantile value  $\tau$  as input and outputs a threshold value  $z$  so that the probability of  $Z$  being less or equal to  $z$  is  $\tau$ . Let  $\eta_\phi^\tau(s)$  be the quantile function at  $\tau \in [0, 1]$  for the random feasibility value at state  $s$ . The TD error between two samples  $\tau, \tau' \sim U([0, 1])$  for the transition  $(s, a, s', r, k)$  is

$$d_{\phi}^{\tau, \tau'} = k'(s) + \gamma \eta^{\tau'}(s') - \eta_\phi^\tau(s)$$

The IQN model can be optimized via the Huber quantile regression loss

$$\mathcal{L}_\tau(d) = |\tau - \mathbb{I}\{d\}| \mathcal{L}_k(d), \quad \text{where } \mathcal{L}_k(d) = \begin{cases} d^2/2k, & |d| < k \\ |d| - k/2, & \text{otherwise} \end{cases} \quad (9)$$

In Figure 8 we compare the Gaussian and IQN approaches for the navigation task. In this experiment, both algorithms use the same hyperparameters. The Gaussian approach slightly outperforms IQN in terms of performance and safety. We theorize that the source of the performance difference is the hyperparameters, which are tuned for the Gaussian assumption. The main difference in the constraint estimation is that the Gaussian approach predicts higher uncertainty leading to higher performance and safety in this environment. To achieve the same similar with IQN, the cost budget or accepted risk has to be decreased. We plan to further investigate the performance of IQN-ATACOM in future work, especially in environments providing only sparse cost feedback.

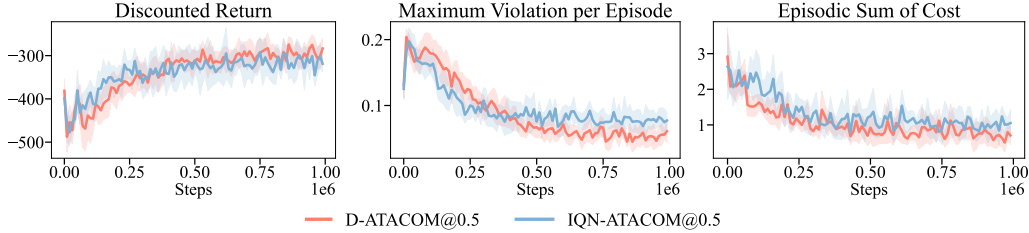


Figure 8: Comparison between the Gaussian distribution assumption and the direct CDF estimation for the navigation task. Both experiments use the same hyperparameters.

## D Hyperparameter tuning

In this section, we report the parameter tuning for all the baselines in all tasks. In general, we test all the methods with different learning rates, cost budgets and safety parameters to ensure the performance of the baseline is optimal. We report all the hyperparameter configurations we tried and indicate which configuration is used for the main evaluation.

Every algorithm is first evaluated with the learning rates of  $1e^{-4}$ ,  $5e^{-4}$  and  $1e^{-3}$ . To keep the computation reasonable, we use the same learning rate for the actor, the critic, the constraints, and the learning rates for the Lagrangian multiplier that are updated every step. We report the results of these experiments for each task in the following sections.

As a second step, we experimented with different cost budgets to get the best trade-off between safety and performance. Our goal is to get the least constraint violations possible while maintaining reasonable behavior. As we show in Section E.1, setting the cost budget too low can have an impact on the performance with no safety benefit.

Lastly, we tuned the cost-dampening parameters of LagSAC and WCSAC using the same principle we used for the cost budget.

### D.1 CartPole

Figure 9 shows the results of the learning rate tuning for the CartPole task. We can see that RCPO and LagSAC have a learning rate that achieves the best performance. For PPOLag and WCSAC, the differences are more nuanced. Table 1 shows all the parameters we tried for the Cartpole task. The resulting best parameters used for the main evaluation can be found in Table 2.

