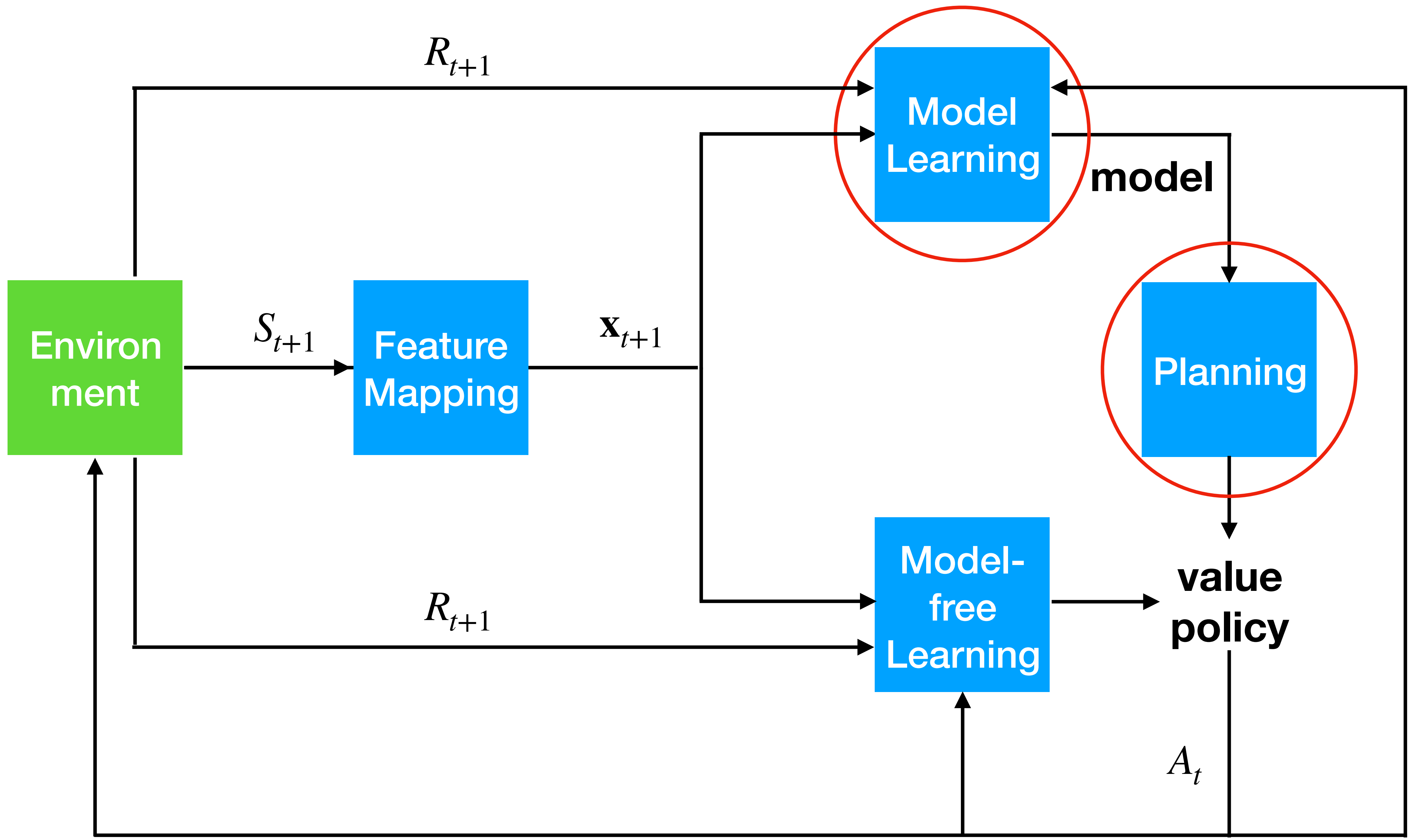
Planning with Expectation Models

Tea Time Talk
2019/06/06

Model-based Reinforcement Learning



Problem Setting

MDP	Criteria	Task
Finite MDP	Discount Reward	Policy Evaluation

$\mathbf{x} = \mathbf{x}(s)$	Feature vector of state s
$\pi(a \mid \mathbf{x})$	Target policy
$b(a \mid \mathbf{x})$	Behavior Policy
$v_\pi(s)$	True value of state s under target policy
$\hat{v}(\mathbf{x}, \mathbf{w})$	Approximate value of state s
$p(s', r \mid s, a)$	Environment Dynamics
$p(\mathbf{x}' \mid s, a), r(s, a)$	True distribution model (for value FA)
$\hat{p}(\mathbf{x}' \mid \mathbf{x}, a), \hat{r}(\mathbf{x}, a)$	Approx. distribution model
$\hat{\mathbf{x}}(\mathbf{x}, a), \hat{r}(\mathbf{x}, a)$	Approx. sample/expectation model

