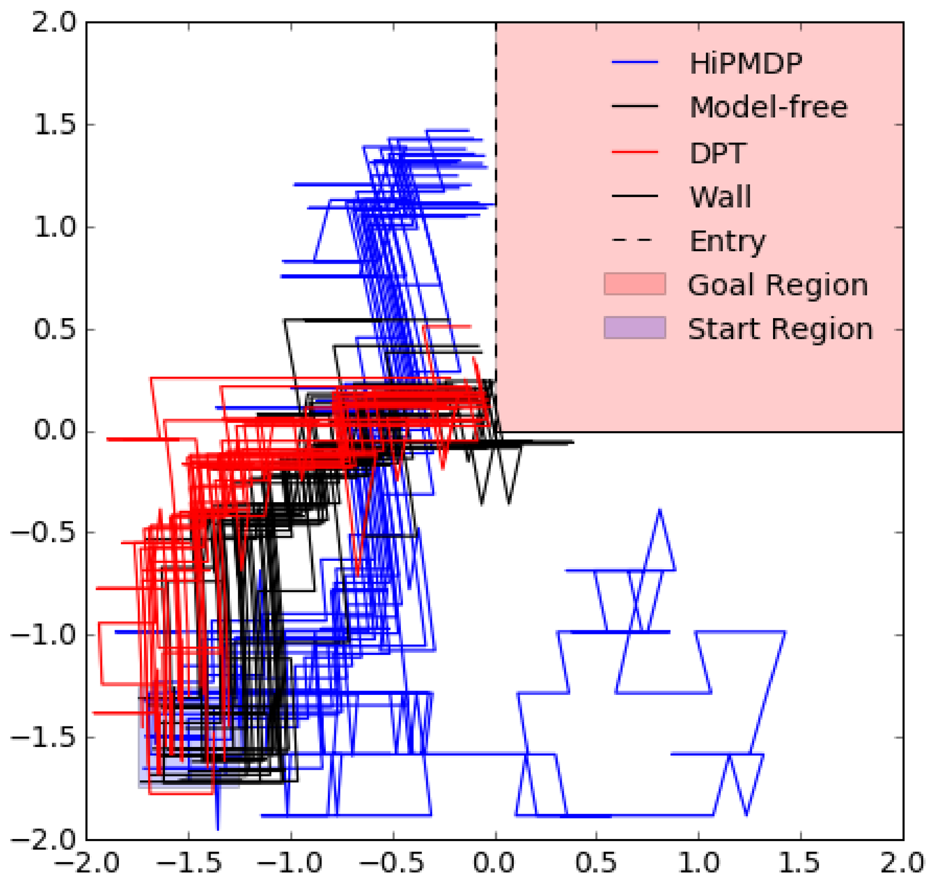
Direct Policy Transfer via Hidden Parameter Markov Decision Processes

Jiayu Yao[1], Taylor Killian [1,2], George Konidaris[3], Finale Doshi-Velez[1]