	RCPO	PPOLag	LagSAC	WCSAC	D-ATACOM
<b>Sweeping parameter</b>					
learning rate actor/critic/constraint			{1e <sup>-3</sup> , 5e <sup>-4</sup> , 1e <sup>-4</sup> }		
cost budget	5	5	{0.1, 5, 25, 40}		
cost dampening	-	-	{1, 10}		
learning rate lagrangien multipliers	0.035	0.035	{1e <sup>-4</sup> , 5e <sup>-4</sup> , 1e <sup>-4</sup> }		
accepted risk	-	-	-	{0.1, 0.5, 0.9}	
<b>Default parameter</b>					
epochs	100	100	100	100	100
steps per epoch	20000	20000	10000	10000	10000
steps per fit	20000	20000	1	1	1
episodes per test	-	-	25	25	25
network size			[128 128]		
batch size	128	64	64	64	64
initial replay size	-	-	2000	2000	2000
max replay size	200000	200000	200000	200000	200000
soft update coefficient	-	-	1e <sup>-3</sup>	1e <sup>-3</sup>	1e <sup>-3</sup>
warm-up transitions	-	-	2000	2000	2000
target kl	0.01	0.02	-	-	-
update iterations	10	40	-	-	-

Table 1: Training Parameters for the CartPole task

	RCPO	PPOLag	LagSAC	WCSAC	D-ATACOM
<b>Sweeping parameter</b>					
learning rate actor/critic/constraint	$5e^{-4}$	$1e^{-4}$	$5e^{-4}$	$5e^{-4}$	$5e^{-4}$
cost budget	5	5	5	5	40
cost dampening	-	-	1	1	-
learning rate lagrangian multipliers	0.035	0.035	$5e^{-4}$	$5e^{-4}$	$5e^{-4}$
accepted risk	-	-	-	0.9	0.9