Model Choices

	model projection	examples	problems
Distribution	$\hat{r}(\mathbf{x}, a) \approx \mathbb{E}_b[R_{t+1} \mid \mathbf{x}_t = \mathbf{x}, A_t = a]$ $\hat{p}(\mathbf{x}' \mid \mathbf{x}, a) \approx \Pr[\mathbf{x}_{t+1} = \mathbf{x}' \mid \mathbf{x}_t = \mathbf{x}, A_t = a]$	Gaussian process Mixture Density Networks Time-varying Gaussian	1. We don't have a method to learn and represent a general distribution in a scalable and efficient way.
Sample	$\hat{r}(\mathbf{x}, a) \approx \mathbb{E}_b[R_{t+1} \mid \mathbf{x}_t = \mathbf{x}, A_t = a]$ $\hat{\mathbf{x}}(\mathbf{x}, a) \sim \hat{p}(\mathbf{x}' \mid \mathbf{x}, a)$	Variational Inference GAN	1. The distribution would still have to be learned and represented.
Expection	$\hat{r}(\mathbf{x}, a) \approx \mathbb{E}_b[R_{t+1} \mid \mathbf{x}_t = \mathbf{x}, A_t = a]$ $\hat{\mathbf{x}}(\mathbf{x}, a) \approx \mathbb{E}_b[\mathbf{x}_{t+1} \mid \mathbf{x}_t = \mathbf{x}, A_t = a]$	Our method	1. Learning is straightforward but in general the information is lost. 2. Rollout is not valid in general

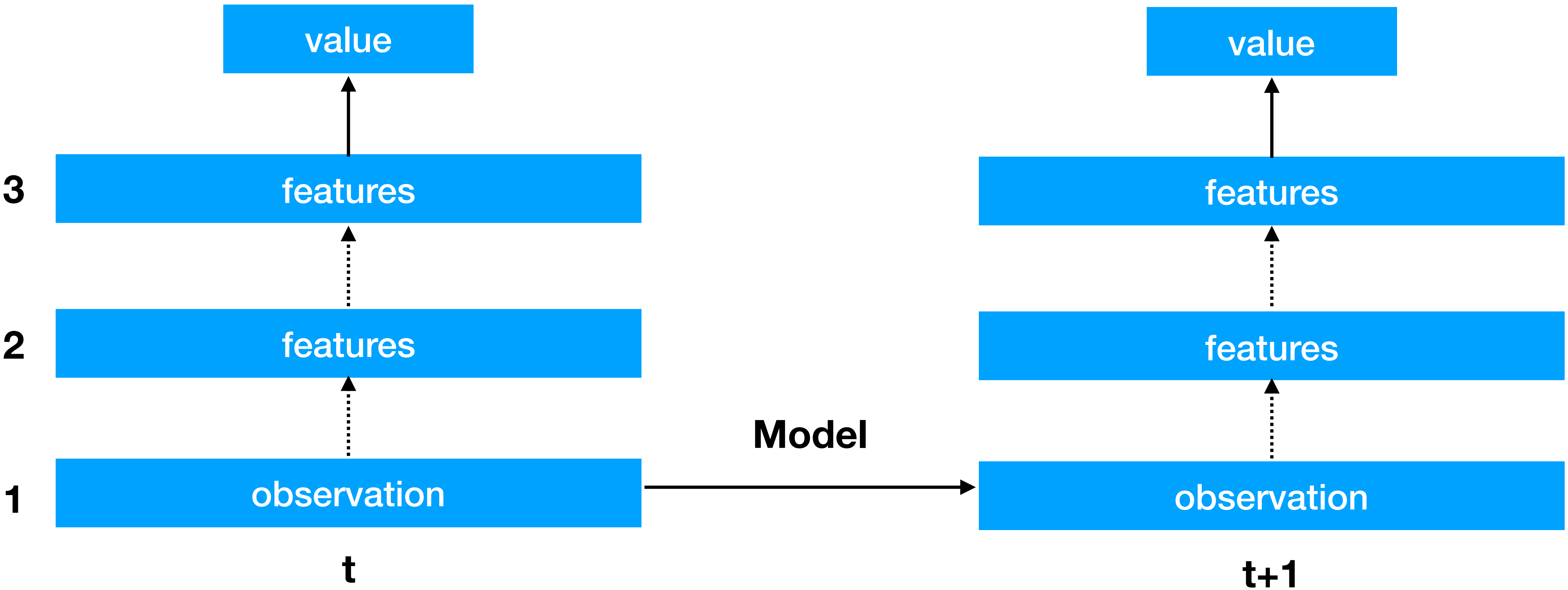
Expectation Models and Linear Value Functions