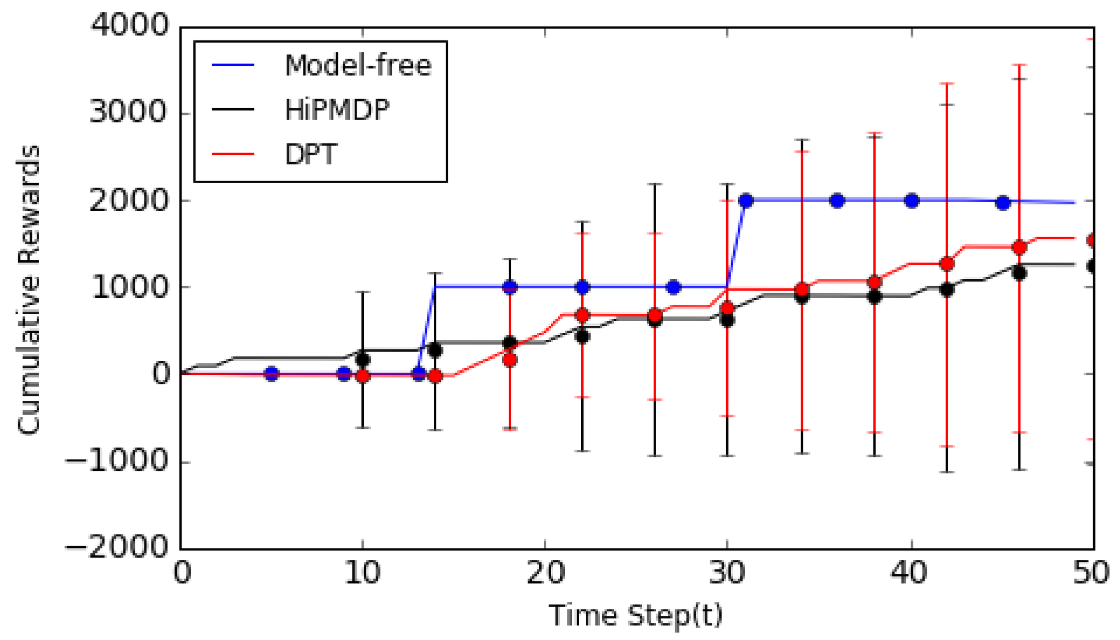
[1]School of Engineering and Applied Sciences, Harvard University,[2] MIT Lincoln Laboratory, [3] Department of Computer Science, Brown University

Introduction

Problem Many applications involve learning from a series of tasks with similar dynamics.
Prior Work The recently-introduced HiP-MDP addresses such situations by characterizing the variation in these dynamics with a few hidden parameters.
Limitation The approach is computationally inefficient since it still needs to train a DDQN. And it requires the estimated transition dynamics to be fairly accurate.
Our work We use these *model-based* parameters for *direct* policy transfer. Given a batch of training tasks, we demonstrate that this direct policy approach requires significantly less samples and computation to learn a policy for a new task.



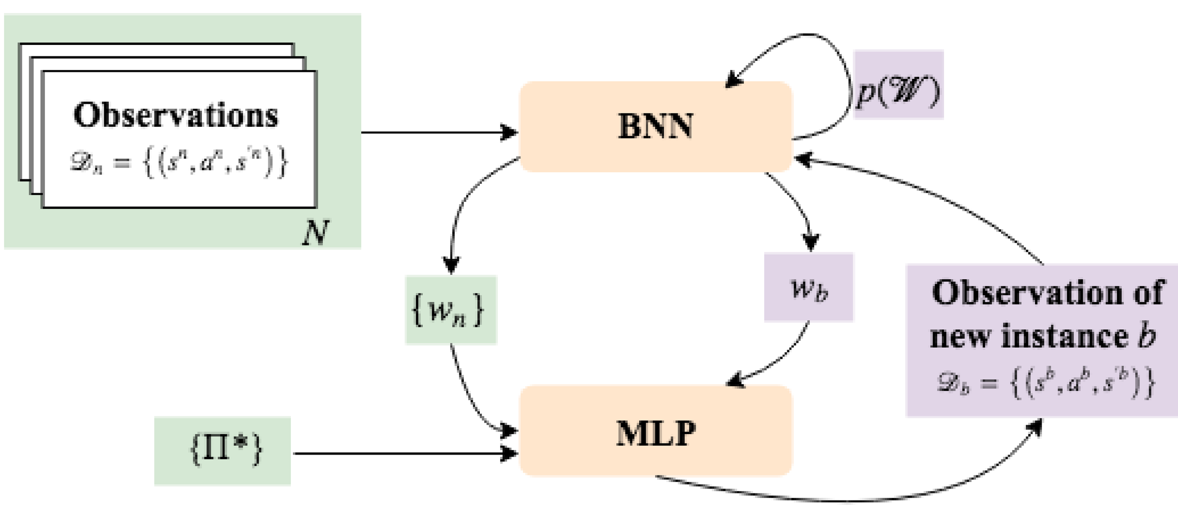
(a) A comparison of epsilon greedy policies π_{DDQN}, π_{HiPMDP}, π_{DPT} ($\epsilon = 0.15$)



