Supplementary Material

1. Task Specifications

Below we provide some specifications for the task observations, actions, and rewards. Please refer to the benchmark source code (https://github.com/rllab/rllab) for complete specification of physics parameters.

1.1. Basic Tasks

Cart-Pole Balancing: In this task, an inverted pendulum is mounted on a pivot point on a cart. The cart itself is restricted to linear movement, achieved by applying horizontal forces. Due to the system's inherent instability, continuous cart movement is needed to keep the pendulum upright. The observation consists of the cart position x, pole angle θ, the cart velocity $\dot{x}$, and the pole velocity $\dot{\theta}$. The 1D action consists of the horizontal force applied to the cart body. The reward function is given by $r(s,a) := 10 - (1 - \cos(\theta)) - 10^{-5}\|a\|_2^2$. The episode terminates when $|x| > 2.4$ or $|\theta| > 0.2$.

Cart-Pole Swing Up: This is a more complicated version of the previous task, in which the system should not only be able to balance the pole, but first succeed in swinging it up into an upright position. This task extends the working range of the inverted pendulum to $360°$. This is a nonlinear extension of the previous task. It has the same observation and action as in balancing. The reward function is given by $r(s,a) := \cos(\theta)$. The episode terminates when $|x| > 3$, with a penalty of -100.

Mountain Car: In this task, a car has to escape a valley by repetitive application of tangential forces. Because the maximal tangential force is limited, the car has to alternately drive up along the two slopes of the valley in order to build up enough inertia to overcome gravity. This brings a challenge of exploration, since before first reaching the goal among all trials, a locally optimal solution exists, which is to drive to the point closest to the target and stay there for the rest of the episode. The observation is given by the horizontal position x and the horizontal velocity $\dot{x}$ of the car. The reward is given by $r(s,a) := -1 + \text{height}$, with height the car's vertical offset. The episode terminates when the car reaches a target height of 0.6. Hence the goal is to reach the target as soon as possible.

Acrobot Swing Up: In this task, an under-actuated, two-link robot has to swing itself into an upright position. It consists of two joints of which the first one has a fixed position and only the second one can exert torque. The goal is to swing the robot into an upright position and stabilize around that position. The controller not only has to swing the pendulum in order to build up inertia, similar to the Mountain Car task, but also has to decelerate it in order to prevent it from tipping over. The observation includes the two joint angles, θ_1 and θ_2, and their velocities, $\dot{\theta}_1$ and $\dot{\theta}_2$. The action is the torque applied at the second joint. The reward is defined as $r(s,a) := -\|\text{tip}(s) - \text{tip}_{\text{target}}\|_2$, where $\text{tip}(s)$ computes the Cartesian position of the tip of the robot given the joint angles. No termination condition is applied.

Double Inverted Pendulum Balancing: This task extends the Cart-Pole Balancing task by replacing the single-link pole by a two-link rigid structure. As in the former task, the goal is to stabilize the two-link pole near the upright position. This task is more difficult than single-pole balancing, since the system is even more unstable and requires the controller to actively maintain balance. The observation includes the cart position x, joint angles (θ_1 and θ_2), and joint velocities ($\dot{\theta}_1$ and $\dot{\theta}_2$). We encode each joint angle as its sine and cosine values. The action is the same as in cart-pole tasks. The reward is given by $r(s,a) = 10 - 0.01x_{\text{tip}}^2 - (y_{\text{tip}} - 2)^2 - 10^{-3} \cdot \dot{\theta}_1^2 - 5 \cdot 10^{-3} \cdot \dot{\theta}_2^2$, where $x_{\text{tip}}, y_{\text{tip}}$ are the coordinates of the tip of the pole. No termination condition is applied. The episode is terminated when $y_{\text{tip}} \leq 1$.

1.2. Locomotion Tasks

Swimmer: The swimmer is a planar robot with 3 links and 2 actuated joints. Fluid is simulated through viscosity forces, which apply drag on each link, allowing the swimmer to move forward. This task is the simplest of all locomotion tasks, since there are no irrecoverable states in which the swimmer can get stuck, unlike other robots which may fall down or flip over. This places less burden on exploration. The 13-dim observation includes the joint angles, joint velocities, as well as

the coordinates of the center of mass. The reward is given by $r(s,a) = v_x - 0.005\|a\|_2^2$, where v_x is the forward velocity. No termination condition is applied.