Policy evaluation (via Approx. DP) with an approximate distribution model

$$
\begin{aligned}
&\forall s \in \mathcal{S}, \mathbf{x} = \mathbf{x}(s) \\
&\hat{v}(\mathbf{x}, \mathbf{w}) \leftarrow \sum_a \pi(a \,|\, \mathbf{x}) \Big[\hat{r}(\mathbf{x}, a) + \gamma \sum_{\mathbf{x}'} \hat{p}(\mathbf{x}' \,|\, \mathbf{x}, a) \hat{v}(\mathbf{x}', \mathbf{w}) \Big] \\
&\qquad = \sum_a \pi(a \,|\, \mathbf{x}) \Big[\hat{r}(\mathbf{x}, a) + \gamma \sum_{\mathbf{x}'} \hat{p}(\mathbf{x}' \,|\, \mathbf{x}, a) \mathbf{x}'^{\top} \mathbf{w} \Big] \\
&\qquad = \sum_a \pi(a \,|\, \mathbf{x}) \big[\hat{r}(\mathbf{x}, a) + \gamma \hat{\mathbf{x}}(\mathbf{x}, a)^{\top} \mathbf{w} \big]
\end{aligned}
$$

Policy evaluation (via Approx. DP) with an approximate expectation model

Where Should We Build Model Upon?



Pro1: Model doesn’t need to capture stochasticity (thus is simpler).
Pr .
Co ’t change features.

Planning doesn’t directly change features,
but may provide information for feature updates.

Linear & Non-Linear Expectation Models

Best Linear Expectation Model

$$\hat{\mathbf{x}}^*(\mathbf{x}, a) = \mathbf{F}_a^* \mathbf{x}$$

$$\hat{r}^*(\mathbf{x}, a) = \mathbf{b}_a^{*\top} \mathbf{x}$$

$$\mathbf{F}_a^* \doteq \arg\min_{\mathbf{G}} \mathbb{E}_b[\mathbb{I}(A_t = a) \|\mathbf{G}\mathbf{x}_t - \mathbf{x}_{t+1}\|_2^2]$$

$$\mathbf{b}_a^* \doteq \arg\min_{\mathbf{u}} \mathbb{E}_b[\mathbb{I}(A_t = a)(\mathbf{u}^\top \mathbf{x}_t - R_{t+1})^2]$$

$$\mathbf{F}_a^* = \mathbb{E}_b[\mathbb{I}(A_t = a)\mathbf{x}_{t+1}\mathbf{x}_t^\top]\mathbb{E}_b[\mathbb{I}(A_t = a)\mathbf{x}_t\mathbf{x}_t^\top]^{-1}$$

