Direct Policy Transfer via Hidden Parameter Markov Decision Processes

Jiayu Yao, Taylor Killian,
George Konidaris, Finale Doshi-Velez

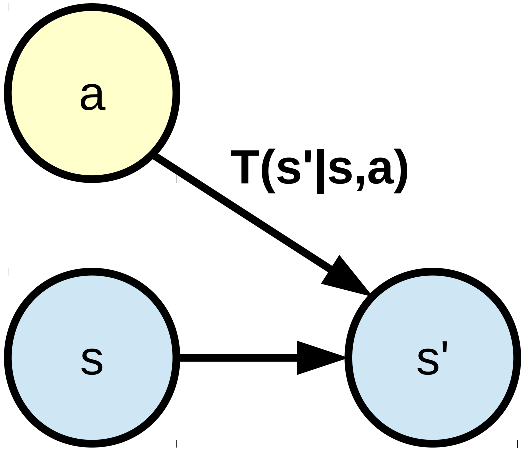
Motivation

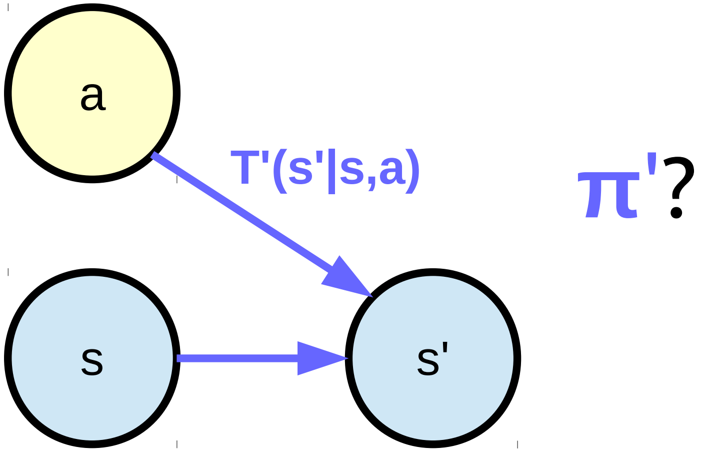
- What if we need to solve a family of related tasks?
  - Picking up objects with different masses/sizes.
  - Driving different vehicles.
  - Treating patients with different physiologies.
- We’ll focus on the situation in which the rewards don’t change but the dynamics change.
- Goal: Still reach near-optimal performance, quickly.

Markov Decision Process

$$(S, A, T, R, \gamma) \rightarrow \pi$$

- S: state space; A: action space
- T(s'|s,a) is the transition model
- R(s,a) is the reward model; **π**(s) is the policy

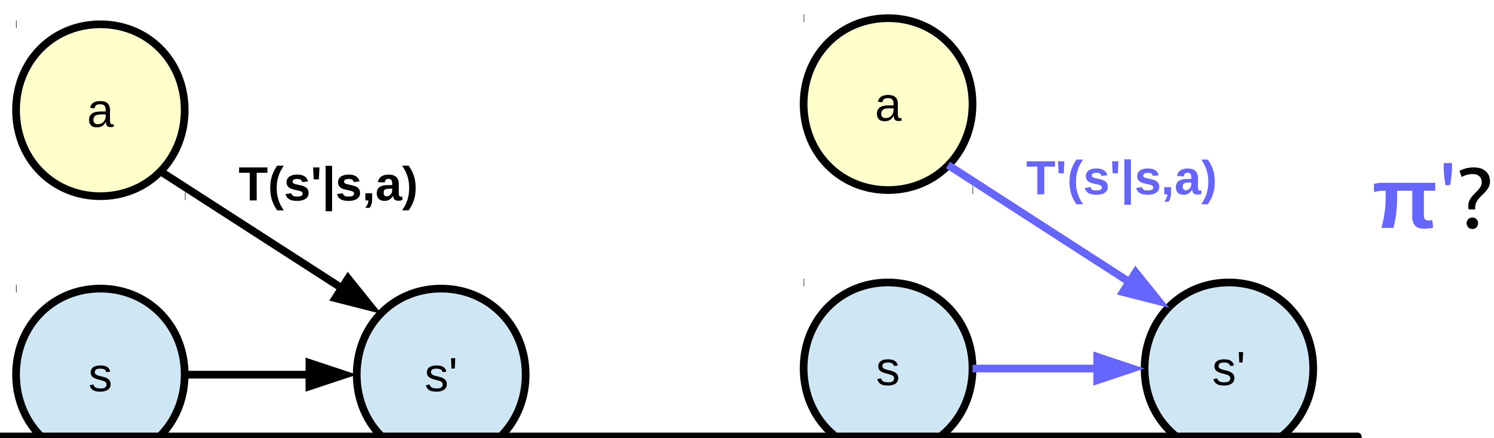




Markov Decision Process

$$(S, A, T, R, \gamma) \rightarrow \pi$$

- S: state space; A: action space
- T(s'|s,a) is the transition model
- R(s,a) is the reward model; **π**(s) is the policy