Hopper: The hopper is a planar monopod robot with 4 rigid links, corresponding to the torso, upper leg, lower leg, and foot, along with 3 actuated joints. More exploration is needed than the swimmer task, since a stable hopping gait has to be learned without falling. Otherwise, it may get stuck in a local optimum of diving forward. The 20-dim observation includes joint angles, joint velocities, the coordinates of center of mass, and constraint forces. The reward is given by $r(s,a) := v_x - 0.005 \cdot \|a\|_2^2 + 1$, where the last term is a bonus for being "alive." The episode is terminated when $z_{body} < 0.7$ where z_{body} is the z-coordinate of the body, or when $|\theta_y| < 0.2$, where θ_y is the forward pitch of the body.

Walker: The walker is a planar biped robot consisting of 7 links, corresponding to two legs and a torso, along with 6 actuated joints. This task is more challenging than hopper, since it has more degrees of freedom, and is also prone to falling. The 21-dim observation includes joint angles, joint velocities, and the coordinates of center of mass. The reward is given by $r(s,a) := v_x - 0.005 \cdot \|a\|_2^2$. The episode is terminated when $z_{body} < 0.8$, $z_{body} > 2.0$, or when $|\theta_y| > 1.0$.

Half-Cheetah: The half-cheetah is a planar biped robot with 9 rigid links, including two legs and a torso, along with 6 actuated joints. The 20-dim observation includes joint angles, joint velocities, and the coordinates of the center of mass. The reward is given by $r(s,a) = v_x - 0.05 \cdot \|a\|_2^2$. No termination condition is applied.

Ant: The ant is a quadruped with 13 rigid links, including four legs and a torso, along with 8 actuated joints. This task is more challenging than the previous tasks due to the higher degrees of freedom. The 125-dim observation includes joint angles, joint velocities, coordinates of the center of mass, a (usually sparse) vector of contact forces, as well as the rotation matrix for the body. The reward is given by $r(s,a) = v_x - 0.005 \cdot \|a\|_2^2 - C_{\text{contact}} + 0.05$, where C_{contact} penalizes contacts to the ground, and is given by $5 \cdot 10^{-4} \cdot \|F_{\text{contact}}\|_2^2$, where F_{contact} is the contact force vector clipped to values between -1 and 1. The episode is terminated when $z_{body} < 0.2$ or when $z_{body} > 1.0$.

Simple Humanoid: This is a simplified humanoid model with 13 rigid links, including the head, body, arms, and legs, along with 10 actuated joints. The increased difficulty comes from the increased degrees of freedom as well as the need to maintain balance. The 102-dim observation includes the joint angles, joint velocities, vector of contact forces, and the coordinates of the center of mass. The reward is given by $r(s,a) = v_x - 5 \cdot 10^{-4}\|a\|_2^2 - C_{\text{contact}} - C_{\text{deviation}} + 0.2$, where $C_{\text{contact}} = 5 \cdot 10^{-6} \cdot \|F_{\text{contact}}\|$, and $C_{\text{deviation}} = 5 \cdot 10^{-3} \cdot (v_y^2 + v_z^2)$ to penalize deviation from the forward direction. The episode is terminated when $z_{body} < 0.8$ or when $z_{body} > 2.0$.

Full Humanoid: This is a humanoid model with 19 rigid links and 28 actuated joints. It has more degrees of freedom below the knees and elbows, which makes the system higher-dimensional and harder for learning. The 142-dim observation includes the joint angles, joint velocities, vector of contact forces, and the coordinates of the center of mass. The reward and termination condition is the same as in the Simple Humanoid model.

1.3. Partially Observable Tasks

Limited Sensors: The full description is included in the main text.

Noisy Observations and Delayed Actions: For all tasks, we use a Gaussan noise with $\sigma = 0.1$. The time delay is as follows: Cart-Pole Balancing 0.15 sec, Cart-Pole Swing Up 0.15 sec, Mountain Car 0.15 sec, Acrobot Swing Up 0.06 sec, and Double Inverted Pendulum Balancing 0.06 sec. This corresponds to 3 discretization frames for each task.

System Identifications: For Cart-Pole Balancing and Cart-Pole Swing Up, the pole length is varied uniformly between, 50% and 150%. For Mountain Car, the width of the valley varies uniformly between 75% and 125%. For Acrobot Swing Up, each of the pole length varies uniformly between 50% and 150%. For Double Inverted Pendulum Balancing, each of the pole length varies uniformly between 83% and 167%. Please refer to the benchmark source code for reference values.