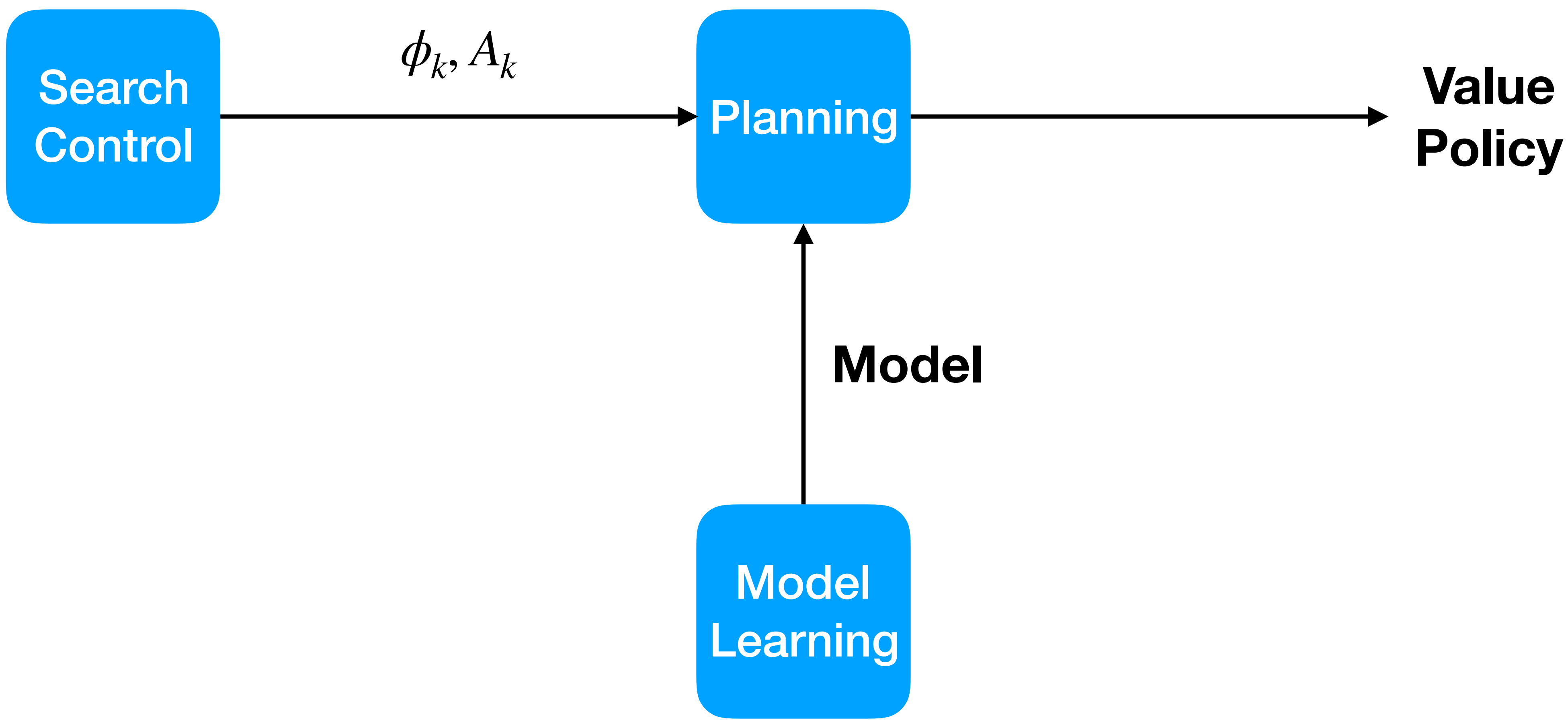
$$\mathbf{b}_a^* = \mathbb{E}_b[\mathbb{I}(A_t = a)\mathbf{x}_t\mathbf{x}_t^\top]^{-1}\mathbb{E}_b[\mathbb{I}(A_t = a)\mathbf{x}_t R_{t+1}]$$

Best Non-Linear Expectation Model

$$\begin{aligned}\hat{\mathbf{x}}^*(\mathbf{x}, a) &\doteq \mathbb{E}_b[\mathbf{x}' \,|\, \mathbf{x}, a] \\ &= \frac{\sum_{s \in H_\mathbf{x}} \eta(s) \mathbb{E}[\mathbf{x}(S') \,|\, S = s, A = a]}{\mu(\mathbf{x})}\end{aligned}$$

$$\begin{aligned}\hat{r}^*(\mathbf{x}, a) &\doteq \mathbb{E}_b[R \,|\, \mathbf{x}, a] \\ &= \frac{\sum_{s \in H_\mathbf{x}} \eta(s) \mathbb{E}[R \,|\, S = s, A = a]}{\mu(\mathbf{x})}\end{aligned}$$

