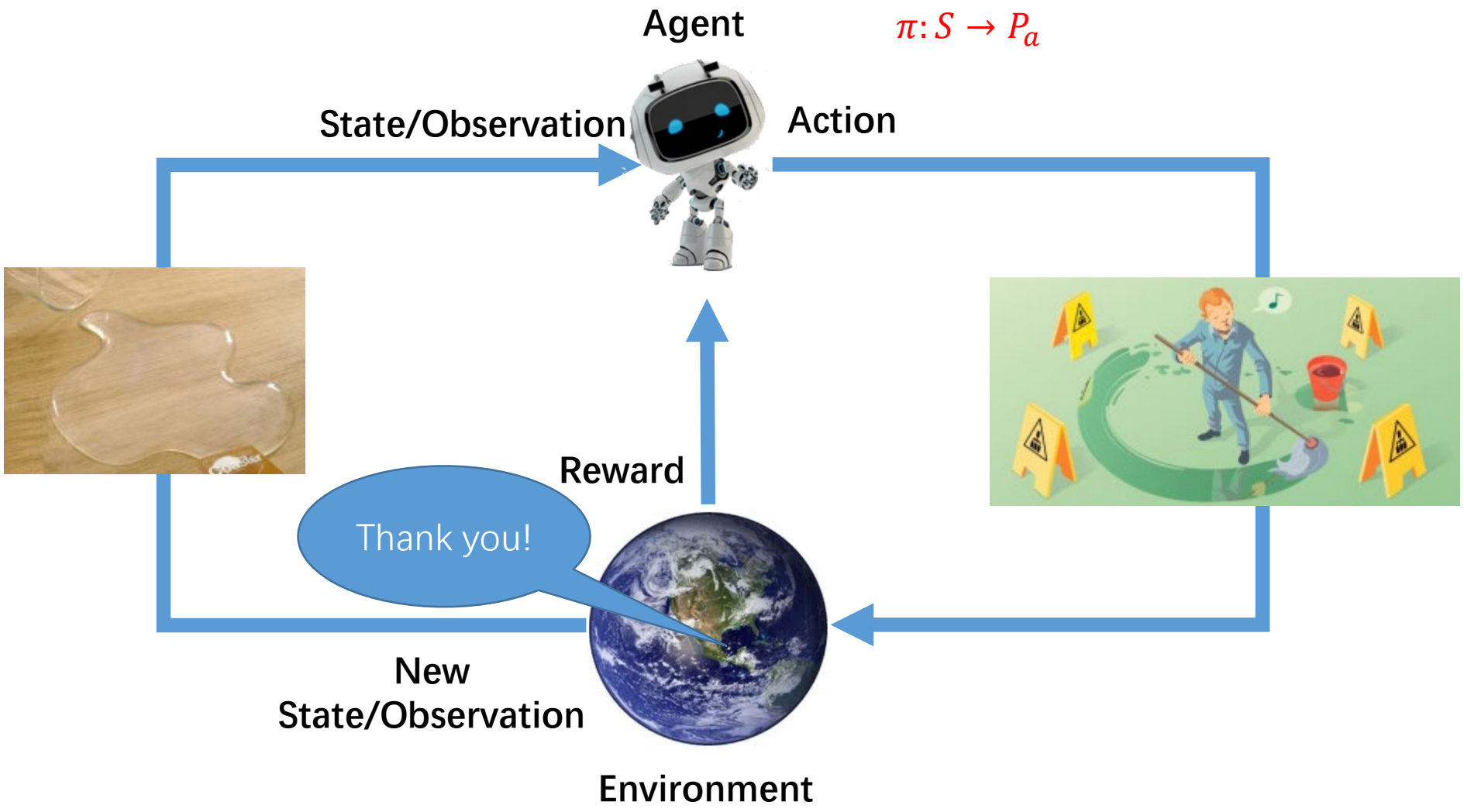
Model-Based Active Exploration

ICML 2019

Reinforcement Learning

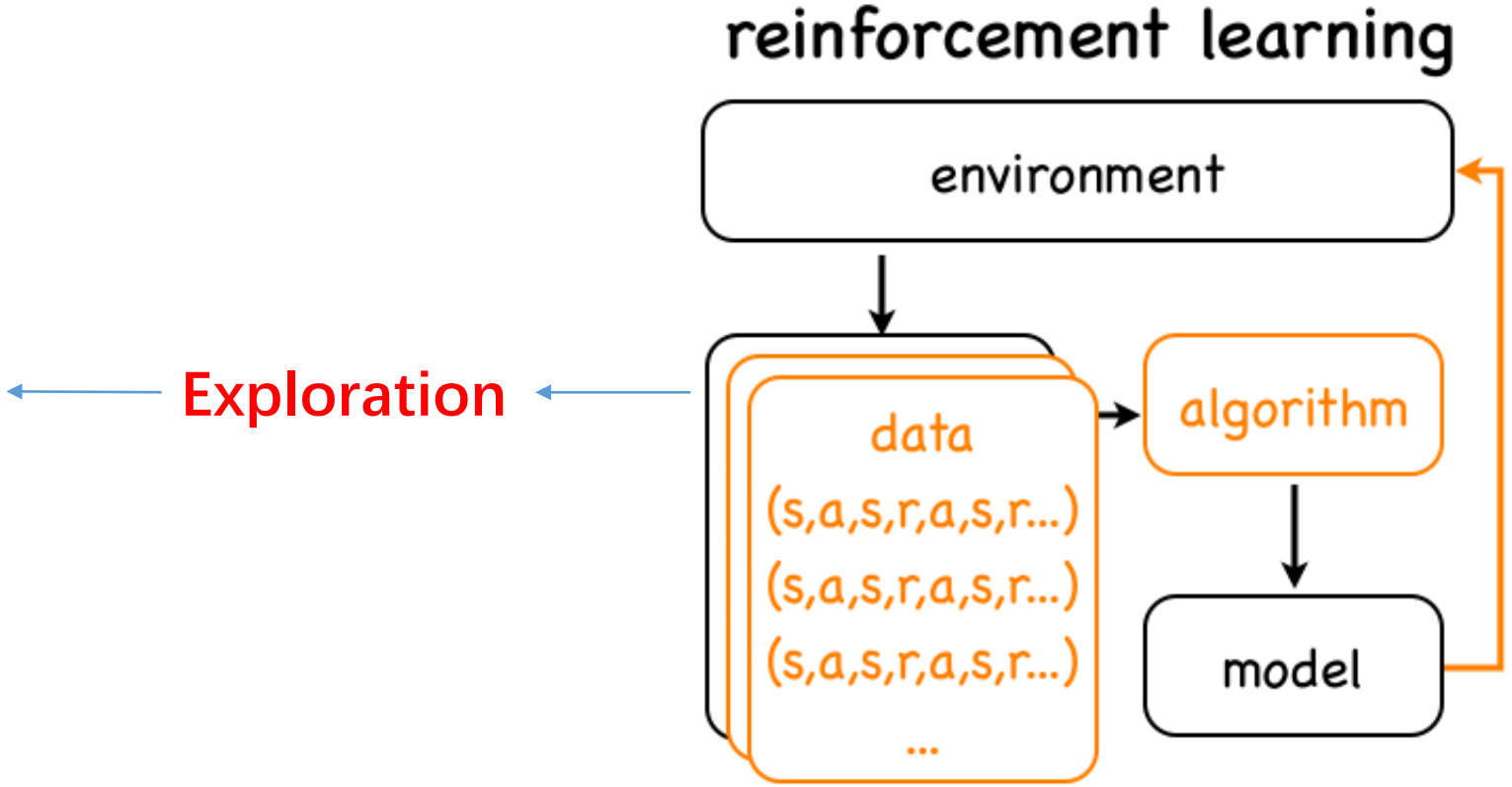


Reinforcement Learning

- Random:
  $\pi + noise$
- Reactive:
  $\pi + f$

where f rewards the agent for fortuitously encounteri-ng novel situations

- Active:
  π_e



Proposed method

T is the space of all possible transition functions.
$\phi = (s, a, s')$

$$\mathrm{IG}(s, a, s') = \mathrm{IG}(\phi) = D_{\mathrm{KL}}(\mathcal{P}(T|\phi) \parallel \mathcal{P}(T))$$

$$\mathrm{IG}(\pi) = \mathbb{E}_{\phi \sim \mathcal{P}(\Phi|\pi)} [\mathrm{IG}(\phi)]$$