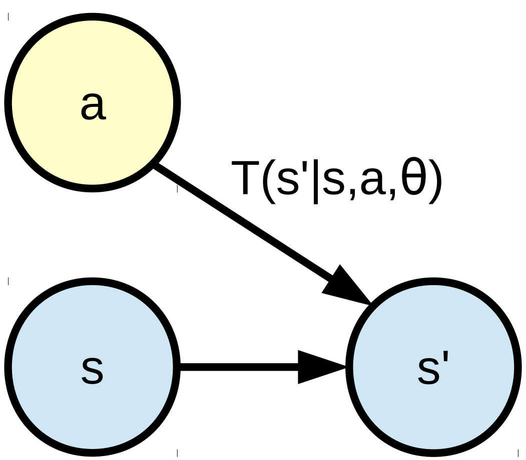


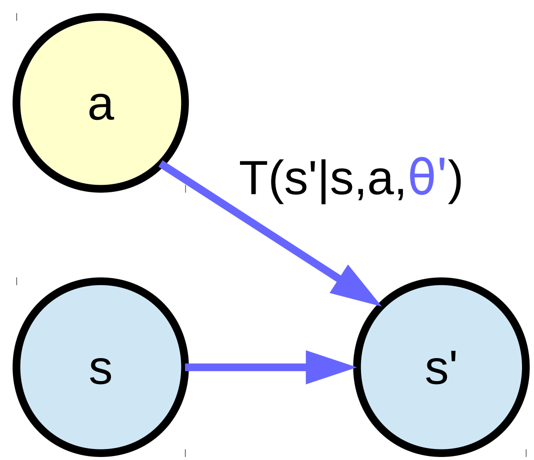
Relationship determines transfer possibilities

HiP-MDPs: Defining related tasks

$$\left(S, A, T_{\theta}, R, \gamma, P_{\theta}\right)$$

- S, A, R as before
- T(s'|s,a,θ) is parameterized by θ
- P_{θ} is the distribution over all possible θ