Table 2: Result of hyperparameter tuning for the CartPole task

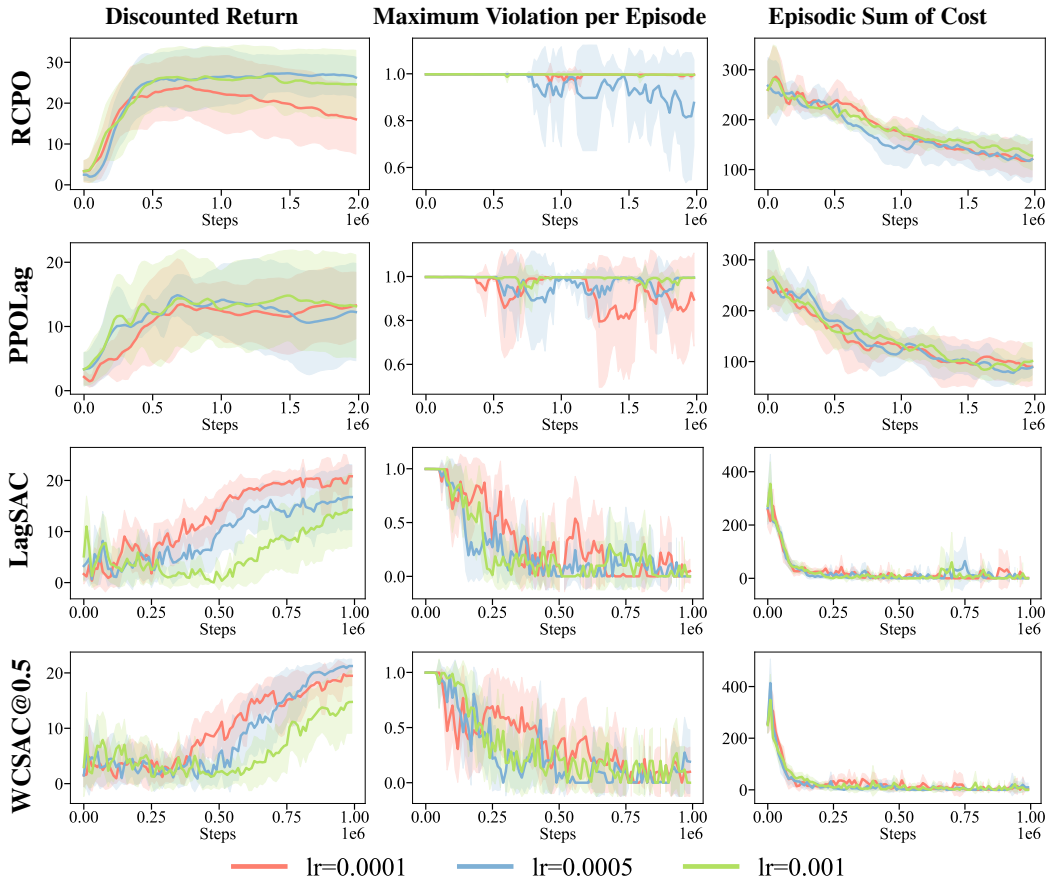


Figure 9: Learning rate ablation study for the Cartpole task. For each experiment we run 10 seeds with all learning rates of the algorithm set to the respective value.

## 520 D.2 Navigation

521 Figure 10 shows the results of the learning rate tuning for the navigation task. We can see WCSAC  
 522 is the only algorithm where the learning rate has a significant impact on the performance. Table 3  
 523 shows all the parameters we tested for the navigation task. The resulting best parameters used for  
 524 the main evaluation can be found in Table 4.

	RCPO	PPOLag	LagSAC	WCSAC	D-ATACOM
<b>Sweeping parameter</b>					
learning rate actor/critic/constraint			{1e <sup>-3</sup> , 5e <sup>-4</sup> , 1e <sup>-4</sup> }		
cost budget	0	0		{0, 1}	
cost dampening	-	-	{1, 10}		-
learning rate lagrangian multipliers	0.035	0.035	{1e <sup>-4</sup> , 5e <sup>-4</sup> , 1e <sup>-4</sup> }		
accepted risk	-	-	-	{0.1, 0.5, 0.9}	
<b>Default parameter</b>					
epochs	100	100	100	100	100
steps per epoch	20000	20000	10000	10000	10000
steps per fit	20000	20000	1	1	1
episodes per test	-	-	25	25	25
network size			[128 128]		
batch size	128	64	64	64	64
initial replay size	-	-	2000	2000	2000
max replay size	200000	200000	200000	200000	200000
soft update coefficient	-	-	1e <sup>-3</sup>	1e <sup>-3</sup>	1e <sup>-3</sup>
warm-up transitions	-	-	2000	2000	2000
target kl	0.01	0.02	-	-	-
update iterations	10	40	-	-	-

Table 3: Training Parameters for the navigation task

	RCPO	PPOLag	LagSAC	WCSAC	D-ATACOM
<b>Sweeping parameter</b>					
learning rate actor/critic/constraint	$1e^{-4}$	$1e^{-4}$	$1e^{-4}$	$1e^{-4}$	$1e^{-4}$
cost budget	0	0	0	0	0
cost dampening	-	-	1	1	-
learning rate lagrangian multipliers	0.035	0.035	$1e^{-4}$	$1e^{-4}$	$1e^{-4}$
accepted risk	-	-	-	0.9	0.5

