This appendix is structured as follows: we introduce the block diagram of the ATACOM in Appendix A. We describe the experiment environment in Appendix B. Then, we show the comparison of ATACOM, *ErrorCorrection*, *Terminated* approach with different RL method in the *CircularMotion* task in Appendix C.1. The comparison of ATACOM with different algorithms in *PlanarDefend* and *IiwaAirHockey* are shown in Appendix C.2 and Appendix C.3, respectively. The the parameters of the environments, the learning algorithms, and the results of the parameter sweep are shown in Appendix D. Finally, we append an extension of ATACOM including the uncontrollable state in Appendix E. The results of the *CollisionAvoidance* environment are illustrated in Appendix E.1.

A Block Diagram of ATACOM

Here we will describe the controlling diagram of the ATACOM using viability constraints. We copy the overall control acceleration in Equation (11) here

$$\begin{bmatrix} \ddot{\boldsymbol{q}} \\ \dot{\boldsymbol{\mu}} \end{bmatrix} = \underbrace{-\boldsymbol{J}_c^{\dagger}(\boldsymbol{q},\boldsymbol{\mu})\left[\boldsymbol{K}_c \boldsymbol{c}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{\mu}) + \boldsymbol{\psi}(\boldsymbol{q},\dot{\boldsymbol{q}})\right]}_{[\ddot{\boldsymbol{q}}_{\text{mm}} \quad \dot{\boldsymbol{\mu}}_{\text{mm}}]^{\mathsf{T}}} + \underbrace{\boldsymbol{N}_c(\boldsymbol{q},\boldsymbol{\mu})\boldsymbol{\alpha}}_{[\ddot{\boldsymbol{q}}_{\text{null}} \quad \dot{\boldsymbol{\mu}}_{\text{null}}]^{\mathsf{T}}}.$$

The first term on the RHS tries to maintain the state on the constraint manifold. The second term tries to explore the tangent space of the constraint manifold based on the agent policy.

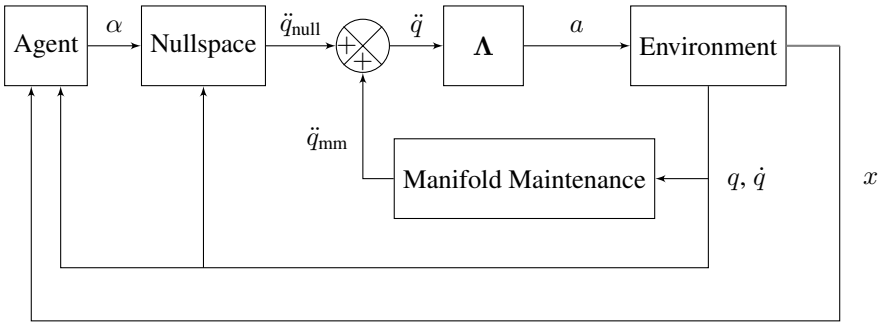


Figure 11: Block Diagram of the proposed method

As shown in Figure 11, the control action is determined by the inverted dynamics $\boldsymbol{a} = \boldsymbol{\Lambda}(\ddot{\boldsymbol{q}})$. In practice, we can apply the integration method to get desired positions and velocities as the input to an accurate tracking controller (e.g., PID controller) to determine the control action. However, this tracking error could potentially bring additional issues, like the interplay between the error in the constraints and tracking performance. In the experiment of this paper, we assume the model of dynamics is perfectly known to exclude the errors caused by the factors that are not considered in this paper.

B Environment Description

B.1 *CircularMotion*

In this environment, shown in Figure 6a, a point is moving on a circle with the following constraints

$$\begin{aligned} f : x^2 + y^2 - 1 = 0, \qquad & g : -y - 0.5 < 0, \\ |\dot{x}| - 1 < 0, \qquad & |\dot{y}| - 1 < 0. \end{aligned}$$

The first constraint ensures that the point moves on the circle. The second one ensures that the point is moving above the height -0.5. The last two constraints limit the velocity of each component. The control action is the acceleration along each axis. The objective is to minimize the distance to the goal $(1, 0)$, shown as the green square

$$r(x, y) = \exp\left(-\sqrt{(x-1)^2 + (y-0)^2}\right)$$

B.2 *PlanarAirHockey*

In this experiment, we solve the air hockey task with a 3-joints planar robot, as illustrated in Figure 6b. In this environment, we consider only inequality constraints, i.e., the robot end-effector should stay inside the table's region, and the joint positions and velocities should not exceed its limits. The constraints are the following

$$\begin{aligned} g_1 : -x_{ee} + x_{table,l} < 0, \qquad & g_2 : -y_{ee} + y_{table,l} < 0, \qquad & g_3 : y_{ee} - y_{table,u} < 0, \\ g_{4,5,6} : q_i^2 - q_{i,l}^2 < 0 \qquad & |\dot{q}_i| - \dot{q}_{i,l} < 0 \qquad & \forall i \in \{1,2,3\} \end{aligned}$$

where (x_{ee}, y_{ee}) is the position of the robot end-effector, $x_{table,l}, y_{table,l}, y_{table,u}$ are the boundaries of the air-hockey table, $q_i, \dot{q}_i$ refers to position and velocity of the i-th joint, and $q_{i,l}, \dot{q}_{i,l}$ are the position and velocity limit for the joint i. The control action is the torque applied on each joint. In this task, the forward kinematics, Jacobian matrix, Hessian matrix, and the inverse dynamics are calculate with the help of the *Pinocchio* rigid bodies dynamics library.

We initialize the puck and the robot in a given configuration. The objective is to hit the puck to the opponent's goal as fast as possible. We define the reward as

$$r = \begin{cases} \exp\left[-8 \cdot \|\boldsymbol{p}_{ee} - \boldsymbol{p}_{puck}\| \cdot \text{clip}(\cos\theta, 0, 1)\right] - \lambda, & \text{if has not hit,} \\ 1 + r_{hit} + 0.1 v_{x,hit} - \lambda, & \text{if has hit,} \\ 150, & \text{if goal is scored.} \end{cases} \tag{12}$$

with $\cos\theta = \langle \frac{\boldsymbol{p}_{puck} - \boldsymbol{p}_{ee}}{\|\boldsymbol{p}_{puck} - \boldsymbol{p}_{ee}\|}, \frac{\boldsymbol{p}_{goal} - \boldsymbol{p}_{puck}}{\|\boldsymbol{p}_{goal} - \boldsymbol{p}_{puck}\|} \rangle$, $\boldsymbol{p}_{ee}, \boldsymbol{p}_{puck}, \boldsymbol{p}_{goal}$ is the 2D position of the end-effector, the puck and the goal, r_{hit} is the reward when hitting occurs, $v_{x,hit}$ is the velocity of the puck along x-direction (longitude direction of the table) at the hitting moment, $\lambda = 0.001 \cdot \|\boldsymbol{a}\|$ is the penalty of the action. If the robot has not hit the puck, the reward encourages to get the end-effector close to the puck along the direction from the puck to the goal. If the hitting occurs, the reward is only influenced by the last reward before hitting and the velocity along the longitude direction, as further actions will not affect the final behavior. When the agent scores a goal, we provide a bonus reward and terminate the episode.

B.3 *IiwaAirHockey*

In this environment, we learn the robot air hockey task with a 7 DoF KUKA IIWA. To increase the reachability of the robot arm, we design a new end-effector. The end-effector is composed of an extension rod, a gas spring, a universal joint, and a mallet. The total length is 0.5m. To ensure that the mallet stays perpendicular to the table surface, we add a separate controller on the 7th joint of the IIWA robot, which forces the axis of the universal joint to be parallel to the table surface. We disabled the collision of the robot with the table to verify that the constraint is actively guaranteed. Therefore, the universal joint is not able to adapt its joint positions passively. We add a position controller in the simulator to ensure the mallet makes proper contact with the table. The control action is a six-dimensional vector representing the torque applied at each joint. As the real-world's KUKA controller enforces the joint velocity limits by default, we force the joint velocity constraints in the simulation.