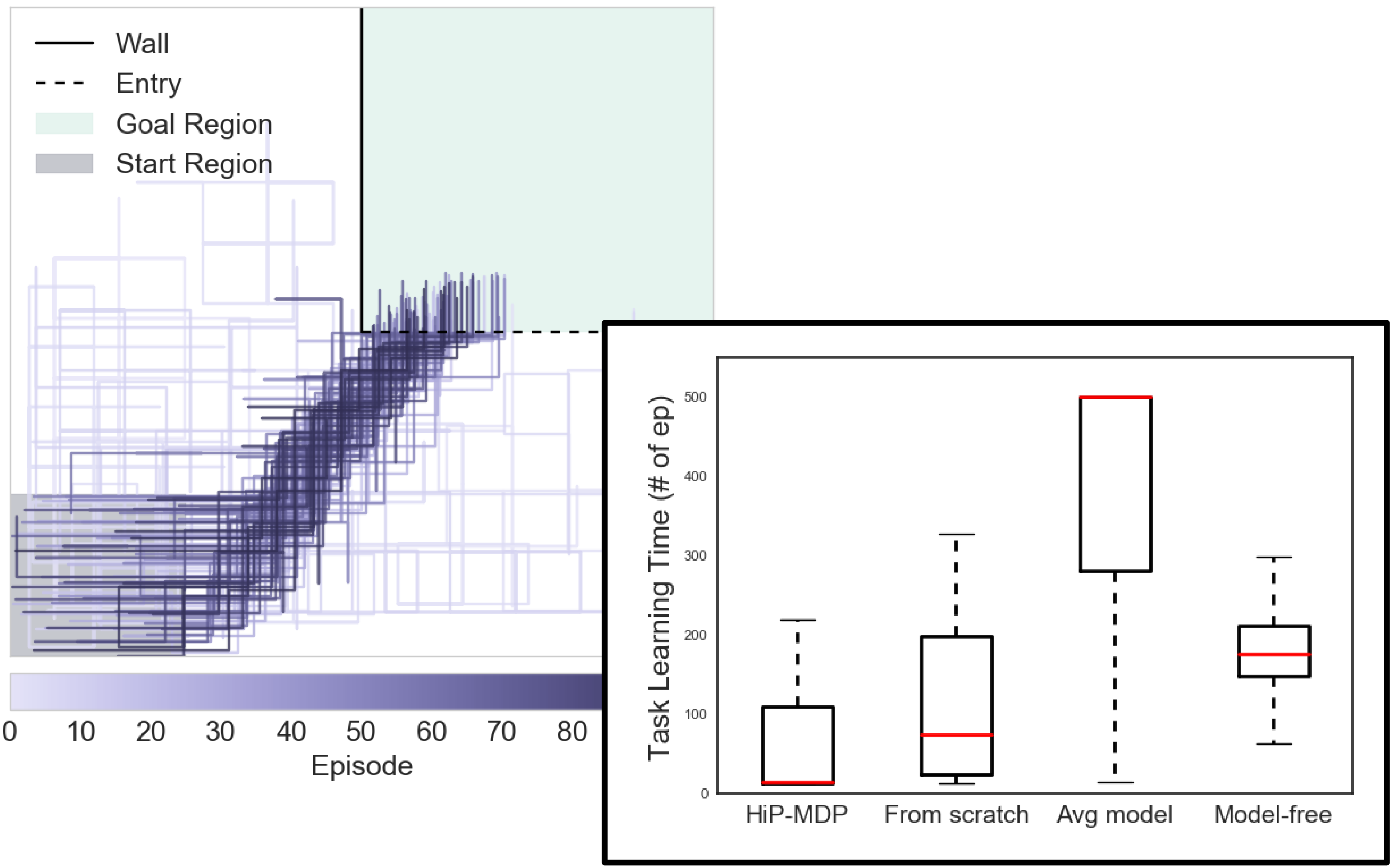
HiP-MDP Approach

- Parameter θ is fixed per task
- Each MDP M_θ is an MDP
- Knowing θ is sufficient for solving the task

Idea: θ is a minimal statistic to characterize the MDP; try to minimize uncertainty in θ and then solve the MDP

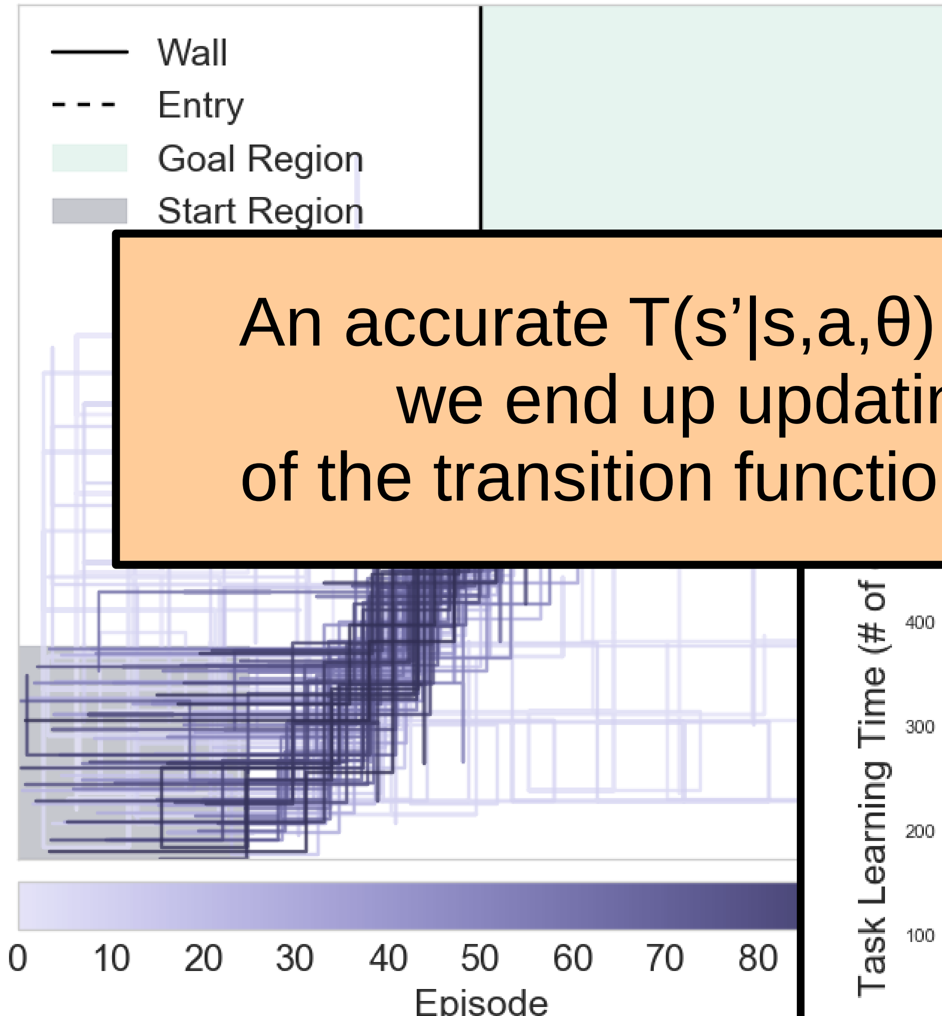
Does it work?