Table 4: Result of hyperparameter tuning for the navigation task

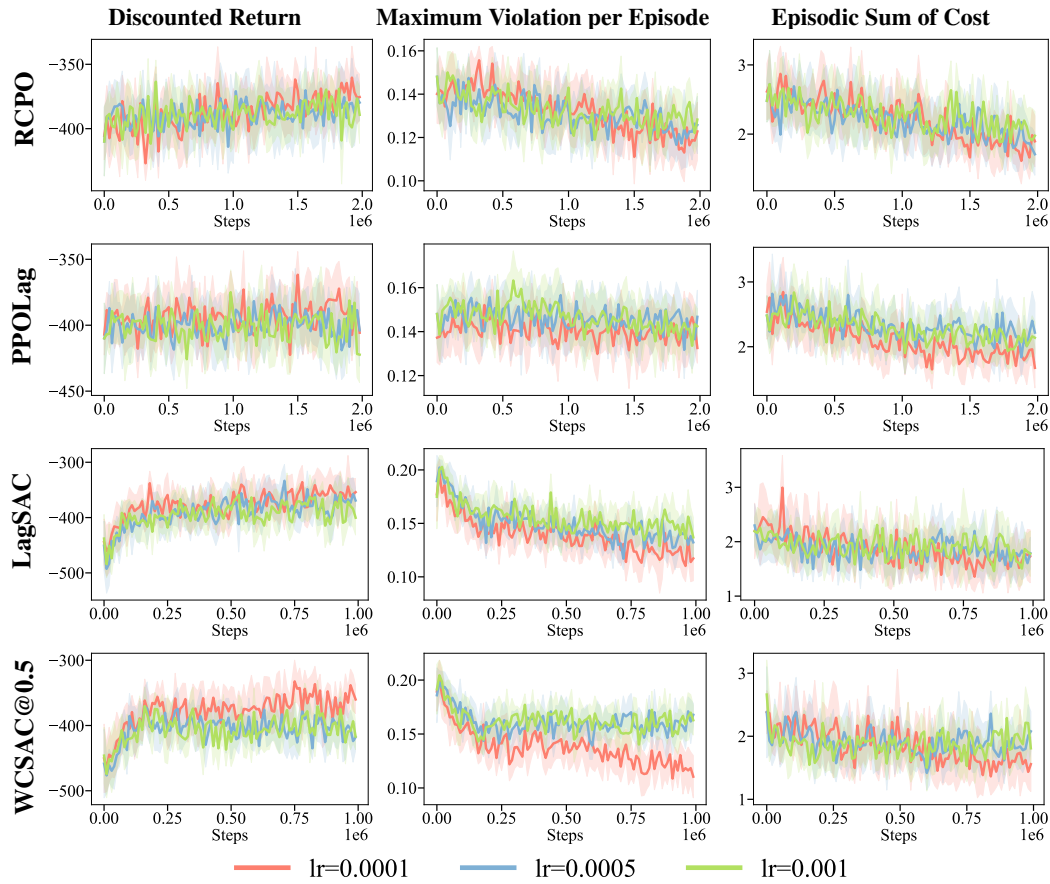


Figure 10: Learning rate ablation study for the Navigation task. For each experiment, we run 10 seeds with all learning rates of the algorithm set to the respective value.



### 525 D.3 Air Hockey

526 Figure 11 shows the results of the learning rate tuning for the air hockey task. We can see that RCPO  
 527 and PPOLag learn safer behaviors compared to LagSAC and WCSAC. However, their discounted  
 528 return is lower, and they need twice as many steps. Table 5 shows all the parameters we tested for  
 529 the air hockey task. The resulting parameters used for the main evaluation can be found in Table 6.

	RCPO	PPOLag	LagSAC	WCSAC	D-ATACOM
<b>Sweeping parameter</b>					
learning rate actor/critic/constraint			{1e <sup>-3</sup> , 5e <sup>-4</sup> , 1e <sup>-4</sup> }		
cost budget	0	0		{0, 1}	
cost dampening	-	-	{1, 10}		-
learning rate lagrangian multipliers	0.035	0.035	{1e <sup>-4</sup> , 5e <sup>-4</sup> , 1e <sup>-4</sup> }		
accepted risk	-	-	-	{0.1, 0.5, 0.9}	
<b>Default parameter</b>					
epochs	100	100	100	100	100
steps per epoch	20000	20000	10000	10000	10000
steps per fit	20000	20000	1	1	1
episodes per test	-	-	25	25	25
network size			[128 128]		
batch size	128	64	64	64	64
initial replay size	-	-	2000	2000	2000
max replay size	200000	200000	200000	200000	200000
soft update coefficient	-	-	1e <sup>-3</sup>	1e <sup>-3</sup>	1e <sup>-3</sup>
warm-up transitions	-	-	2000	2000	2000
target kl	0.01	0.02	-	-	-
update iterations	10	40	-	-	-

