# RL for continuous spaces, function approximation

Sébastien Gros

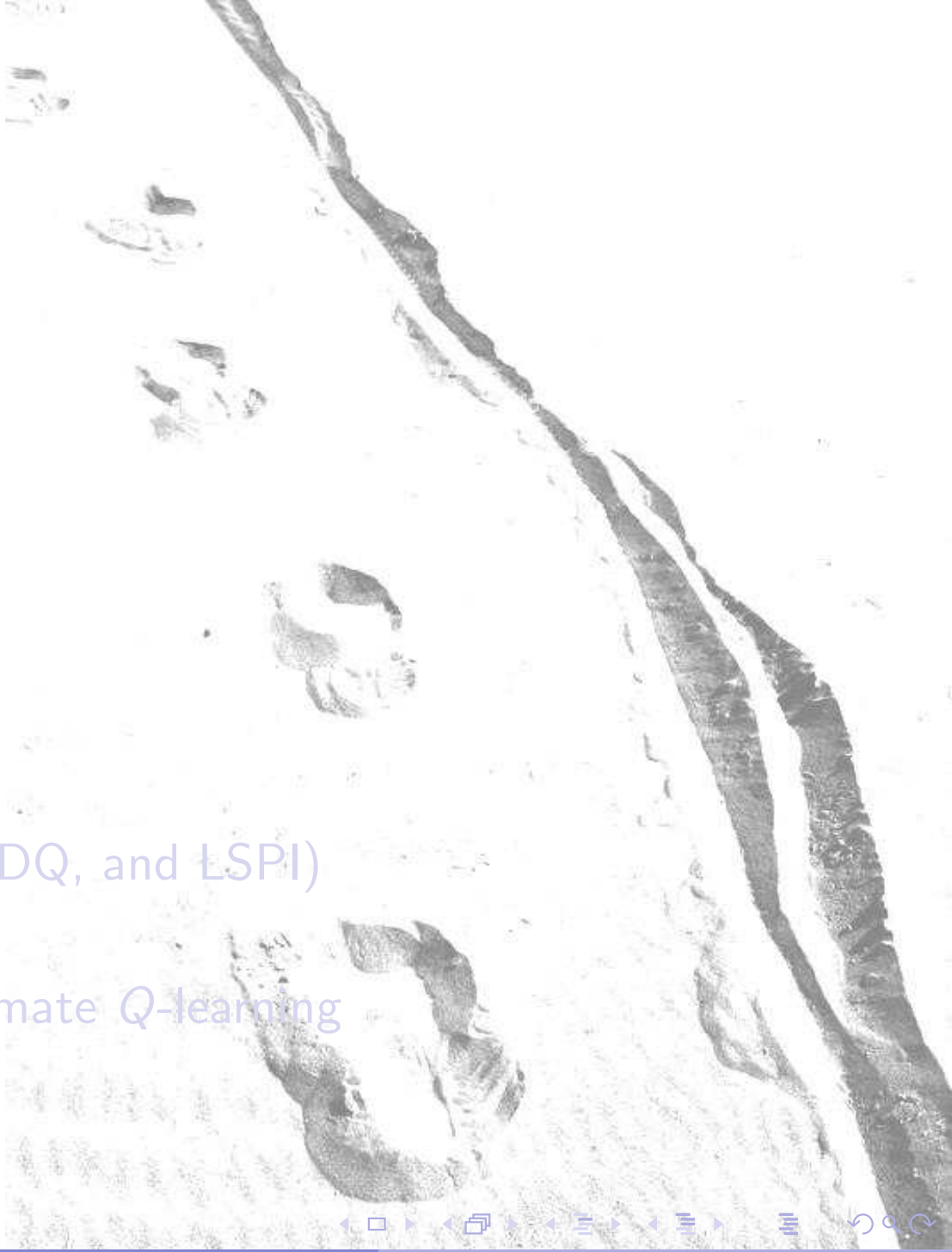
Cybernetic, NTNU
Elec. Eng., Chalmers

TUM lectures on RL

# Outline

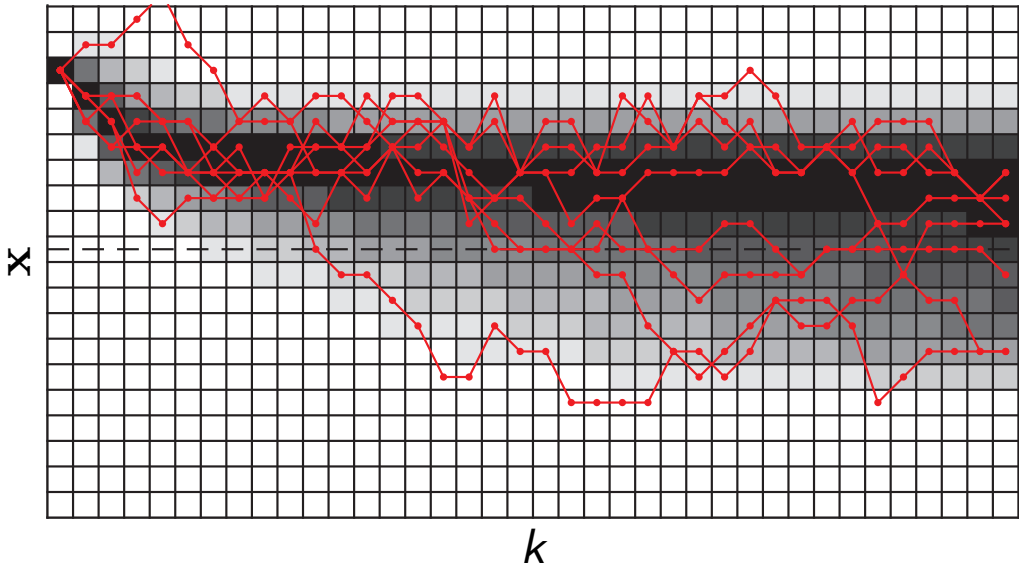
# Outline

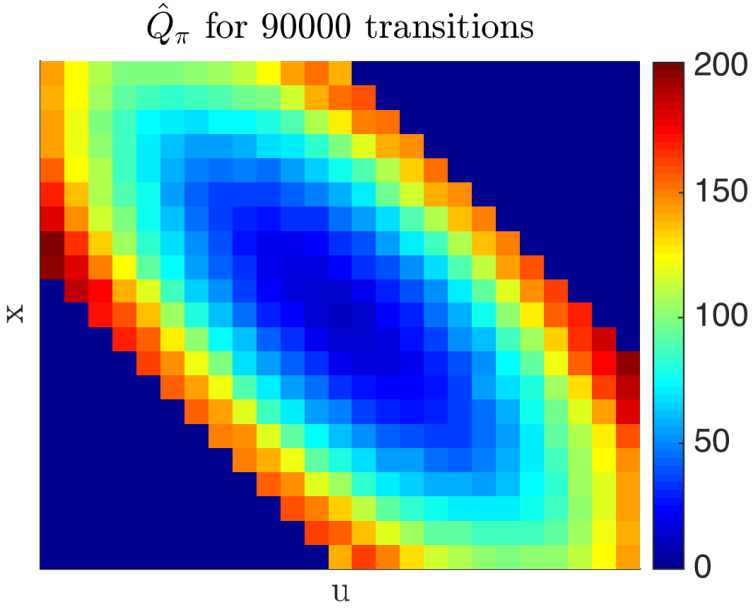




# Why function approximations?

We have so far looked at...




- Example with small discrete state and input spaces, i.e. e.g.

$$\mathbf{x} \in \{-10, -9, \ldots, 9, 10\}, \qquad \mathbf{u} \in \{-12, -11, \ldots, 11, 12\}$$

- Value functions $\hat{V}_\pi$, $\hat{Q}_\pi$, $\hat{V}_\star$, $\hat{Q}_\star$, policy $\pi(\mathbf{x})$, $\pi_\star(\mathbf{x})$ can be handled in a **tabular** fashion, e.g.

$$\hat{V}_\pi(-10) = 2.39, \quad \hat{V}_\pi(-9) = 5.43, \ldots$$

**This approach works only for small discrete spaces, for large and/or continuous spaces, we need to represent all these functions differently**

# Approximating the value functions - Examples

- **Polynomials**, e.g.

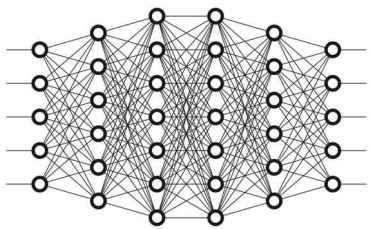
$$\hat{V}_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top S_{\boldsymbol{\theta}}\mathbf{x} + \mathbf{f}_{\boldsymbol{\theta}}^\top \mathbf{x} + c_{\boldsymbol{\theta}}$$

- **Radial Basis Functions**, e.g.

$$\hat{V}(\mathbf{x}) = \sum_{k=1}^{N} \boldsymbol{\theta}_k e^{-\frac{1}{2}\|\mathbf{x}-\mathbf{x}_k\|^2}$$

  where $\mathbf{x}_{1,\ldots,N}$ is a grid of the state space

- **DNN**



- **What else?** *Stay tune...*

# Approximating the value functions - Examples

- **Polynomials**, e.g.

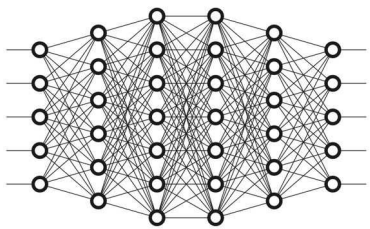
$$\hat{V}_\theta(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top S_\theta \mathbf{x} + \mathbf{f}_\theta^\top \mathbf{x} + c_\theta$$

- **Radial Basis Functions**, e.g.

$$\hat{V}(\mathbf{x}) = \sum_{k=1}^{N} \theta_k e^{-\frac{1}{2}\|\mathbf{x}-\mathbf{x}_k\|^2}$$

  where $\mathbf{x}_{1,\ldots,N}$ is a grid of the state space

- **DNN**



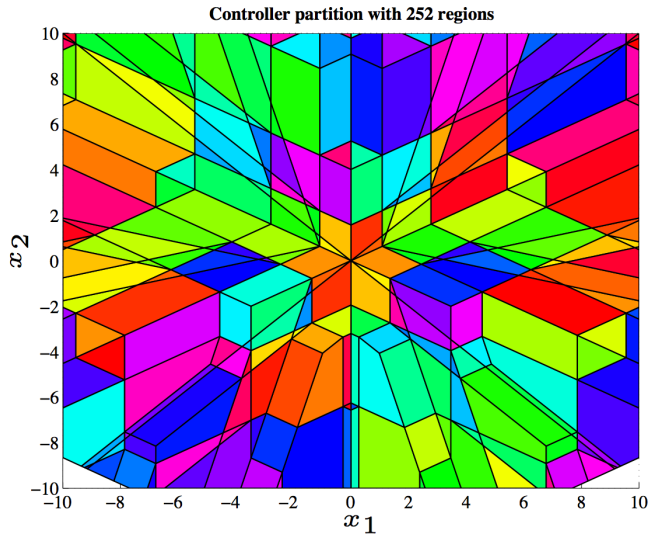
- **What else?** *Stay tune...*

**Remark**: suppose we have

- Linear, deterministic dynamics
- Quadratic cost
- Affine constraints

*then $V_\star$ is piecewise quadratic*

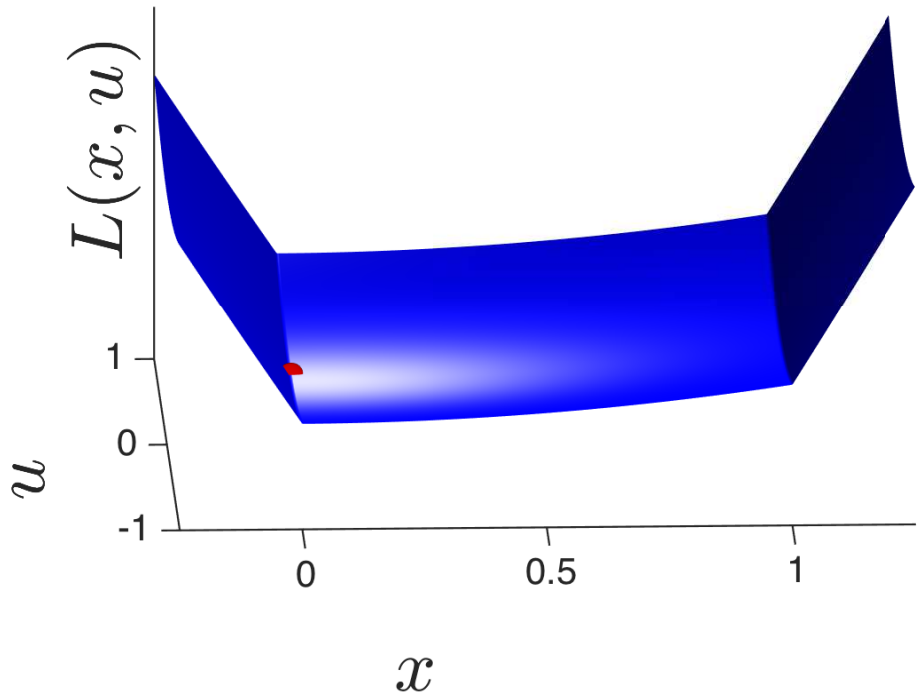
**Regions can be arbitrarily complex...**



# Let's use the following example

- State $\mathbf{x} \in \mathbb{R}$ and input $\mathbf{u} \in [-1, 1]$
- Dynamics: $\mathbf{x}_+ = \mathbf{x} + 0.1\mathbf{u} + \mathbf{e}$ with $\mathbf{e} \sim \mathcal{U}([-0.1, 0])$
- Stage cost: $L(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^2 + \frac{1}{2}\mathbf{u}^2$ + strong penalty for $\mathbf{x} \notin [0, 1]$
- Discount $\gamma = 0.9$
- Noise $\mathbf{e}$ "pushing" $\mathbf{x}$ "against left wall"

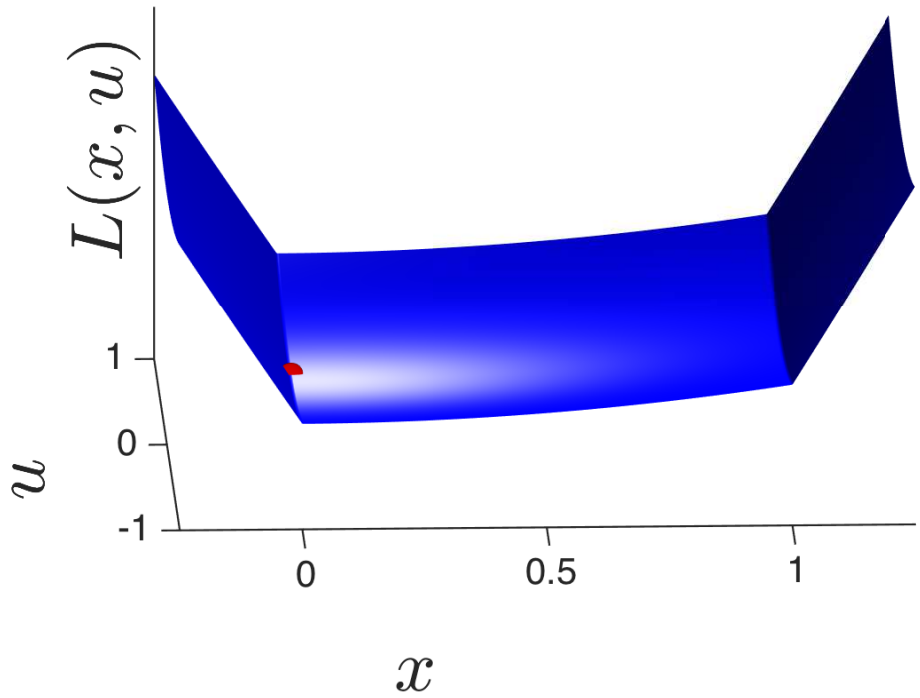
**Stage cost**



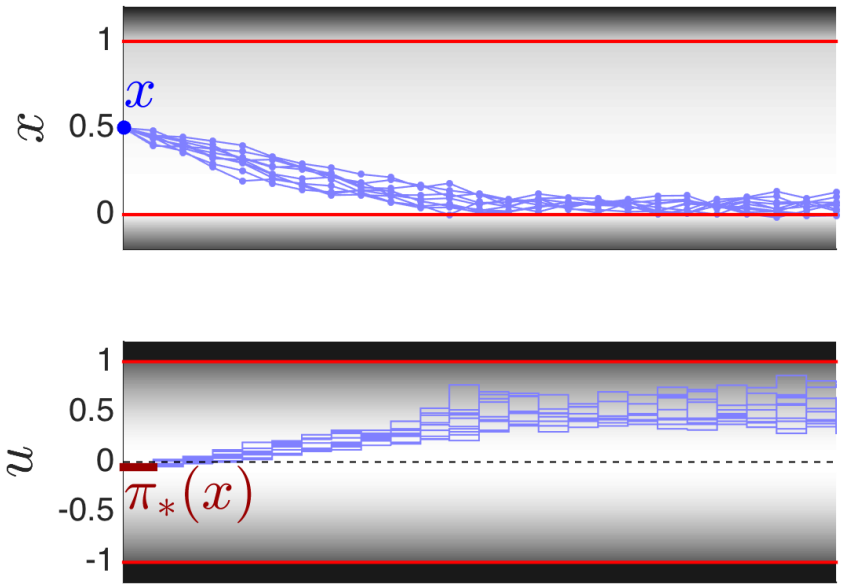
# Let's use the following example

- State $\mathbf{x} \in \mathbb{R}$ and input $\mathbf{u} \in [-1, 1]$
- Dynamics: $\mathbf{x}_+ = \mathbf{x} + 0.1\mathbf{u} + \mathbf{e}$ with $\mathbf{e} \sim \mathcal{U}([-0.1, 0])$
- Stage cost: $L(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^2 + \frac{1}{2}\mathbf{u}^2$ + strong penalty for $\mathbf{x} \notin [0, 1]$
- Discount $\gamma = 0.9$
- Noise $\mathbf{e}$ "pushing" $\mathbf{x}$ "against left wall"