Kinda… Toy Example

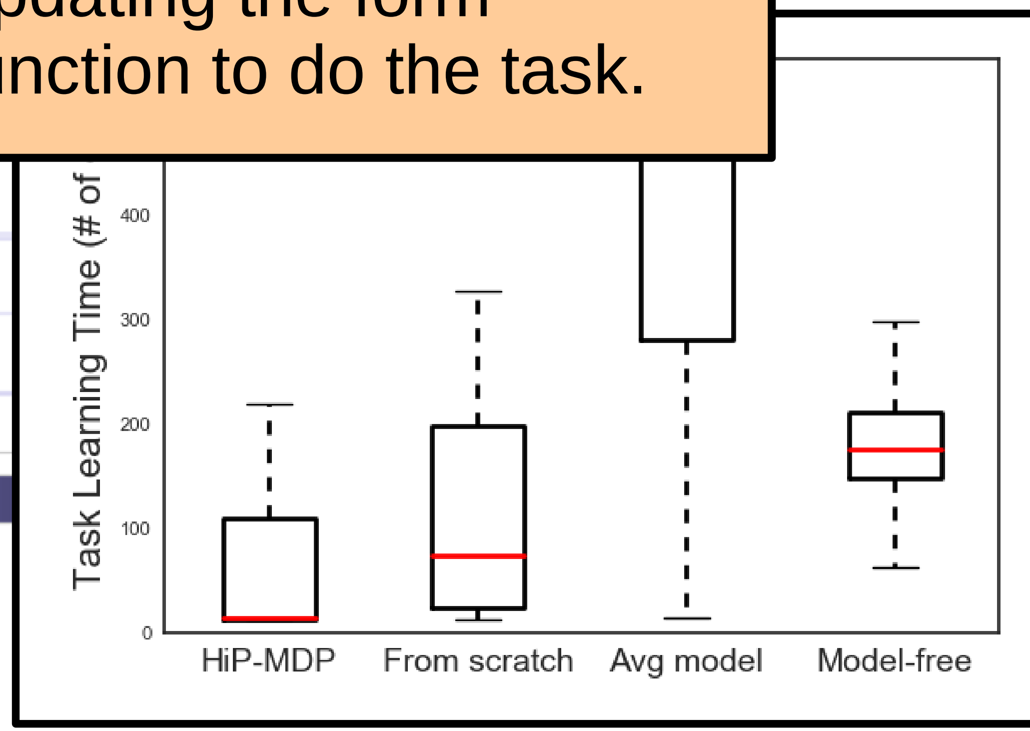


Killian et al. NIPS 2017

Kinda… Toy Example



An accurate T(s’|s,a,θ) is hardto learn;
we end up updating the form
of the transition function to do the task.



