Robust and Efficient Transfer Learning with Hidden Parameter Markov Decision Processes

Taylor Killian* [1] Samuel Daulton* [1]

George Konidaris [2] Finale Doshi-Velez [1]

NIPS 2017 | Long Beach, CA
6 December 2017

[1] HARVARD
John A. Paulson
School of Engineering
and Applied Sciences

[2] BROWN

Motivation

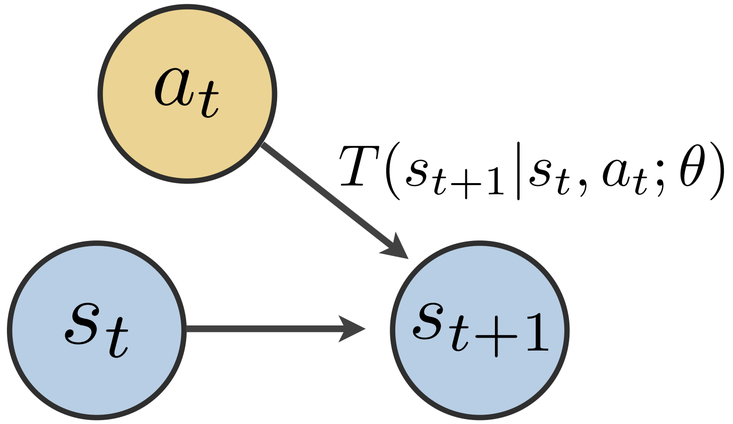
Real-world tasks are often repeated—but not exactly

Variations in physical interactions often require subtle, yet important, adjustments in order to successfully complete unique instances of the same task

Markov Decision Processes (MDP)

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

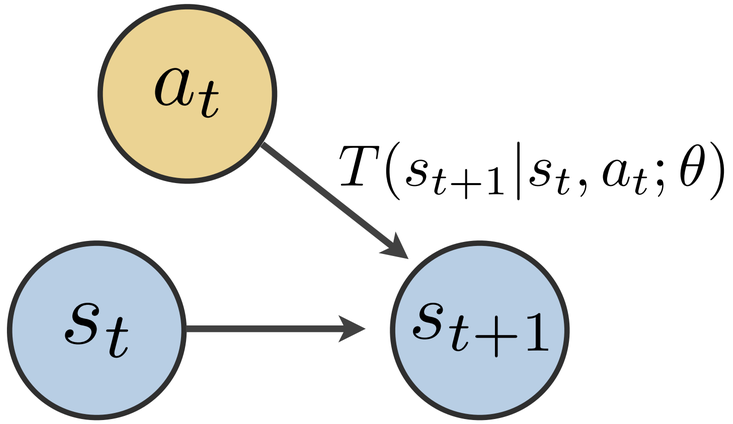
- S: state space; A: action space
- $T(s_{t+1}|s_t, a_t; \theta)$ is the transition model
- $R(s_t, a_t)$ is the reward model with discount factor γ
- $\pi(s_t) \rightarrow a_t$ is the policy mapping states to actions

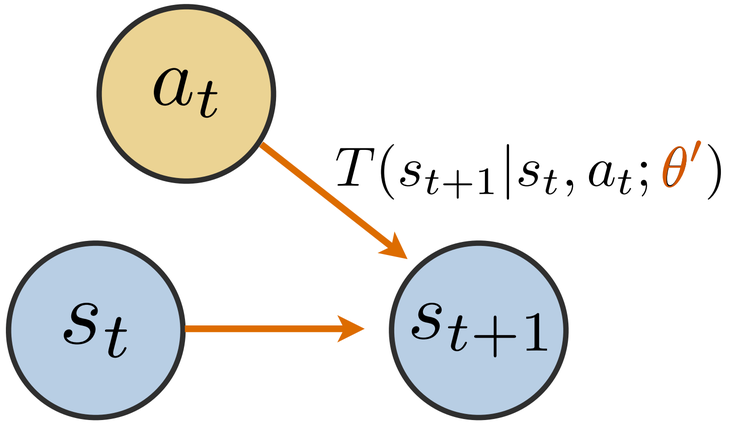


Markov Decision Processes (MDP)

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

- S: state space; A: action space
- $T(s_{t+1}|s_t, a_t; \theta)$ is the transition model
- $R(s_t, a_t)$ is the reward model with discount factor γ
- $\pi(s_t) \rightarrow a_t$ is the policy mapping states to actions



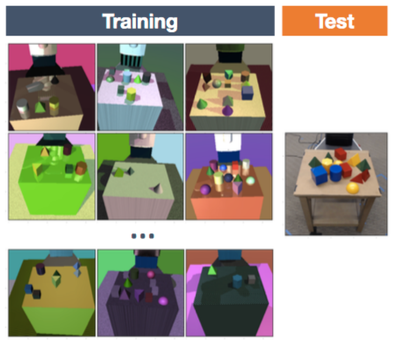


Learning Across Related MDPs

The objective of learning optimal control policies across related MDPs introduces an intriguing application of transfer learning

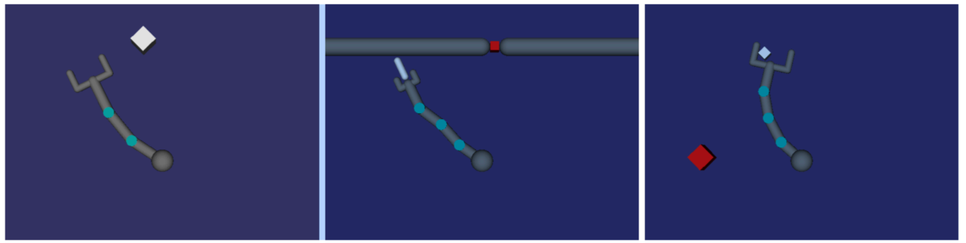
Environment Randomization

[Yahya, et al. 2016]

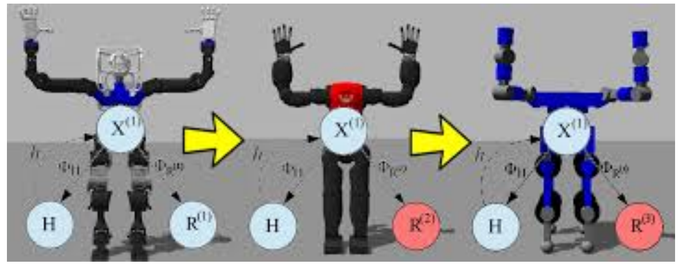


[Tobin, et al. 2017]

Creation of an Invariant Subspace

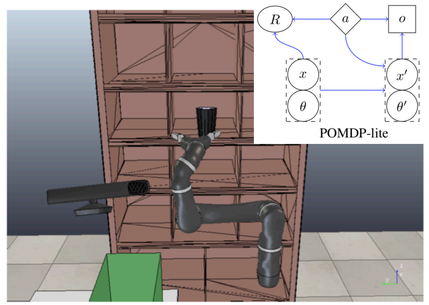


[Gupta, et al. 2017]

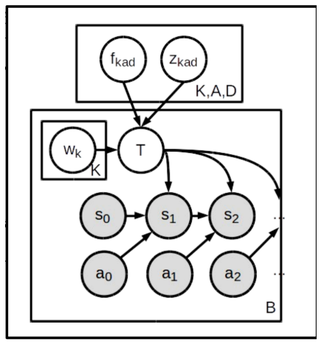


[Delhaisse, et al. 2017]

Latent Variable Modeling



[Chen, et al. 2016]

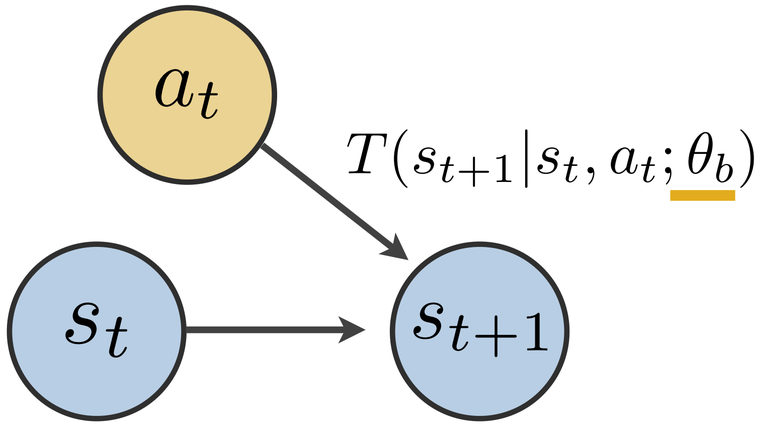


[Doshi-Velez and Konidaris 2016]

Hidden Parameter Markov Decision Processes (HiP-MDP)

Introduced by Doshi-Velez and Konidaris (2016) to account for related, yet distinct, MDPs when learning control policies

- Hidden parameters θ_b estimated by latent, low-dimensional representation w_b
  - θ_b is fixed per task instance and fully parameterizes the task



Hidden Parameter Markov Decision Processes (HiP-MDP)

Introduced by Doshi-Velez and Konidaris (2016) to account for related, yet distinct, MDPs when learning control policies

- Transition dynamics are approximated by a linear combination of Gaussian Processes
  - The parameters w_b are used as weights
- Limitations of this model choice:
  - Cannot accurately approximate nonlinear dynamics
  - No interaction between state and latent weights
  - Concerns about scalability due to GP bases