**Stage cost**



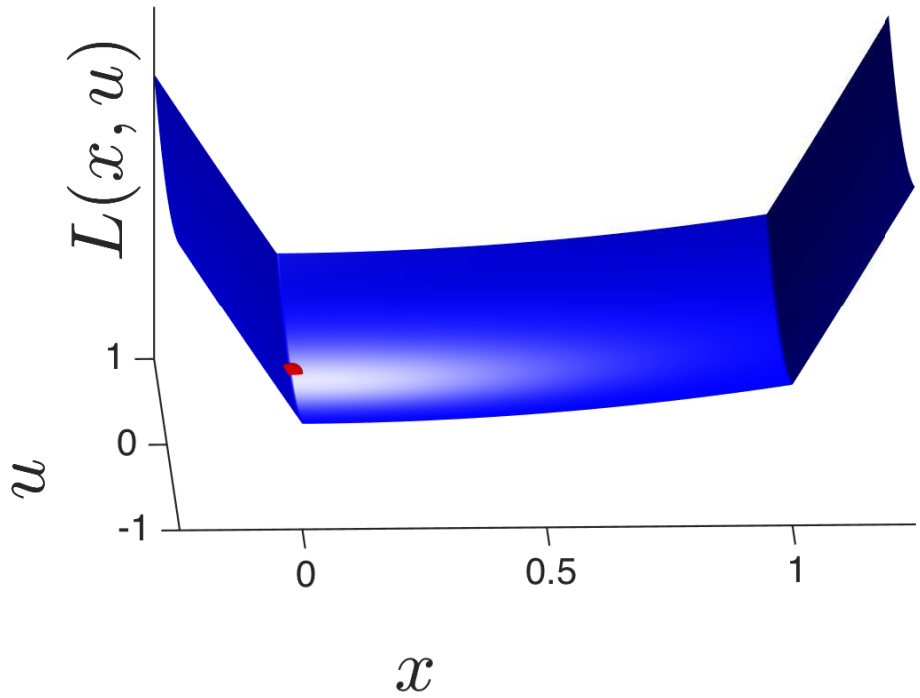

**Trajectories**



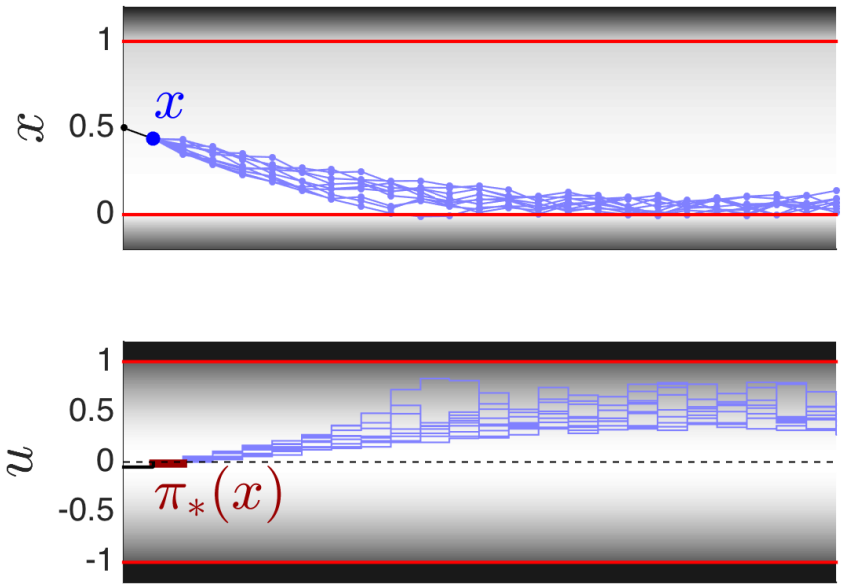
# Let's use the following example

- State $\mathbf{x} \in \mathbb{R}$ and input $\mathbf{u} \in [-1, 1]$
- Dynamics: $\mathbf{x}_+ = \mathbf{x} + 0.1\mathbf{u} + \mathbf{e}$ with $\mathbf{e} \sim \mathcal{U}([-0.1, 0])$
- Stage cost: $L(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^2 + \frac{1}{2}\mathbf{u}^2$ + strong penalty for $\mathbf{x} \notin [0, 1]$
- Discount $\gamma = 0.9$
- Noise $\mathbf{e}$ "pushing" $\mathbf{x}$ "against left wall"

**Stage cost**



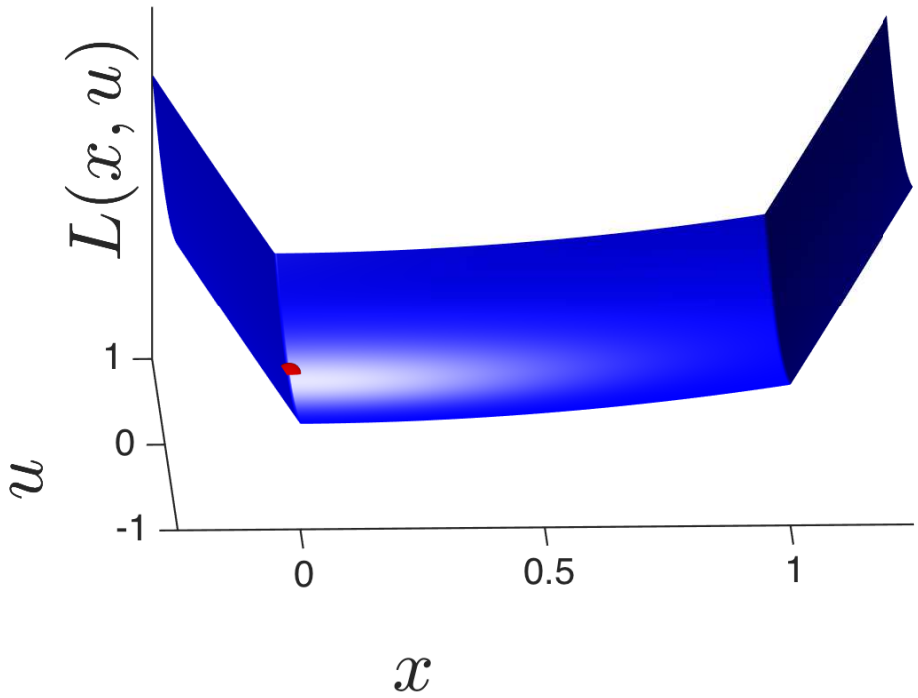

**Trajectories**



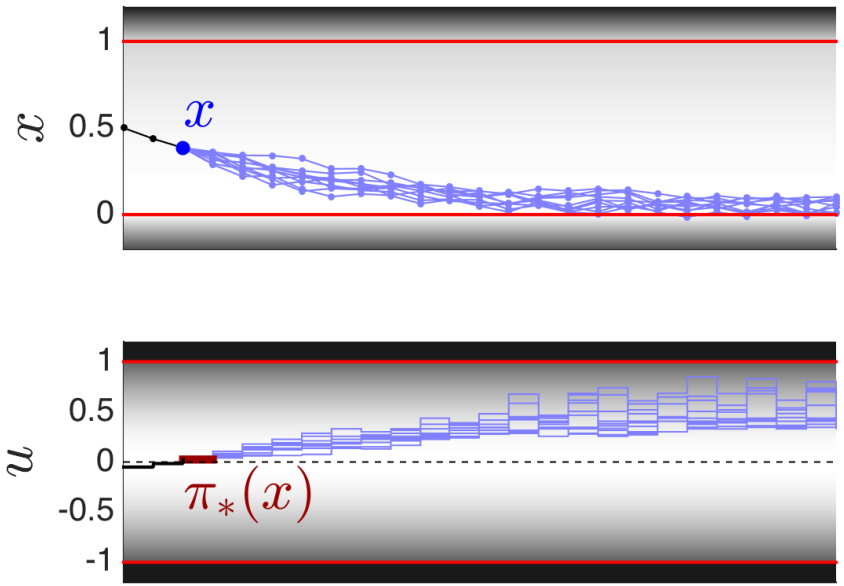
# Let's use the following example

- State $\mathbf{x} \in \mathbb{R}$ and input $\mathbf{u} \in [-1, 1]$
- Dynamics: $\mathbf{x}_+ = \mathbf{x} + 0.1\mathbf{u} + \mathbf{e}$ with $\mathbf{e} \sim \mathcal{U}([-0.1, 0])$
- Stage cost: $L(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^2 + \frac{1}{2}\mathbf{u}^2$ + strong penalty for $\mathbf{x} \notin [0, 1]$
- Discount $\gamma = 0.9$
- Noise $\mathbf{e}$ "pushing" $\mathbf{x}$ "against left wall"

**Stage cost**



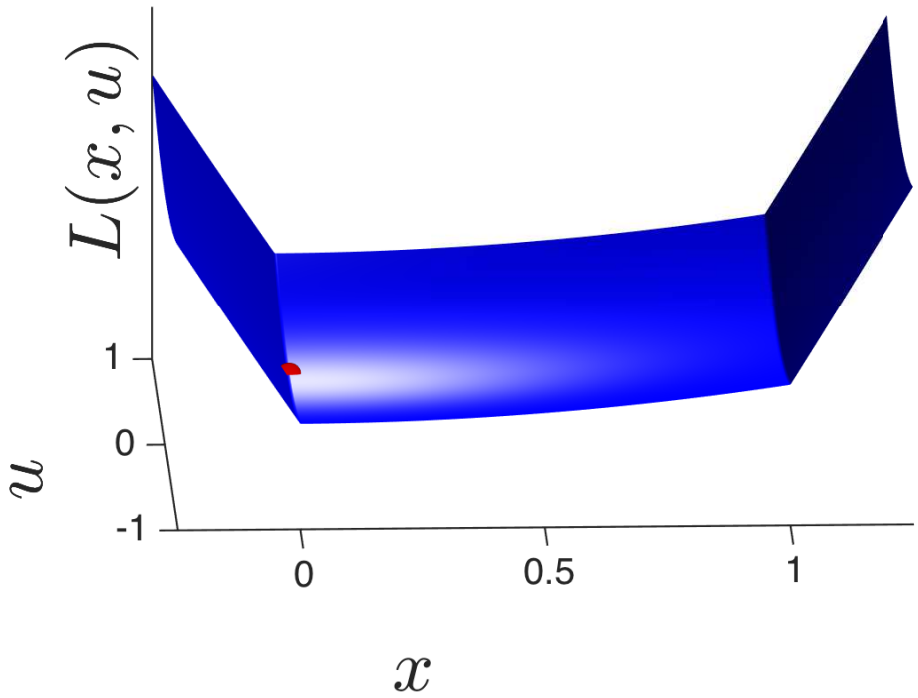

**Trajectories**



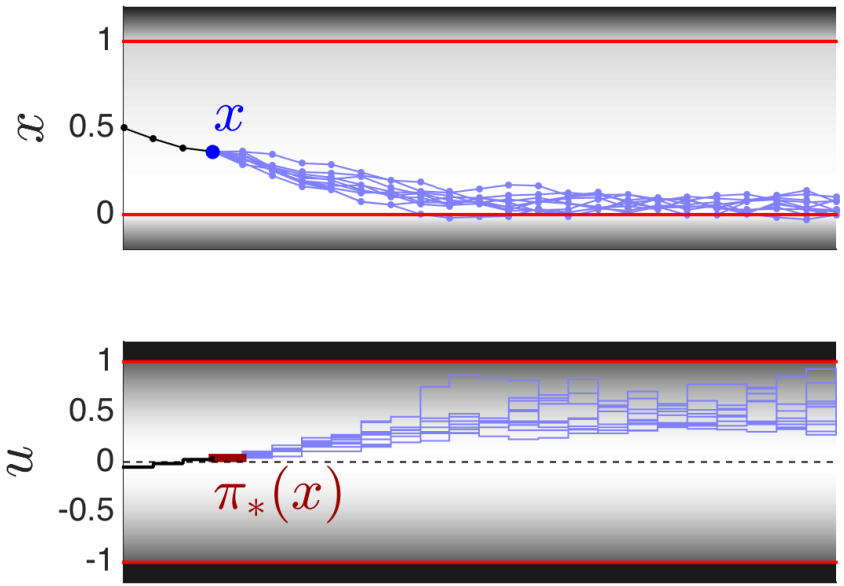
# Let's use the following example

- State $\mathbf{x} \in \mathbb{R}$ and input $\mathbf{u} \in [-1, 1]$
- Dynamics: $\mathbf{x}_+ = \mathbf{x} + 0.1\mathbf{u} + \mathbf{e}$ with $\mathbf{e} \sim \mathcal{U}([-0.1, 0])$
- Stage cost: $L(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^2 + \frac{1}{2}\mathbf{u}^2$ + strong penalty for $\mathbf{x} \notin [0, 1]$
- Discount $\gamma = 0.9$
- Noise $\mathbf{e}$ "pushing" $\mathbf{x}$ "against left wall"

**Stage cost**



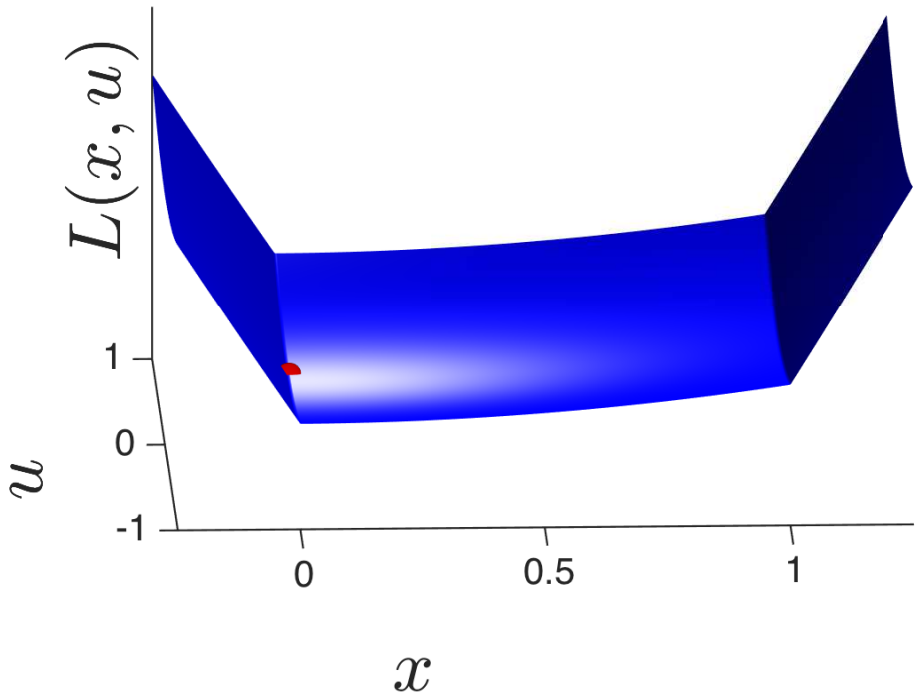

**Trajectories**



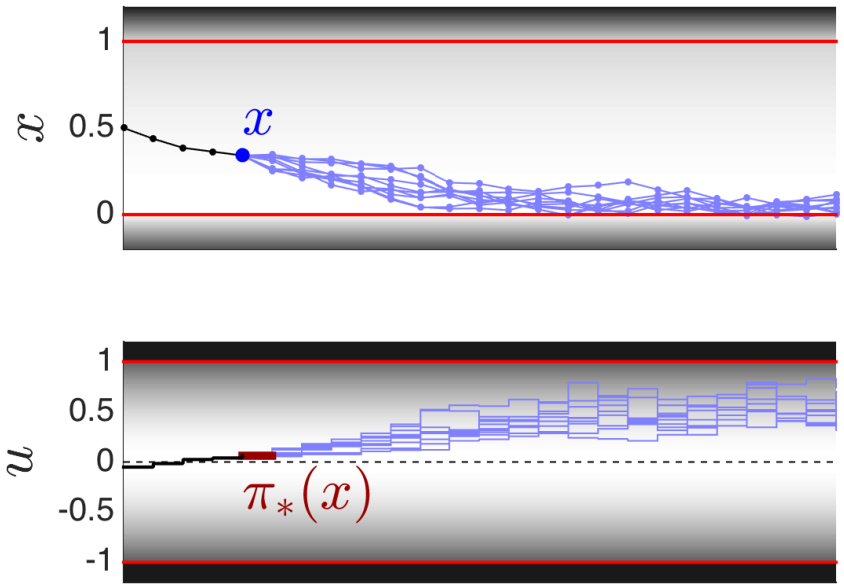
# Let's use the following example

- State $\mathbf{x} \in \mathbb{R}$ and input $\mathbf{u} \in [-1, 1]$
- Dynamics: $\mathbf{x}_+ = \mathbf{x} + 0.1\mathbf{u} + \mathbf{e}$ with $\mathbf{e} \sim \mathcal{U}([-0.1, 0])$
- Stage cost: $L(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^2 + \frac{1}{2}\mathbf{u}^2$ + strong penalty for $\mathbf{x} \notin [0, 1]$
- Discount $\gamma = 0.9$
- Noise $\mathbf{e}$ "pushing" $\mathbf{x}$ "against left wall"