Proposed method

$$\mathrm{IG}(\pi) = \mathbb{E}_{t\sim\mathcal{P}(T)}\left[\mathbb{E}_{s,a\sim\mathcal{P}(\mathcal{S},\mathcal{A}|\pi,t)}\left[u(s,a)\right]\right],$$

where

$$u(s,a) = \int_T \int_{\mathcal{S}} \mathrm{IG}(s,a,s')p(s'|a,s,t)p(t)\,ds'\,dt.$$

Exploration reward:

$$u(s,a) = \mathrm{JSD}\{\mathcal{P}(\mathcal{S}|s,a,t) \mid t \sim \mathcal{P}(T)\}$$

Proposed method

$$u(s,a) \simeq H\left(\frac{1}{N}\sum_{i=1}^{N}\mathcal{P}(\mathcal{S}|s,a,t_i)\right) - \frac{1}{N}\sum_{i=1}^{N}H(\mathcal{P}(\mathcal{S}|s,a,t_i)).$$

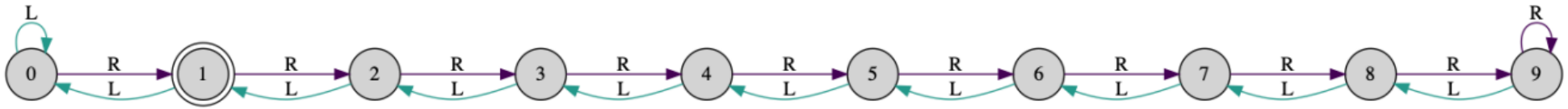
Proposed method

Algorithm 1 MODEL-BASED ACTIVE EXPLORATION

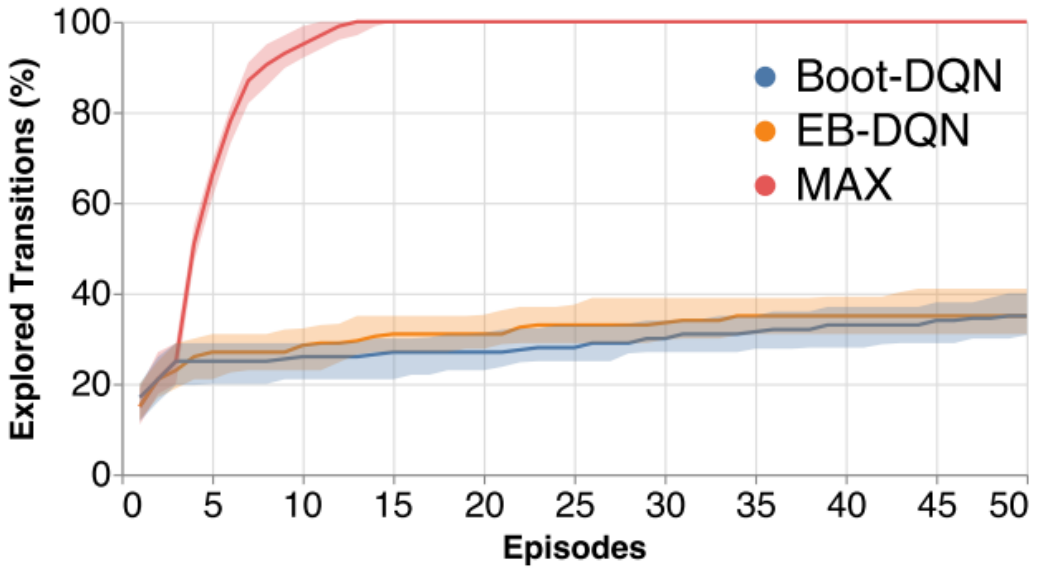
Initialize: Transitions dataset D, with random policy
Initialize: Model ensemble, $\tilde{T} = \{t_1, t_2, \cdots, t_N\}$
repeat
  while episode not complete **do**
    ExplorationMDP $\leftarrow (\mathcal{S}, \mathcal{A}, \text{Uniform}\{\tilde{T}\}, u, \delta(s_\tau))$
    $\pi \leftarrow$ SOLVE(ExplorationMDP)
    $a_\tau \sim \pi(s_\tau)$
    act in environment: $s_{\tau+1} \sim \mathcal{P}(\mathcal{S}|s_\tau, a_\tau, t^*)$
    $D \leftarrow D \cup \{(s_\tau, a_\tau, s_{\tau+1})\}$
    Train t_i on D for each t_i **in** $\tilde{T}$
  end while
until computation budget exhausted