Killian et al. NIPS 2017

Our approach: Direct Policy Transfer

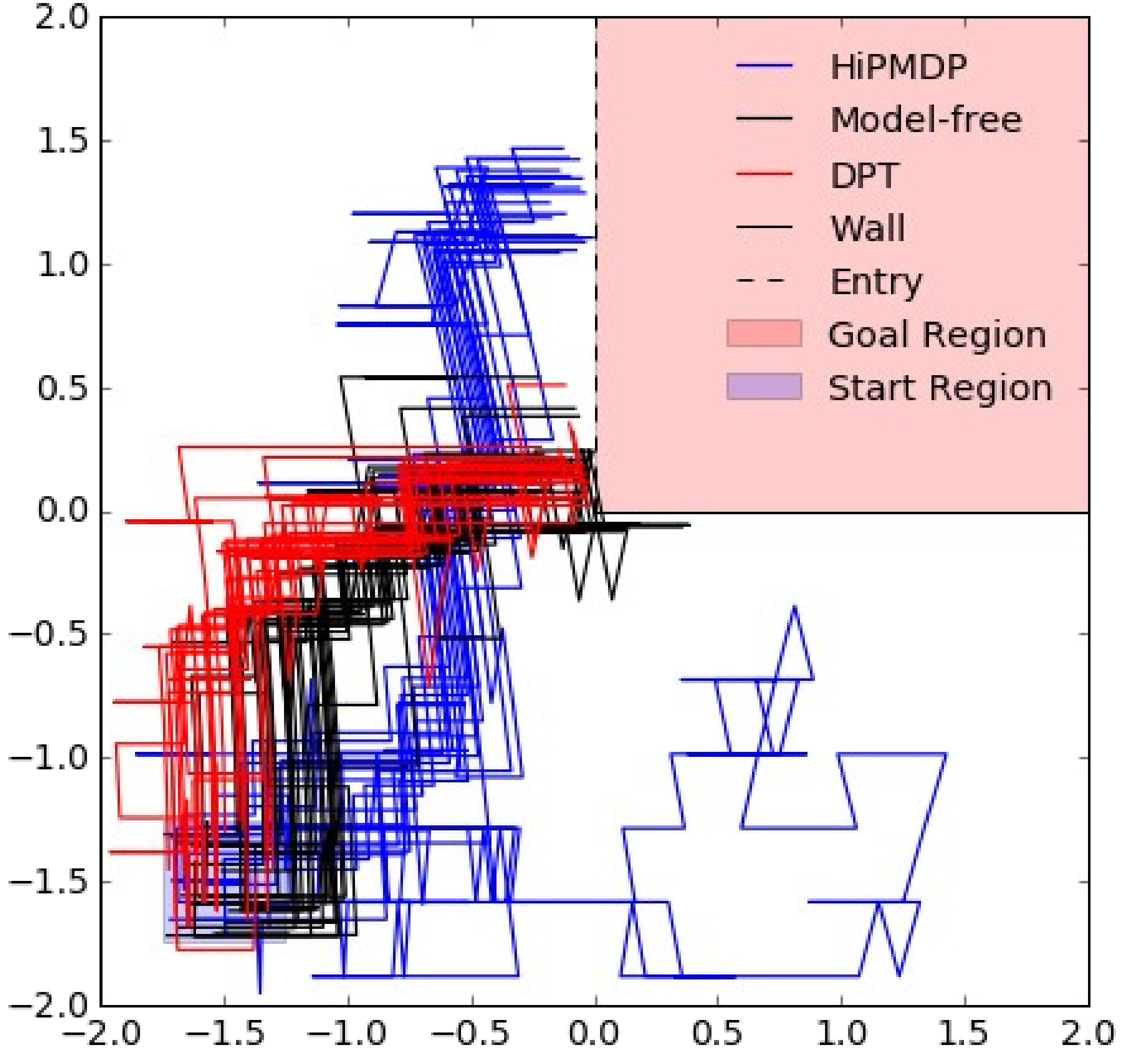
- Assume a batch of available data, with near-optimal policies. (Common in many real scenarios where we have observational data.)
- Use the batch to learn the functional form of $T(s'|s,a,\theta)$ and P_θ ; solve for each θ. Learn a form for the policy $\pi(a|s,\theta)$.
- Given interactions from a new instance, quickly identify θ; then follow the policy $\pi(a|s,\theta)$.