Dyna-style Planning



Limitation of Linear Models

If

$$\phi_k \sim d_b(\,\cdot\,)$$
$$A_k \sim \pi(\,\cdot\,|\,\phi_k)$$

then in general

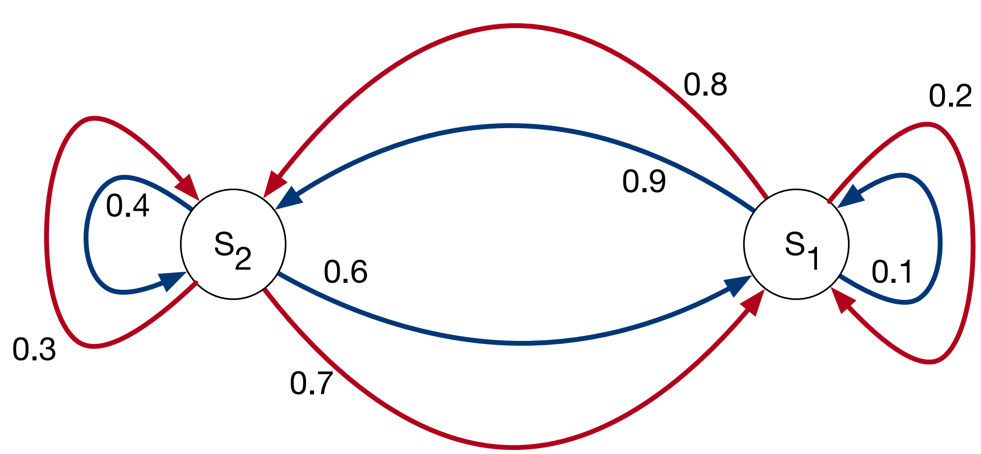
$$\mathbf{w}_{\textbf{linear}} \neq \mathbf{w}_{\textbf{non-linear}} = \mathbf{w}_{\textbf{real}}$$

where

$$\mathbf{w}_{\textbf{linear}} = (\mathbf{I} - \gamma \mathbf{F}^{*\top})^{-1}\mathbf{b}*, \mathbf{F}* = \mathbb{E}[\mathbf{F}^*_{A_k}\phi_k\phi_k^\top]\mathbb{E}[\phi_k\phi_k^\top]^{-1}, \mathbf{b}* = \mathbb{E}[\phi_k\phi_k^\top]^{-1}\mathbb{E}[\phi_k\phi_k^\top\mathbf{b}^*_{A_k}]$$

$$\mathbf{w}_{\textbf{non-linear}} = \mathbb{E}[\phi_k(\phi_k - \gamma\hat{\mathbf{x}}*(\phi_k, A_k))^\top]^{-1}\mathbb{E}[r*(\phi_k, A_k), \phi_k]$$

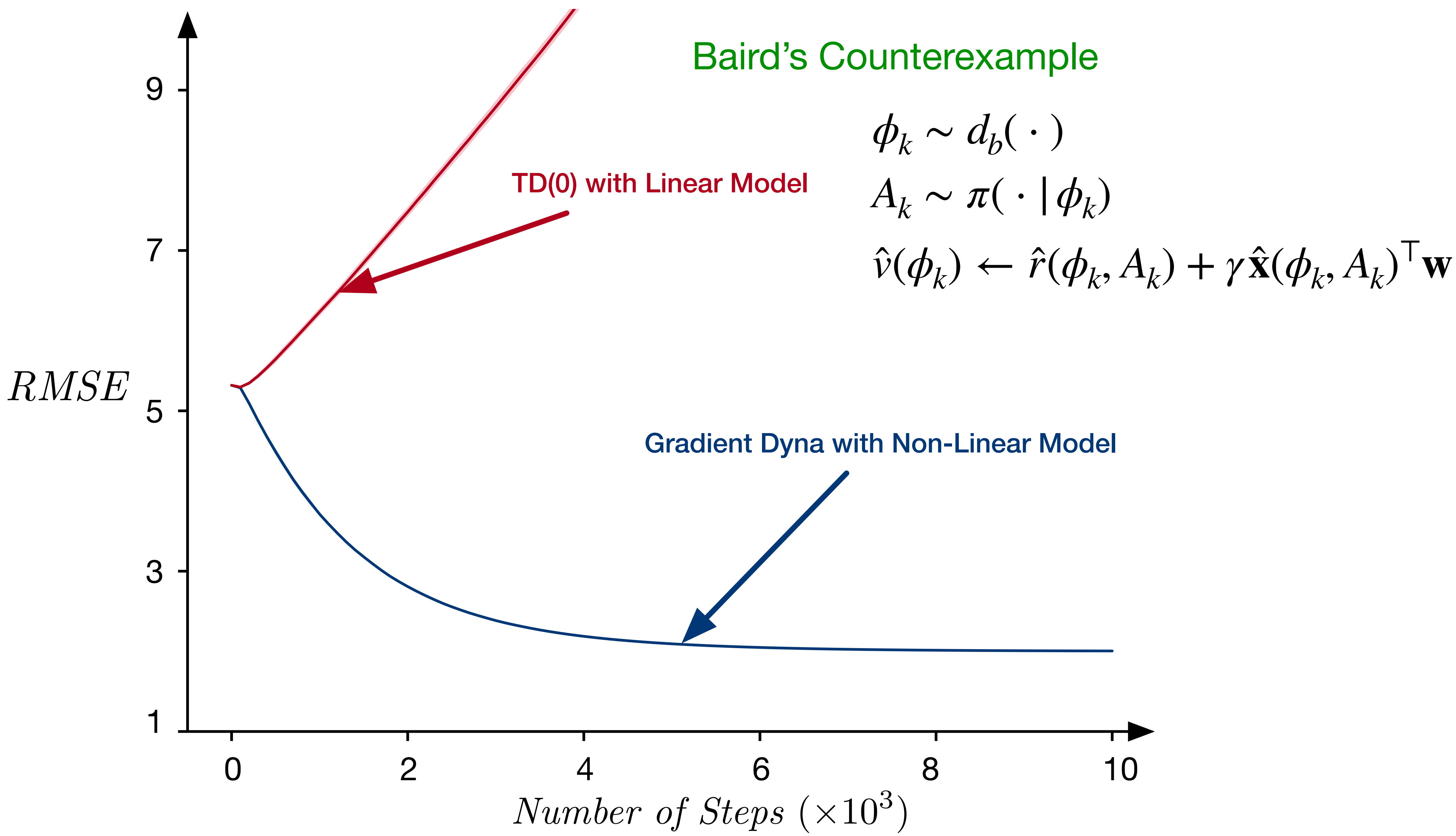
$$\mathbf{w}_{\textbf{real}} = \mathbb{E}[\rho_t\mathbf{x}_t(\mathbf{x}_t - \gamma\mathbf{x}_t)^\top]^{-1}\mathbb{E}[\rho_t R_{t+1}\mathbf{x}_t]$$