(b) A comparison of cumulative rewards of multiple runs following the three policies

Figure 1: Demonstration

Model



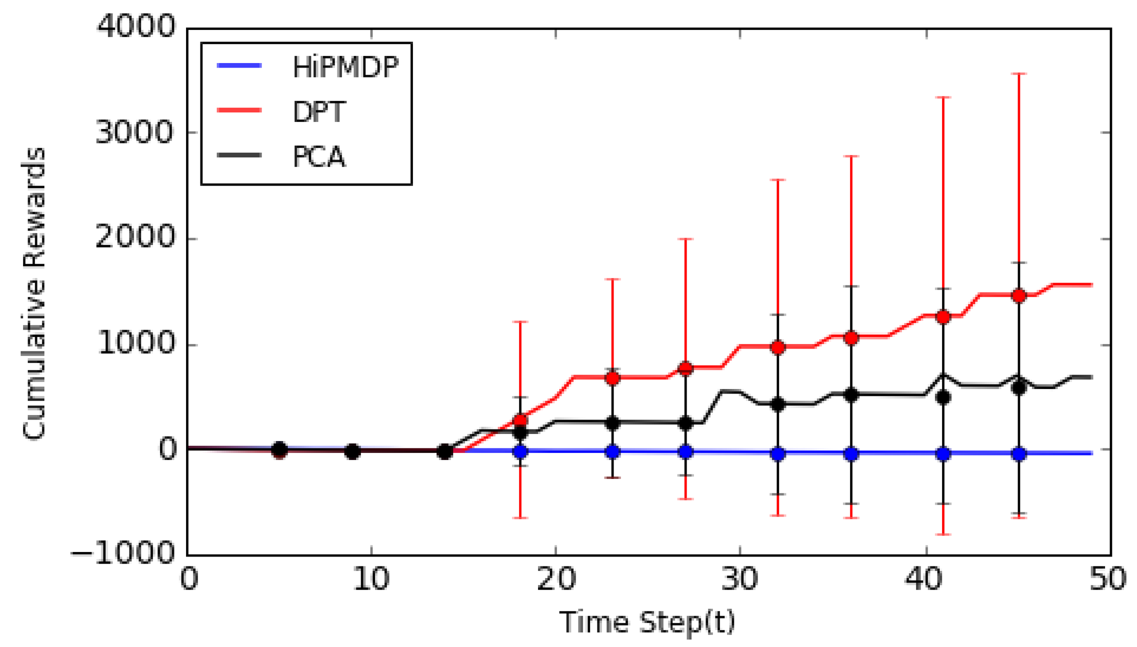
Training Phase

1. Collect initial observations $\mathcal{D} = \{\mathcal{D}_n\}_{n=1}^N$
2. Estimate the transition function and latent variables by iterativiely updating $p(\mathcal{W}|s^n, a^n, s'^n, w_n) \approx \Pi_i q(w_i)$ and $\hat{w}_n^{MLE}$
3. Learn a general policy $\pi(s, w_n; \mathcal{Y})$ by training a MLP to predict $a^* = \pi_n(s_n)$