Our approach: Direct Policy Transfer

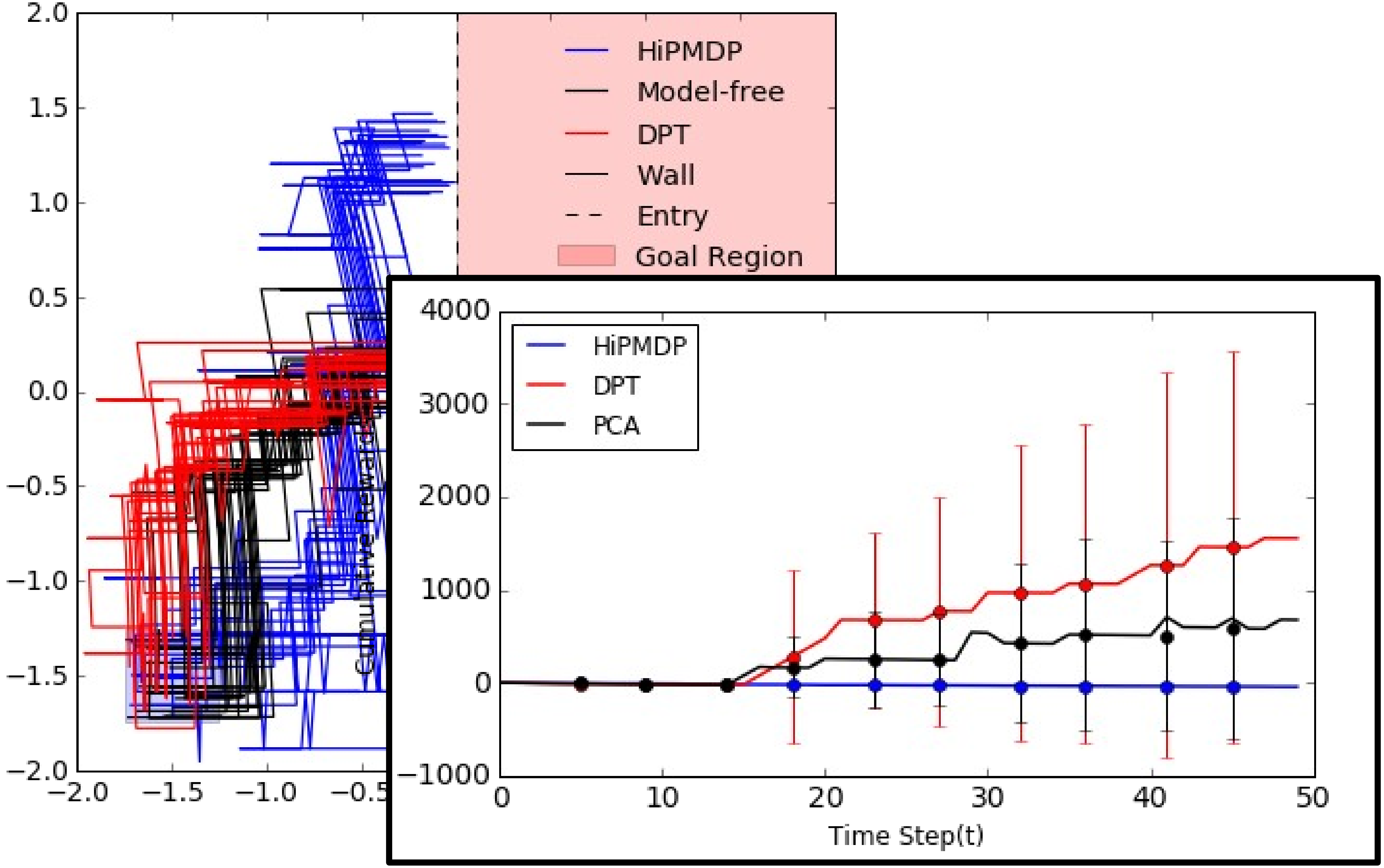
- Assume a batch of available data, with near-optimal policies. (Common in many real scenarios where we have observational data.)
- Use the batch to learn the functional form of $T(s'|s,a,\theta)$ and P_θ ; solve for each θ. Learn a form for the policy $\pi(a|s,\theta)$.
- Given interactions from a new instance, quickly identify θ; then follow the policy $\pi(a|s,\theta)$.

Hypothesis: θ may not be sufficient for planning but may be sufficient to key a near-optimal policy.