**Stage cost**



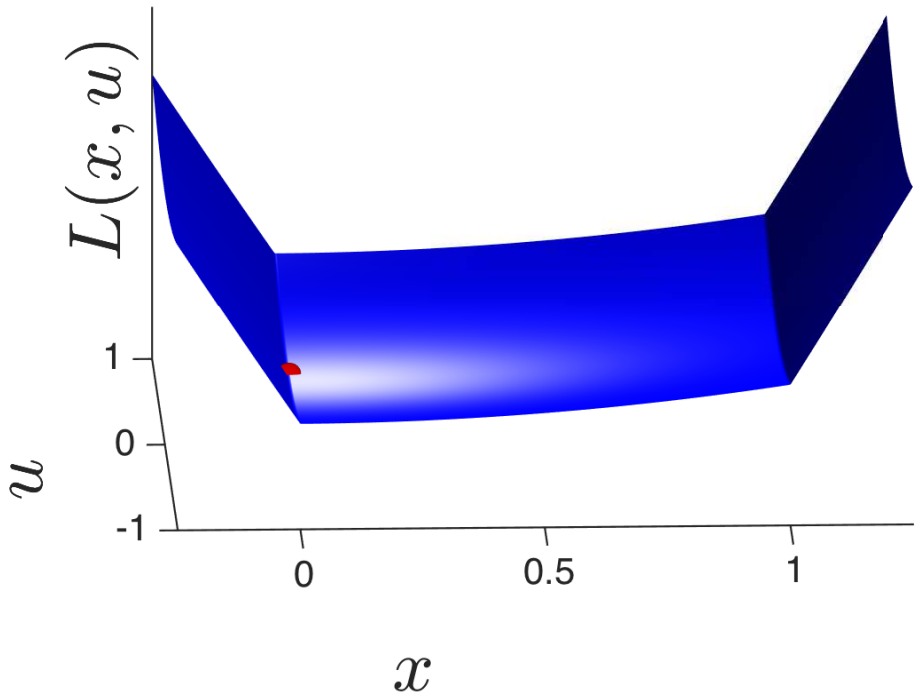

**Trajectories**



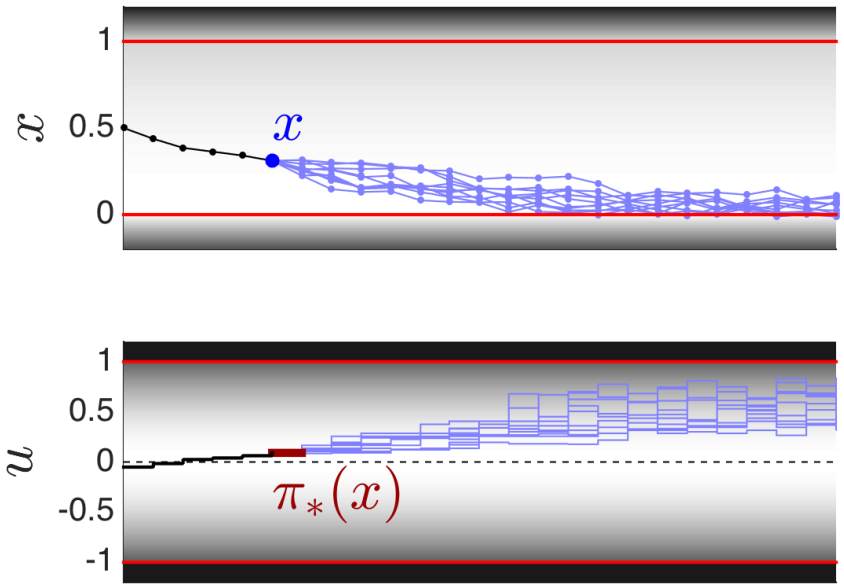
# Let's use the following example

- State $\mathbf{x} \in \mathbb{R}$ and input $\mathbf{u} \in [-1, 1]$
- Dynamics: $\mathbf{x}_+ = \mathbf{x} + 0.1\mathbf{u} + \mathbf{e}$ with $\mathbf{e} \sim \mathcal{U}([-0.1, 0])$
- Stage cost: $L(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^2 + \frac{1}{2}\mathbf{u}^2$ + strong penalty for $\mathbf{x} \notin [0, 1]$
- Discount $\gamma = 0.9$
- Noise $\mathbf{e}$ "pushing" $\mathbf{x}$ "against left wall"

**Stage cost**



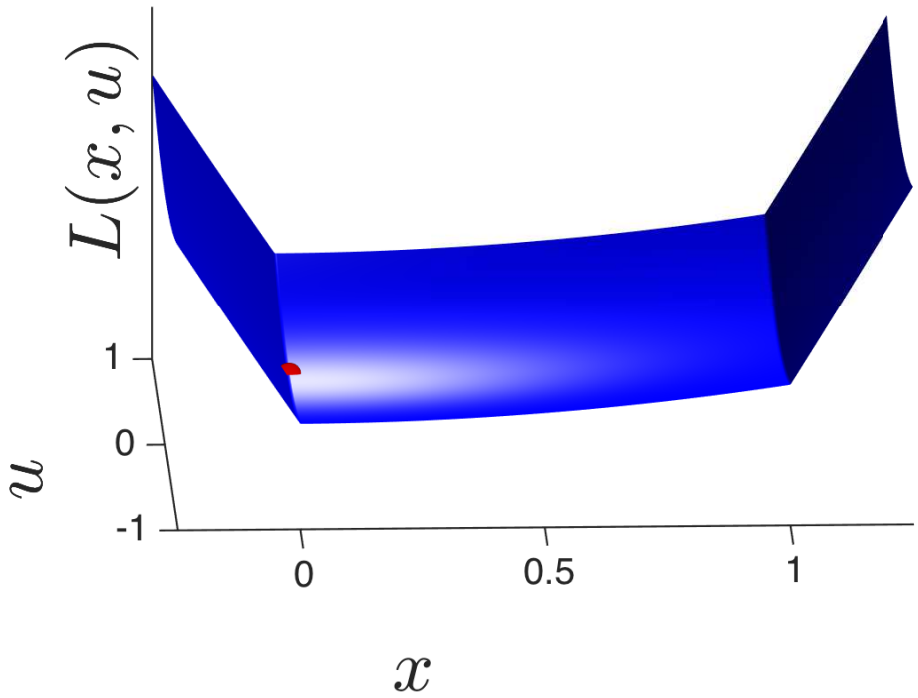

**Trajectories**



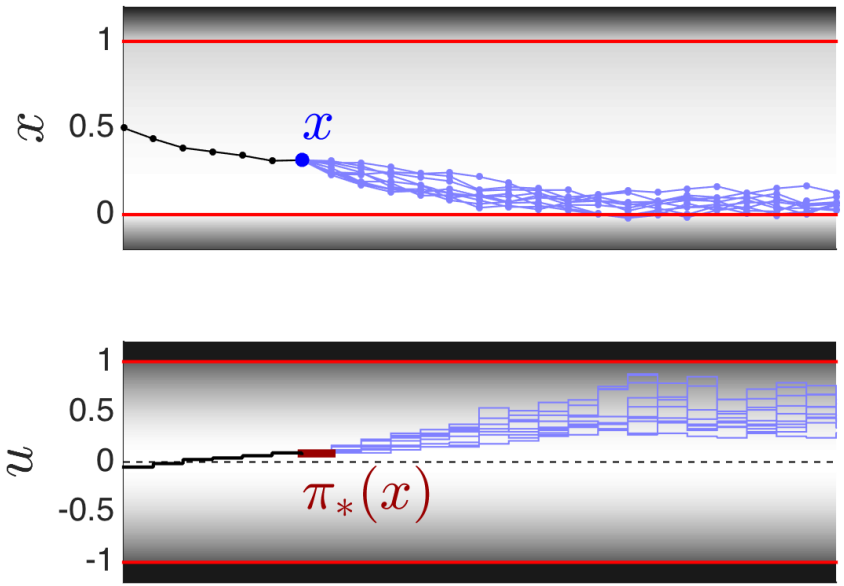
# Let's use the following example

- State $\mathbf{x} \in \mathbb{R}$ and input $\mathbf{u} \in [-1, 1]$
- Dynamics: $\mathbf{x}_+ = \mathbf{x} + 0.1\mathbf{u} + \mathbf{e}$ with $\mathbf{e} \sim \mathcal{U}([-0.1, 0])$
- Stage cost: $L(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^2 + \frac{1}{2}\mathbf{u}^2$ + strong penalty for $\mathbf{x} \notin [0, 1]$
- Discount $\gamma = 0.9$
- Noise $\mathbf{e}$ "pushing" $\mathbf{x}$ "against left wall"

**Stage cost**



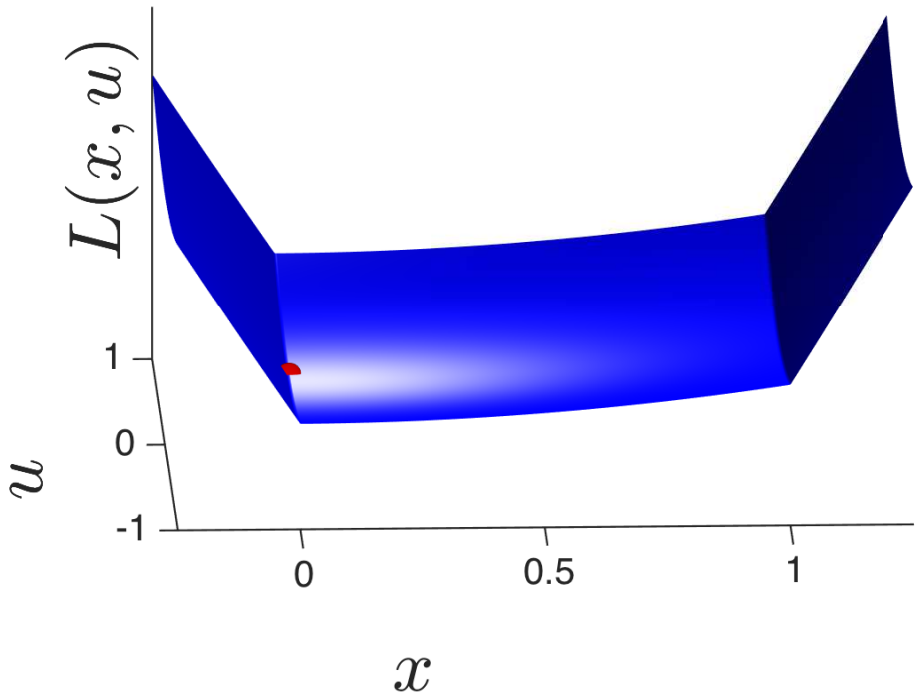

**Trajectories**



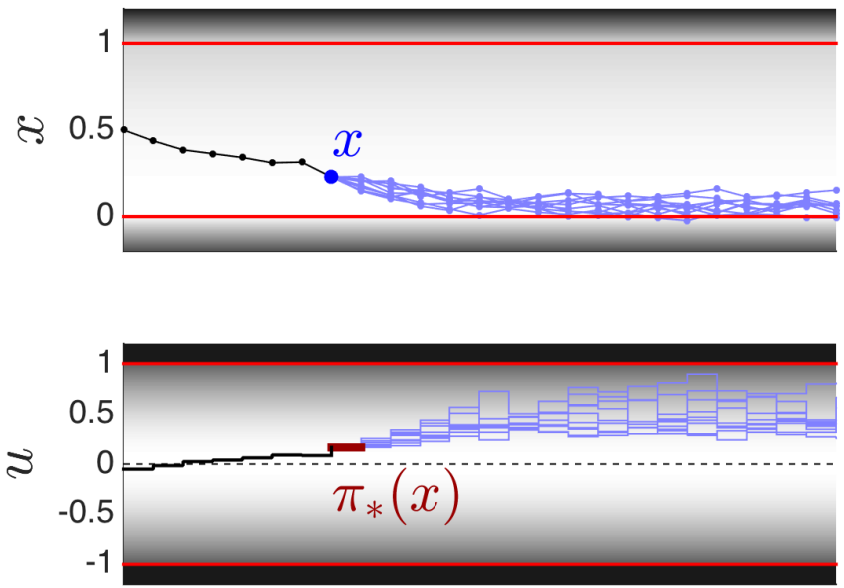
# Let's use the following example

- State $\mathbf{x} \in \mathbb{R}$ and input $\mathbf{u} \in [-1, 1]$
- Dynamics: $\mathbf{x}_+ = \mathbf{x} + 0.1\mathbf{u} + \mathbf{e}$ with $\mathbf{e} \sim \mathcal{U}([-0.1, 0])$
- Stage cost: $L(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^2 + \frac{1}{2}\mathbf{u}^2$ + strong penalty for $\mathbf{x} \notin [0, 1]$
- Discount $\gamma = 0.9$
- Noise $\mathbf{e}$ "pushing" $\mathbf{x}$ "against left wall"

**Stage cost**



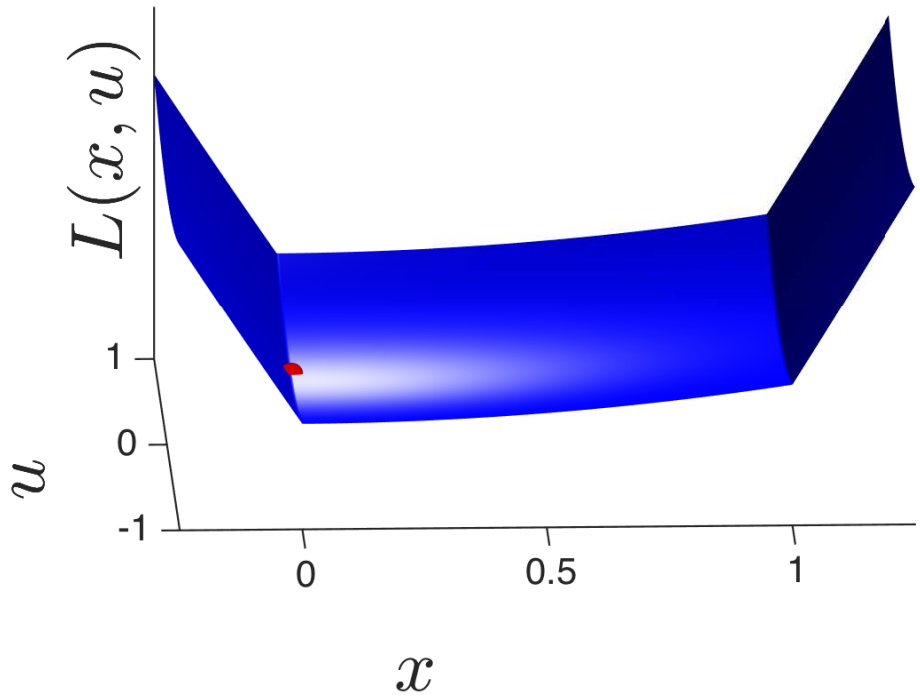

**Trajectories**



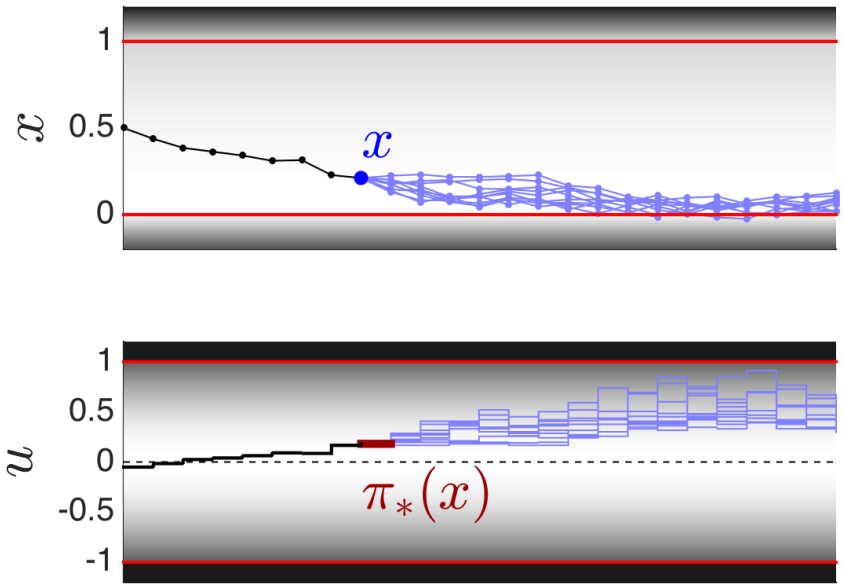
# Let's use the following example

- State $\mathbf{x} \in \mathbb{R}$ and input $\mathbf{u} \in [-1, 1]$
- Dynamics: $\mathbf{x}_+ = \mathbf{x} + 0.1\mathbf{u} + \mathbf{e}$ with $\mathbf{e} \sim \mathcal{U}([-0.1, 0])$
- Stage cost: $L(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^2 + \frac{1}{2}\mathbf{u}^2$ + strong penalty for $\mathbf{x} \notin [0, 1]$
- Discount $\gamma = 0.9$
- Noise $\mathbf{e}$ "pushing" $\mathbf{x}$ "against left wall"