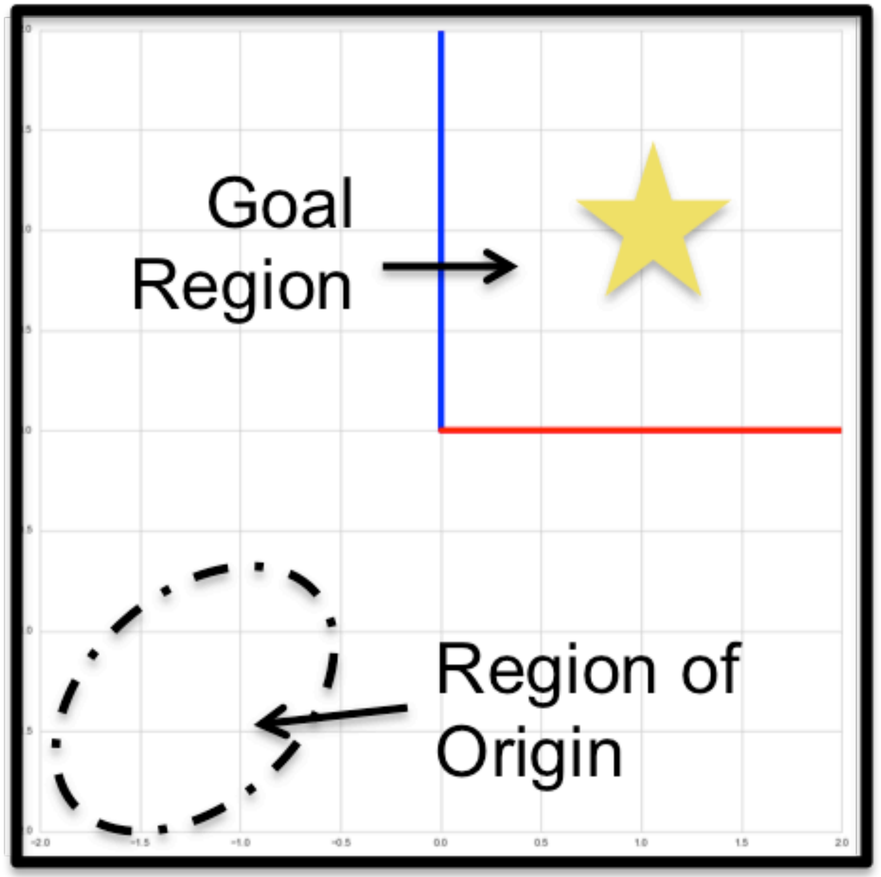
$$s_{t+1}^d \approx \sum_{k=1}^{K} w_{kb}\ \hat{T}_{kad}(s_t) + \epsilon$$

$$w_{kb} \sim \mathcal{N}(\mu_{w_k}, \sigma_w^2)$$

$$\epsilon \sim \mathcal{N}(0, \sigma_{nad}^2)$$

Evaluating the HiP-MDP

A Simple Toy Domain



$S : [-2, 2]^2 \subset \mathbb{R}^2$

$A : \leftarrow, \rightarrow, \uparrow, \downarrow$

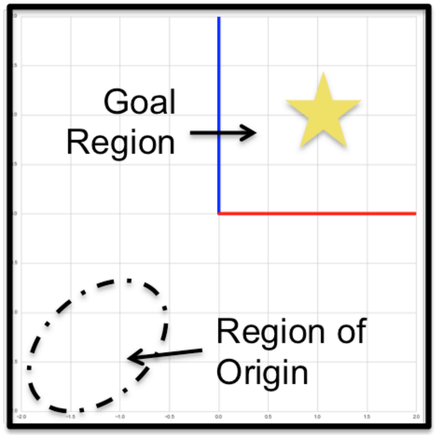
with randomized step size

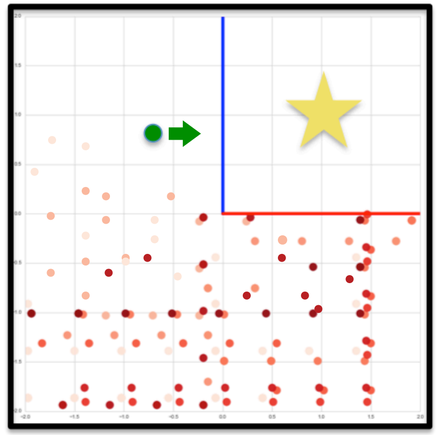
$$R(s,a) = \begin{cases} +++, & \text{if in goal region} \\ --, & \text{if run into wall} \\ -, & \text{otherwise} \end{cases}$$

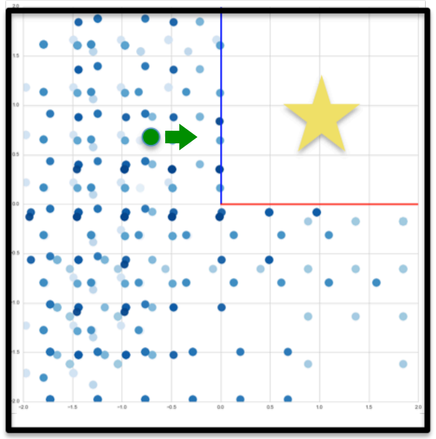
w_b : Numerical estimation of dynamics present between blue / red instances

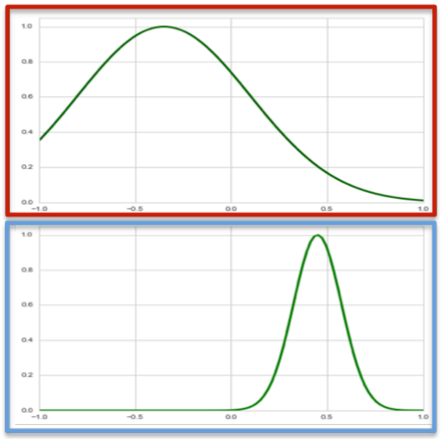
Evaluating the HiP-MDP

Limitations of Original HiP-MDP









$$s^d_{t+1} \approx \sum_{k=1}^{K} w_{kb}\ \hat{T}_{kad}(s_t) + \epsilon$$

$$w_{kb} \sim \mathcal{N}(\mu_{w_k}, \sigma^2_w)$$

$$\epsilon \sim \mathcal{N}(0, \sigma^2_{nad})$$

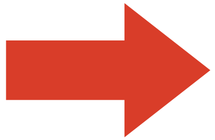
- Learning the w_b requires that observations from separate task instances needed to overlap to differentiate between the observed dynamics
  - While reasonable in some domains (e.g. robotics), it is not feasible in more complex settings (e.g. human patients)

Reformulating the HiP-MDP

$$s^d_{t+1} \approx \sum_{k=1}^{K} w_{kb}\ \hat{T}_{kad}(s_t) + \epsilon$$

$$w_{kb} \sim \mathcal{N}(\mu_{w_k}, \sigma^2_w)$$

$$\epsilon \sim \mathcal{N}(0, \sigma^2_{nad})$$



$$s_{t+1} \approx \hat{T}(s_t, a_t, w_b) + \epsilon$$

$$w_b \sim \mathcal{N}(\mu_w, \Sigma_b)$$

$$\epsilon \sim \mathcal{N}(0, \sigma^2_n)$$

By embedding the parameters w_b with the input to the transition function, we allow for direct interaction between the state and the latent dynamics encoded in the w_b

Reformulating the HiP-MDP

Selecting a Transition Model

- What we want:
  - More efficiency and scalability
  - Nonlinear interactions between latent weights and state
  - Retain probabilistic measure of model certainty

$$s_{t+1} \approx \hat{T}(s_t, a_t, w_b) + \epsilon$$
$$w_b \sim \mathcal{N}(\mu_w, \Sigma_b)$$
$$\epsilon \sim \mathcal{N}(0, \sigma_n^2)$$

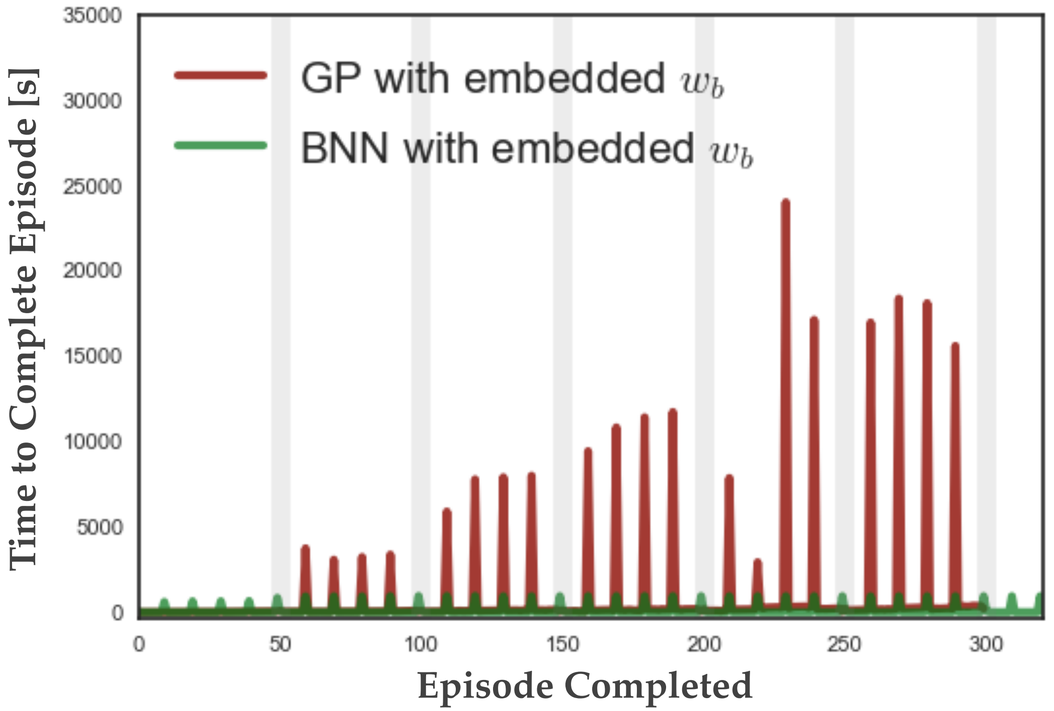
To satisfy the desired performance requirements of our reformulation of the HiP-MDP, we replace the GP basis functions with a Bayesian Neural Network, trained using α-divergence minimization†