In addition to the inequality constraints described in the *PlanarAirHockey*, we add an equality constraint to ensure the end-effector is moving on the table surface, two inequality constraints on the 4th link and the 6th link in order to prevent the collision. The final constraint set is

$$\begin{aligned} f : \quad & z_{ee} - z_{table} = 0, \quad & g_1 : \quad & -z_4 + \hat{z}_{4,l} < 0, \quad & g_2 : & -z_6 + \hat{z}_{6,l} < 0, \\ g_3 : \quad & -x_{ee} + x_{table,l} < 0, \quad & g_4 : \quad & -y_{ee} + y_{table,l} < 0, \quad & g_5 : & y_{ee} - y_{table,u} < 0, \\ g_{6,7,\cdots 11} : \quad & q_i^2 - q_{i,l}^2 < 0, & & \forall i \in \{1, 2, \cdots 6\} & & \end{aligned}$$

where f ensures the end-effector's height z_{ee} is the same as table's height z_{table}. g_1, g_2 constraint the height of the 4-th link z_4 and the 6-th link z_6 above its limit $\hat{z}_{4,l}$ and $\hat{z}_{6,l}$, respectively. g_3, g_4, g_5 ensure the end-effector is moving inside the table's range. $g_{6,7,\cdots,11}$ are the joint position limits constraints. In the *IiwaHit* task, the reward is same as *PlanarHit* specified in (12).

C Additional Experiments

C.1 *CircularMotion*

In this section, we compare ATACOM with the *ErrorCorrection* and the *Terminated* in the *CircularMotion* environment. The *ErrorCorrection* approach adds the acceleration calculated by (10) at each time step and the *Terminated* approach terminate the episode when the constraint violations bigger than a threshold. The details of the experiment parameter can be found in Appendix D.

Figure 12 illustrated comparison of different approaches in each RL algorithms. We select the best performed parameter for each algorithm after the parameter sweep. Here we plot the total return instead of discounted return, as the difference will be more evident. It is clear that the ATACOM has better performance on each algorithms and less constraint violations.

C.2 *PlanarDefend*

In this section, we demonstrate the the task *PlanarDefend*. In this task, the puck and the robot is initialized at a certain position and with the same initial velocity. The constraints and environment parameters are same as the hitting task. The ultimate objective is to stop the puck's at the line $x = -0.6$. The reward is designed as following

$$r = \begin{cases} \exp\left(-3\|\boldsymbol{p}_{des} - \boldsymbol{p}_{ee}\|\right), & \text{if no short sides of the table has been hit and the puck has not been hit,} \\ 1 + \exp\left(-5|x_{puck} + 0.6|\right) + 5\exp\left(-5\|\boldsymbol{v}_{puck}\|\right), & \text{if no short sides of the table has been hit and the puck has been hit,} \\ 0, & \text{if any short sides of the table has been hit,} \\ -50, & \text{if the puck is in the defender goal.} \end{cases}$$

Figure 13 shows the comparison of the five RL algorithms. In this task, the policy is very hard to obtain. Small changes of the end-effector's position will result in complete different movements on the puck. The constraint violations remains small as expected.

C.3 *IiwaAirHockey*

In this subsection, we test the *IiwaAirHockey* task with five RL algorithms (DDPG, TD3, SAC, TRPO, PPO). In Figure 14 we present the results of the best parameter of each algorithms after the parameter sweep. These results show that all algorithms are able to improve the policy while maintaining the constraint violation low.

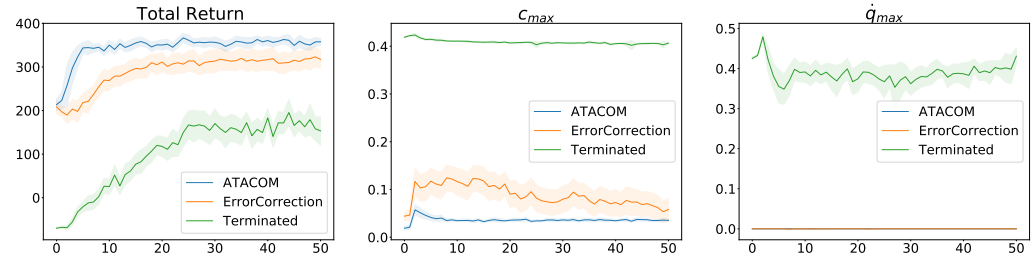
C.4 Comparison with Riemannian Motion Policies

In the following section, we compare our method with Riemannian Motion Policies (RMP) [37] in the *IiwasAirHockey* environment. We follow the dynamics of the collision avoidance as described in [26]. The collision avoidance policy is used to avoid the collision between the End-Effector and the table's boundary, the wrist link and the table, as well as the elbow link and the table. In addition, we also apply the joint limits avoidance policy described in [37]. Unfortunately, RMP does not consider equality constraints. Therefore, to maintain the end-effector on the table surface, we deploy a simple PD-controller to maintain the height of the end-effector z_{ee} at the table's height z_{table} as:

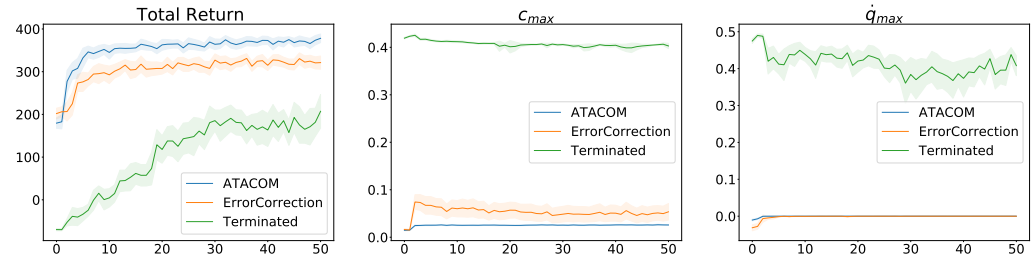
$$\ddot{z}_{ee} = -P(z_{ee} - z_{table}) - D\dot{z}_{ee}$$

with P, D the controller gain. Finally, we add a learning policy on top of the composed RMP policy. The action space of the learning agent is the joint acceleration in each dimension.

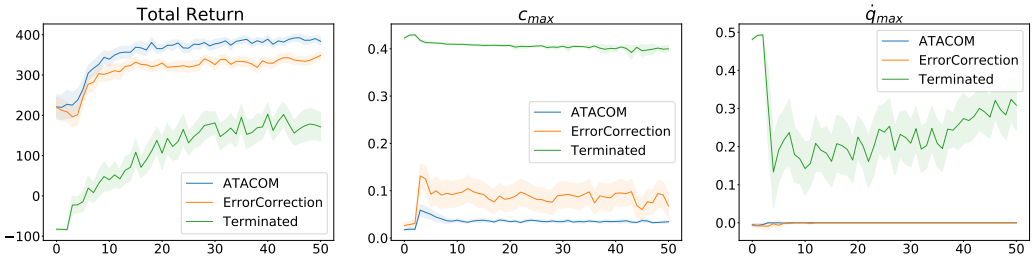
Figure 15 shows the comparison between RMP and ATACOM. ATACOM can learn faster since it explores only on the constraint manifold, instead of on the whole joint space as in RMP. ATACOM keeps the constraint violations small during the whole learning process, differently from the RMP approach. These higher violations are due to the fact that the potential fields of RMP model soft constraints and not hard ones. While the final performance of RMP is slightly better than the one achieved by ATACOM, the learned trajectories violate the constraints, allowing for faster movements that are not possible on real robots.