Table 5: Training Parameters for the air hockey task

	RCPO	PPOLag	LagSAC	WCSAC	D-ATACOM
<b>Sweeping parameter</b>					
learning rate actor/critic/constraint	$5e^{-4}$	$1e^{-3}$	$5e^{-4}$	$5e^{-4}$	$5e^{-4}$
cost budget	0	0	0	0	1
cost dampening	-	-	1	1	-
learning rate lagrangian multipliers	0.035	0.035	$5e^{-4}$	$5e^{-4}$	$5e^{-4}$
accepted risk	-	-	-	0.9	0.9

Table 6: Result of hyperparameter tuning for the air hockey task

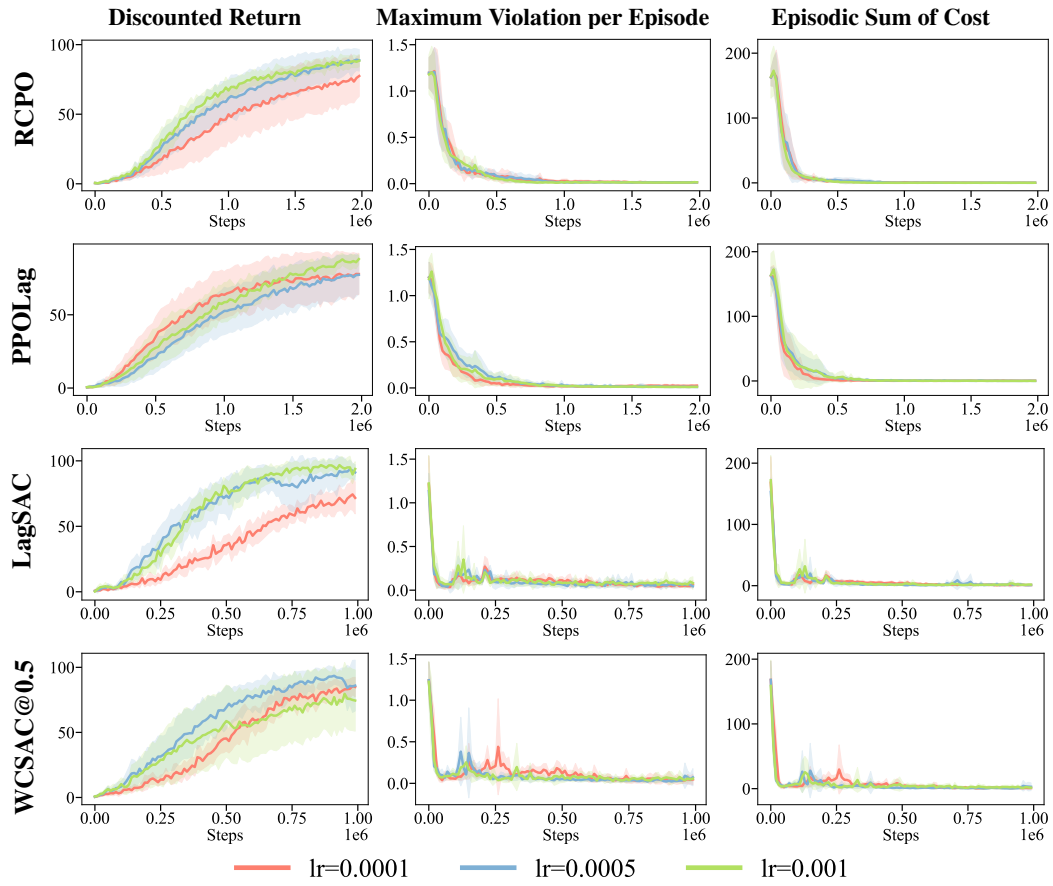


Figure 11: Learning rate ablation study for the Air Hockey task. For each experiment, we run 10 seeds with all learning rates of the algorithm set to the respective value.