**Stage cost**



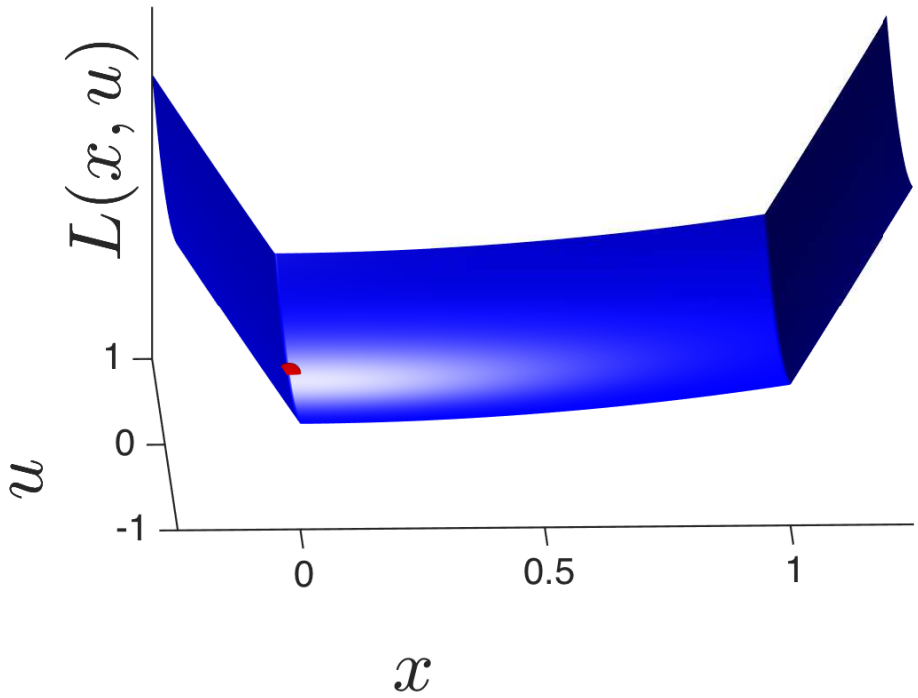

**Trajectories**



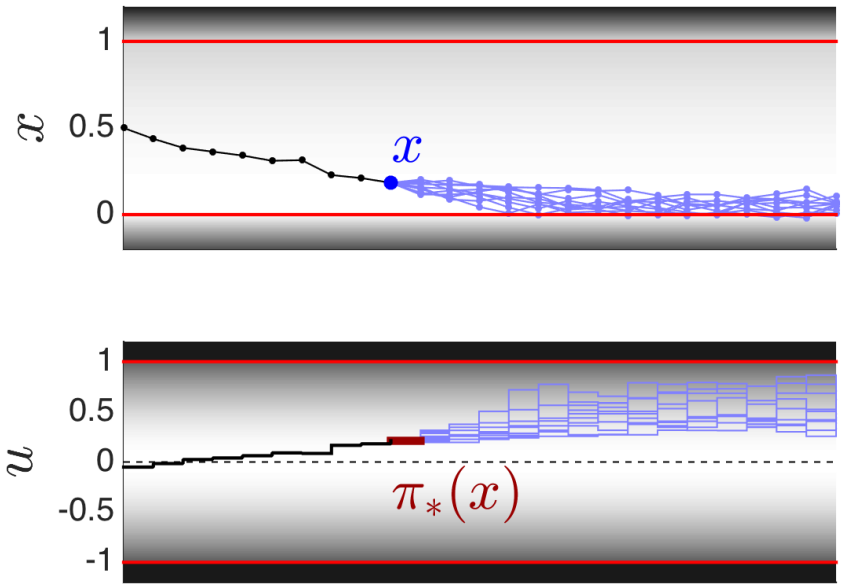
# Let's use the following example

- State $\mathbf{x} \in \mathbb{R}$ and input $\mathbf{u} \in [-1, 1]$
- Dynamics: $\mathbf{x}_+ = \mathbf{x} + 0.1\mathbf{u} + \mathbf{e}$ with $\mathbf{e} \sim \mathcal{U}([-0.1, 0])$
- Stage cost: $L(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^2 + \frac{1}{2}\mathbf{u}^2$ + strong penalty for $\mathbf{x} \notin [0, 1]$
- Discount $\gamma = 0.9$
- Noise $\mathbf{e}$ "pushing" $\mathbf{x}$ "against left wall"

**Stage cost**



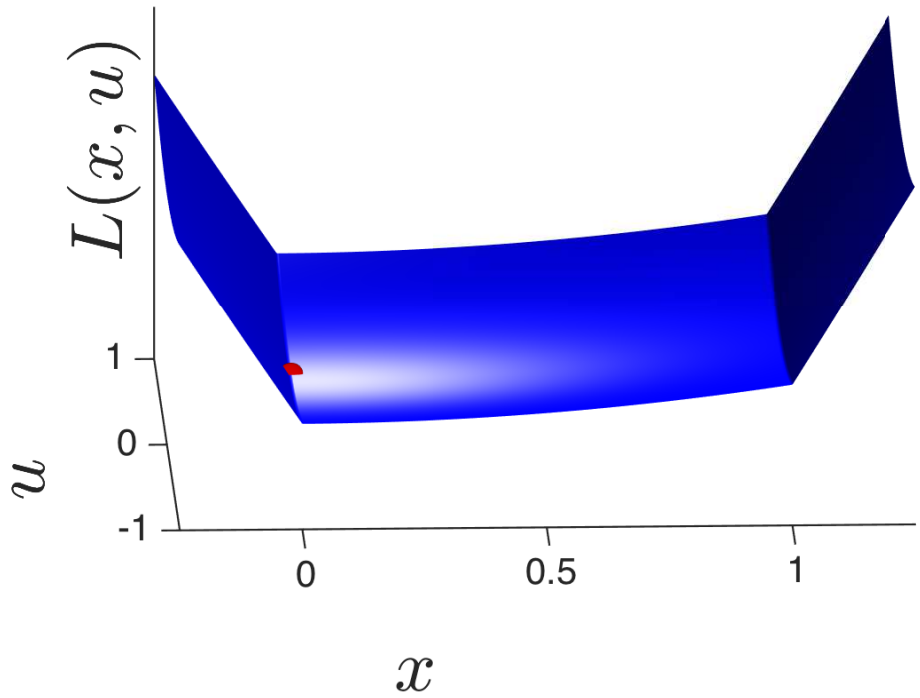

**Trajectories**



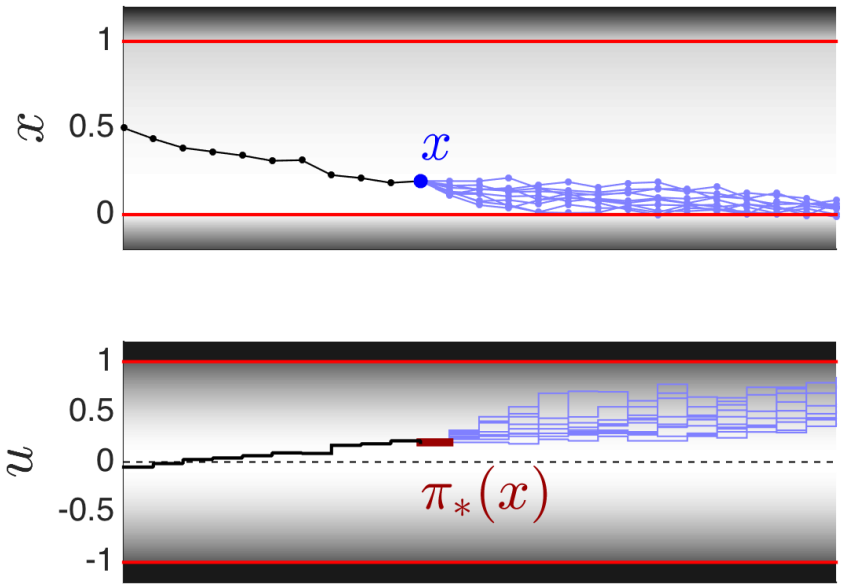
# Let's use the following example

- State $\mathbf{x} \in \mathbb{R}$ and input $\mathbf{u} \in [-1, 1]$
- Dynamics: $\mathbf{x}_+ = \mathbf{x} + 0.1\mathbf{u} + \mathbf{e}$ with $\mathbf{e} \sim \mathcal{U}([-0.1, 0])$
- Stage cost: $L(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^2 + \frac{1}{2}\mathbf{u}^2$ + strong penalty for $\mathbf{x} \notin [0, 1]$
- Discount $\gamma = 0.9$
- Noise $\mathbf{e}$ "pushing" $\mathbf{x}$ "against left wall"

**Stage cost**



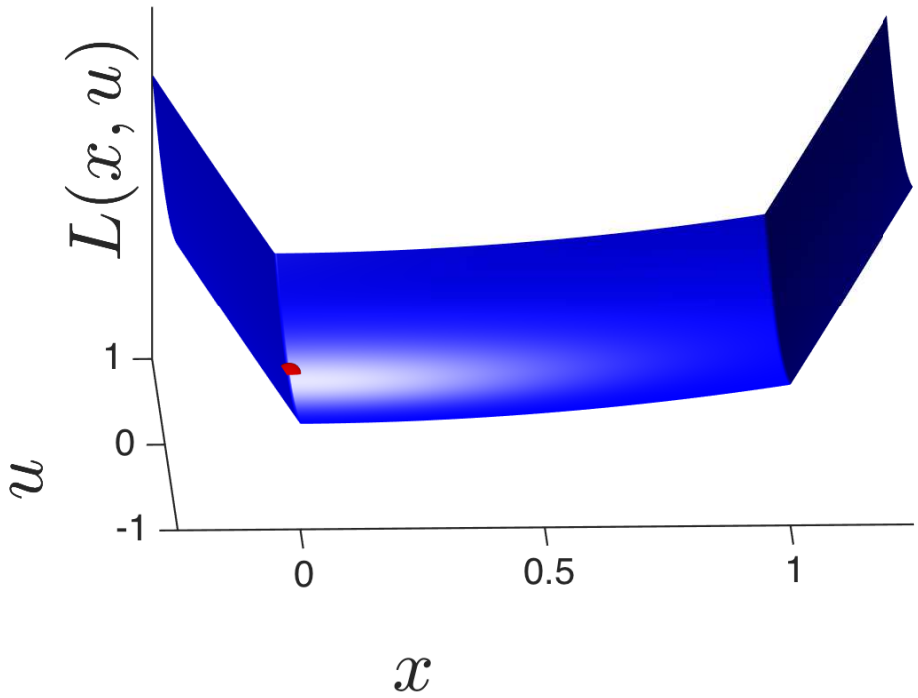

**Trajectories**



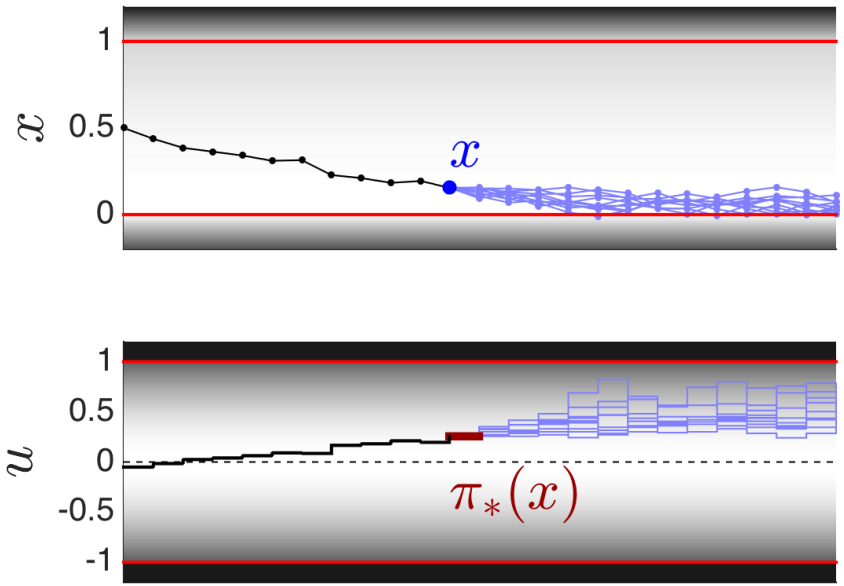
# Let's use the following example

- State $\mathbf{x} \in \mathbb{R}$ and input $\mathbf{u} \in [-1, 1]$
- Dynamics: $\mathbf{x}_+ = \mathbf{x} + 0.1\mathbf{u} + \mathbf{e}$ with $\mathbf{e} \sim \mathcal{U}([-0.1, 0])$
- Stage cost: $L(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^2 + \frac{1}{2}\mathbf{u}^2$ + strong penalty for $\mathbf{x} \notin [0, 1]$
- Discount $\gamma = 0.9$
- Noise $\mathbf{e}$ "pushing" $\mathbf{x}$ "against left wall"

**Stage cost**



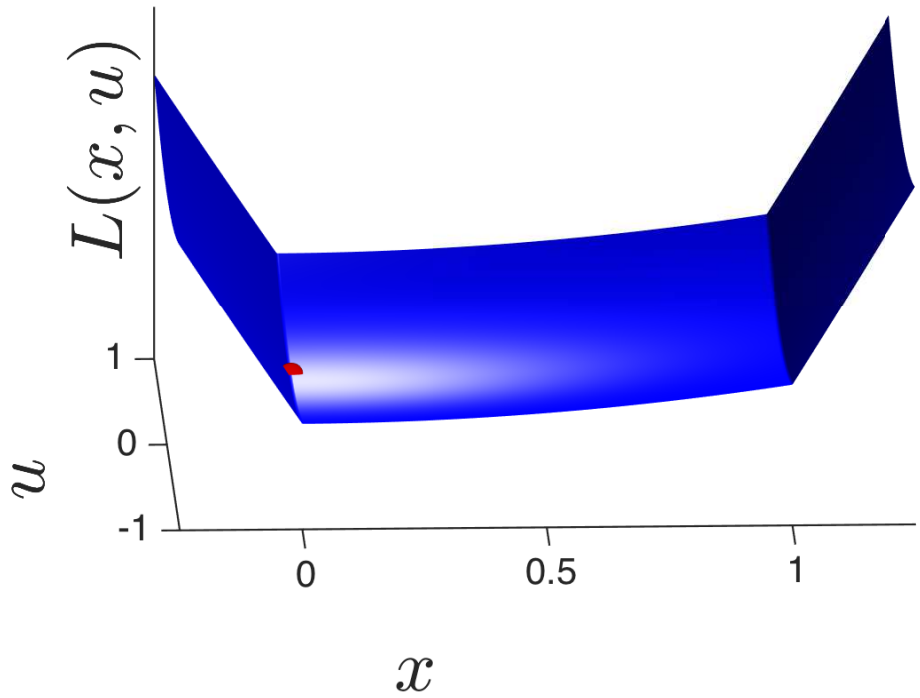

**Trajectories**



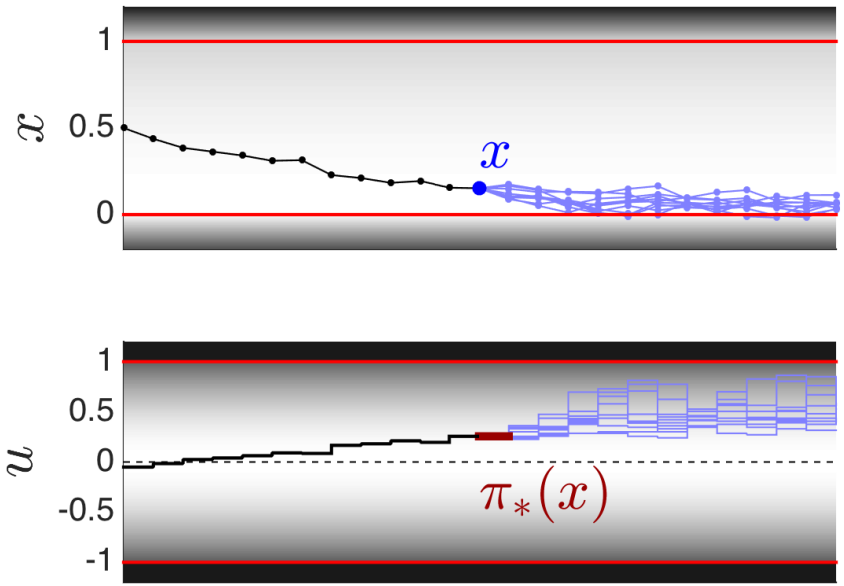
# Let's use the following example

- State $\mathbf{x} \in \mathbb{R}$ and input $\mathbf{u} \in [-1, 1]$
- Dynamics: $\mathbf{x}_+ = \mathbf{x} + 0.1\mathbf{u} + \mathbf{e}$ with $\mathbf{e} \sim \mathcal{U}([-0.1, 0])$
- Stage cost: $L(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^2 + \frac{1}{2}\mathbf{u}^2$ + strong penalty for $\mathbf{x} \notin [0, 1]$
- Discount $\gamma = 0.9$
- Noise $\mathbf{e}$ "pushing" $\mathbf{x}$ "against left wall"