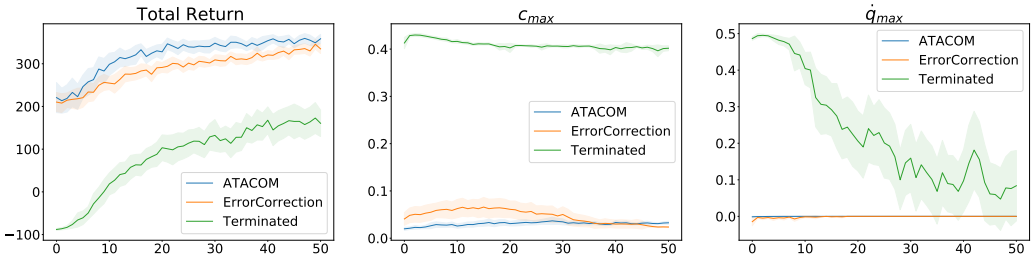
(a) DDPG



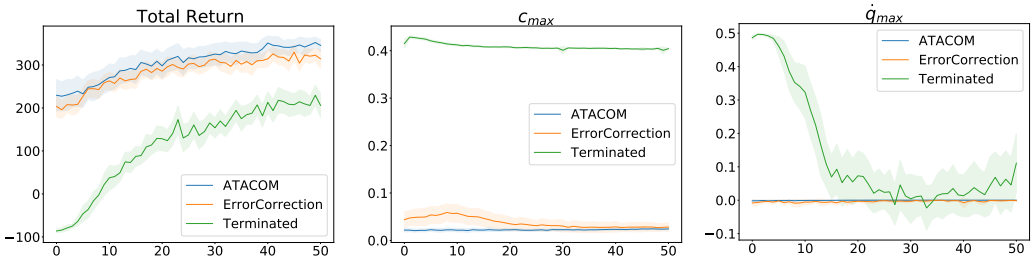
(b) TD3



(c) SAC



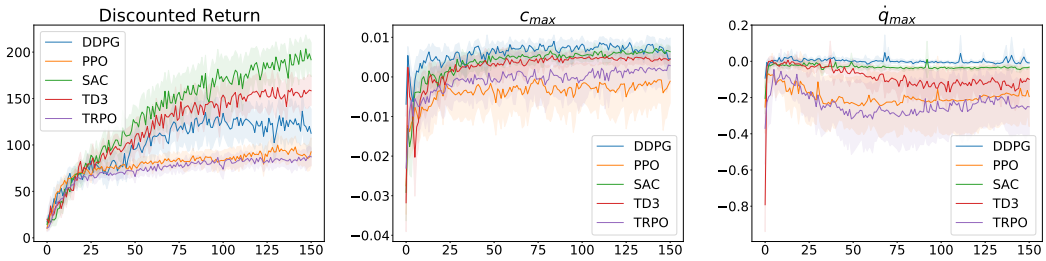
(d) TRPO



Total Return | Max Constraint Violations | Max Velocity Limits Violations

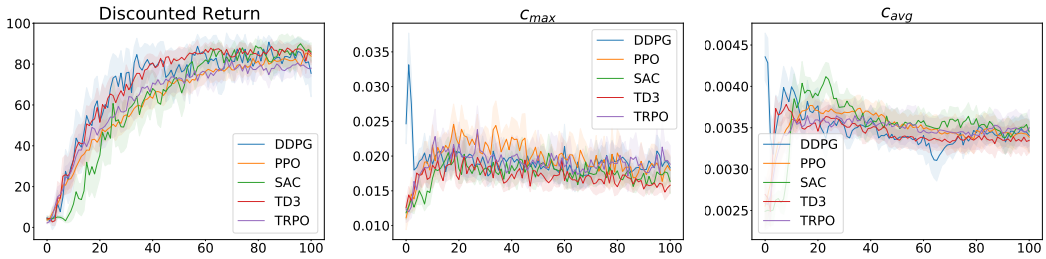
(e) PPO

Figure 12: Comparison of ATACOM, *ErrorCorrection* and *Terminated* in *CicularMotion*.



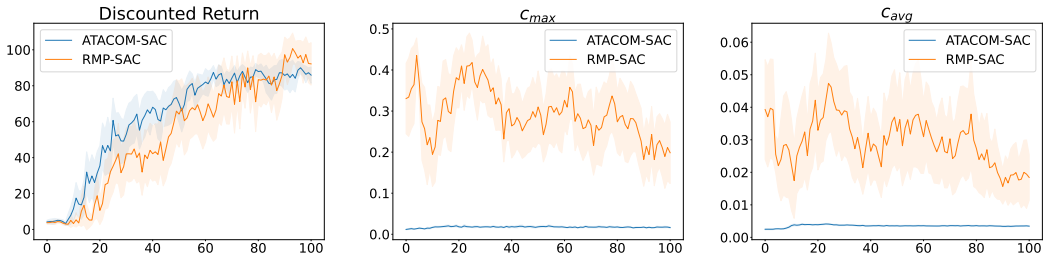
(a) Discounted Reward (b) Max Constraint Violations (c) Max Velocity Limits Violations

Figure 13: Comparison of RL algorithms in *PlanarDefend*.



(a) Discounted Reward (b) Max Constraint Violations (c) Average Constraint Violations

Figure 14: Comparison of RL algorithms in *IiwaAirHockey*.



(a) Discounted Reward (b) Max Constraint Violations (c) Average Constraint Violations

Figure 15: Comparison of RMP with ATACOM in *IiwaAirHockey*.

D Parameters Sweep and Results

In this section, we listed all the parameters and the results of the parameter sweep of each experiment. We launched 25 independent runs for each sweeping parameter.

D.1 *CircularMotion*

We compare ATACOM with the *ErrorCorrection* and the *Terminated* approaches. The *ErrorCorrection* approach applies only the error correction (10) at each time step and the *Terminated* approach terminate the environment when the constraint violations bigger than a threshold. The hyperparameters for the experiments are shown in Table 1 and Table 2.

Environment Parameter	ATACOM	*ErrorCorrection*	*Terminated*
episode duration	5s	5s	5s
discount factor	0.99	0.99	0.99
simulation step size	0.01s	0.01s	0.01s
acceleration limit $\boldsymbol{a}_{\max}$	10	10	10
$\boldsymbol{K}_c$ in (10)	$\mathrm{diag}([100, 100])$	$\mathrm{diag}([100, 100])$	-
$\boldsymbol{K}_f$ for the equality constraint	$\mathrm{diag}([0.1])$	$\mathrm{diag}([0.1])$	-
$\boldsymbol{K}_g$ for the inequality constraint	$\mathrm{diag}([2])$	$\mathrm{diag}([2])$	-
$\boldsymbol{K}_a$ for the joint accelerations	$\mathrm{diag}([20, 20])$	$\mathrm{diag}([20, 20])$	-
error correction frequency	100	100	-
termination tolerance	-	-	0.4
action space	1D tangent space	2D acceleration	2D acceleration

Table 1: Parameters for *CircularMotion* Environment

	DDPG	TD3	SAC	TRPO	PPO
Sweeping parameter					
actor/critic learning rate	$\{1e^{-3}, 5e^{-4}, 1e^{-4}\}$			-	-
clipping coefficient	-	-	-	{0.1, 0.05, 0.01}	-
maximum kl	-	-	-	-	{0.2, 0.1, 0.05}
Default parameter					
epochs	50	50	50	50	50
steps per epoch	5000	5000	5000	5000	5000
steps per fit	1	1	1	2000	2000
episodes per test	25	25	25	25	25
actor/critic network size	[80 80]	[80 80]	[32 32]	[32 32]	[32 32]
batch size	64	64	64	64	64
initial policy covariance	0.2	1.0	-	0.25	0.25
initial replay size	5000	5000	5000	-	-
max replay size	200000	200000	200000	-	-
soft updates coefficient	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	-	-
warm-up transitions	-	-	10000	-	-
learning rate alpha	-	-	0.0003	-	-
target entropy	-	-	-6	-	-
GAE* update coefficient	-	-	-	0.95	0.95
entropy regularisation	-	-	-	5e-5	5e-5
line searches per fit	-	-	-	10	-
CG** steps per fit	-	-	-	10	-
CG damping	-	-	-	$1e^{-2}$	-
CG tolerance	-	-	-	$1e^{-10}$	-

Table 2: Training Parameter for Algorithms. GAE*: Generalized Advantage Estimation. CG**: Conjugate Gradient

D.2 *PlanarAirHockey*

For this task, we have demonstrated two tasks, *PlanarHit* and *PlanarDefend*. The Parameter choice of the environment and RL algorithms are shown in Table 3 and Table 4

Environment Parameter	*PlanarHit*	*PlanarDefend*
episode duration	2s	3s
discount factor	0.99	
simulation step size	1 / 240s	
acceleration limit $\boldsymbol{a}_{\max}$	$[10, 10, 10]$	
velocity limit $\boldsymbol{v}_{\max}$	$[2.3562, 2.3562, 2.3562]$	
$\boldsymbol{K}_c$ in (10)	$\mathrm{diag}([240])$	
$\boldsymbol{K}_f$ for the equality constraint	-	
$\boldsymbol{K}_g$ for the inequality constraint	$\mathrm{diag}([0.5, 0.5, 0.5, 1, 1, 1])$	
$\boldsymbol{K}_a$ for the joint accelerations	$\mathrm{diag}(2 \cdot \boldsymbol{a}_{\max}/\boldsymbol{v}_{\max})$	
error correction frequency	240	
control frequency	60	
maximum simulated joint velocity	$1.5 \cdot \boldsymbol{v}_{\max}$	

