A Algorithms

Trajectory optimization for receding horizon curiosity (RHC) is implemented using CasADi [20], a control and auto-differentiation toolbox. The number of Fourier features in the learned dynamics model varies across environments: MountainCar 20, Pendulum 90, CartPole 80.

As a model-free baseline RL algorithm, soft actor-critic (SAC) is used [23], as implemented in Stable Baselines [40], with the following parameters across all environments

$$\begin{aligned} \gamma &= 0.99, \\ \tau &= 0.005, \\ \text{learning rate} &= 0.0003, \\ \text{buffer size} &= 50000, \\ \text{batch size} &= 64. \end{aligned}$$

The exploration bonus based on the *squared prediction error* is defined as follows

$$r_{\text{pe}}(s_t, a_t) = \left(s_{t+1} - \mathbb{E}_p[s_{t+1} \mid s_t, a_t]\right)^2,$$

where p denotes the model trained on the data from previous episodes. The exploration bonus based on the *information gain* is defined as the reduction of entropy

$$r_{\text{ig}}(s_t, a_t) = \mathbb{H}(\boldsymbol{\theta} \mid \mathcal{X}_t) - \mathbb{H}(\boldsymbol{\theta} \mid \mathcal{X}_{t+1}),$$

where $\mathcal{X}_t$ denotes the set of observations until time step t, $\mathbb{H}$ is the entropy, and $\boldsymbol{\theta}$ denotes the model parameters.

B Environments

MountainCar. The implementation from OpenAI Gym [41] is modified as follows to accommodate the episodic exploration setting. Car power is set to 10^{-3} and the speed limit is removed. Upon reset, the car starts at the center of the valley with zero velocity. An episode ends when the car reaches the environment bounds or after 130 time steps. The evaluation task is to drive the car on top of the mountain as dictated by the stage cost $c = 10(x - x_{\text{goal}})^2 + 0.001a^2$, where x is the position of the car, x_{goal} is the goal location, and a the action.

Pendulum. The implementation from DeepMind Control Suite [42] with observations $[\cos\theta, \sin\theta, \dot{\theta}]$ is used with the following modifications. The pendulum is initialized handging down with zero velocity. Each episode consists of 100 time steps of 80ms duration each. The evaluation task is to swing the pendulum up as dictated by the stage cost $c = 100(1 - \cos\theta)^2 + 0.1\sin^2\theta + 0.1\dot{\theta}^2 + 0.001a^2$.

CartPole. The implementation from DeepMind Control Suite [42] with observations $[x, \cos\theta, \sin\theta, \dot{x}, \dot{\theta}]$ is used. Each episode starts with the cart at the center and the pole hanging down, both having zero velocity. The system is simulated at 50Hz. An episode ends when the cart reaches the state limits or after 100 time steps. The evaluation task is to swing the pole up, $c = 100x^2 + 100(1 - \cos\theta)^2 + 0.1\sin^2\theta + 0.1\dot{x}^2 + 0.1\dot{\theta}^2 + 0.1a^2$.

C Runtimes

Table 1 shows how long one run of each algorithm depicted in Fig. 1 on average takes. One run consists of 20 episodes (x-axis in Fig. 1). Evaluation of RHC EVR was only possible on the MountainCar environment due to its high memory demands. Evaluations were run on a machine with an Intel Xeon E5-2670 processor.

	MountainCar	Pendulum	CartPole
RHC EVR	1.51	-	-
RHC US	0.03	0.80	0.54
SAC PE	0.03	0.20	0.45
SAC IG	0.03	0.21	0.48
RAND	0.01	0.09	0.30
MAX	9.49	11.17	5.84

Table 1: Average wall-clock-time (in hours) for evaluated exploration algorithms (see Fig. 1).