$u + r$

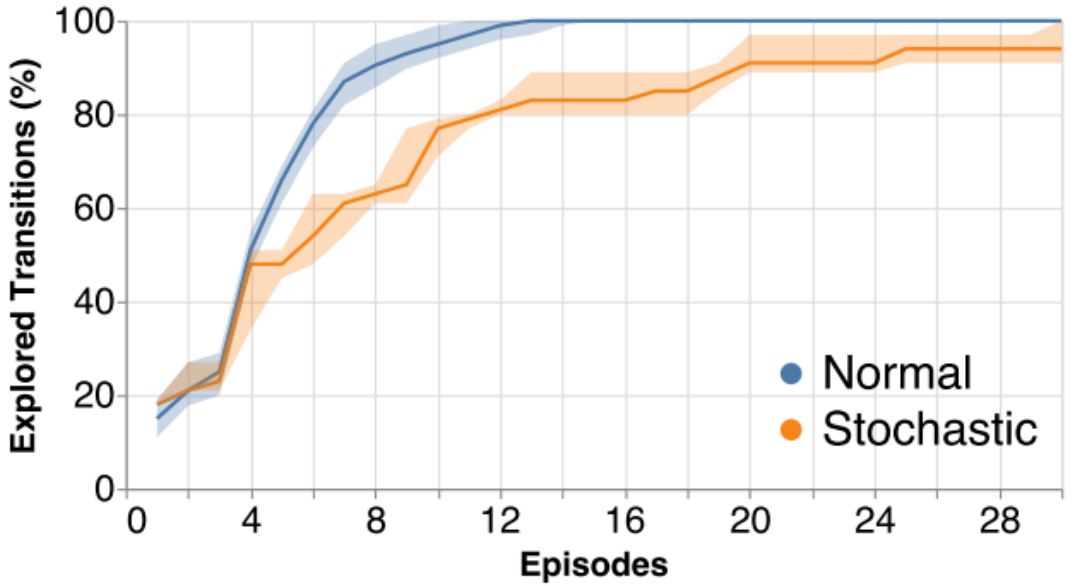
Experiments



(a) Chain environment of length 10.

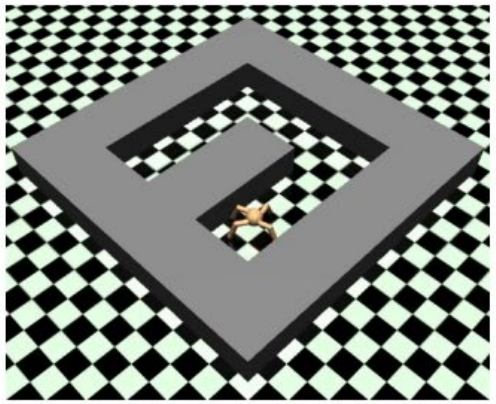


(b) 50-state chain.