- Naturally guarantees interaction between latent weights and state transitions
- Provides opportunity for direct transfer via online computation vs retaining/caching data
- More readily scalable to accommodate higher volumes of data and more complex transition dynamics

† Hernández-Lobato, et al. (2016, *ICML*)

Reformulating the HiP-MDP

Selecting a Transition Model

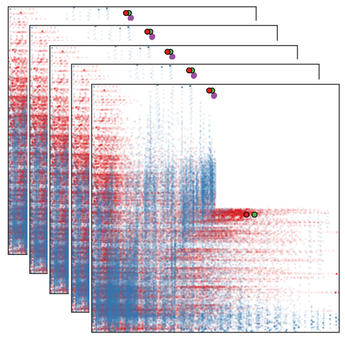


Comparing GP and BNN approaches for $\hat{T}$, both with embedded latent parameters w_b:

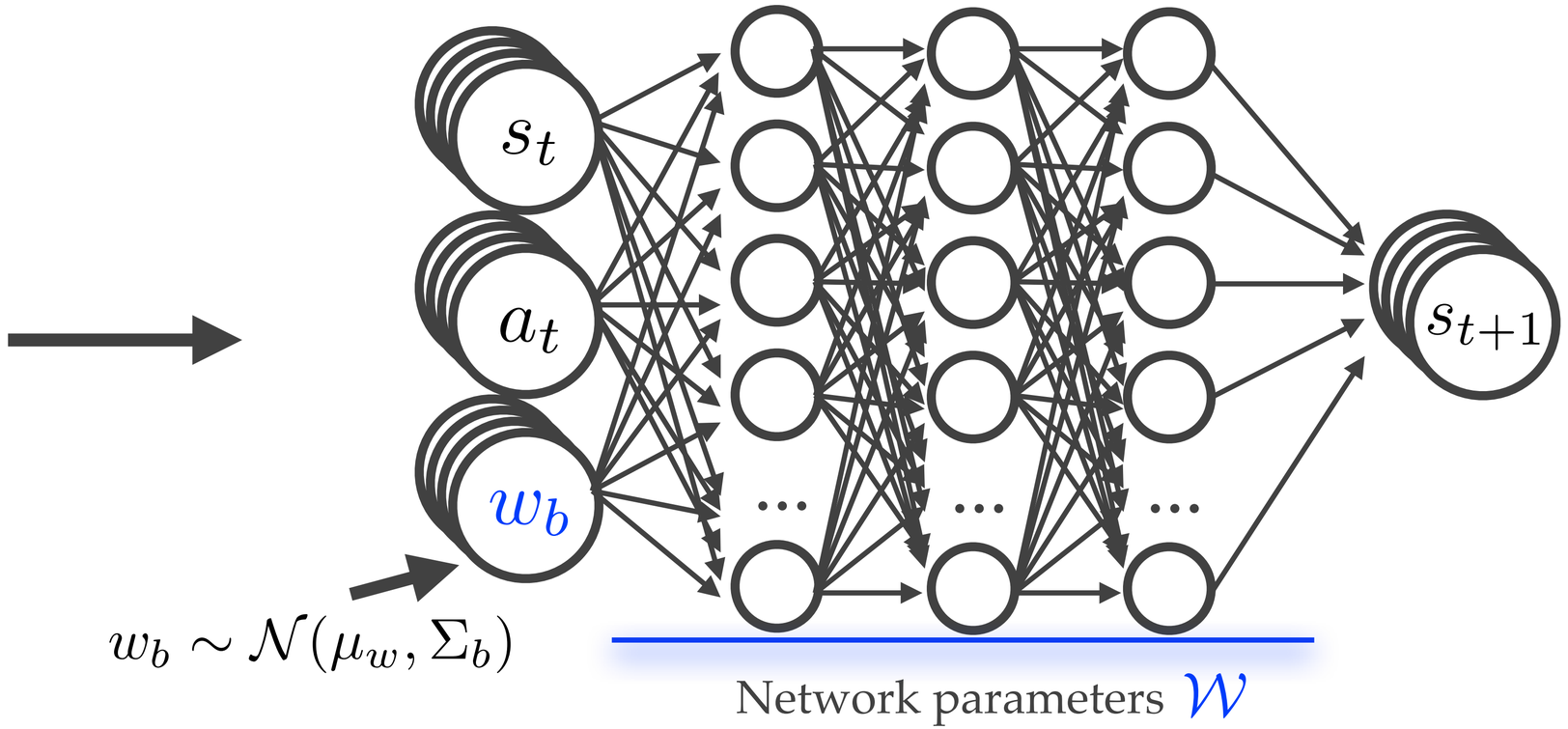
- With Toy 2D Navigation Domain: 6 task instances, 50 episodes per instance
- Transition model updated every 10 episodes

† Snelson and Ghahramani (2005, *NIPS*)
Hernández-Lobato, et al. (2016, *ICML*)

Training the BNN



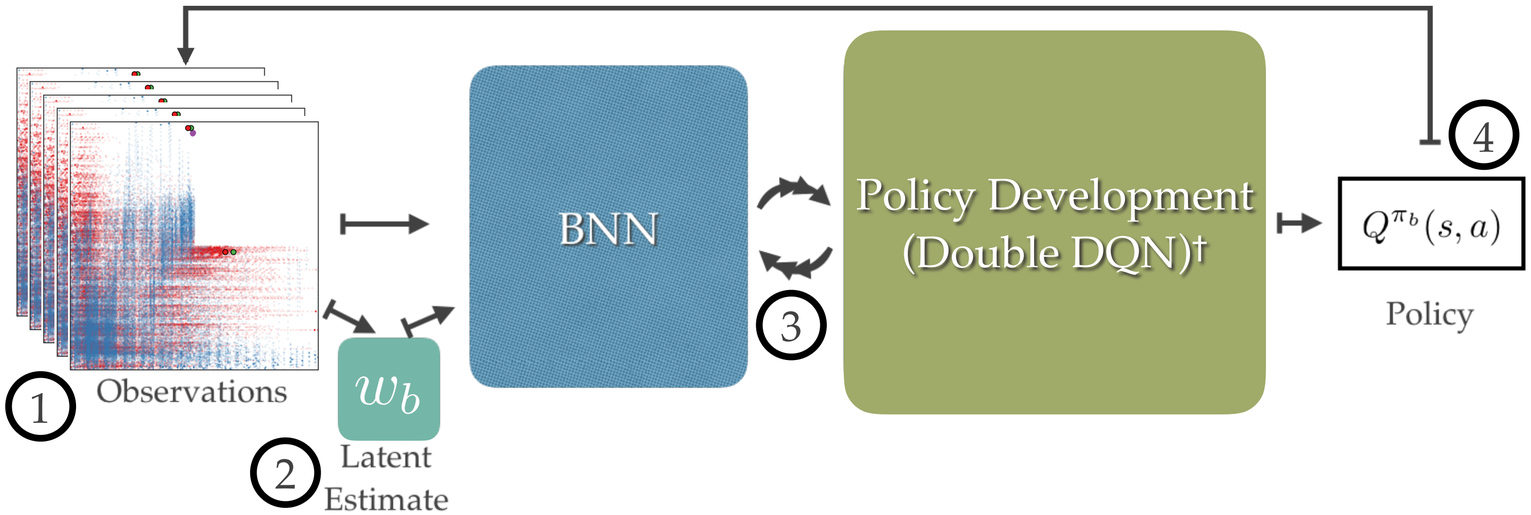
Observe $\{s_t, a_t, r_t, s_{t+1}\}$
from multiple θ_b



$\hat{T}(s_{t+1}|s_t, a_t; w_b)$ is trained by iteratively updating w_b and the network parameters $\mathcal{W}$ using α-divergence minimization†

† Hernández-Lobato, et al. (2016, *ICML*)