Table 3: Parameters for *PlanarAirHockey* Environment

	DDPG	TD3	SAC	TRPO	PPO
Sweeping parameter					
actor/critic learning rate	$\{1e^{-3}, 5e^{-4}, 1e^{-4}\}$			-	-
clipping coefficient	-	-	-	{0.1, 0.05, 0.01}	-
maximum kl	-	-	-	-	{0.2, 0.1, 0.05}
Default parameter (*PlanrHit*/*PlanrDefend*)					
epochs	50 / 150				
steps per epoch	5000 / 12000				
steps per fit	1	1	1	600	600
episodes per test	25				
actor/critic network size	[80 80]	[80 80]	[64 64]	[64 64]	[64 64]
batch size	64				
initial policy covariance	0.2	1.0	-	0.25	0.25
initial replay size	5000	5000	5000	-	-
max replay size	200000	200000	200000	-	-
soft updates coefficient	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	-	-
warm-up transitions	-	-	10000	-	-
learning rate alpha	-	-	0.0003	-	-
target entropy	-	-	-6	-	-
GAE* update coefficient	-	-	-	0.95	0.95
entropy regularisation	-	-	-	5e-5	5e-5
line searches per fit	-	-	-	10	-
CG** steps per fit	-	-	-	10	-
CG damping	-	-	-	$1e^{-2}$	-
CG tolerance	-	-	-	$1e^{-10}$	-

Table 4: Training Parameter for *PlanarHit*/*PlanarDefend*. GAE*: Generalized Advantage Estimation. CG**: Conjugate Gradient

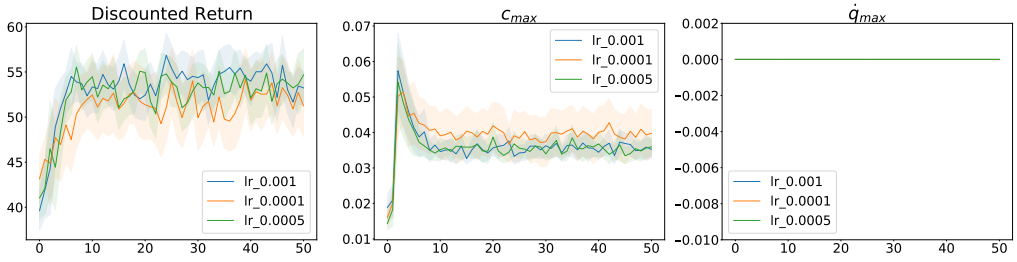
D.3 *IiwaAirHockey*

Environment Parameter	*IiwaAirHockey*
Sweeping Parameter (SAC)	
simulation step size	$[1/50\text{s}, 1/250\text{s}, 1/500\text{s}, 1/1000\text{s}]$
Default Parameter	
episode duration	2s
discount factor	0.99
acceleration limit $\boldsymbol{a}_{\max}$	$[10, 10, 10, 10, 10, 10]$
velocity limit $\boldsymbol{v}_{\max}$	$[1.4835, 1.4835, 1.7453, 1.3090, 2.2689, 2.3562]$
$\boldsymbol{K}_c$ in (10)	$\text{diag}([500])$
$\boldsymbol{K}_f$ for the equality constraint	$\text{diag}([0.1])$
$\boldsymbol{K}_g$ for the inequality constraint	$\text{diag}([0.5, 0.5, 0.5, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0])$
$\boldsymbol{K}_a$ for the joint accelerations	-
control frequency	50
maximum simulated joint velocity	$\boldsymbol{v}_{\max}$

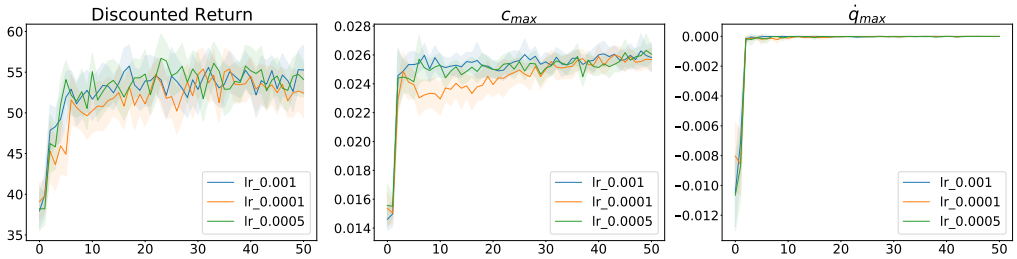
Table 5: Parameters for *IiwaAirHockey* Environment

	DDPG	TD3	SAC	TRPO	PPO
Sweeping parameter					
actor/critic learning rate	$\{1e^{-3}, 5e^{-4}, 1e^{-4}\}$			-	-
clipping coefficient	-	-	-	{0.1, 0.05, 0.01}	-
maximum kl	-	-	-	-	{0.2, 0.1, 0.05}
Default parameter (*PlanrHit*/*PlanrDefend*)					
epochs	50 / 150				
steps per epoch	5000 / 12000				
steps per fit	1	1	1	600	600
episodes per test	25				
actor/critic network size	[80 80]	[80 80]	[64 64]	[64 64]	[64 64]
batch size	64				
initial policy covariance	0.2	1.0	-	0.25	0.25
initial replay size	5000	5000	5000	-	-
max replay size	200000	200000	200000	-	-
soft updates coefficient	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	-	-
warm-up transitions	-	-	10000	-	-
learning rate alpha	-	-	0.0003	-	-
target entropy	-	-	-6	-	-
GAE* update coefficient	-	-	-	0.95	0.95
entropy regularisation	-	-	-	5e-5	5e-5
line searches per fit	-	-	-	10	-
CG** steps per fit	-	-	-	10	-
CG damping	-	-	-	$1e^{-2}$	-
CG tolerance	-	-	-	$1e^{-10}$	-

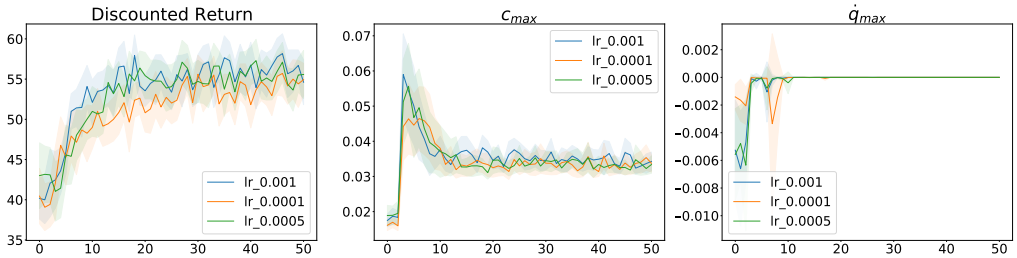
Table 6: Training Parameter for *IiwaAirHockey*. GAE*: Generalized Advantage Estimation. CG**: Conjugate Gradient