**Stage cost**



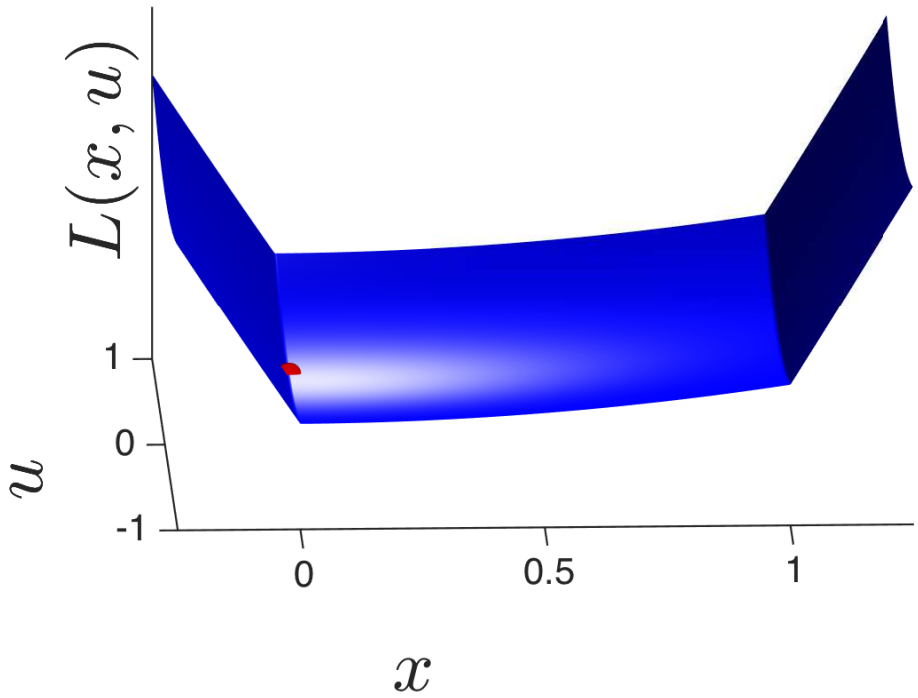

**Trajectories**



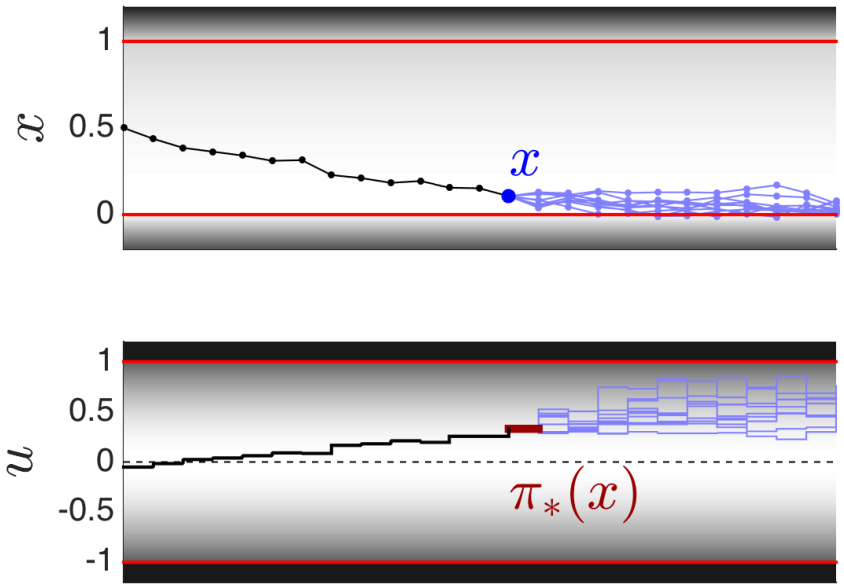
# Let's use the following example

- State $\mathbf{x} \in \mathbb{R}$ and input $\mathbf{u} \in [-1, 1]$
- Dynamics: $\mathbf{x}_+ = \mathbf{x} + 0.1\mathbf{u} + \mathbf{e}$ with $\mathbf{e} \sim \mathcal{U}([-0.1, 0])$
- Stage cost: $L(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^2 + \frac{1}{2}\mathbf{u}^2$ + strong penalty for $\mathbf{x} \notin [0, 1]$
- Discount $\gamma = 0.9$
- Noise $\mathbf{e}$ "pushing" $\mathbf{x}$ "against left wall"

**Stage cost**



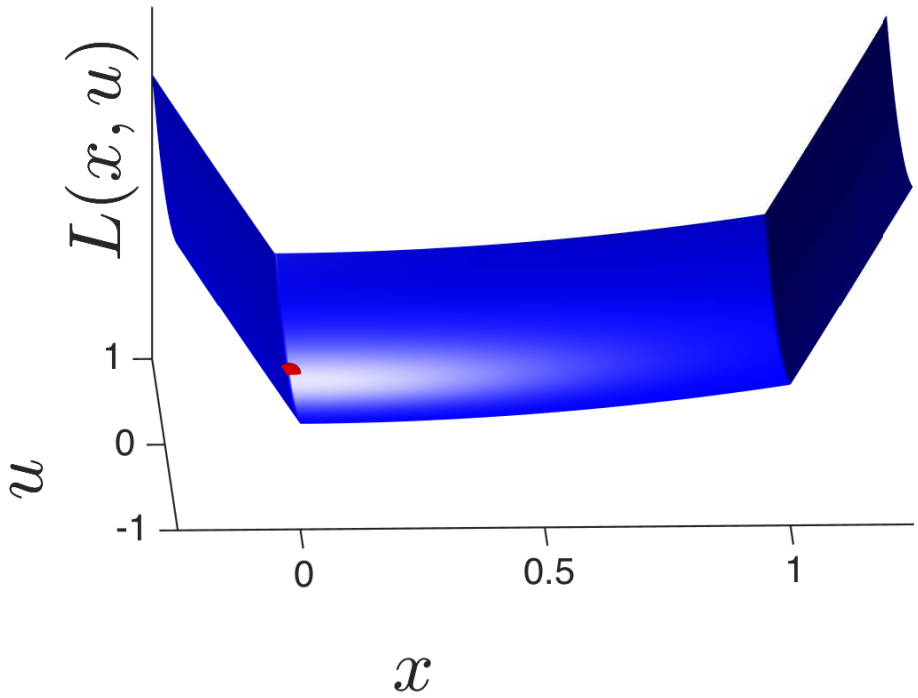

**Trajectories**



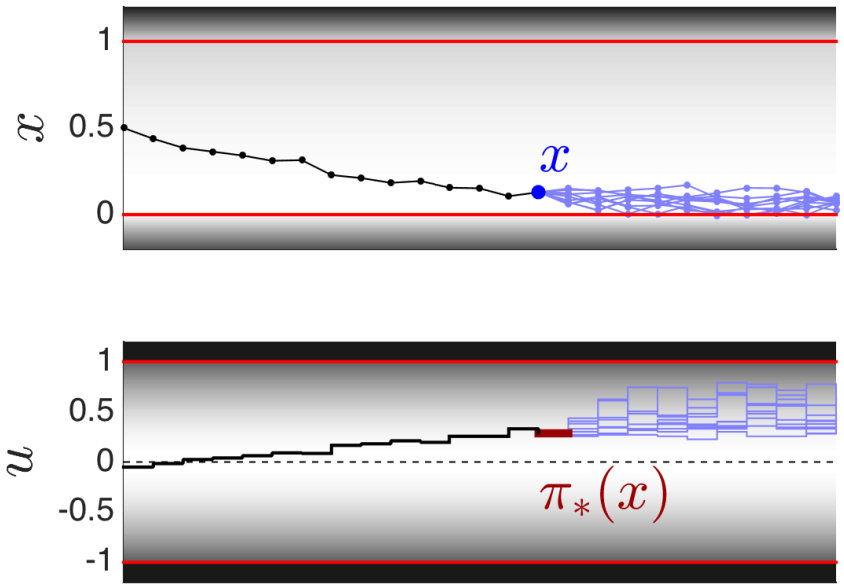
# Let's use the following example

- State $\mathbf{x} \in \mathbb{R}$ and input $\mathbf{u} \in [-1, 1]$
- Dynamics: $\mathbf{x}_+ = \mathbf{x} + 0.1\mathbf{u} + \mathbf{e}$ with $\mathbf{e} \sim \mathcal{U}([-0.1, 0])$
- Stage cost: $L(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^2 + \frac{1}{2}\mathbf{u}^2$ + strong penalty for $\mathbf{x} \notin [0, 1]$
- Discount $\gamma = 0.9$
- Noise $\mathbf{e}$ "pushing" $\mathbf{x}$ "against left wall"

**Stage cost**



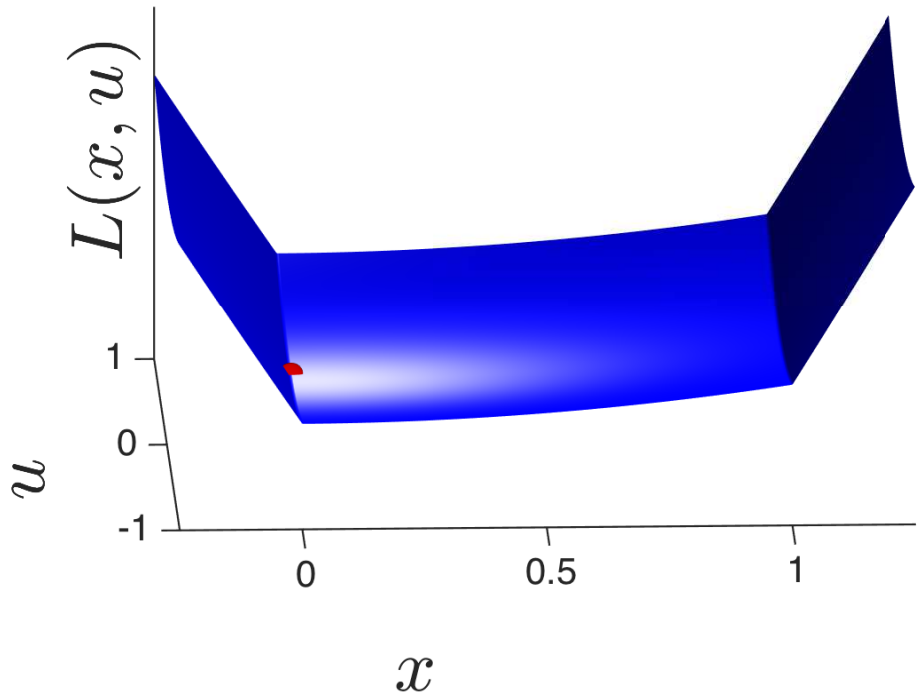

**Trajectories**



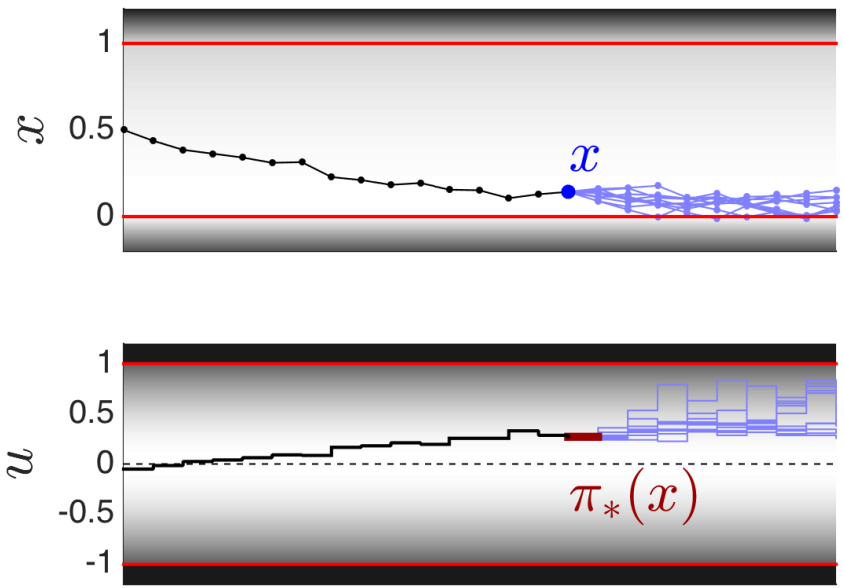
# Let's use the following example

- State $\mathbf{x} \in \mathbb{R}$ and input $\mathbf{u} \in [-1, 1]$
- Dynamics: $\mathbf{x}_+ = \mathbf{x} + 0.1\mathbf{u} + \mathbf{e}$ with $\mathbf{e} \sim \mathcal{U}([-0.1, 0])$
- Stage cost: $L(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^2 + \frac{1}{2}\mathbf{u}^2$ + strong penalty for $\mathbf{x} \notin [0, 1]$
- Discount $\gamma = 0.9$
- Noise $\mathbf{e}$ "pushing" $\mathbf{x}$ "against left wall"

**Stage cost**



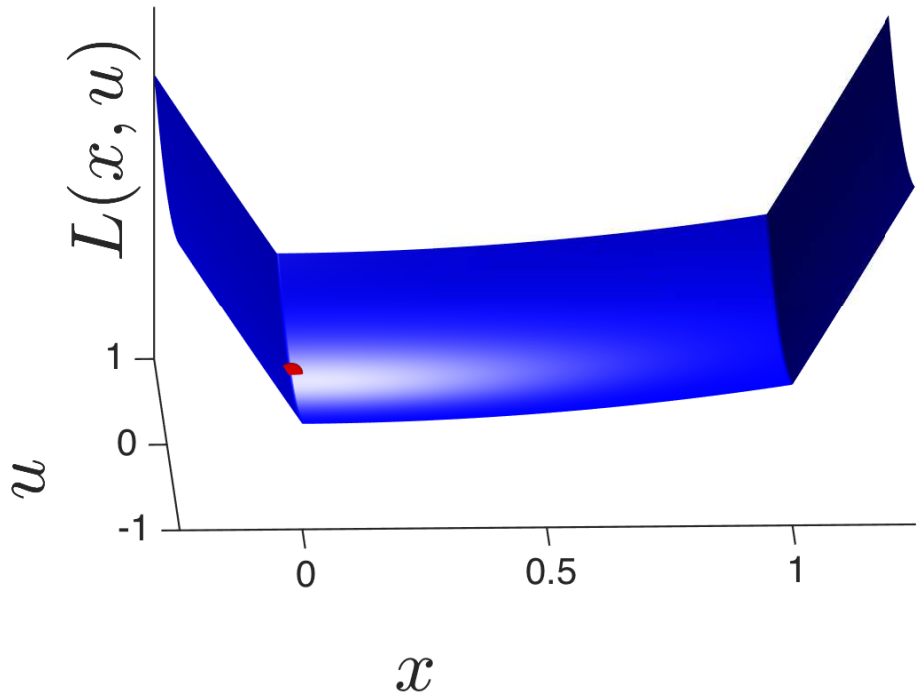

**Trajectories**



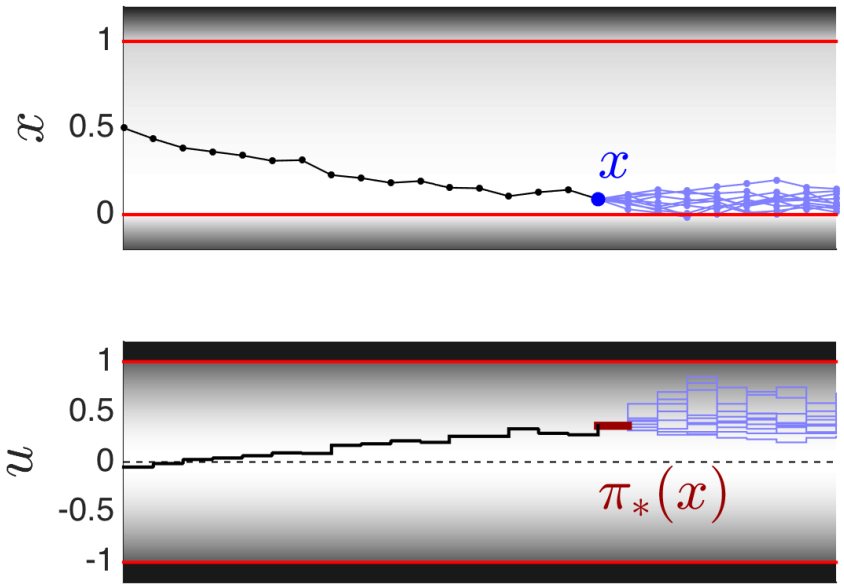
# Let's use the following example

- State $\mathbf{x} \in \mathbb{R}$ and input $\mathbf{u} \in [-1, 1]$
- Dynamics: $\mathbf{x}_+ = \mathbf{x} + 0.1\mathbf{u} + \mathbf{e}$ with $\mathbf{e} \sim \mathcal{U}([-0.1, 0])$
- Stage cost: $L(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^2 + \frac{1}{2}\mathbf{u}^2$ + strong penalty for $\mathbf{x} \notin [0, 1]$
- Discount $\gamma = 0.9$
- Noise $\mathbf{e}$ "pushing" $\mathbf{x}$ "against left wall"