Algorithmic Methodology

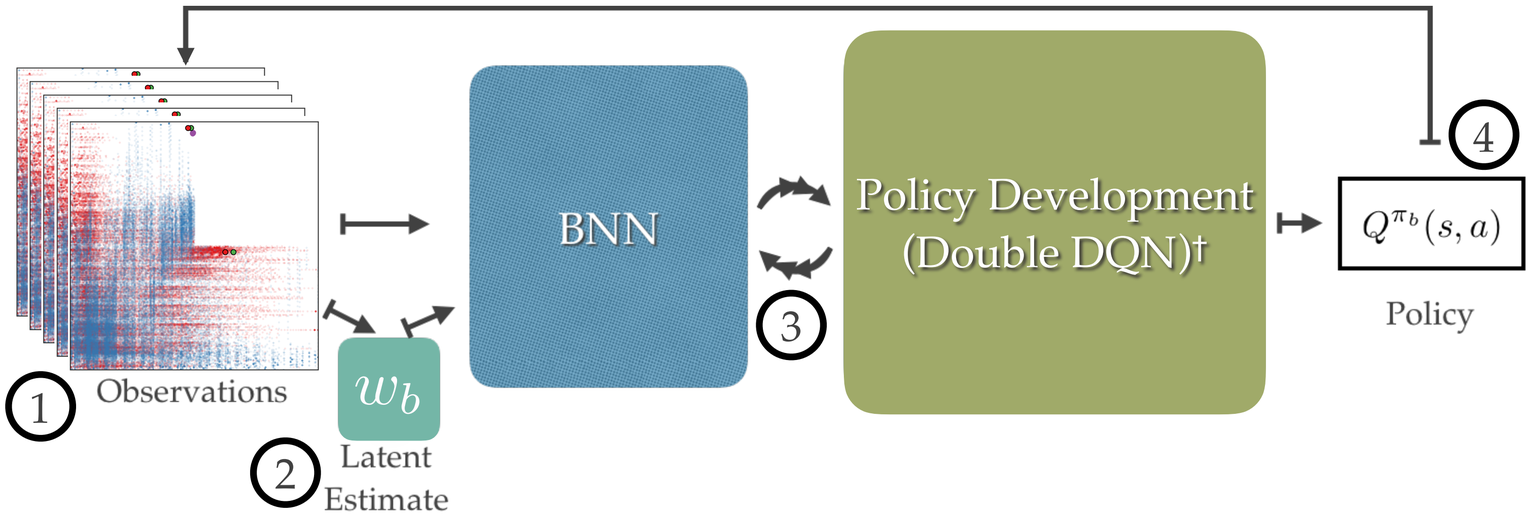


With a trained BNN, on a newly initialized task instance:

1. **Initial exploratory episode**
2. **Estimate w_b and refine the BNN model**
3. **Train a control policy π_b**
4. **Execute π_b in subsequent episodes**

† van Hasselt, et al. (2016, *AAAI*)

Algorithmic Methodology



With a trained BNN, on a newly initialized task instance:

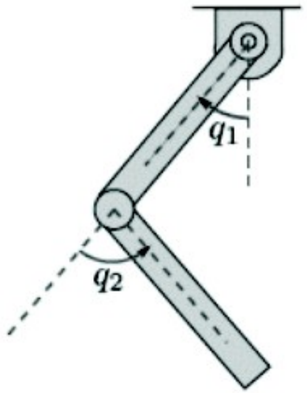
1. Initial exploratory episode
2. Estimate w_b and refine the BNN model
3. Train a control policy π_b
4. Execute π_b in subsequent episodes

Evaluated against 4 baselines:

1. Model-free
2. Model averaged over all observed instances
3. Model trained only on the current instance
4. Model with latent weights used as a linear output layer for BNN predictions

† van Hasselt, et al. (2016, *AAAI*)

Performance Comparison



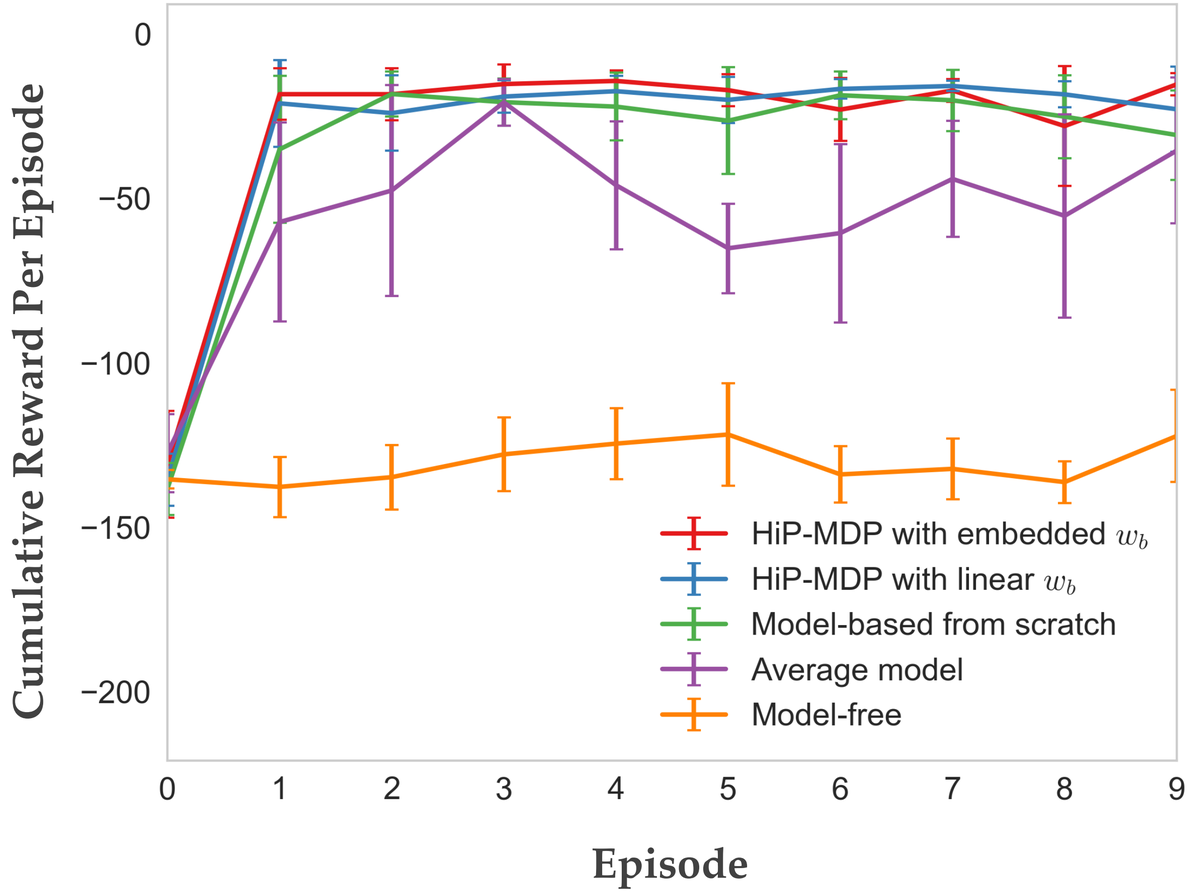
Acrobot

$S \in \mathbb{R}^4$

$|A| = 3$

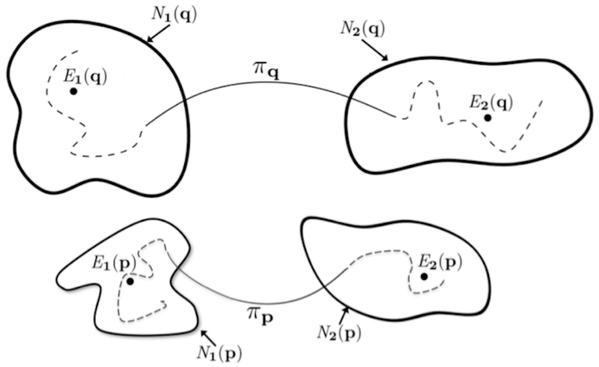
$w_b \in \mathbb{R}^5$

Acrobot



Performance Comparison

Simulated HIV Treatment†



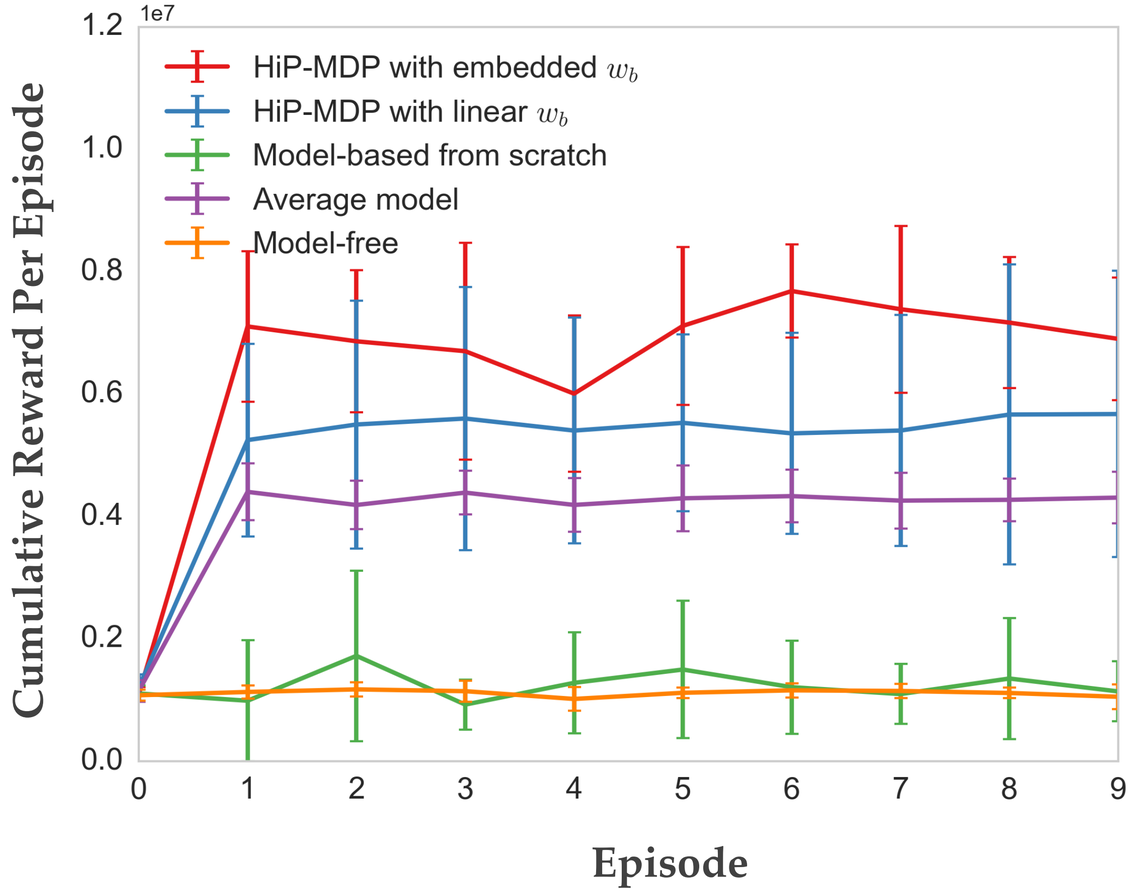
Adapted from Adams, et al. (2004)