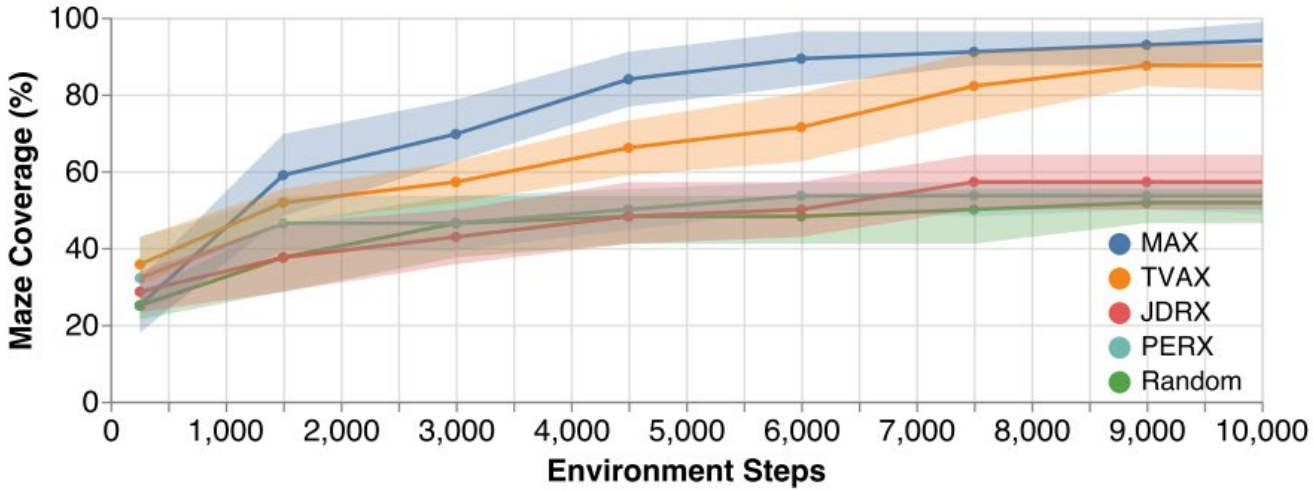


(d) Stochastic trap

Experiments



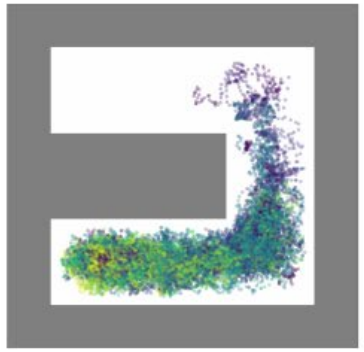
(a) Ant Maze Environment



(b) Maze Exploration Performance

(c) 300 steps

(d) 600 steps

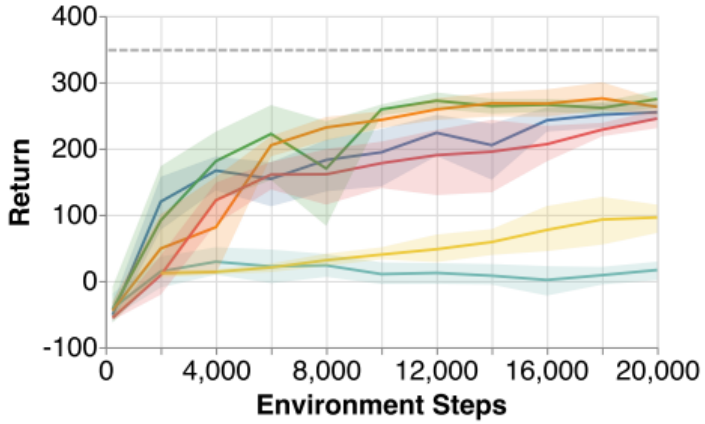


(e) 3600 steps

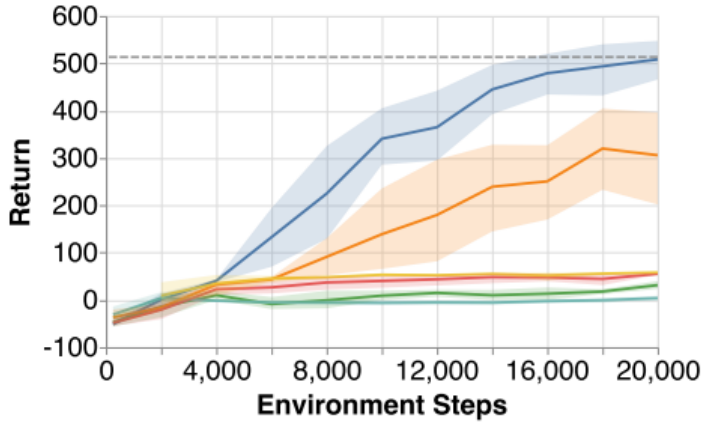
(f) 12000 steps

Experiments

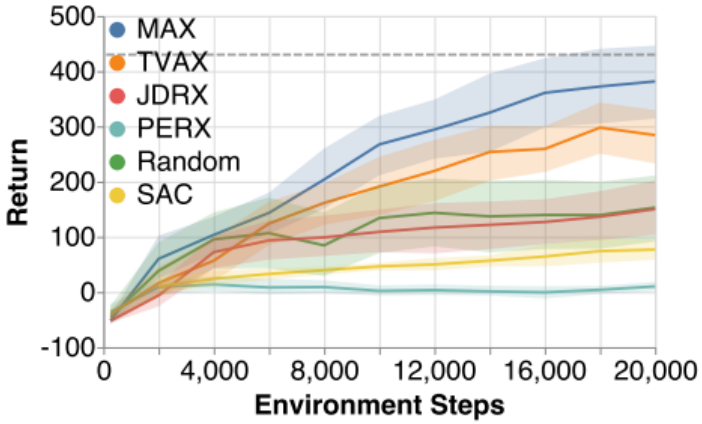
Soft-Actor Critic (Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor)