**Stage cost**



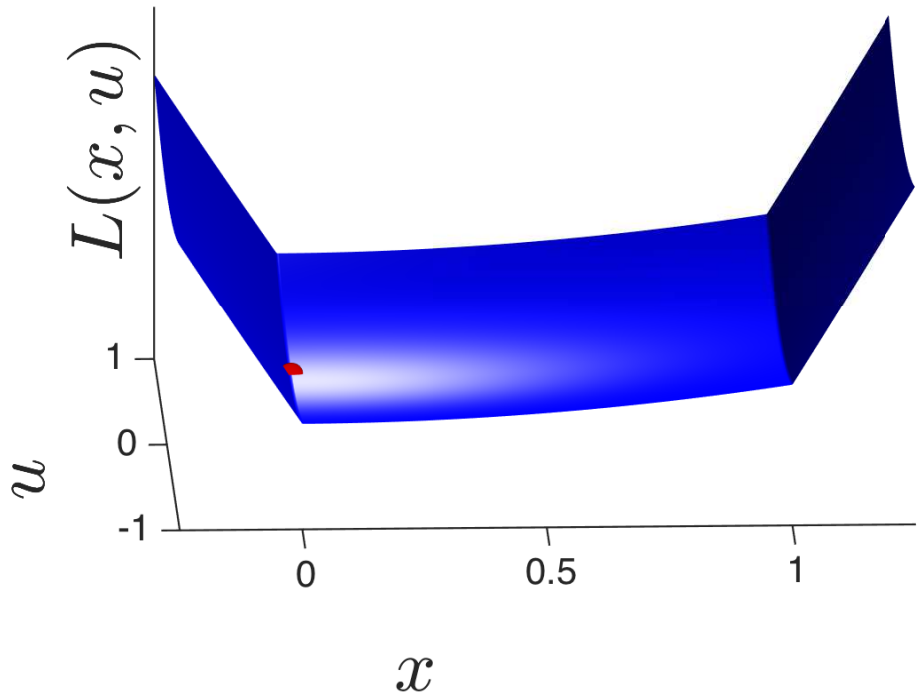

**Trajectories**



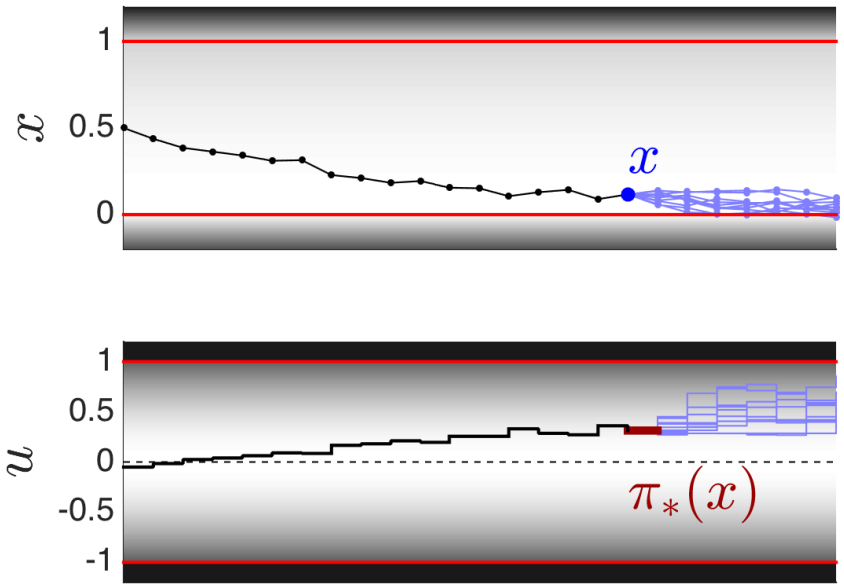
# Let's use the following example

- State $\mathbf{x} \in \mathbb{R}$ and input $\mathbf{u} \in [-1, 1]$
- Dynamics: $\mathbf{x}_+ = \mathbf{x} + 0.1\mathbf{u} + \mathbf{e}$ with $\mathbf{e} \sim \mathcal{U}([-0.1, 0])$
- Stage cost: $L(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^2 + \frac{1}{2}\mathbf{u}^2$ + strong penalty for $\mathbf{x} \notin [0, 1]$
- Discount $\gamma = 0.9$
- Noise $\mathbf{e}$ "pushing" $\mathbf{x}$ "against left wall"

**Stage cost**



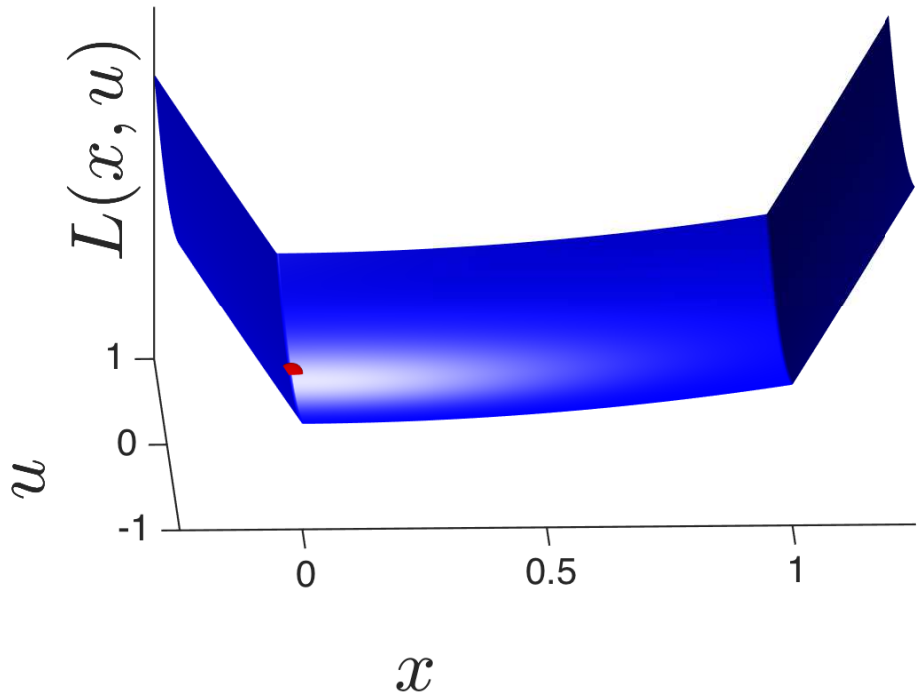

**Trajectories**



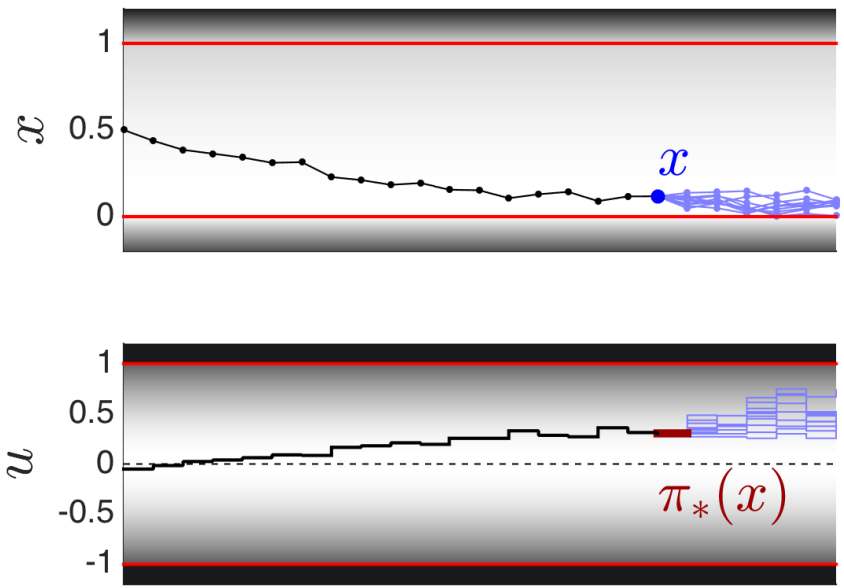
# Let's use the following example

- State $\mathbf{x} \in \mathbb{R}$ and input $\mathbf{u} \in [-1, 1]$
- Dynamics: $\mathbf{x}_+ = \mathbf{x} + 0.1\mathbf{u} + \mathbf{e}$ with $\mathbf{e} \sim \mathcal{U}([-0.1, 0])$
- Stage cost: $L(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^2 + \frac{1}{2}\mathbf{u}^2$ + strong penalty for $\mathbf{x} \notin [0, 1]$
- Discount $\gamma = 0.9$
- Noise $\mathbf{e}$ "pushing" $\mathbf{x}$ "against left wall"

**Stage cost**



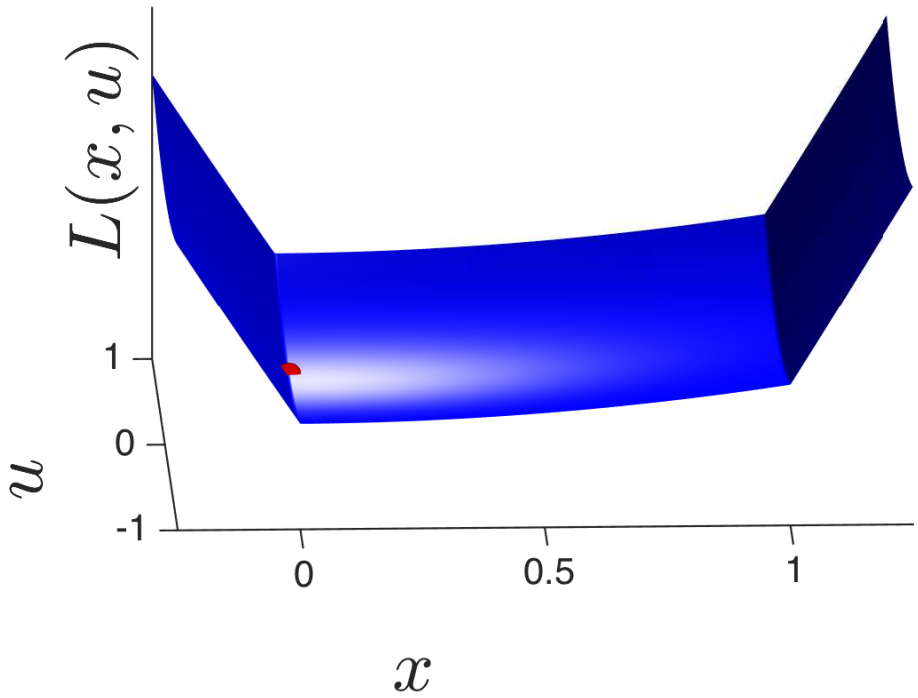

**Trajectories**



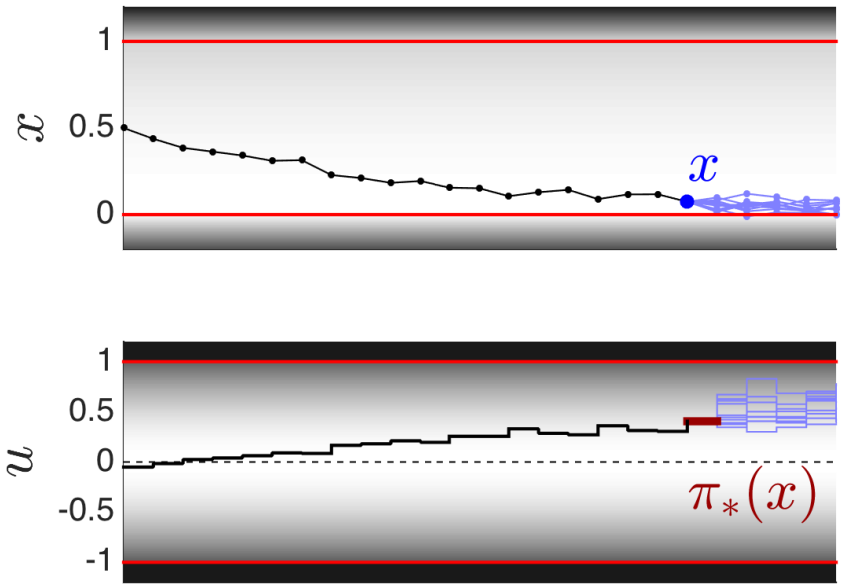
# Let's use the following example

- State $\mathbf{x} \in \mathbb{R}$ and input $\mathbf{u} \in [-1, 1]$
- Dynamics: $\mathbf{x}_+ = \mathbf{x} + 0.1\mathbf{u} + \mathbf{e}$ with $\mathbf{e} \sim \mathcal{U}([-0.1, 0])$
- Stage cost: $L(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^2 + \frac{1}{2}\mathbf{u}^2$ + strong penalty for $\mathbf{x} \notin [0, 1]$
- Discount $\gamma = 0.9$
- Noise $\mathbf{e}$ "pushing" $\mathbf{x}$ "against left wall"

**Stage cost**



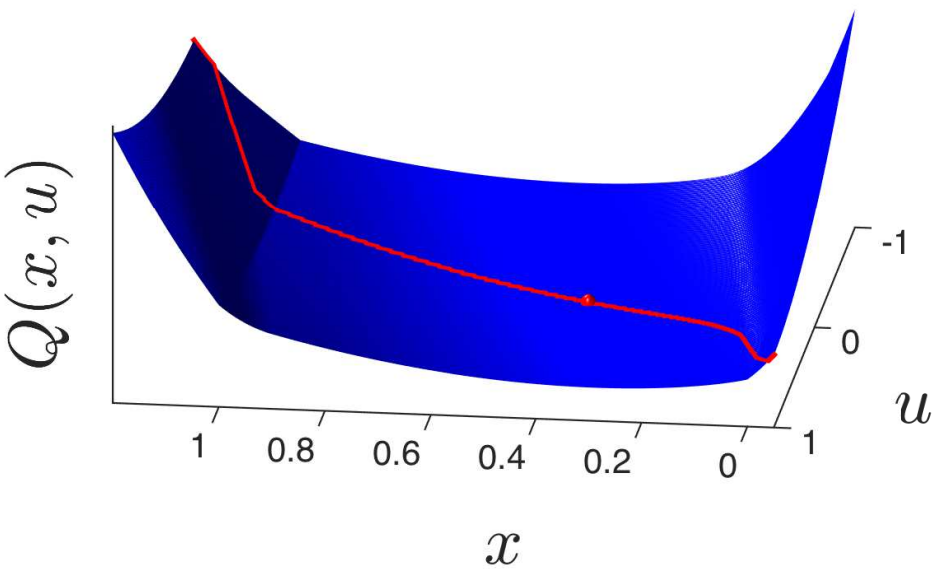

**Trajectories**



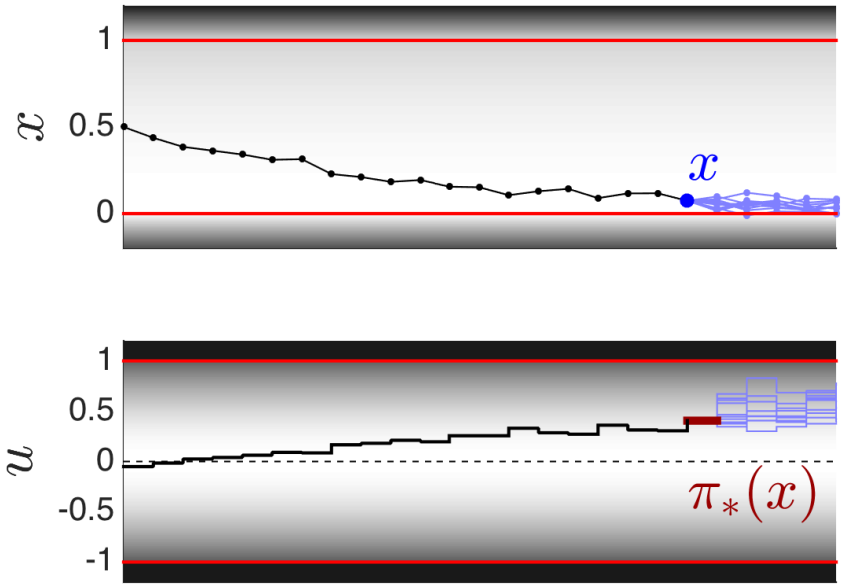
# Let's use the following example

- State $\mathbf{x} \in \mathbb{R}$ and input $\mathbf{u} \in [-1, 1]$
- Dynamics: $\mathbf{x}_+ = \mathbf{x} + 0.1\mathbf{u} + \mathbf{e}$ with $\mathbf{e} \sim \mathcal{U}([-0.1, 0])$
- Stage cost: $L(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^2 + \frac{1}{2}\mathbf{u}^2$ + strong penalty for $\mathbf{x} \notin [0, 1]$
- Discount $\gamma = 0.9$
- Noise $\mathbf{e}$ "pushing" $\mathbf{x}$ "against left wall"