HIV Treatment

$S \in \mathbb{R}^6$

$|A| = 4$

$w_b \in \mathbb{R}^5$



† Ernst, et al. (2006)

Conclusion

- The HiP-MDP provides a framework for robust and efficient transfer learning
  - Facilitated by a latent embedding to an approximated dynamic model of the environment

- Embedding the latent estimation of the environment *with* the input is more advantageous in domains with highly complex and nonlinear dynamics
  - This motivates further extension to even more complicated and realistic applications

- Further improvements to the HiP-MDP will contribute to a general transfer learning framework capable of addressing the most nuanced and complex control problems

Please visit us at poster #36 this evening. We're looking forward to meeting you.

Contact

taylorkillian@g.harvard.edu

@tw_killian

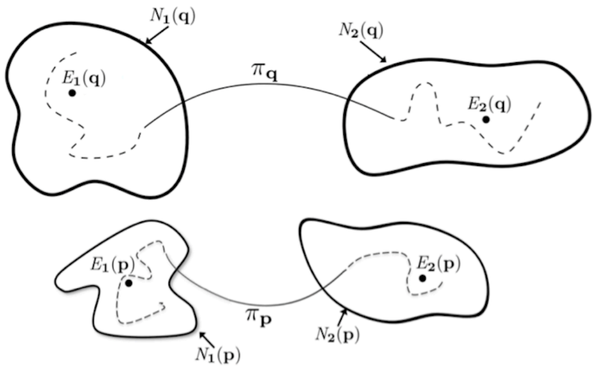
sdaulton@g.harvard.edu

https://github.com/dtak/hip-mdp-public

Please visit us at poster #36 this evening. We're looking forward to meeting you.

BACKUP

Simulated HIV Treatment



Adapted from Adams, et al. (2004)

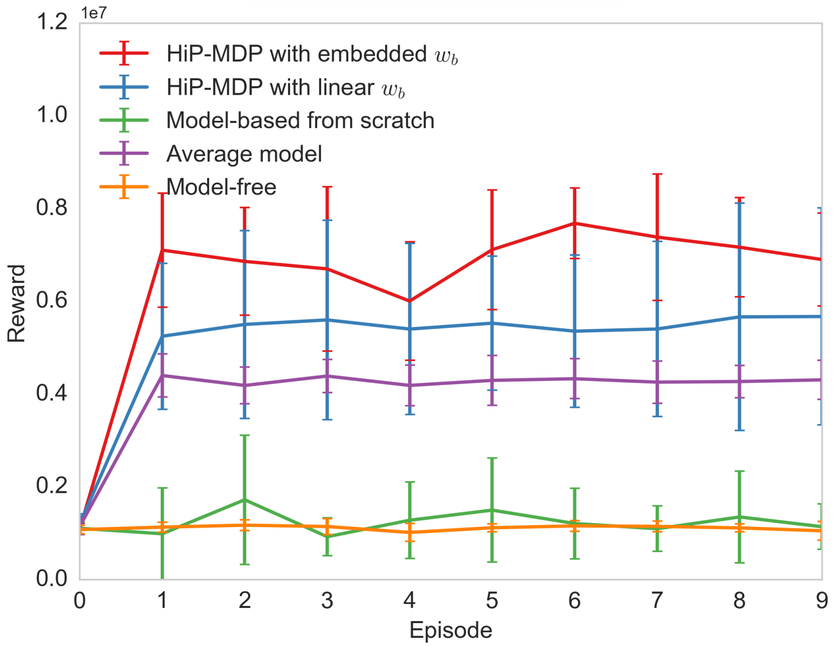
HIV Treatment

$S \in \mathbb{R}^6$

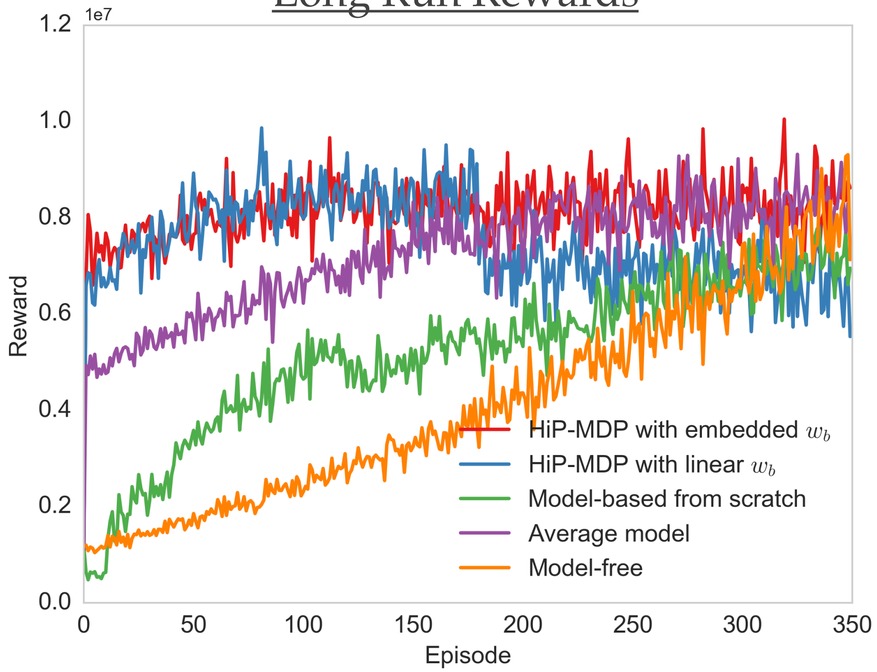
$|A| = 4$

$w_b \in \mathbb{R}^5$

Cumulative Reward

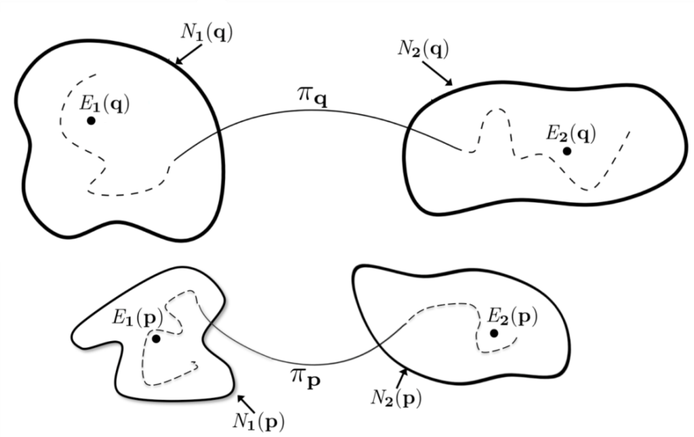


Long Run Rewards



Modeling Patient Response to HIV Treatment

- Adams, et al. (2004) modeled a patient's response to HIV treatment with a system of nonlinear equations
  - Defined by 22 physical parameters
- Ernst, et al. (2006) instituted a RL framework to develop effective treatment policies for HIV patients
- Perturbations of the underlying parameters admit subtle variations in the dynamics of patient response
  - Each variation has its own optimal policy



Notional transitions from unhealthy steady states to healthy steady states, defined by a patient's individual physiological response to treatment.