## E Additional Experiments

### E.1 CartPole with different Cost Budget

In this experiment, we will compare the impact of the cost budget parameter on D-ATACOM and WCSAC. We chose the CartPole task for this comparison because both algorithms do not learn a completely safe policy. Figure 12 shows the performance of D-ATACOM and WCSAC with different cost budgets. We can observe that the performance of D-ATACOM is more sensitive to the cost budget parameter compared to WCSAC. When the policy cannot achieve the given cost budget the performance of D-ATACOM degrades significantly. This performance drop occurs because the delta eventually will converge towards zero, which results in a very conservative policy. The behavior for D-ATACOM with the cost budgets of 0.1 and 5 is balancing to the pole in its initial position because the policy is too conservative to move towards the goal, as this will lead to constraint violations.

On the other hand, WCSAC is more robust w.r.t. the cost budget parameter. An unreasonable cost budget will increase the Lagrange multiplier, giving more weight to the constraint. The difference is that the Lagrange multiplier does not set an explicit limit to the constraint like the delta does in D-ATACOM. Instead, WCSAC gives more weight to the constraint violations in the optimization problem, which has less impact on policy performance. It is worth noting that, depending on the application, one of the two behaviors would be preferable. In safety-critical applications, having an algorithm that strongly enforces the constraint violation, independently of the performance, is preferable. Instead, when partial constraint satisfaction is enough, it may be better to choose a lagrangian-based algorithm.

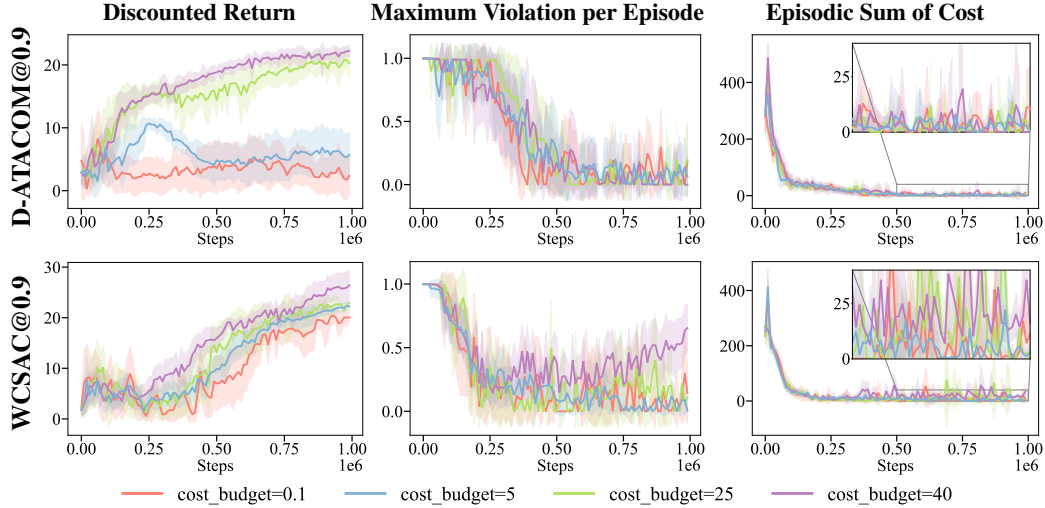


Figure 12: Impact of the cost budget parameter on D-ATACOM and WCSAC performance in the CartPole task

## 550 E.2 Experiment with different Accepted Risk

551 In the distributional setting, the parameter accepted risk determines how much of the tail of the  
 552 distribution we are willing to violate, i.e., how much risk we want to take. However, this is not the  
 553 only parameter that influences the safety of a policy. Usually, there is another parameter that is tuned  
 554 with a given cost budget that also influences how safe the behavior is. For WCSAC this parameter is  
 555 the Lagrange multiplier  $\beta$ , and for D-ATACOM it is the learned  $\delta$ . To show the complete impact  
 556 of the accepted risk, we fix  $\delta$  to a constant value such that it cannot compensate for the difference in  
 557 the accepted risk. Figure 13 shows the performance of D-ATACOM with a fixed delta and different  
 558 levels of accepted risk in the Navigation task. Clearly, a lower accepted risk leads to safer behavior.