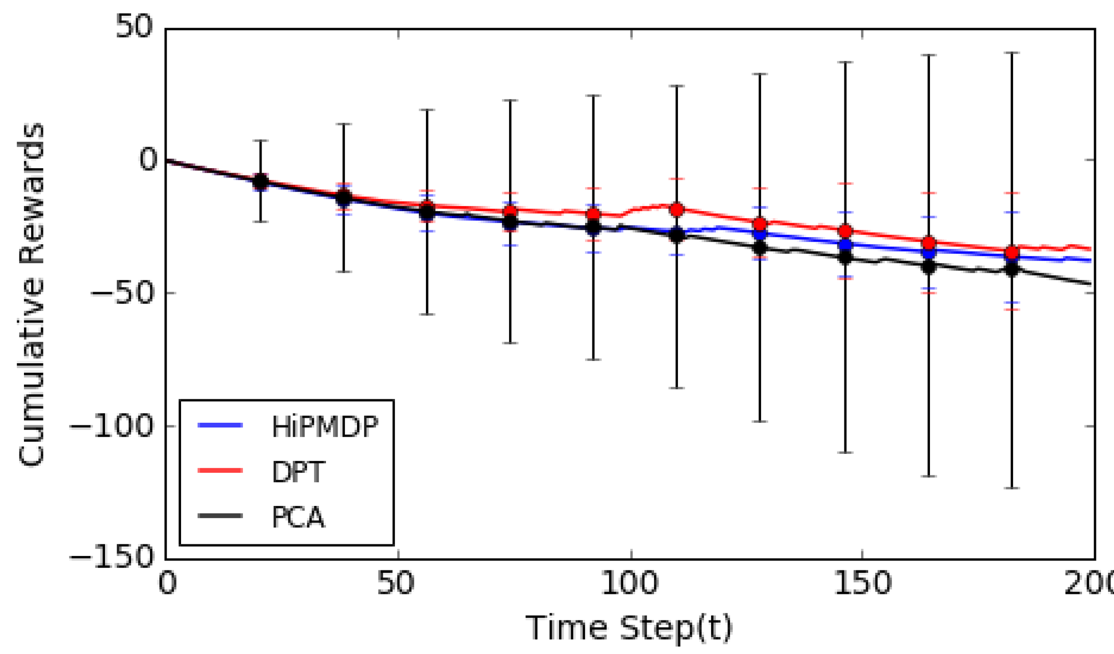
Testing Phase

1. Initialize $w_b = E[w_n]$
2. Generate transitions $\mathcal{D}_b$ with $\pi(s, w_b; \mathcal{Y})$
3. Update w_b with $\mathcal{D}_b$ by minimizing α-divergence of $q(\mathcal{W})$ and $p(\mathcal{W}|s^n, a^n, s'^n, w_n)$
4. Repeat step 2 until π stabilizes