$$\mathbf{w}_{\textbf{linear}} = [0.953]^\top$$
$$\mathbf{w}_{\textbf{real}} = [8.89]^\top$$

Use non-linear expectation models instead of linear ones!

Limitation of TD(0) Planning with Linear Value Functions



Gradient Dyna

Mean Square Projected Bellman Error (MB-MSPBE)

$$\textbf{MSPBE}(\mathbf{w}) = \mathbb{E}[\rho_t \delta_t \mathbf{x}_t]^\top \mathbb{E}[\mathbf{x}_t \mathbf{x}_t^\top]^{-1} \mathbb{E}[\rho_t \delta_t \mathbf{x}_t]$$

$$\delta_t = R_{t+1} + \gamma \mathbf{w}^\top \mathbf{x}_{t+1} - \mathbf{w}^\top \mathbf{x}_t$$

Model-Based Mean Square Projected Bellman Error (MB-MSPBE)

$$\textbf{MB-MSPBE}(\mathbf{w}) = \mathbb{E}[\Delta_k \phi_k]^\top \mathbb{E}[\phi_k \phi_k^\top]^{-1} \mathbb{E}[\Delta_k \phi_k]$$

$$\Delta_k = \hat{r}(\phi_k, A_k) + \gamma \mathbf{w}^\top \hat{\mathbf{x}}(\phi_k, A_k) - \mathbf{w}^\top \phi_k$$

MB-MSPBE = MSPBE, if

$$\phi_k \sim d_b(\,\cdot\,)$$
$$A_k \sim \pi(\,\cdot \mid \phi_k)$$
$$\hat{r} = \hat{r}^*, \hat{\mathbf{x}} = \hat{\mathbf{x}}^*$$

Gradient Dyna

$$\nabla \textbf{MB-MSPBE}(\mathbf{w}) = \mathbb{E}[(\gamma \hat{\mathbf{x}}(\phi_k, A_k) - \phi_k)\phi_k^\top]\mathbb{E}[\phi_k\phi_k^\top]^{-1}\mathbb{E}[\Delta_k\phi_k]$$

Algorithm 1 Gradient Dyna Algorithm

Input: $\mathbf{w}_0$, policy π, feature vector distribution ζ, expectation model $\{\hat{\mathbf{x}}, \hat{r}\}$, stepsizes α_k, β_k for $k = 1, 2, \cdots$