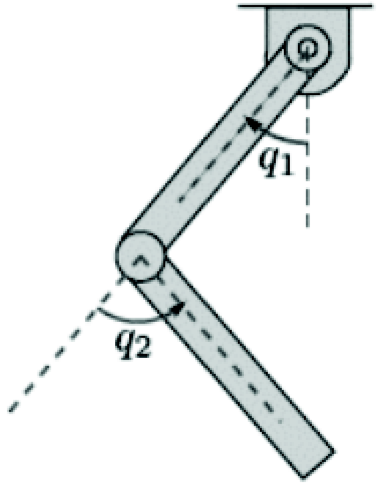
Toy Example, One Episode



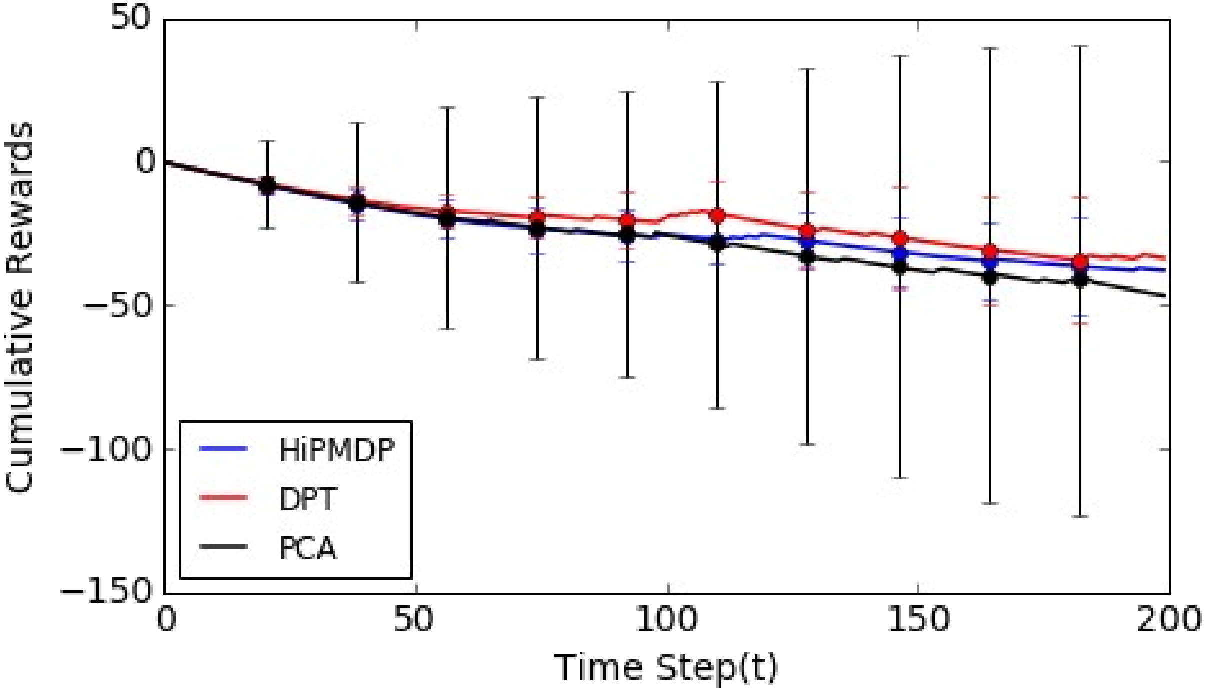
Toy Example, One Episode



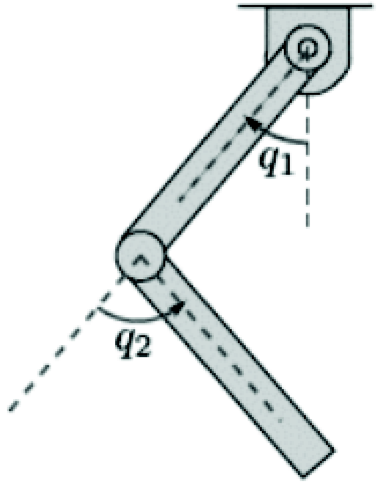
Acrobot



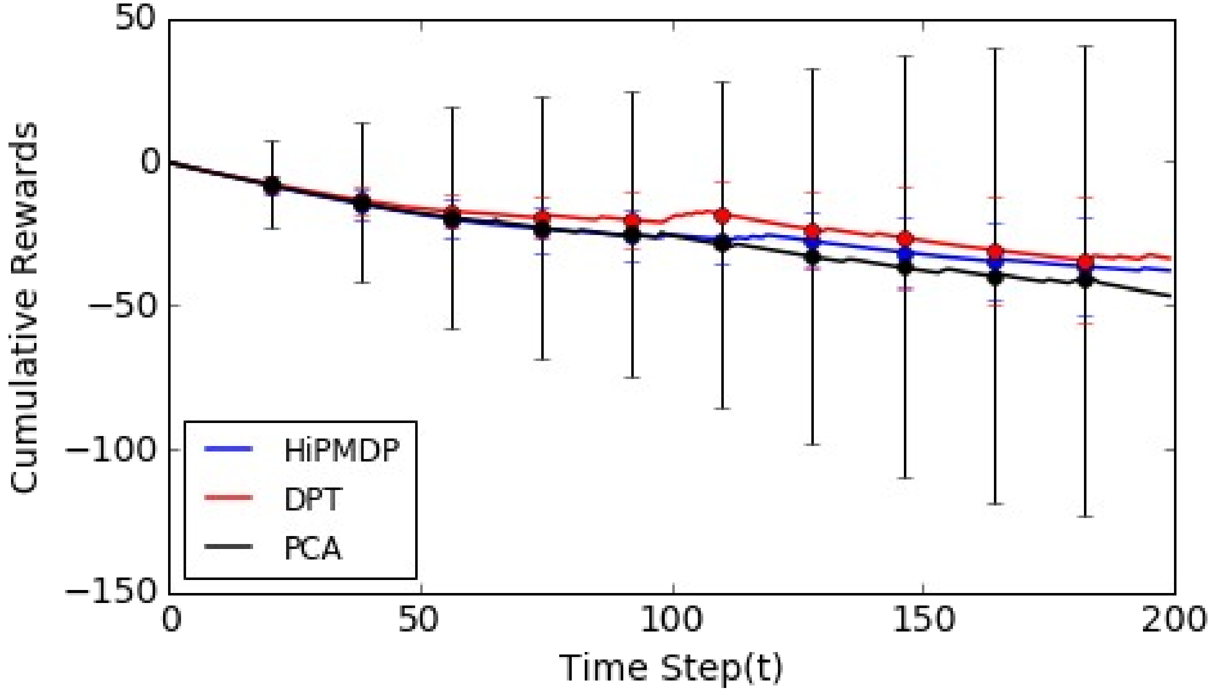
Goal: Swing up
Action: torque@1
Varied: masses