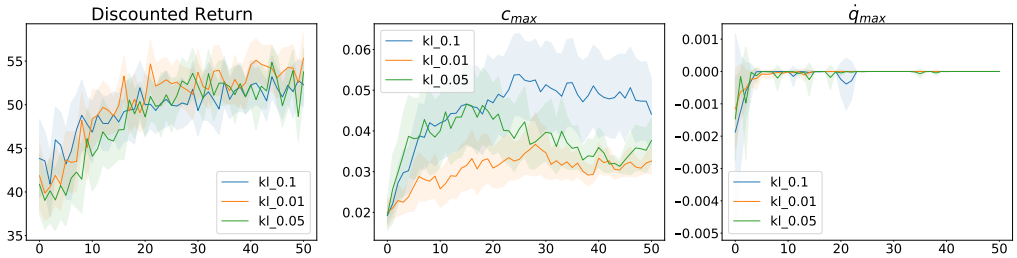
(a) DDPG



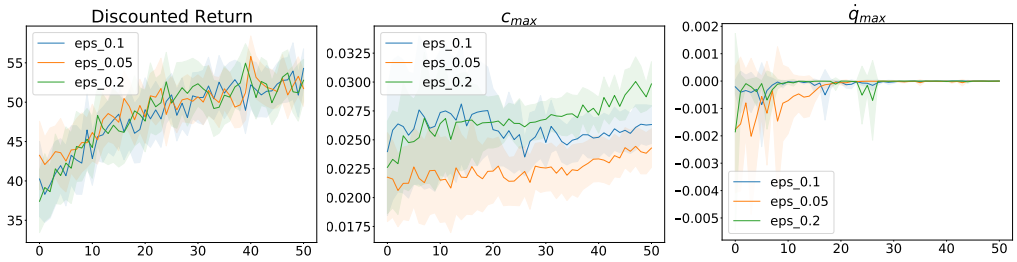
(b) TD3



(c) SAC



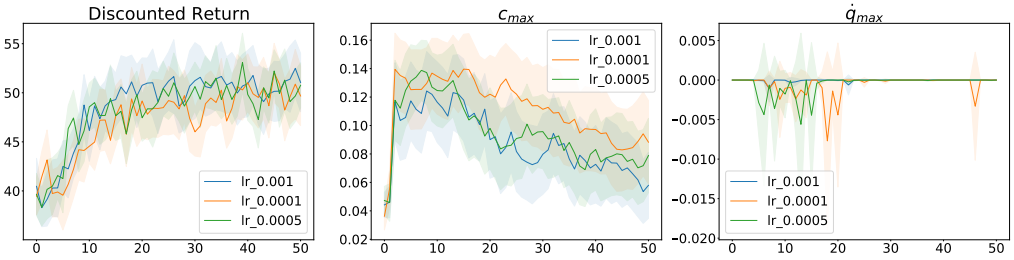
(d) TRPO



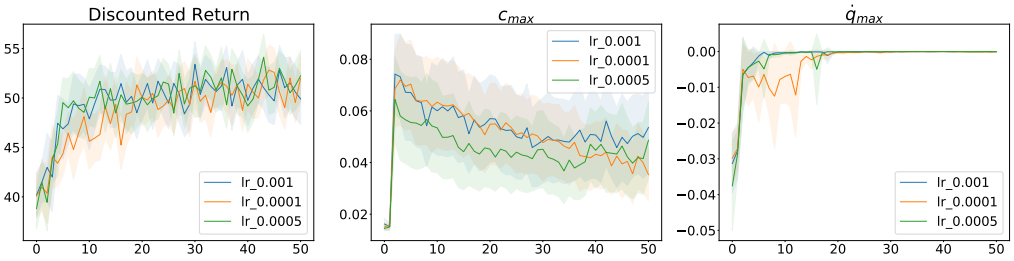
Discounted Return | Max Constraint Violations | Max Velocity Limits Violations

(e) PPO

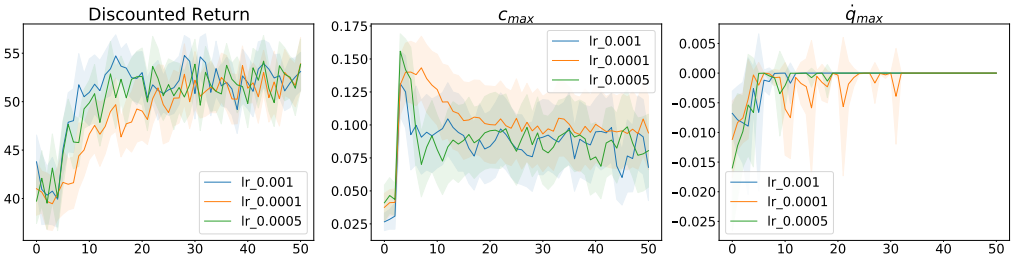
Figure 16: Parameter sweep of ATACOM in *CicularMotion*.



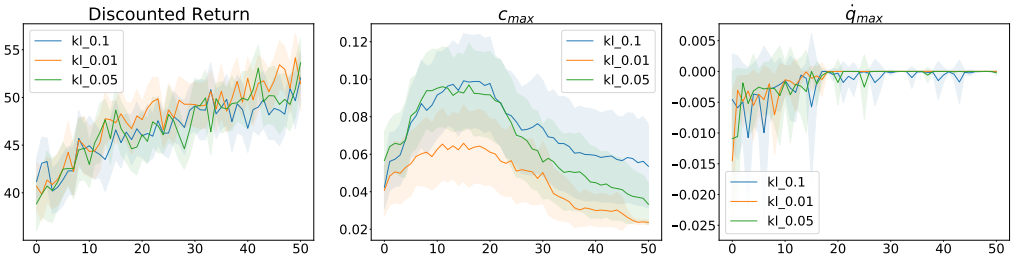
(a) DDPG



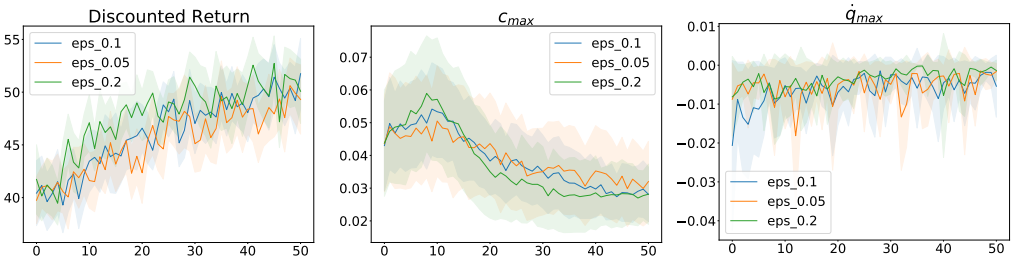
(b) TD3



(c) SAC



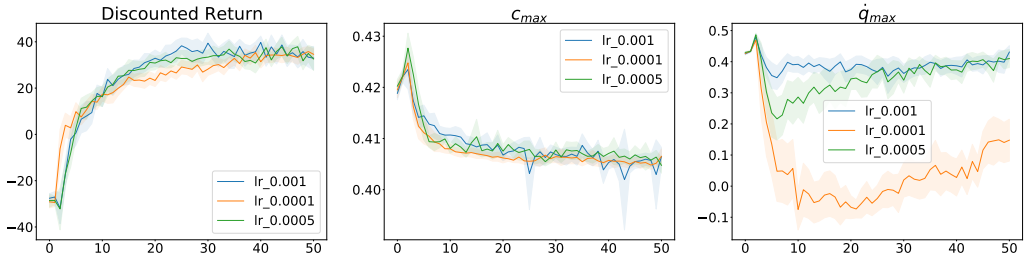
(d) TRPO



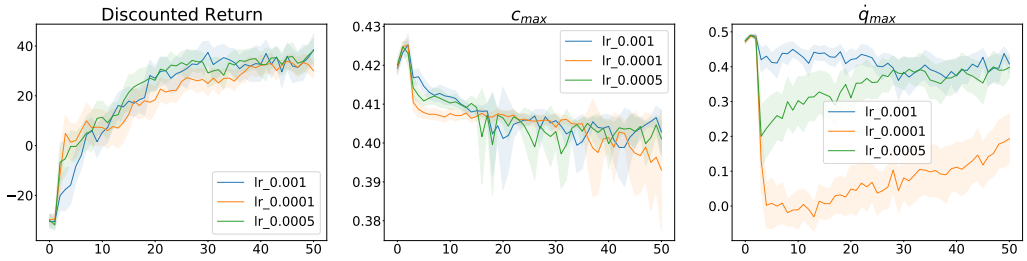
Discounted Return | Max Constraint Violations | Max Velocity Limits Violations

(e) PPO

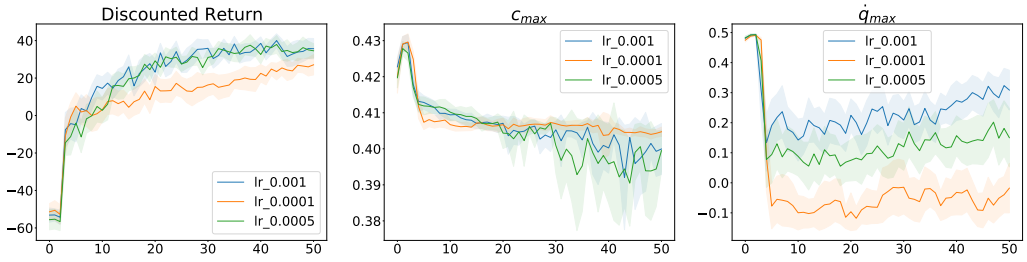
Figure 17: Parameter sweep of *ErrorCorrection* in *CicularMotion*.