**Action-value function**



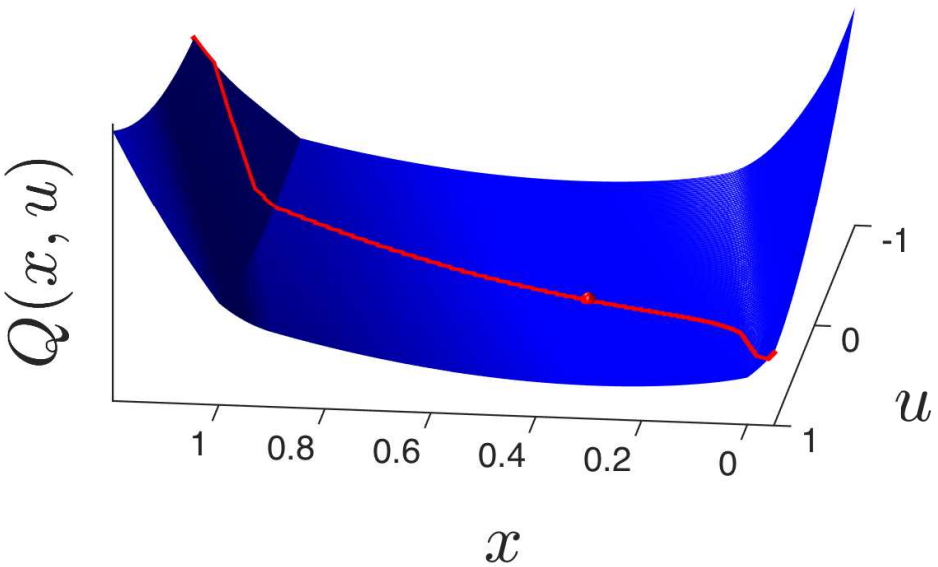

**Trajectories**



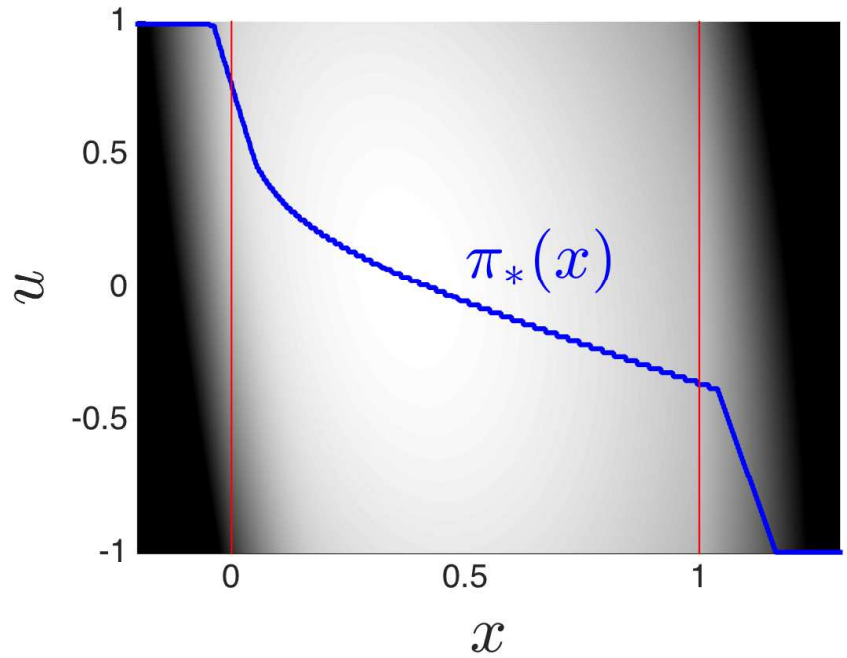
# Let's use the following example

- State $\mathbf{x} \in \mathbb{R}$ and input $\mathbf{u} \in [-1, 1]$
- Dynamics: $\mathbf{x}_+ = \mathbf{x} + 0.1\mathbf{u} + \mathbf{e}$ with $\mathbf{e} \sim \mathcal{U}([-0.1, 0])$
- Stage cost: $L(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^2 + \frac{1}{2}\mathbf{u}^2$ + strong penalty for $\mathbf{x} \notin [0, 1]$
- Discount $\gamma = 0.9$
- Noise $\mathbf{e}$ "pushing" $\mathbf{x}$ "against left wall"

**Action-value function**



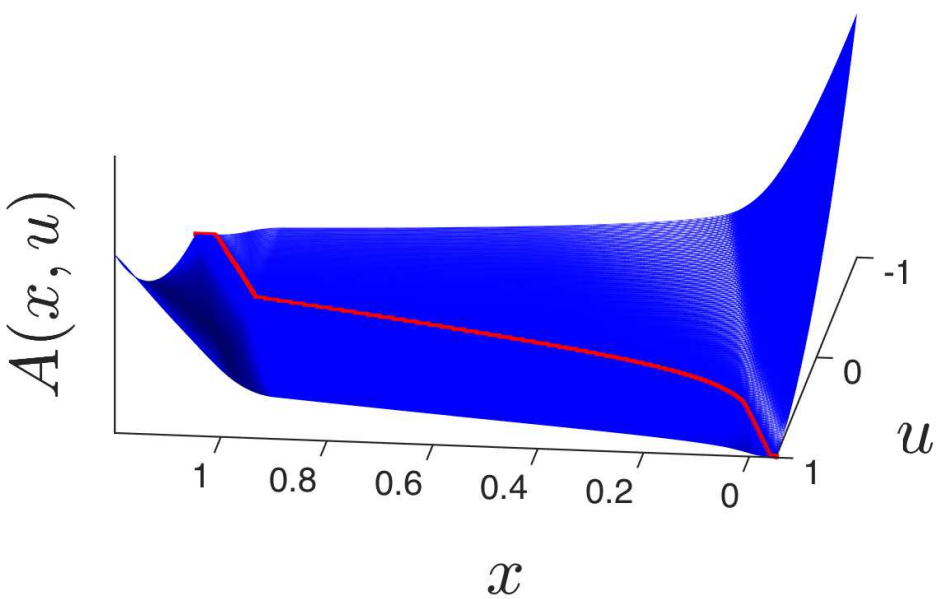

**Optimal policy**



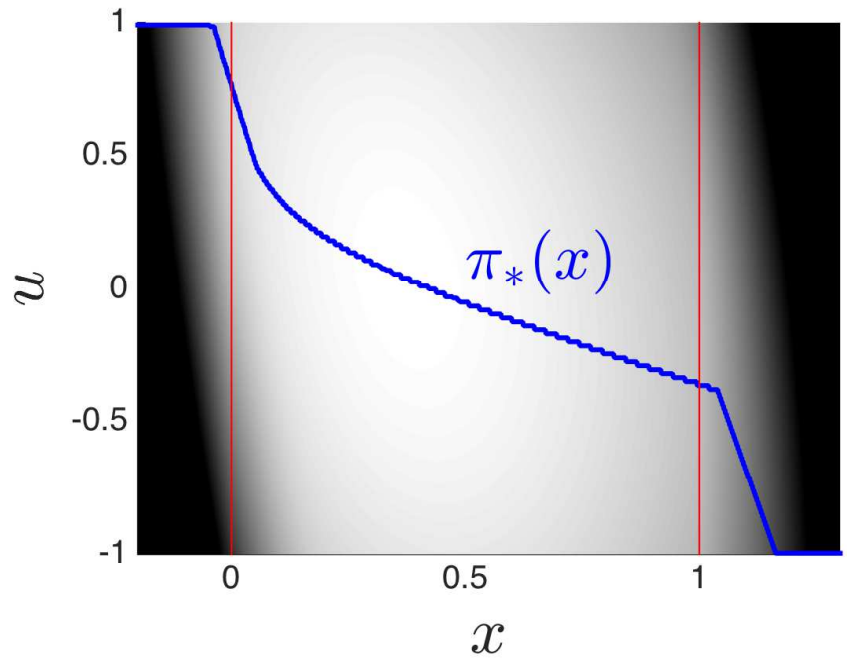
# Let's use the following example

- State $\mathbf{x} \in \mathbb{R}$ and input $\mathbf{u} \in [-1, 1]$
- Dynamics: $\mathbf{x}_+ = \mathbf{x} + 0.1\mathbf{u} + \mathbf{e}$ with $\mathbf{e} \sim \mathcal{U}([-0.1, 0])$
- Stage cost: $L(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^2 + \frac{1}{2}\mathbf{u}^2$ + strong penalty for $\mathbf{x} \notin [0, 1]$
- Discount $\gamma = 0.9$
- Noise $\mathbf{e}$ "pushing" $\mathbf{x}$ "against left wall"

**Advantage function**



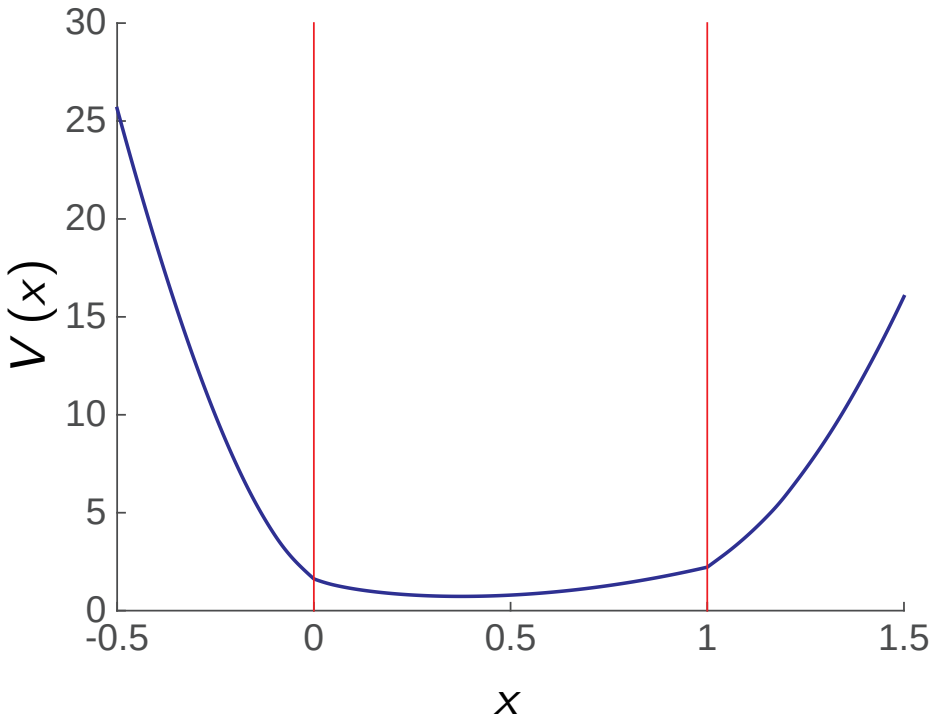

**Optimal policy**



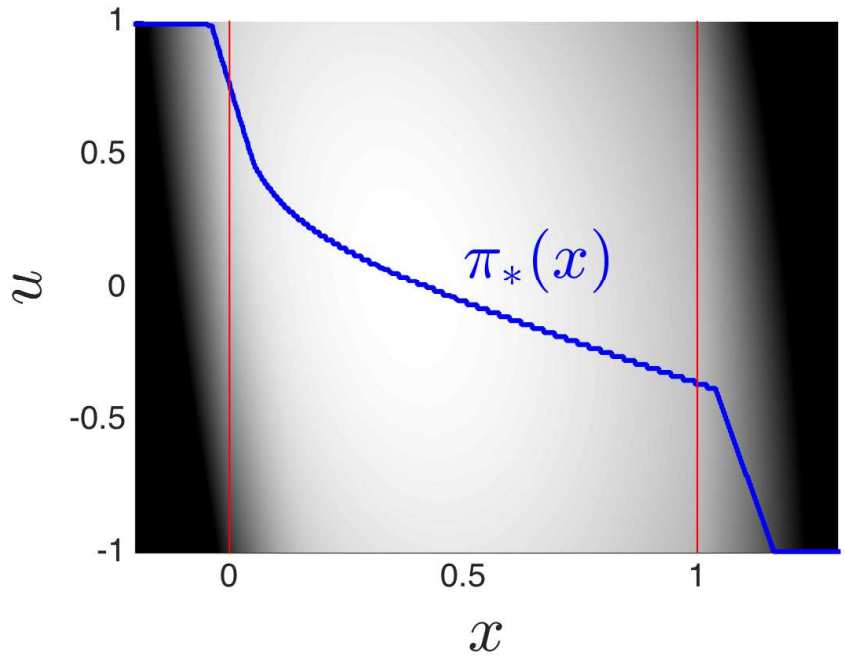
# Let's use the following example

- State $\mathbf{x} \in \mathbb{R}$ and input $\mathbf{u} \in [-1, 1]$
- Dynamics: $\mathbf{x}_+ = \mathbf{x} + 0.1\mathbf{u} + \mathbf{e}$ with $\mathbf{e} \sim \mathcal{U}([-0.1, 0])$
- Stage cost: $L(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^2 + \frac{1}{2}\mathbf{u}^2$ + strong penalty for $\mathbf{x} \notin [0, 1]$
- Discount $\gamma = 0.9$
- Noise $\mathbf{e}$ "pushing" $\mathbf{x}$ "against left wall"