Acrobot



Goal: Swing up
Action: torque@1
Varied: masses



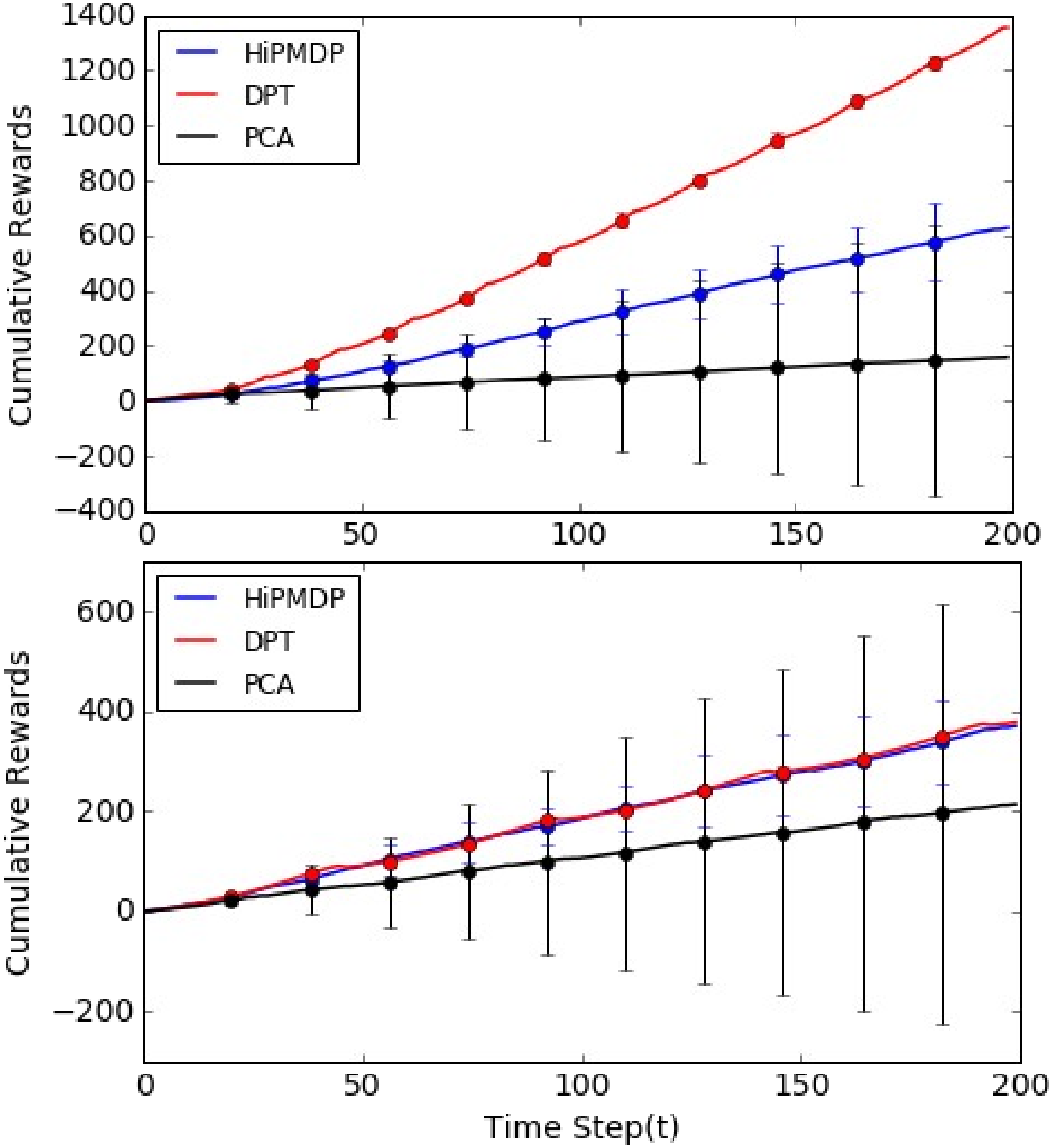
Note: Even if the policy has similar performance, much faster at test time! Only requires solving for θ! (In our experiments, at least 10x faster.)

HIV Simulator

- Take the HIV simulator from Adams et al (2004), used in Earnst et al. (2005) – only two drugs, six measured variables.
- Each patient now has a different dynamical system model.
- Goal: given several patients, quickly learn a model for a new patient.

HIV Simulator

Examples from two different test patients



Summary

- Working toward faster adaptation to new but similar dynamics.
- Currently: Use the dynamics to create a statistic of the problem; use the statistic to key a policy.
- Future work: Reducing constraints on the observational data (optimal policies available), more robust learning.