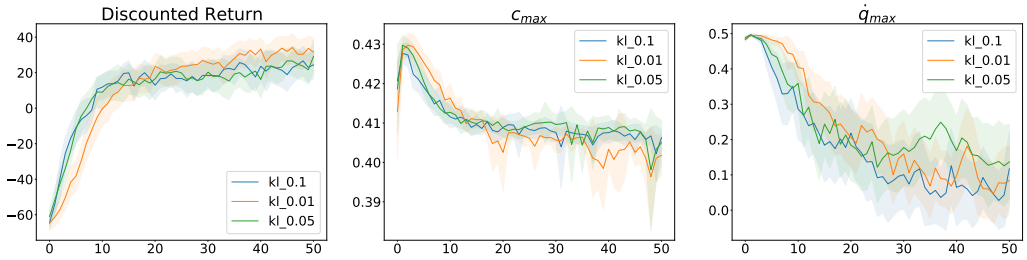
(a) DDPG



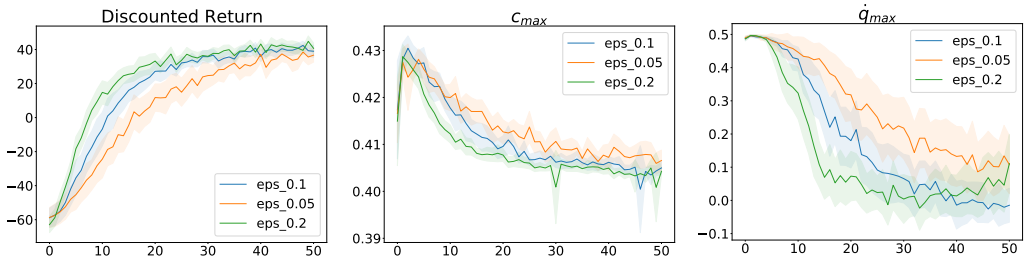
(b) TD3



(c) SAC



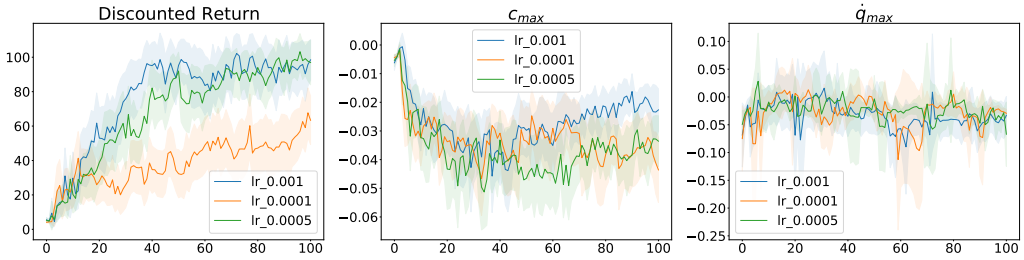
(d) TRPO



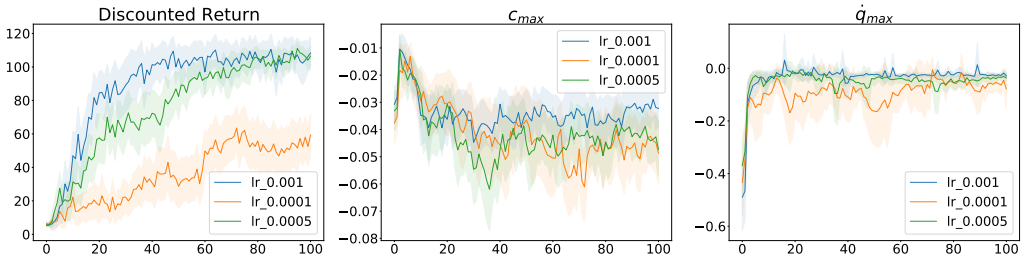
Discounted Return | Max Constraint Violations | Max Velocity Limits Violations

(e) PPO

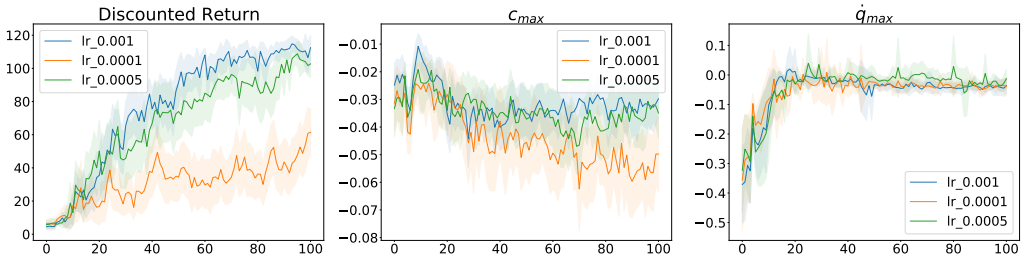
Figure 18: Parameter sweep of *Terminated* in *CicularMotion*.



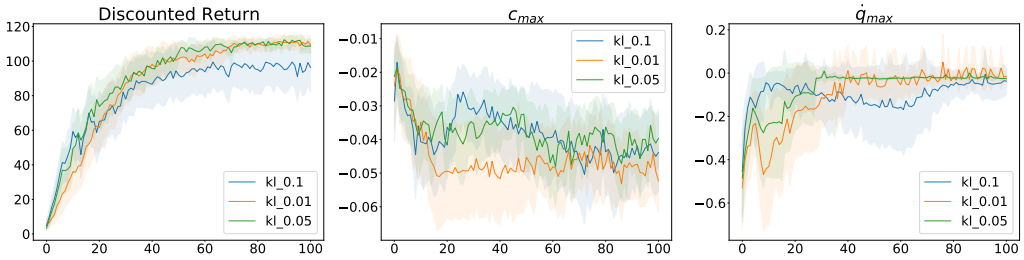
(a) DDPG



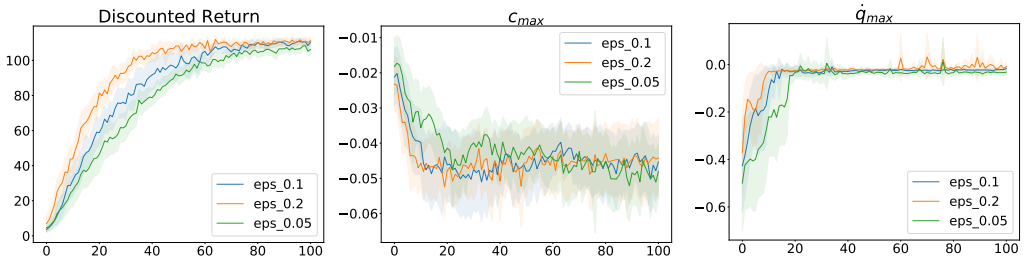
(b) TD3



(c) SAC



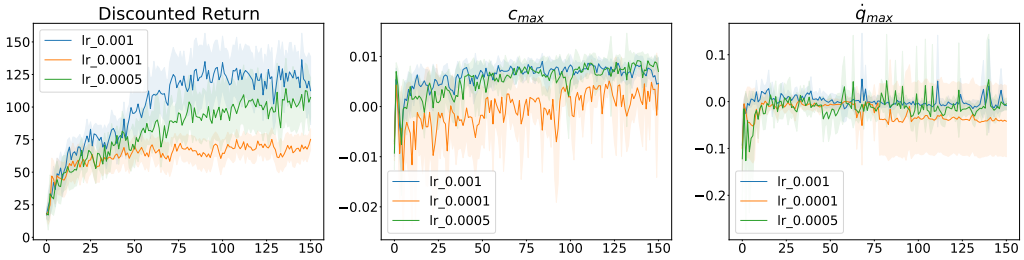
(d) TRPO



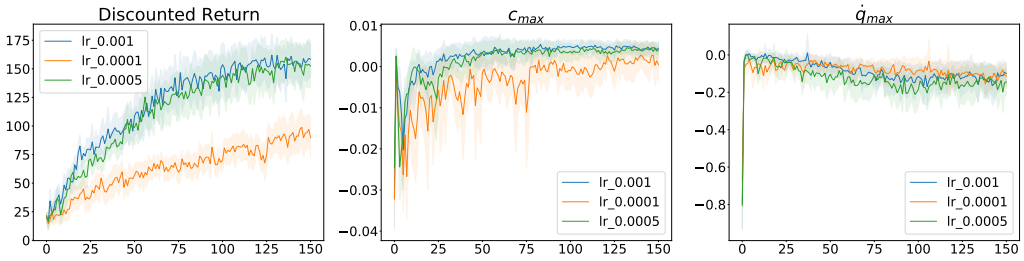
Discounted Return | Max Constraint Violations | Max Velocity Limits Violations