State Space

Six Indicators of Patient Health

- Healthy CD4+ T-lymphocytes
- Healthy Macrophages
- Infected CD4+ T-lymphocytes
- Infected Macrophages
- Free virus particles
- HIV-specific cytotoxic T-cells

Action Space

- No Treatment
- Protease Inhibitor (PI)
- Reverse Transcriptase Inhibitor (RTI)
- PI + RTI

Reward Function $R(s_t, a_t)$:

Weighted combination of number of healthy versus infected cells along with penalty for side effects introduced by each treatment

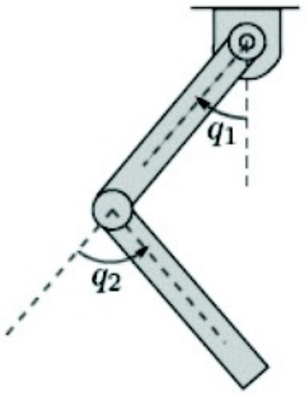
Modeling Patient Response to HIV Treatment

- Adams, et al. (2004) modeled a patient's response to HIV treatment with a system of nonlinear equations
  - Defined by 22 physical parameters
- Ernst, et al. (2006) instituted a RL framework to develop effective treatment policies for HIV patients
- Perturbations of the underlying parameters admit subtle variations in the dynamics of patient response
  - Each variation has its own optimal policy

parameter	value	units	description
λ_1	10,000	$\frac{cells}{mL \cdot day}$	target cell type 1 production (source) rate
d_1	0.01**	$\frac{1}{day}$	target cell type 1 death rate
ϵ_1	$\in [0,1)$	–	efficacy of reverse transcriptase inhibitor
ϵ_2	$\in [0,1)$	–	efficacy of protease inhibitor
k_1	8.0×10^{-7}	$\frac{mL}{virions \cdot day}$	population 1 infection rate
λ_2	31.98	$\frac{cells}{mL \cdot day}$	target cell type 2 production (source) rate
d_2	0.01**	$\frac{1}{day}$	target cell type 2 death rate
f	0.34 ($\in [0,1]$)	–	treatment efficacy reduction in population 2
k_2	1×10^{-4}	$\frac{mL}{virions \cdot day}$	population 2 infection rate
δ	0.7*	$\frac{1}{day}$	infected cell death rate
m_1	1.0×10^{-5}	$\frac{mL}{cells \cdot day}$	immune-induced clearance rate for population 1
m_2	1.0×10^{-5}	$\frac{mL}{cells \cdot day}$	immune-induced clearance rate for population 2
N_T	100*	$\frac{virions}{cell}$	virions produced per infected cell
c	13*	$\frac{1}{day}$	virus natural death rate
ρ_1	1	$\frac{virions}{cell}$	average number virions infecting a type 1 cell
ρ_2	1	$\frac{virions}{cell}$	average number virions infecting a type 2 cell
λ_E	1	$\frac{cells}{mL \cdot day}$	immune effector production (source) rate
b_E	0.3	$\frac{1}{day}$	maximum birth rate for immune effectors
K_b	100	$\frac{cells}{mL}$	saturation constant for immune effector birth
d_E	0.25	$\frac{1}{day}$	maximum death rate for immune effectors
K_d	500	$\frac{cells}{mL}$	saturation constant for immune effector death
δ_E	0.1*	$\frac{1}{day}$	natural death rate for immune effectors

Table 1: Parameters used in model (2.1). Those in the top section of the table are taken directly from Callaway and Perelson. Parameters in the bottom section of the table are adapted from those in Bonhoeffer, *et al.*. The superscripts * denote parameters the authors indicated were estimated from human data and ** denote those estimated from macaque data.

Acrobot



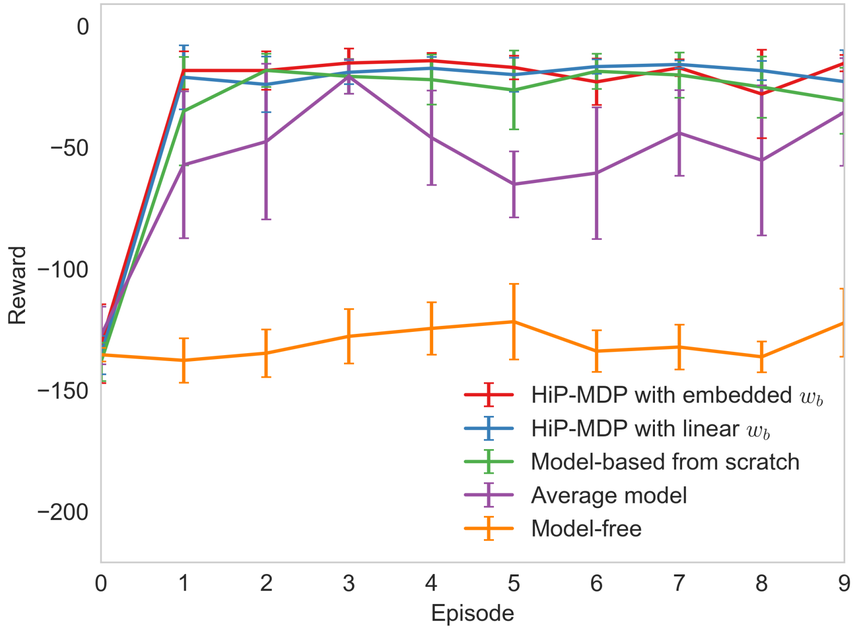
Acrobot

$S \in \mathbb{R}^4$

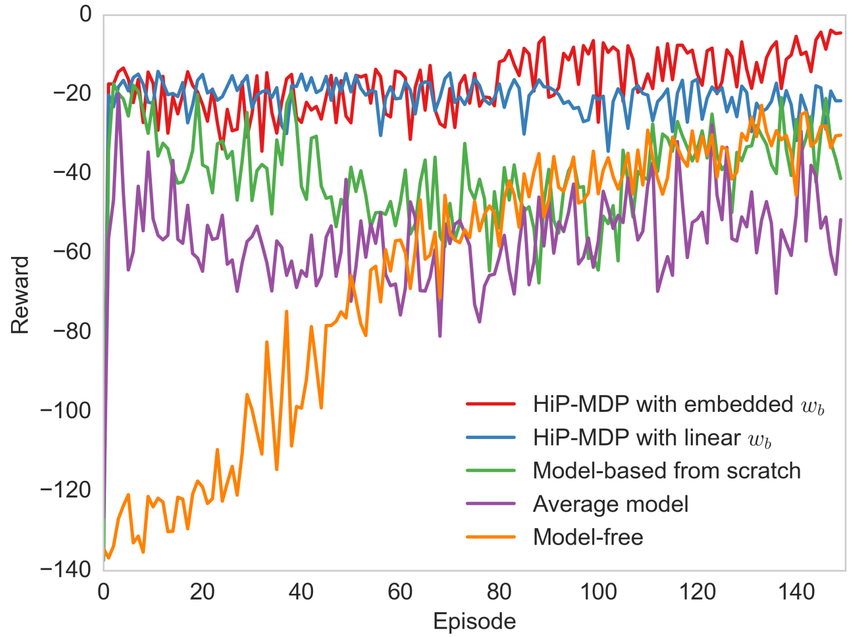
$|A| = 3$

$w_b \in \mathbb{R}^5$

Cumulative Reward

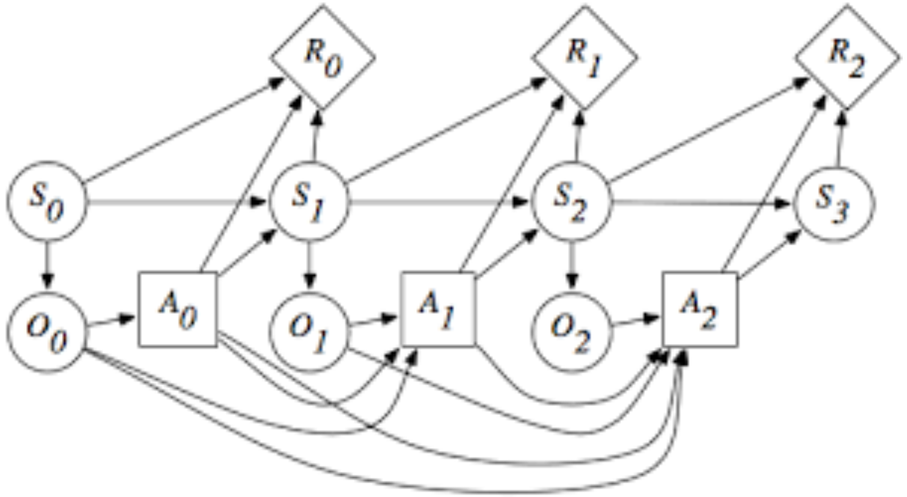


Long Run Rewards



Partially Observable Markov Decision Processes

- POMDPs are a generalization of MDPs where either the system dynamics or state representation are not fully observed
- States (or transition dynamics) are represented by distributions rather than discrete quantities
  - Current developments in RL decompose these distributions into a set of options, with an explicit [Bacon et al., 2016] or latent [Chen et al., 2017] representation and then solve as a discrete MDP



Gaussian Processes

When provided data $\mathbf{X} \in \mathbb{R}^D$, GPs are fully specified by a mean $m(\mathbf{X})$ and covariance function $k(\mathbf{X}, \mathbf{X}')$ of some underlying true process $f(\mathbf{X})$

Then, when given test data $\mathbf{X}_*$, the posterior prediction of the output values can be represented as:

$$f_*|\mathbf{X}_*, \mathbf{X}, f \sim \mathcal{N}\left(K(X_*, X)K(X, X)^{-1}f,\ K(X_*, X_*) - K(X_*, X)K(X, X)^{-1}K(X, X_*)\right)$$