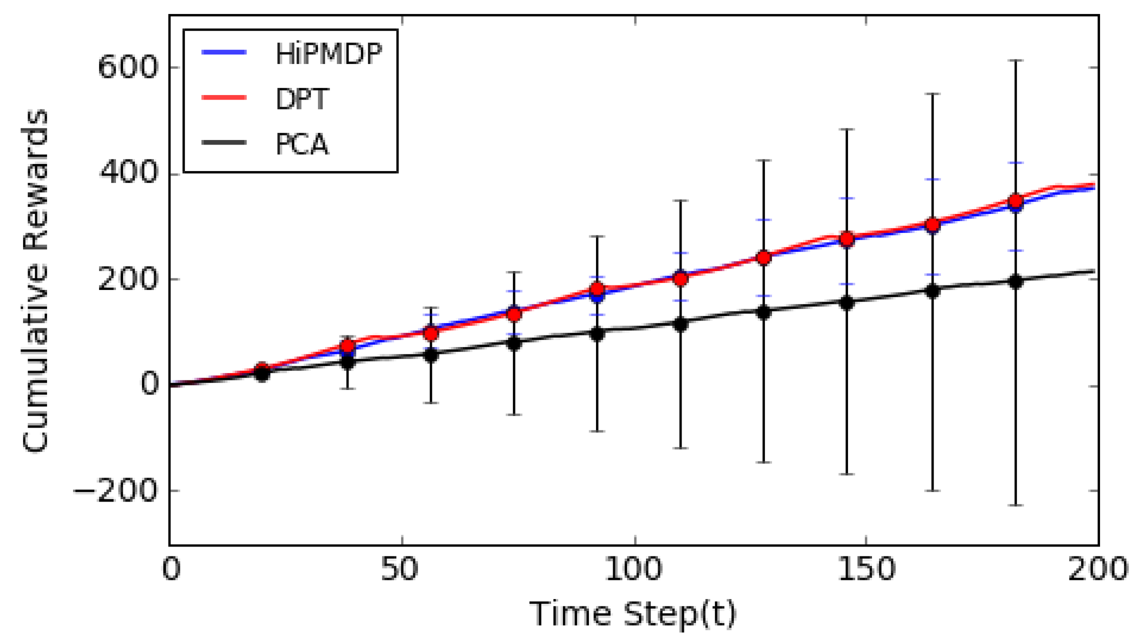
Results



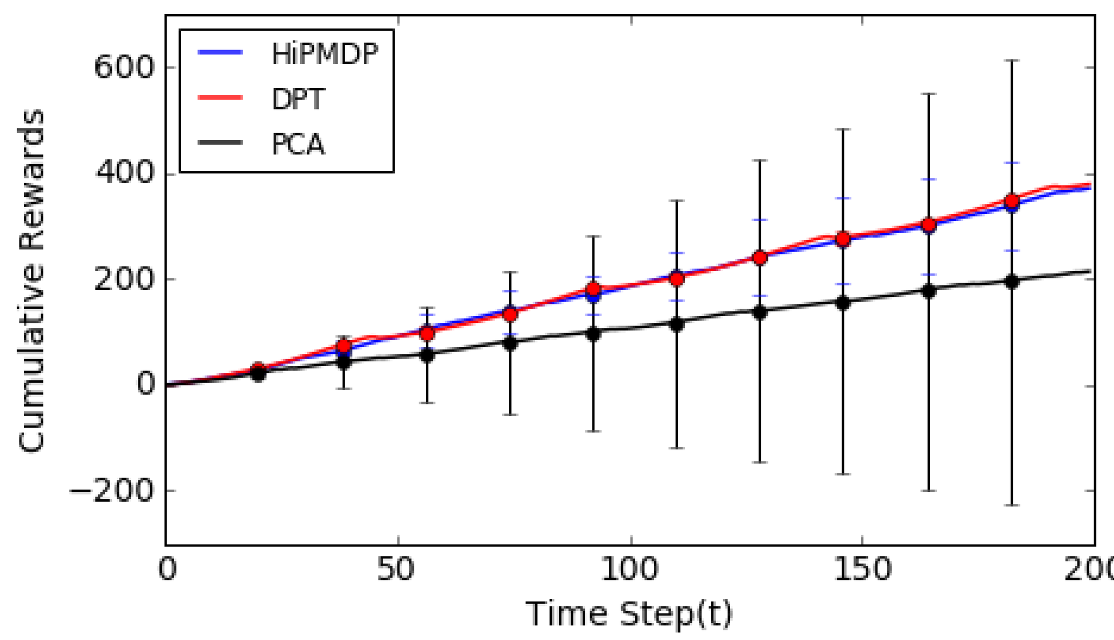
(a) 2D Navigation



(b) Acrobot



(c) HIV Test Patient 1



(d) HIV Test Patient 2

Figure 2: Cumulative rewards achieved throughout the initial episode of a newly encountered instance. The PCA baseline also uses a latent-representation to parametrize the policy, learned through a dimensionality reduction of the transition statistics. Denote the transition statistics of observed instances as Φ_N. Decompose $\Phi_N = U_\Phi S_\Phi V_\Phi^\intercal$. Then $w_b = \phi_b \cdot V_\Phi$

	Computation Time			Cumulative Rewards			
	2D Nav	Acrobot	HIV	2D Nav	Acrobot	HIV	
PCA	**17.4s±0.52**	**56.3s±1.49**	**180.6s±4.43**	317.9±207.8	-42.7±38.89	100.8±12.8	207.8±1.53
HiPMDP	1.0×10^4s	1.9×10^4s	1.0×10^4s	809.9±35	-30.8±33.2	726.7±59.8	**580.0±21.9**
DPT	1.1×10^3s	1.2×10^3s	1.2×10^3s	**891.9±319**	**-27.7±49.5**	**1425.0±5.6**	562.2±4.2

Table 1: Experimental results where DPT is evaluated against HiP-MDP and PCA baseline

Conclusions

- The latent variable is **sufficient** to capture differences in the dynamics of an environment and can be used to parametrize policy diretly
- The DPT approach is **computationally-efficient** and generates **better** policies than HiP-MDP and PCA baselines
- For safety-critical applications, such as healthcare, a rough transition model and generally optimal policy, may provide a way to **safe-guard** against truly poor actions