Output: $\mathbf{w}_k$

1: **for** $k = 1, 2, \cdots$ **do**

2: Sample $\boldsymbol{\phi}_k \sim \zeta(\cdot)$

3: Sample $A_k \sim \pi(\cdot|\boldsymbol{\phi}_k)$

4:

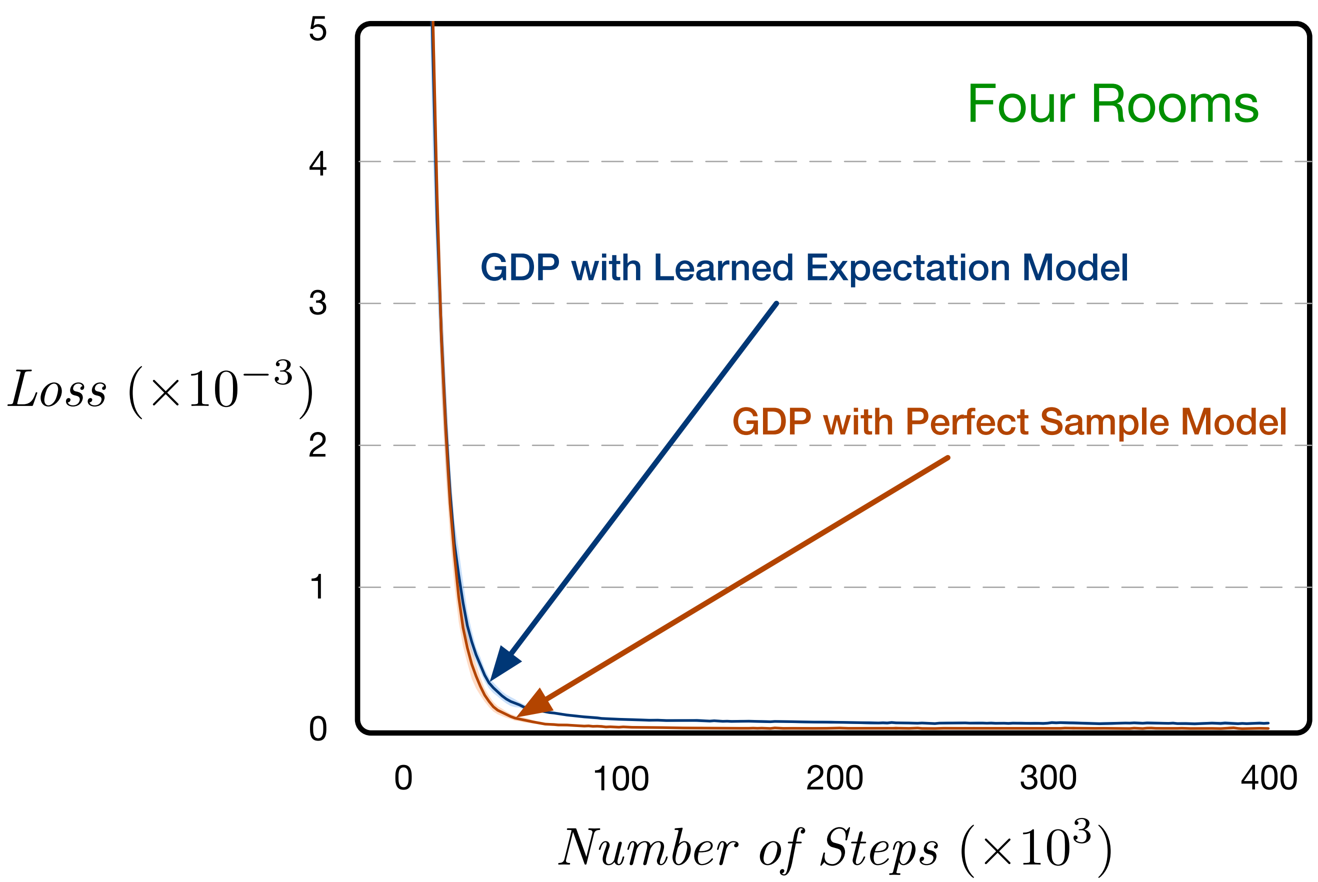
$$\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k - \alpha_k \mathbf{V}_k \Delta_k \boldsymbol{\phi}_k$$