That is,

$$m(X_*) = K(X_*, X)K(X, X)^{-1}f$$
$$k(X, X') = K(X_*, X_*) - K(X_*, X)K(X, X)^{-1}K(X, X_*)$$

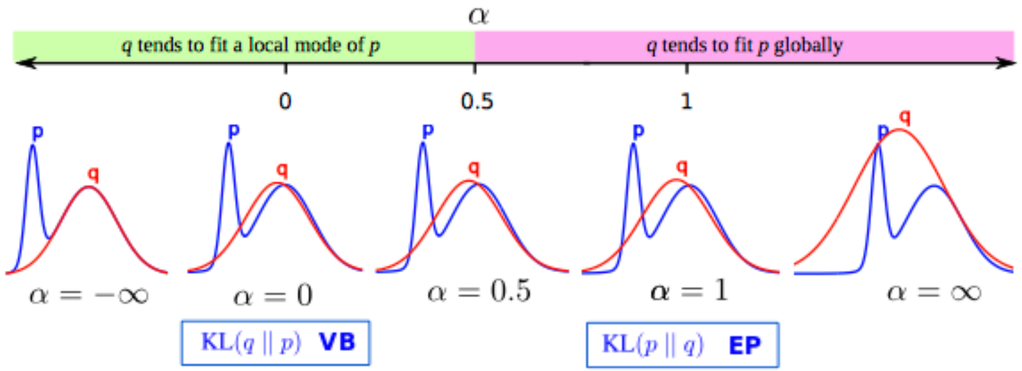
Typically,

$$k(x, x') = \sigma^2 \exp\left(-\frac{(x - x')^2}{2\ell^2}\right)$$

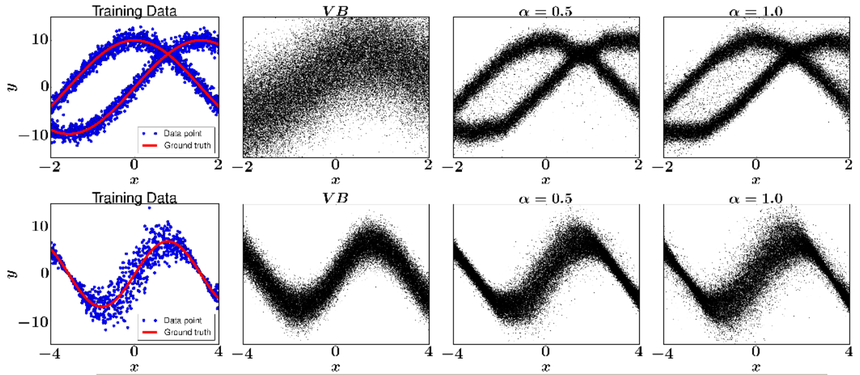
Bayesian Neural Networks (BNN)

Alpha Divergence Minimization [Hernández-Lobato et al., 2015]

- Alpha-Divergence Minimization is an approximate inference technique for estimating the posterior network parameter distributions of the BNN
  - Used to approximate the intractable calculation of (details in backup): $p(\theta|\mathcal{D}) \propto \left[\prod_{n=1}^{N} p(\mathbf{x}_n|\theta)\right] p_0(\theta)$
- Alpha-divergence trained BNN transition functions are both scalable and expressive, a perfect match for our needs



Visualizing effect of alpha parameter when approximating different distributions
"Black-Box alpha-Divergence Minimization" [Hernández-Lobato et al., 2015]



Example Regression Performance of BB-alpha trained BNN
"Learning and Policy Search in Stochastic Dynamical Systems with Bayesian Neural Networks" [Depeweg et al., 2016]

Bayesian Neural Networks

Variational Inference: Alpha Divergence Minimization

We aim to solve for the posterior distribution of our parameter, given some observations: $p(\theta|\mathcal{D}) \propto \left[\prod_{n=1}^{N} p(\mathbf{x}_n|\theta)\right] p_0(\theta)$

This is typically intractable as the form of the distribution p is usually unknown. In variational inference, we approximate this posterior by constructing a separate distribution q and then try to optimize its parameters such that it is "close" to p

Alpha Divergence minimization seeks to minimize the distance between p and q via:

$$D_\alpha[p||q] = \frac{1}{\alpha(1-\alpha)}\left(1 - \int p(\theta)^\alpha q(\theta)^{1-\alpha} d\theta\right)$$

Then by matching moments and by linearly approximating the parameters of q we solve what is known as the energy function of Power Expectation Propagation:

$$E(\lambda_0, \{\lambda_n\}) = \log Z(\lambda_0) + \left(\frac{N}{\alpha} - 1\right) \log Z(\lambda_q) - \frac{1}{\alpha} \sum_{n=1}^{N} \log \int p(x_n|\theta)^\alpha \exp\left\{s(\theta)^T(\lambda_q - \alpha\lambda_n)\right\} d\theta$$

Developing a Policy

- [Deisenroth and Rasmussen (2011)] introduced a data-efficient methodology (PILCO) that utilizes a model (approximated or derived) of observed states to learn optimal control policies in a paired online/offline fashion
  - Was recently updated by [Gal and Rasmussen (2017)] to incorporate deep structures
- After this fashion, with a large batch of previously run data, we execute the following:

```
# Observe randomly initialized instance of system (e.g. receive a new patient)
# Repeat for N episodes
    Observe system according to current policy
    # Periodically update latent weighting of current instance
    # Update policy using Double Deep Q-Network with approximated
# Update GP hyperparameters
```

$T(s' | s, a, \theta_b)$

A Use Case for Transfer Learning

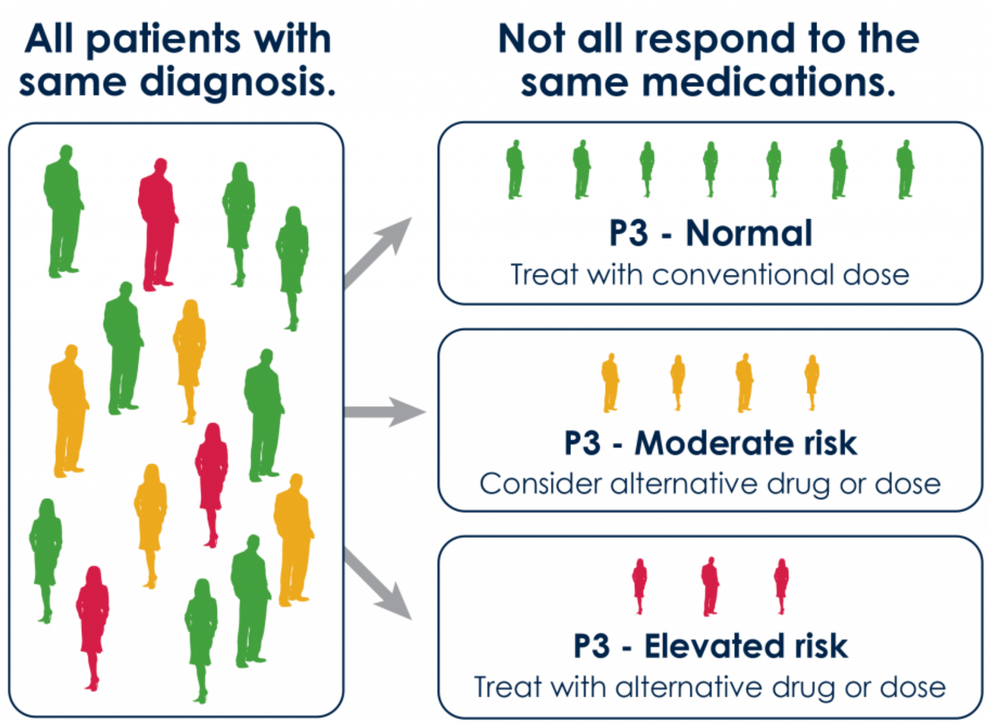


Image courtesy: http://arcpointos.com/pharmacogenetics-testing/

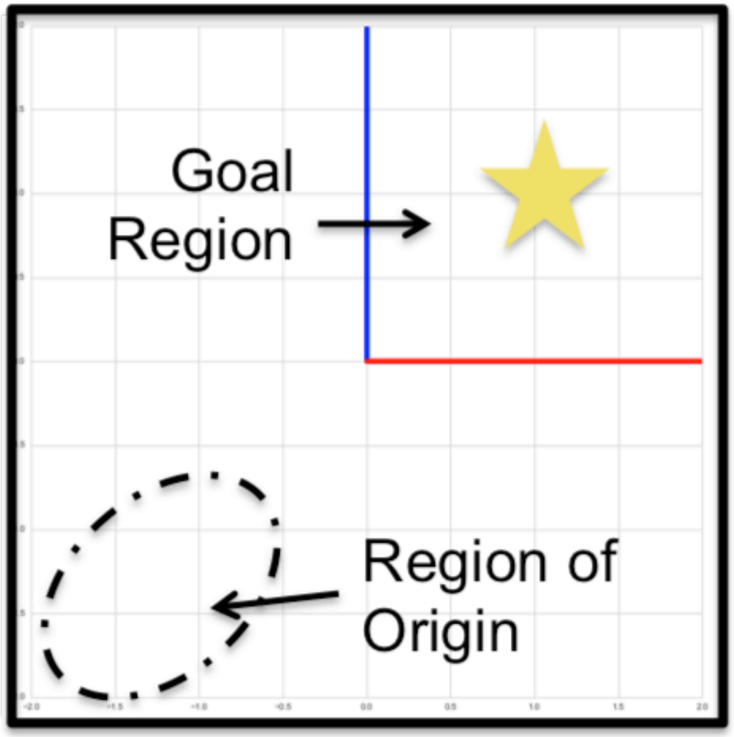
- Individual response to medical treatments can vary across the patient population
- Some treatments can lead to no response or potentially harmful side effects
- Significant challenge arises when patient is diagnosed with aggressive, life-altering illness
  - e.g. HIV / AIDS, Diabetes, Cancer, etc.