559 The impact of the accepted risk on the safety shrinks for D-ATACOM when the delta is learned.  
 560 Delta can compensate for a high accepted risk, resulting in the same safe policy as a lower accepted  
 561 risk would produce. The accepted risk has an impact in this setting toward the beginning of the  
 562 training when delta is not yet converged. Thus accepted risk determines how risky the exploration  
 563 at the beginning of the training will be. Figure 14 shows the impact of different accepted risk  
 564 settings on the air hockey task. The lower accepted risk explores slower, thus the discounted return  
 565 converges slower. The maximum violation and sum of cost are comparable for all accepted risk  
 566 settings because, in the air hockey task, the constraint does not majorly affect the optimal policy.

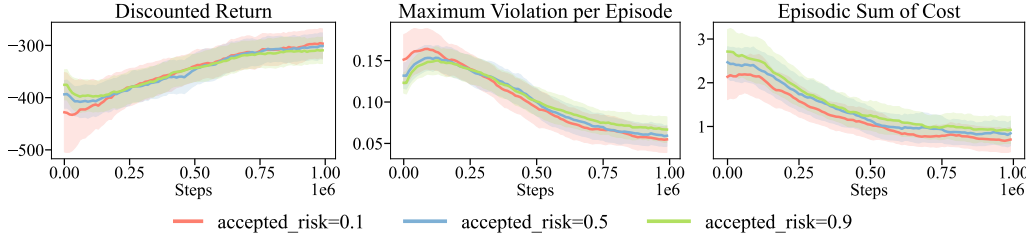


Figure 13: Impact of accepted risk on performance in the Navigation task with a fixed delta. The plots are smoothed via the exponential moving average with 0.9 weight

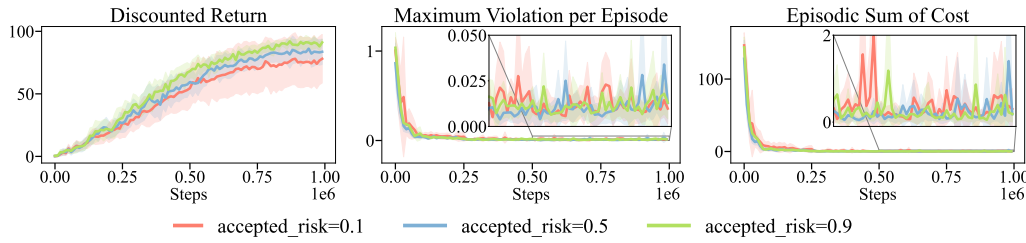


Figure 14: Impact of accepted risk on performance the air hockey task

### 567 E.3 Analysis of Air Hockey

568 In the air hockey task, D-ATACOM cannot reach the same discounted return as LagSAC and WC-  
 569 SAC. We investigate the final performance of the policies to understand the differences that lead to  
 570 the performance gap. As D-ATACOM results in a safer policy, we theorize that performance is lost  
 571 when the puck is initialized too close to the edge of the table. To test this hypothesis, we evaluate  
 572 the performance of the final policies with an adjusted region for the initial puck position, that omits  
 573 these critical positions. Figure 15 shows the performance for the original and adjusted regions and  
 574 the difference between them.