(a) Running task performance



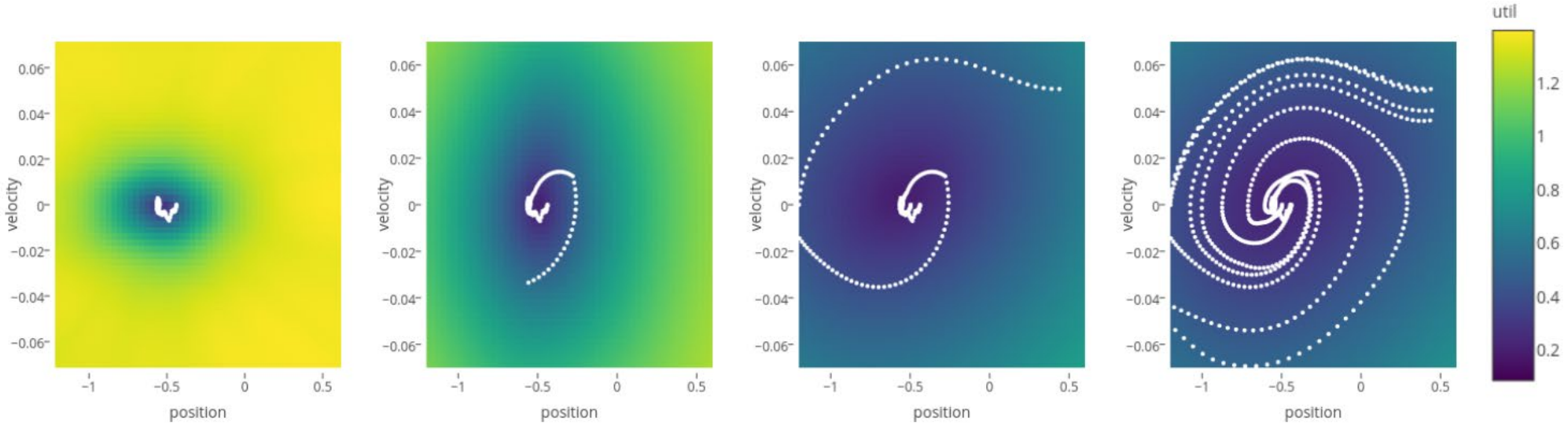
(b) Flipping task performance



(c) Average performance

Experiments

Continuous Mountain Car environment



(a) 100 steps (b) 150 steps (c) 220 steps (d) 800 steps

Appendix

$$u(s,a) = \int_{\mathcal{S}} \mathrm{IG}(s,a,s')p(s'|a,s)\,ds',$$

$$= \int_{\mathcal{S}} D_{\mathrm{KL}}(\mathcal{P}(T|\phi) \parallel \mathcal{P}(T))\, p(s'|s,a)\,ds'$$

$$u(s,a) = \int_{\mathcal{S}} \int_{T} p(t|\phi) \log\left(\frac{p(t|\phi)}{p(t)}\right) p(s'|s,a)\,dt\,ds' \longleftarrow D_{\mathrm{KL}}(\mathcal{P}_1 \parallel \mathcal{P}_2) = \int_{Z} p_1(z) \log\left(\frac{p_1(z)}{p_2(z)}\right) dz$$

$$u(s,a) = \int_{\mathcal{S}} \int_{T} \frac{p(s'|s,a,t)p(t)}{p(s'|s,a)} \log\left(\frac{p(s'|s,a,t)}{p(s'|s,a)}\right) p(s'|s,a)\,dt\,ds'$$

$$= \int_{\mathcal{S}} \int_{T} p(s'|s,a,t)p(t) \log\left(\frac{p(s'|s,a,t)}{p(s'|s,a,\bar{t})}\right) dt\,ds'.$$

$$p(t|s',a,s) = \frac{p(s'|s,a,t)p(t|s,a)}{p(s'|s,a)}$$

$$p(t|s,a) = p(t)$$

$$u(s,a) = \int_{T} \int_{\mathcal{S}} p(s'|s,a,t) \log\left(p(s'|s,a,t)\right)p(t)\,ds'\,dt$$

$$- \int_{\mathcal{S}} \int_{T} p(s'|s,a,t)p(t)\,dt \log\left(\int_{T} p(s'|s,a,t)p(t)\,dt\right) ds'$$