1.4. Hierarchical Tasks

Locomotion + Food Collection: During each episode, 8 food units and 8 bombs are placed in the environment. Collecting a food unit gives $+1$ reward, and collecting a bomb gives -1 reward. Hence the best cumulative reward for a given episode is 8.

Locomotion + Maze: During each episode, a $+1$ reward is given when the robot reaches the goal. Otherwise, the robot receives a zero reward throughout the episode.

2. Experiment Parameters

For all batch gradient-based algorithms, we use the same time-varying feature encoding for the linear baseline:

$$\phi_{s,t} = \text{concat}(s, s \odot s, 0.01t, (0.01t)^2, (0.01t)^3, 1)$$

where s is the state vector and $\odot$ represents element-wise product.

Table 1 shows the experiment parameters for all four categories. We will then detail the hyperparameter search range for the selected tasks and report best hyperparameters, shown in Tables 2, 3, 4, 5, 6, and 7.

Table 1. Experiment Setup

	Basic & Locomotion	Partially Observable	Hierarchical
Sim. steps per Iter.	50,000	50,000	50,000
Discount(λ)	0.99	0.99	0.99
Horizon	500	100	500
Num. Iter.	500	300	500

Table 2. Learning Rate α for REINFORCE

	Search Range	Best
Cart-Pole Swing Up	$[1 \times 10^{-4}, 1 \times 10^{-1}]$	5×10^{-3}
Double Inverted Pendulum	$[1 \times 10^{-4}, 1 \times 10^{-1}]$	5×10^{-3}
Swimmer	$[1 \times 10^{-4}, 1 \times 10^{-1}]$	1×10^{-2}
Ant	$[1 \times 10^{-4}, 1 \times 10^{-1}]$	5×10^{-3}

Table 3. Step Size δ_{KL} for TNPG

	Search Range	Best
Cart-Pole Swing Up	$[1 \times 10^{-3}, 5 \times 10^{0}]$	5×10^{-2}
Double Inverted Pendulum	$[1 \times 10^{-3}, 5 \times 10^{0}]$	3×10^{-2}
Swimmer	$[1 \times 10^{-3}, 5 \times 10^{0}]$	1×10^{-1}
Ant	$[1 \times 10^{-3}, 5 \times 10^{0}]$	3×10^{-1}

Table 4. Step Size δ_{KL} for TRPO

	Search Range	Best
Cart-Pole Swing Up	$[1 \times 10^{-3}, 5 \times 10^{0}]$	5×10^{-2}
Double Inverted Pendulum	$[1 \times 10^{-3}, 5 \times 10^{0}]$	1×10^{-3}
Swimmer	$[1 \times 10^{-3}, 5 \times 10^{0}]$	5×10^{-2}
Ant	$[1 \times 10^{-3}, 5 \times 10^{0}]$	8×10^{-2}

Table 5. Step Size δ_{KL} for REPS

	Search Range	Best
Cart-Pole Swing Up	$[1 \times 10^{-3}, 5 \times 10^{0}]$	1×10^{-2}
Double Inverted Pendulum	$[1 \times 10^{-3}, 5 \times 10^{0}]$	8×10^{-1}
Swimmer	$[1 \times 10^{-3}, 5 \times 10^{0}]$	3×10^{-1}
Ant	$[1 \times 10^{-3}, 5 \times 10^{0}]$	8×10^{-1}

Table 6. Initial Extra Noise for CEM

	Search Range	Best
Cart-Pole Swing Up	$[1 \times 10^{-3}, 1]$	1×10^{-2}
Double Inverted Pendulum	$[1 \times 10^{-3}, 1]$	1×10^{-1}
Swimmer	$[1 \times 10^{-3}, 1]$	1×10^{-1}
Ant	$[1 \times 10^{-3}, 1]$	1×10^{-1}

Table 7. Initial Standard Deviation for CMA-ES

	Search Range	Best
Cart-Pole Swing Up	$[1 \times 10^{-3}, 1 \times 10^{3}]$	1×10^{3}
Double Inverted Pendulum	$[1 \times 10^{-3}, 1 \times 10^{3}]$	3×10^{-1}
Swimmer	$[1 \times 10^{-3}, 1 \times 10^{3}]$	1×10^{-1}
Ant	$[1 \times 10^{-3}, 1 \times 10^{3}]$	1×10^{-1}