575 For the original region D-ATACOM has significant outliers in the discounted return compared to  
 576 WCSAC and LagSAC. However, LagSAC and WCSAC have more outliers in the maximum viola-  
 577 tion and sum of cost. Thus, WCSAC and LagSAC sacrifice safety to gain a stable performance.  
 578 The safe exploration of D-ATACOM results in the opposite behavior, where the policy will sacrifice  
 579 performance to ensure safety.

580 When we evaluate the performance with the adjusted region, we can observe that the discounted  
 581 return of D-ATACOM increases more compared to WCSAC and LagSAC. Additionally, the decrease  
 582 in maximum violation and sum of cost is more significant for LagSAC and WCSAC. This result  
 583 confirms our hypothesis that D-ATACOM does not properly hit the puck when it is too close to the  
 584 edge of the table because it is not possible to do so safely. WCSAC and LagSAC learn to hit the  
 585 puck in these critical positions, but this comes at the cost of safety.

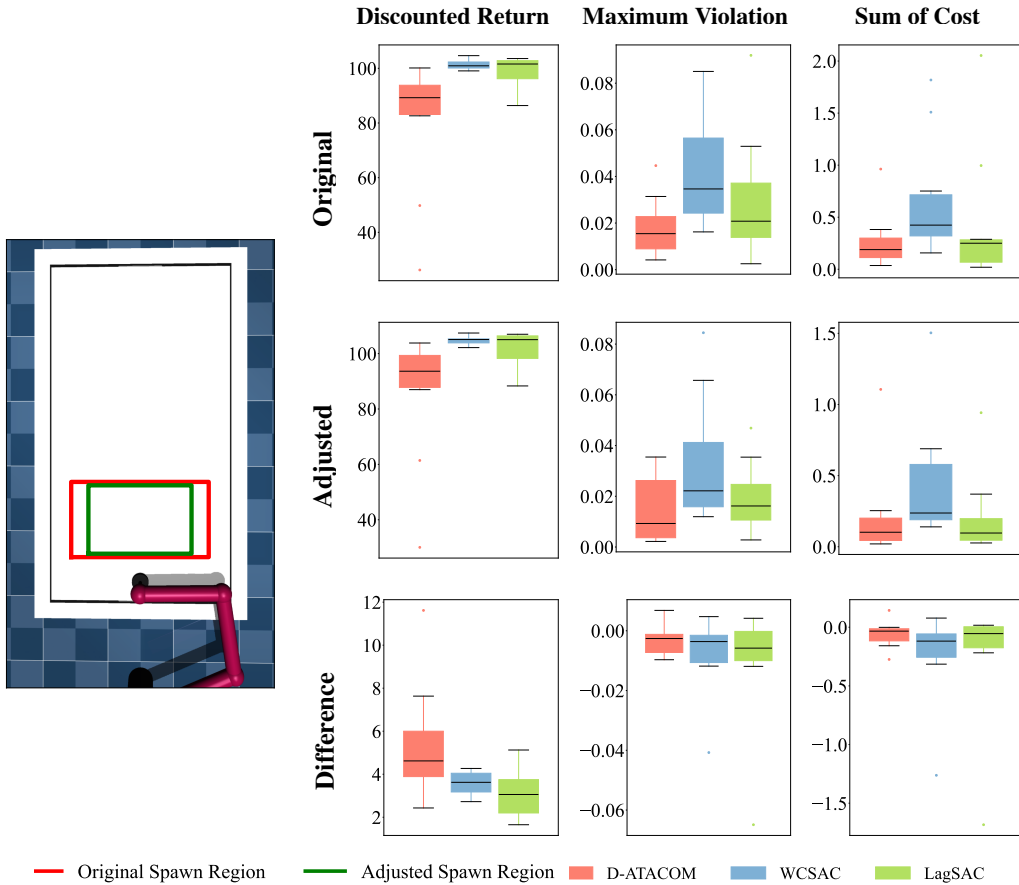


Figure 15: Performance of the final policy from D-ATACOM, WCSAC, and LagSAC in the air hockey task. The performance is evaluated with the original and an adjusted region for the initial puck position.