Can we determine an optimal treatment policy for any patient according to their individual genetic characteristics, in diagnosis and throughout administration?

Transfer Learning

Intertask variation: a more subtle environment for transfer

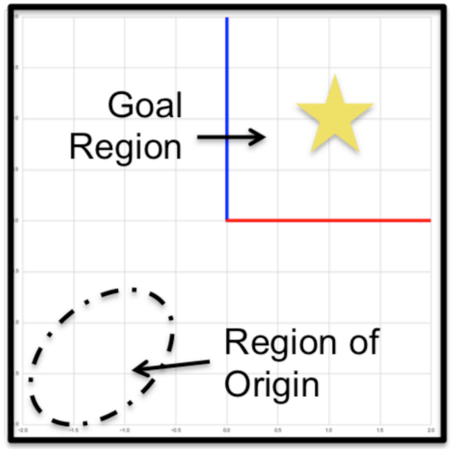
- Key to the transfer between varied instances of the same task is in the construction and estimation of an invariant feature space
- To aid the development of a robust and efficient transfer algorithm in such scenarios we introduce a simple 2D navigation domain:
  - Hidden latent parameter determines how agent can transition to Goal Region
  - Agent must learn separate control policies based on this latent parametrization



State space: $[s_1, s_2] \in [-2, 2] \subset \mathbb{R}^2$
Actions: Left, Right, Up, Down

$$R(s,a) = \begin{cases} 1000 & \text{if } \texttt{agent} \text{ reaches Goal Region} \\ -5 & \text{if } \texttt{agent} \text{ hits wall or attempts invalid transition} \\ -0.1 & \text{otherwise} \end{cases}$$

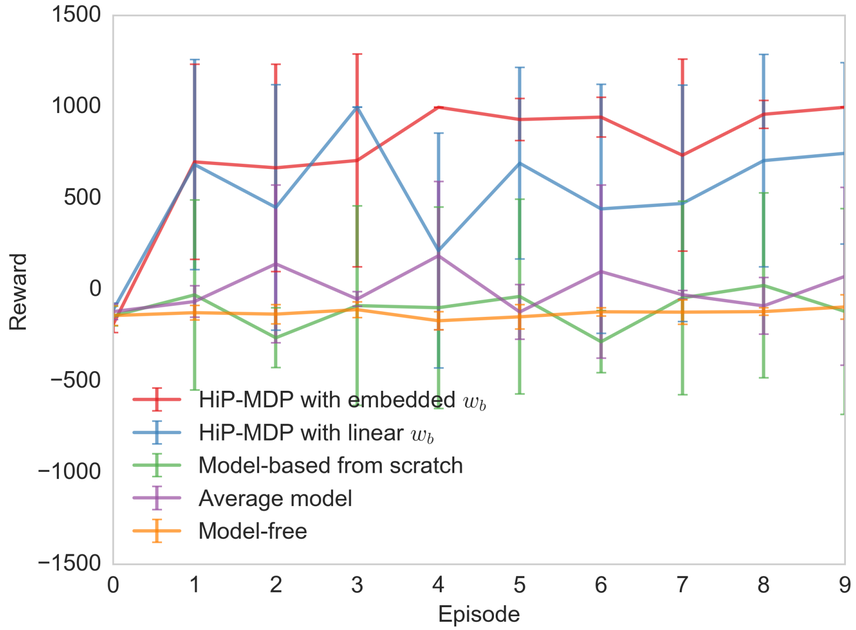
Toy Problem: 2D Navigation



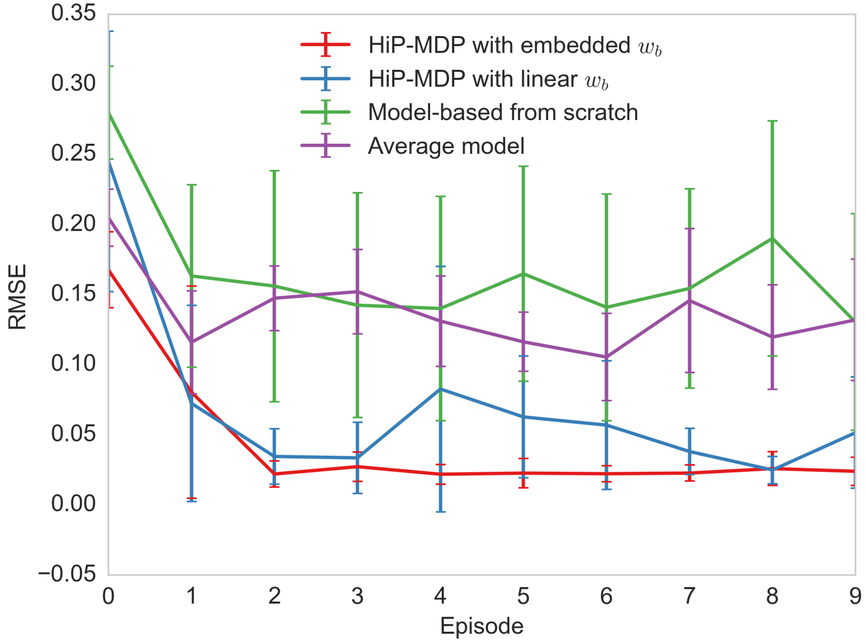
Toy 2D Navigation

$$S \in R^2$$
$$|A| = 4$$
$$w_b \in R^3$$

Cumulative Reward



Model Prediction Error

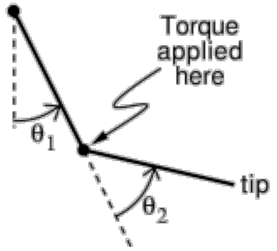


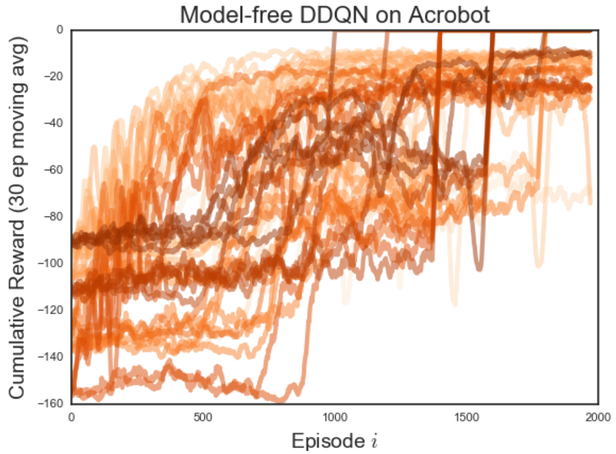
Deploying the HiP-MDP

Extending to larger domains

The Acrobot

Goal: Raise tip above line





State Space

Four angular meas. of pendulum

- Hinge angular displacement
- Hinge angular velocity
- Tip angular displacement
- Tip angular velocity

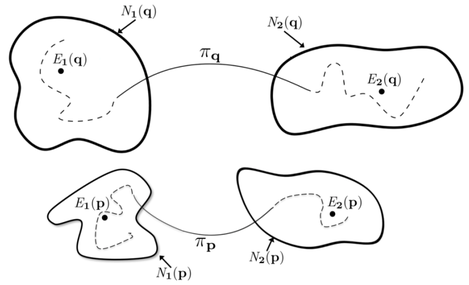
Action Space

- Apply torque left
- Apply torque right
- Do nothing

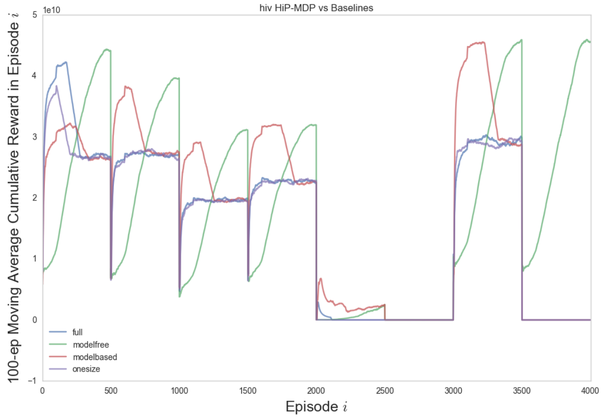
Reward Function

$$R(s_t, a_t) = \begin{cases} -1 & \text{if tip not above line} \\ 10 & \text{if tip above line} \end{cases}$$

HIV Treatment



Notional transitions from unhealthy steady states to healthy steady states, defined by a patient's individual physiological response to treatment.



State Space

Six Indicators of Patient Health

- Healthy CD4+ T-lymphocytes
- Healthy Macrophages
- Infected CD4+ T-lymphocytes
- Infected Macrophages
- Free virus particles
- HIV-specific cytotoxic T-cells

Action Space

- No Treatment
- Protease Inhibitor (PI)
- Reverse Transcriptase Inhibitor (RTI)
- PI + RTI

Reward Function $R(s_t, a_t)$:

Weighted combination of number of healthy versus infected cells along with penalty for side effects introduced by each treatment