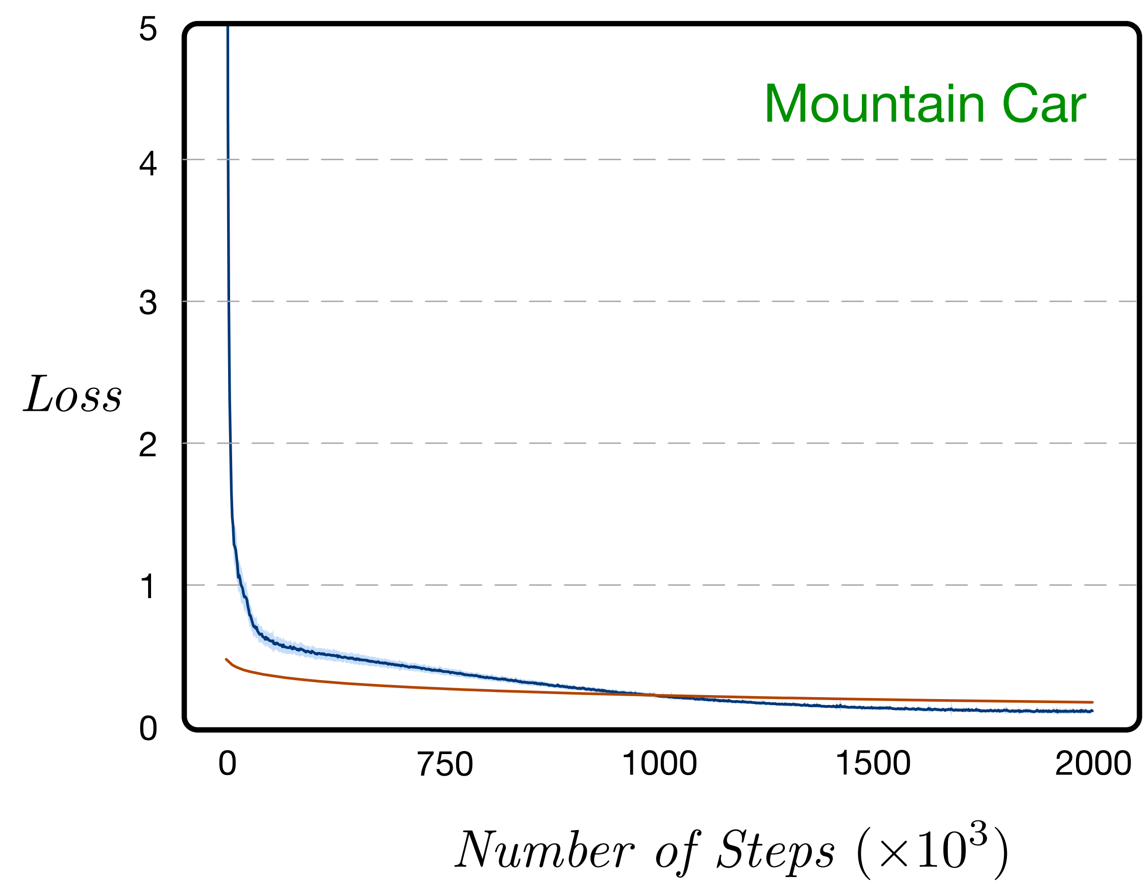
$$\mathbf{V}_{k+1} \leftarrow \mathbf{V}_k + \beta_k((\gamma\hat{\mathbf{x}}(\boldsymbol{\phi}_k, A_k) - \boldsymbol{\phi}_k)\boldsymbol{\phi}_k^\top - \mathbf{V}_k\boldsymbol{\phi}_k\boldsymbol{\phi}_k^\top)$$

5: **end for**

Gradient Dyna

$$\mathbf{A}_{\text{LSTD}} = \mathbb{E}[\rho_t \mathbf{x}_t(\mathbf{x}_t - \gamma \mathbf{x}_t)^\top]$$
$$\mathbf{c}_{\text{LSTD}} = \mathbb{E}[\rho_t R_{t+1} \mathbf{x}_t]$$
$$\text{loss} = \|\mathbf{A}_{\text{LSTD}} \mathbf{w} - \mathbf{c}_{\text{LSTD}}\|_2^2$$





Take-home Messages

1) if the dynamics is stochastic and you want to use expectation model, then in general you need to use linear state value function
2) you want to use non-linear expectation model instead of linear one
3) Gradient Dyna-style planning converges to min MB-MSPBE even if your model is bad and the model training data distribution and model testing data distribution are different.
4) if your model is perfect and there is no such distribution mismatch, then MB-MSPBE = MSPBE