(e) PPO

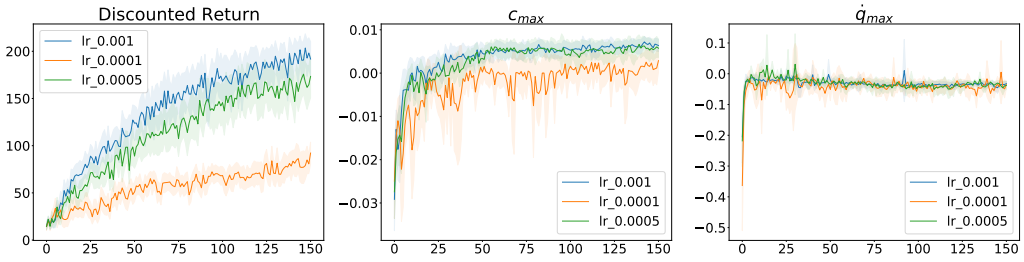
Figure 19: Parameter sweep of ATACOM in *PlanarHit*.



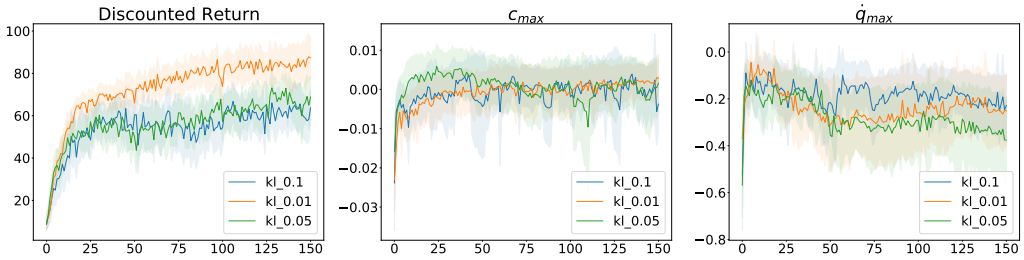
(a) DDPG



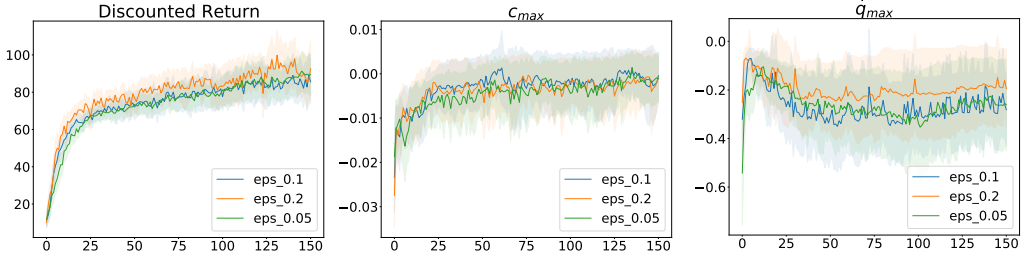
(b) TD3



(c) SAC



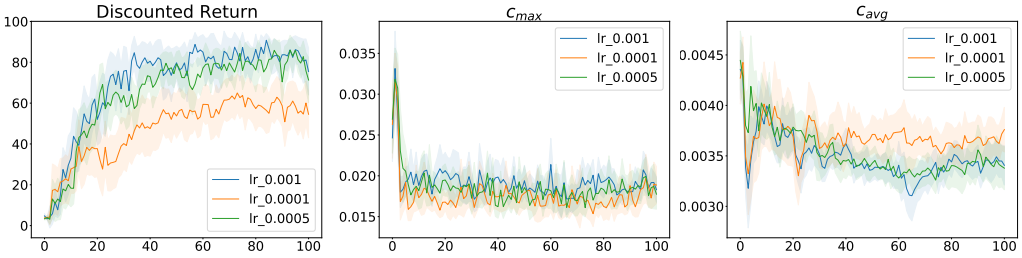
(d) TRPO



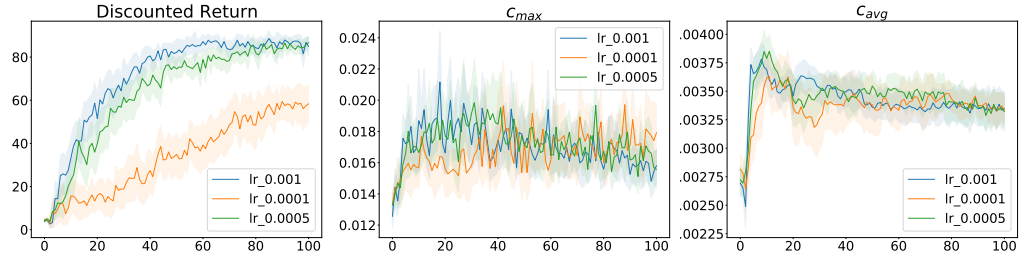
Discounted Return | Max Constraint Violations | Max Velocity Limits Violations

(e) PPO

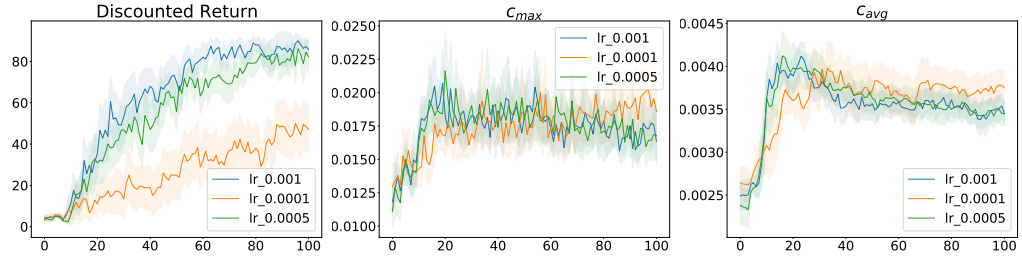
Figure 20: Parameter sweep of ATACOM in *PlanarDefend*.



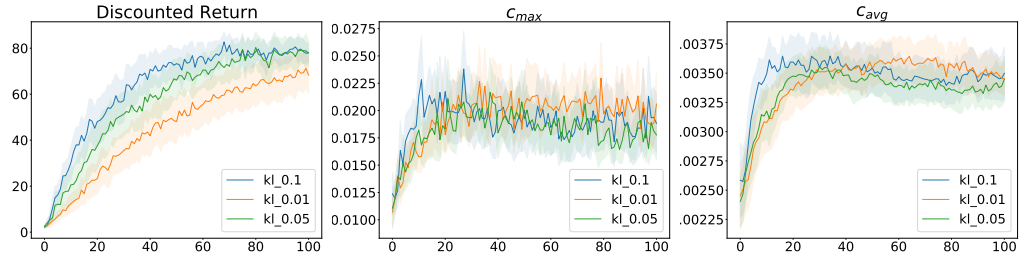
(a) DDPG



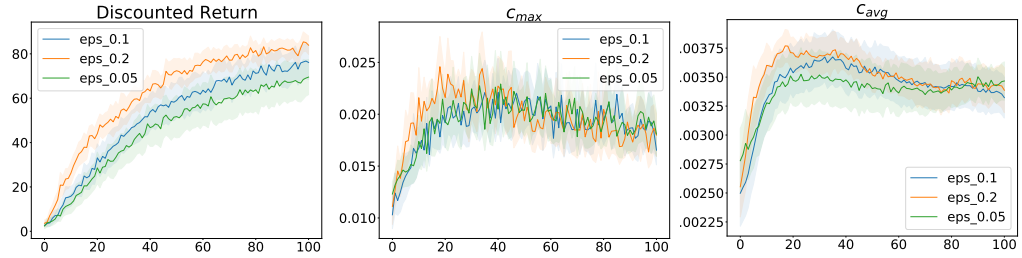
(b) TD3



(c) SAC



(d) TRPO



Discounted Return | Max Constraint Violations | Average Constraint Violations

(e) PPO

Figure 21: Parameter sweep of ATACOM in *IiwaAirHockey*.

E Constraint with Uncontrollable State

The proposed method ATACOM can be easily extended to the constraints with uncontrollable states. We assume that the velocity of uncontrollable state $\dot{\boldsymbol{x}}$ can be estimated and the acceleration of controllable state $\ddot{\boldsymbol{x}} = \mathbf{0}$. Here we briefly introduce the extension of ATACOM of the inequality constraints with $\boldsymbol{g}(\boldsymbol{q}, \boldsymbol{x}) \leq \mathbf{0}$. The viability constraint set is

$$\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{x}, \dot{\boldsymbol{x}}, \boldsymbol{\mu}) = \boldsymbol{g}(\boldsymbol{q}, \boldsymbol{x}) + \boldsymbol{K}(\boldsymbol{J}_q(\boldsymbol{q}, \boldsymbol{x})\dot{\boldsymbol{q}} + \boldsymbol{J}_x(\boldsymbol{q}, \boldsymbol{x})\dot{\boldsymbol{x}}) + \frac{1}{2}\boldsymbol{\mu}^2 = \mathbf{0}$$

with the partial derivatives $\boldsymbol{J}_q(\boldsymbol{q}, \boldsymbol{x}) = \frac{\partial}{\partial q}\boldsymbol{g}(\boldsymbol{q}, \boldsymbol{x})$ and $\boldsymbol{J}_x(\boldsymbol{q}, \boldsymbol{x}) = \frac{\partial}{\partial x}\boldsymbol{g}(\boldsymbol{q}, \boldsymbol{x})$. We use $\boldsymbol{J}_q, \boldsymbol{J}_x$ to simplify the notation. The time derivative of the viability constraint is

$$\dot{\boldsymbol{c}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{x}, \dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}, \boldsymbol{\mu}, \dot{\boldsymbol{\mu}}) = \boldsymbol{J}_q\dot{\boldsymbol{q}} + \boldsymbol{J}_x\dot{\boldsymbol{x}} + \boldsymbol{K}\boldsymbol{J}_q\ddot{\boldsymbol{q}} + \boldsymbol{K}\dot{\boldsymbol{J}}_q\dot{\boldsymbol{q}} + \boldsymbol{K}\boldsymbol{J}_x\ddot{\boldsymbol{x}} + \boldsymbol{K}\dot{\boldsymbol{J}}_x\dot{\boldsymbol{x}} + \operatorname{diag}(\boldsymbol{\mu})\dot{\boldsymbol{\mu}}$$

with the time derivative of i-th Jacobian $\dot{\boldsymbol{J}}_{i,q} = \dot{\boldsymbol{q}}^\mathsf{T}\boldsymbol{H}_{i,qq} + \dot{\boldsymbol{x}}^\mathsf{T}\boldsymbol{H}_{i,xq}$ and $\dot{\boldsymbol{J}}_{i,x} = \dot{\boldsymbol{x}}^\mathsf{T}\boldsymbol{H}_{i,xx} + \dot{\boldsymbol{q}}^\mathsf{T}\boldsymbol{H}_{i,qx}$. $\boldsymbol{H}_i$ is the hessian matrix w.r.t i-th constraint.

As mentioned before, we assume $\dot{\boldsymbol{x}}$ is known and $\ddot{\boldsymbol{x}} = \mathbf{0}$. By setting $\dot{\boldsymbol{c}} = \mathbf{0}$, we have

$$\underbrace{\begin{bmatrix}\boldsymbol{K}\boldsymbol{J}_q & \operatorname{diag}(\boldsymbol{\mu})\end{bmatrix}}_{\boldsymbol{J}_c}\begin{bmatrix}\ddot{\boldsymbol{q}}\\ \dot{\boldsymbol{\mu}}\end{bmatrix} + \underbrace{\boldsymbol{J}_q\dot{\boldsymbol{q}} + \boldsymbol{J}_x\dot{\boldsymbol{x}} + \boldsymbol{K}\dot{\boldsymbol{J}}_q\dot{\boldsymbol{q}} + \boldsymbol{K}\dot{\boldsymbol{J}}_x\dot{\boldsymbol{x}}}_{\boldsymbol{\psi}} = \mathbf{0}$$

The overall control action is

$$\begin{bmatrix}\ddot{\boldsymbol{q}}\\ \dot{\boldsymbol{\mu}}\end{bmatrix} = -\boldsymbol{J}_c^\dagger(\boldsymbol{\psi} + \boldsymbol{K}_c\boldsymbol{c}) + \boldsymbol{N}_c\boldsymbol{\alpha}$$

with the error correction gain $\boldsymbol{K}_c$. We validate our method in the *CollisionAvoidance* environment which is described in Appendix E.1.

E.1 *CollisionAvoidance*

In this experiment, we demonstrate a collision avoidance task with four moving obstacles in a 2d environment shown in Figure 22. The objective is to move the agent (blue circle) to the target (green square) while avoiding the collision with four random moving obstacles (red circle). In this environment, the velocities of the obstacles are known. The agent and obstacles have a radius of 0.3. The control action is the acceleration along x, y direction. The collision avoidance constraint is

$$g_i : 0.6^2 - (q_x - o_{i,x})^2 - (q_2 - o_{i,y})^2 < 0, \quad i \in \{1, 2, 3, 4\}.$$

q_x and q_y are agent's positions and $o_{i,x}, o_{i,y}$ are positions of i-th object. The reward function is

$$r(q_x, q_y) = -\frac{1}{c}\|\boldsymbol{q}_{goal} - \boldsymbol{q}\|.$$

with a scale constant c.

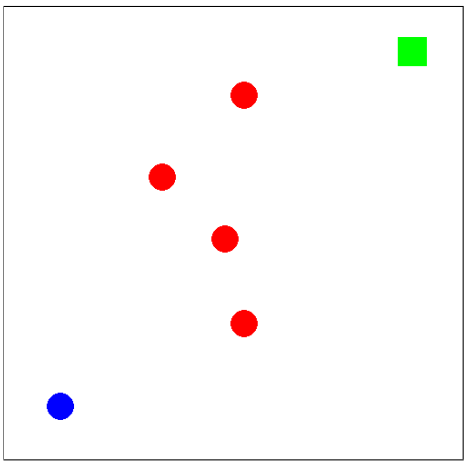


Figure 22: Collision Avoidance Environment

	SAC
default parameter	
actor/critic learning rate	$3e^{-4}$
epochs	100
steps per epoch	10000
steps per fit	11
episodes per test	25
actor/critic network size	[64 64]
batch size	64
initial policy covariance	-
initial replay size	5000
max replay size	200000
soft updates coefficient	$1e^{-3}$
warm-up transitions	10000
learning rate alpha	0.0003
target entropy	-4

Table 7: Training Parameter for Collision Avoidance

E.2 Experiment of Collision Avoidance

In this section, we demonstrate the preliminary result of collision avoidance. We applied the SAC with default parameter setup as shown in Table 7. As shown in Figure 23, the agent learns to reach the goal while the maximum constraint violations throughout the learning process remain low. Small constraint violations during the learning process occur if the agent is surrounded by the obstacles in the corner, and there exists no feasible action to avoid the collision.

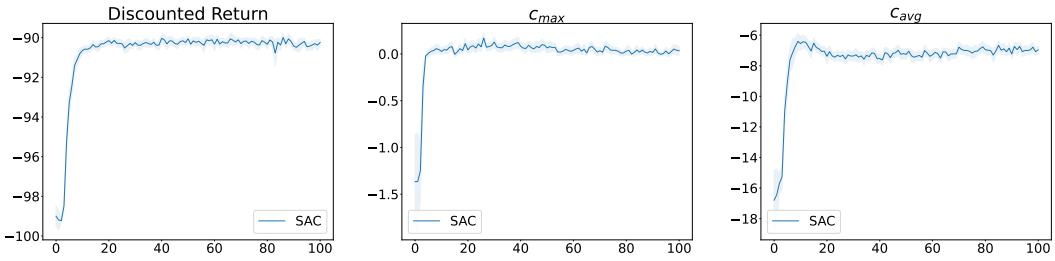


Figure 23: Learning Curve of ATACOM-SAC in *CollisionAvoidance* Environment