**Value function**

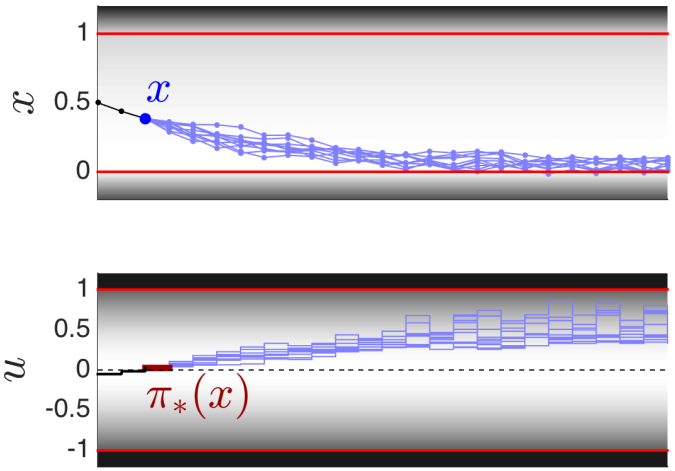


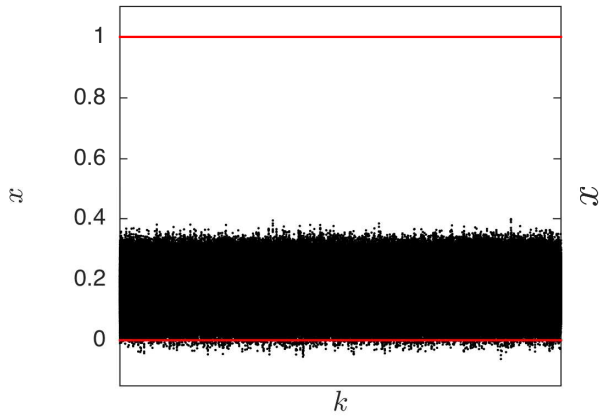

**Optimal policy**

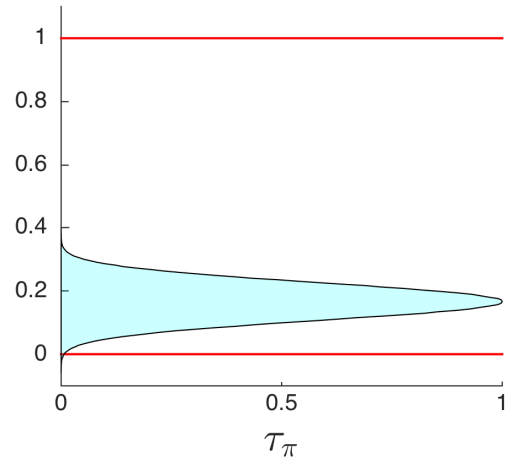


# Outline

# Reminder - Expected value over Markov chains



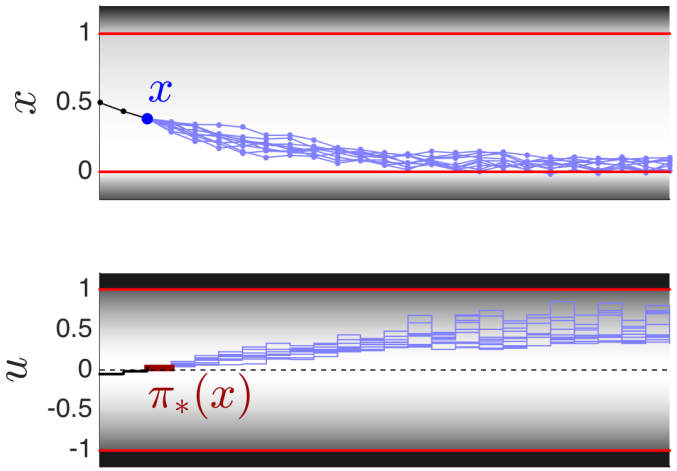



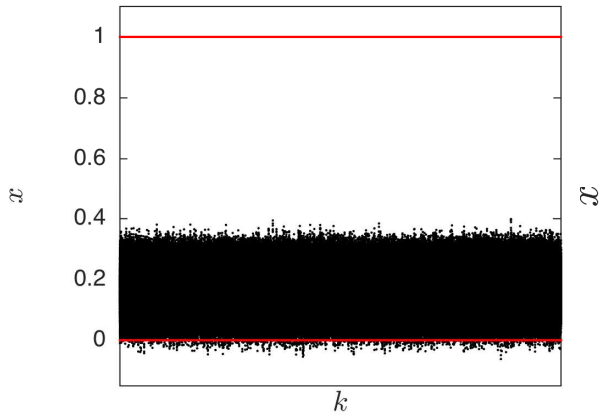

$$\mathbb{P}\left[\mathbf{x}_k \mid \mathbf{x}_0\right] = \int \prod_{i=0}^{k-1} \mathbb{P}\left[\mathbf{x}_{i+1} \mid \mathbf{x}_i,\ \boldsymbol{\pi}(\mathbf{x}_i)\right] \mathrm{d}\mathbf{x}_1 \ldots \mathrm{d}\mathbf{x}_{k-1}$$

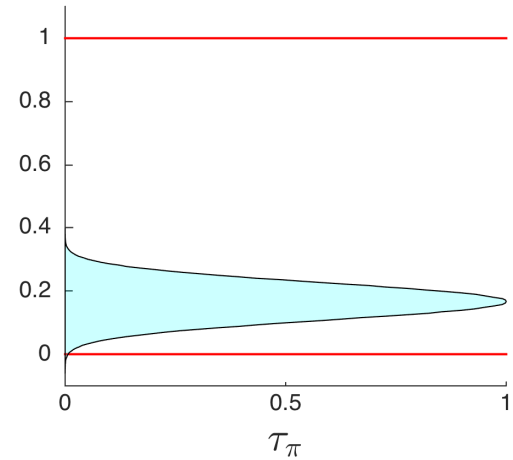
$$V(\mathbf{x}_0) = \sum_{k=0}^{\infty} \int \gamma^k L(\mathbf{x}_k, \boldsymbol{\pi}(\mathbf{x}_k)) \mathbb{P}\left[\mathbf{x}_k \mid \mathbf{x}_0\right] \mathrm{d}\mathbf{x}_k$$

$$J(\boldsymbol{\pi}) = \mathbb{E}\left[V(\mathbf{x}_0)\right] = \int V(\mathbf{x}_0) \mathbb{P}\left[\mathbf{x}_0\right] \mathrm{d}\mathbf{x}_0 = \mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[L(\mathbf{x}, \mathbf{u})\right]$$

# Reminder - Expected value over Markov chains




$$\mathbb{P}\left[\mathbf{x}_k \mid \mathbf{x}_0\right] = \int \prod_{i=0}^{k-1} \mathbb{P}\left[\mathbf{x}_{i+1} \mid \mathbf{x}_i,\ \boldsymbol{\pi}(\mathbf{x}_i)\right] \mathrm{d}\mathbf{x}_1 \ldots \mathrm{d}\mathbf{x}_{k-1}$$

$$V(\mathbf{x}_0) = \sum_{k=0}^{\infty} \int \gamma^k L(\mathbf{x}_k, \boldsymbol{\pi}(\mathbf{x}_k)) \mathbb{P}\left[\mathbf{x}_k \mid \mathbf{x}_0\right] \mathrm{d}\mathbf{x}_k$$

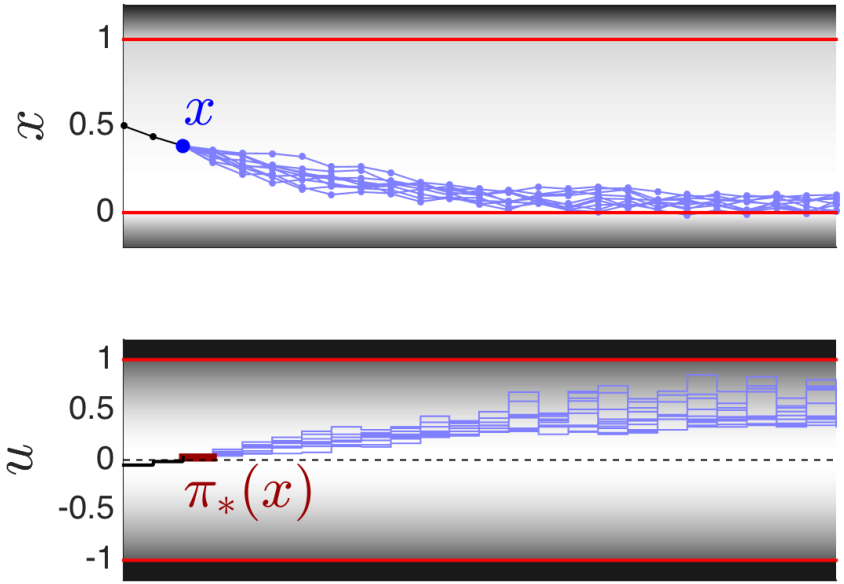
$$J(\boldsymbol{\pi}) = \mathbb{E}\left[V(\mathbf{x}_0)\right] = \int V(\mathbf{x}_0) \mathbb{P}\left[\mathbf{x}_0\right] \mathrm{d}\mathbf{x}_0 = \mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[L(\mathbf{x}, \mathbf{u})\right]$$

When we approximate the functions $(V, Q, \boldsymbol{\pi})$, "where" we build them (i.e. under which distribution) will matter

# Core principle - Function fitting

**Value function fitting:**

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{2}\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[\left(V_{\boldsymbol{\pi}}(\mathbf{x}) - V_{\boldsymbol{\theta}}(\mathbf{x})\right)^2\right]$$



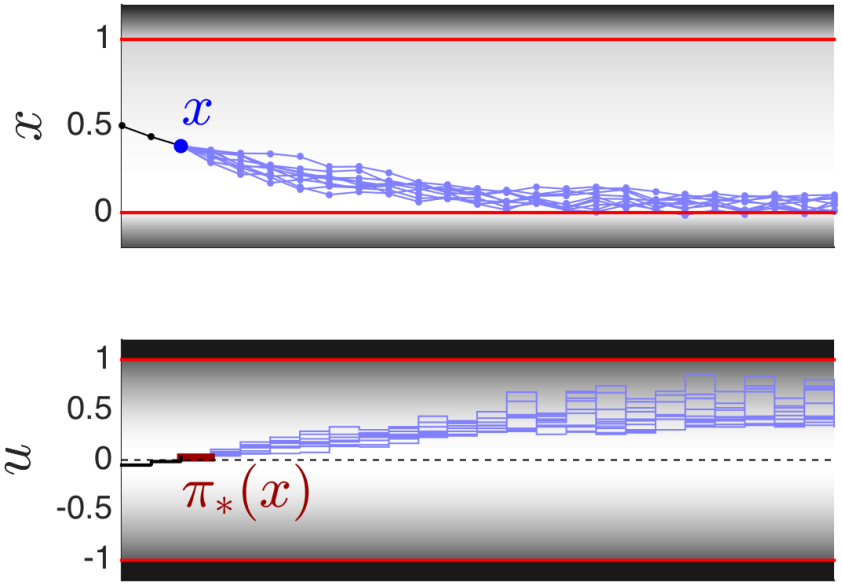
# Core principle - Function fitting

**Value function fitting:**

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{2}\mathbb{E}_{\tau_\pi}\left[\left(V_\pi(\mathbf{x}) - V_\theta(\mathbf{x})\right)^2\right]$$

**Optimal parameter $\hat{\theta}$ satisfies:**

$$\mathbb{E}_{\tau_\pi}\left[\left(V_\pi(\mathbf{x}) - V_\theta(\mathbf{x})\right)\nabla_\theta V_\theta(\mathbf{x})\right] = 0$$



# Core principle - Function fitting

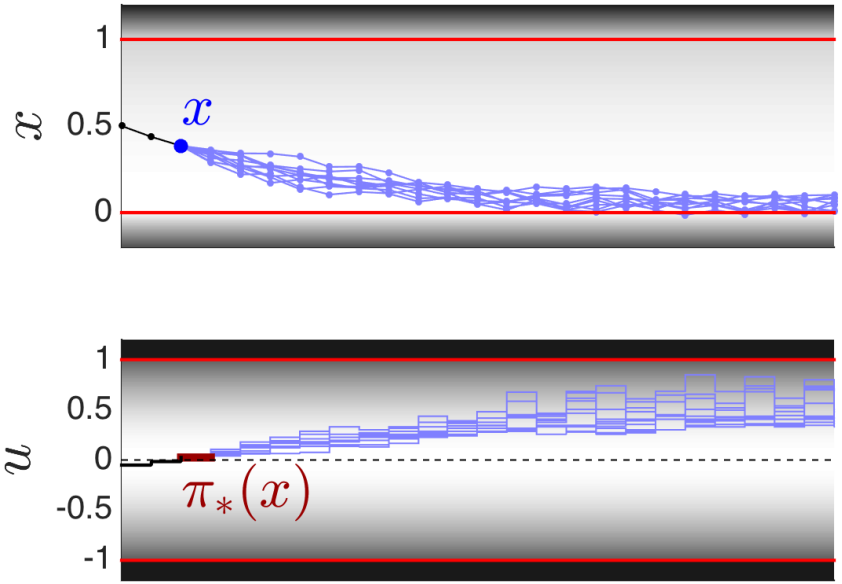
**Value function fitting:**

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{2}\mathbb{E}_{\tau_\pi}\left[\left(V_\pi(\mathbf{x}) - V_\theta(\mathbf{x})\right)^2\right]$$

**Optimal parameter $\hat{\theta}$ satisfies:**

$$\mathbb{E}_{\tau_\pi}\left[\left(V_\pi(\mathbf{x}) - V_\theta(\mathbf{x})\right)\nabla_\theta V_\theta(\mathbf{x})\right] = 0$$

**Where do we get $V_\pi$ from?**



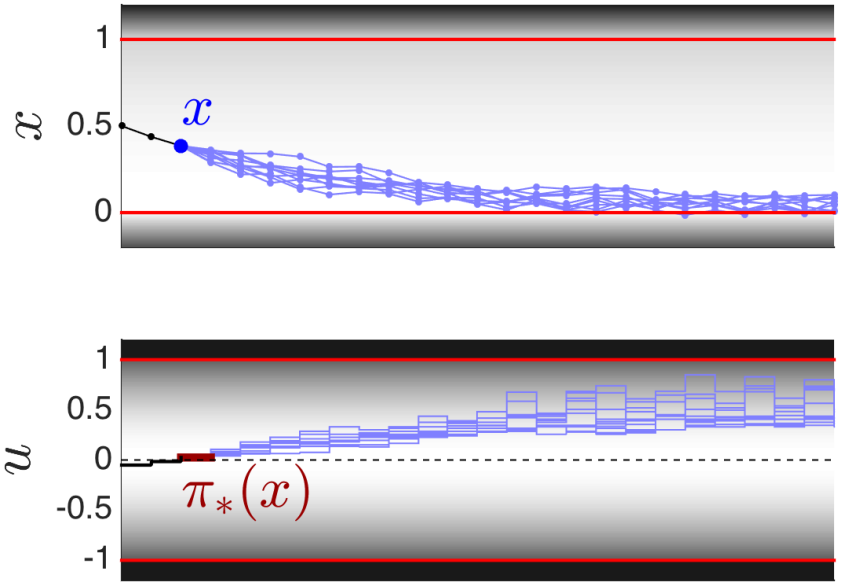
# Core principle - Function fitting

**Value function fitting:**

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{2}\mathbb{E}_{\tau_\pi}\left[\left(V_\pi(\mathbf{x}) - V_\theta(\mathbf{x})\right)^2\right]$$

**Optimal parameter $\hat{\theta}$ satisfies:**

$$\mathbb{E}_{\tau_\pi}\left[\left(V_\pi(\mathbf{x}) - V_\theta(\mathbf{x})\right)\nabla_\theta V_\theta(\mathbf{x})\right] = 0$$




**Where do we get $V_\pi$ from?**

$$V_\pi(\mathbf{x}) = \mathbb{E}\left[\sum_{k=0}^{\infty}\gamma^k L(\mathbf{x}_k, \boldsymbol{\pi}(\mathbf{x}_k)) \,\middle|\, \mathbf{x}_0 = \mathbf{x}\right] = L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma\mathbb{E}\left[V_\pi(\mathbf{x}_+) \mid \mathbf{x}\right]$$

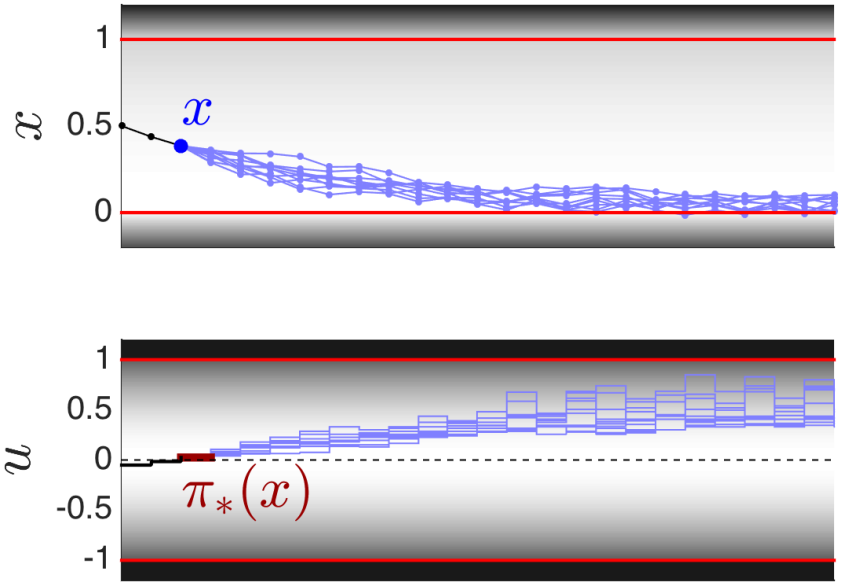
# Core principle - Function fitting

**Value function fitting:**

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{2}\mathbb{E}_{\tau_\pi}\left[\left(V_\pi(\mathbf{x}) - V_\theta(\mathbf{x})\right)^2\right]$$

**Optimal parameter $\hat{\theta}$ satisfies:**

$$\mathbb{E}_{\tau_\pi}\left[\left(V_\pi(\mathbf{x}) - V_\theta(\mathbf{x})\right)\nabla_\theta V_\theta(\mathbf{x})\right] = 0$$




**Where do we get $V_\pi$ from?**

$$V_\pi(\mathbf{x}) = \mathbb{E}\left[\sum_{k=0}^{\infty}\gamma^k L(\mathbf{x}_k, \pi(\mathbf{x}_k)) \,\middle|\, \mathbf{x}_0 = \mathbf{x}\right] = L(\mathbf{x}, \pi(\mathbf{x})) + \gamma\mathbb{E}\left[V_\pi(\mathbf{x}_+) \mid \mathbf{x}\right]$$

**MC evaluation**

$$V_\pi(\mathbf{x}) \approx \sum_{i=0}^{\infty}\gamma^i L(\mathbf{x}_i, \pi(\mathbf{x}_i))$$

(noisy)

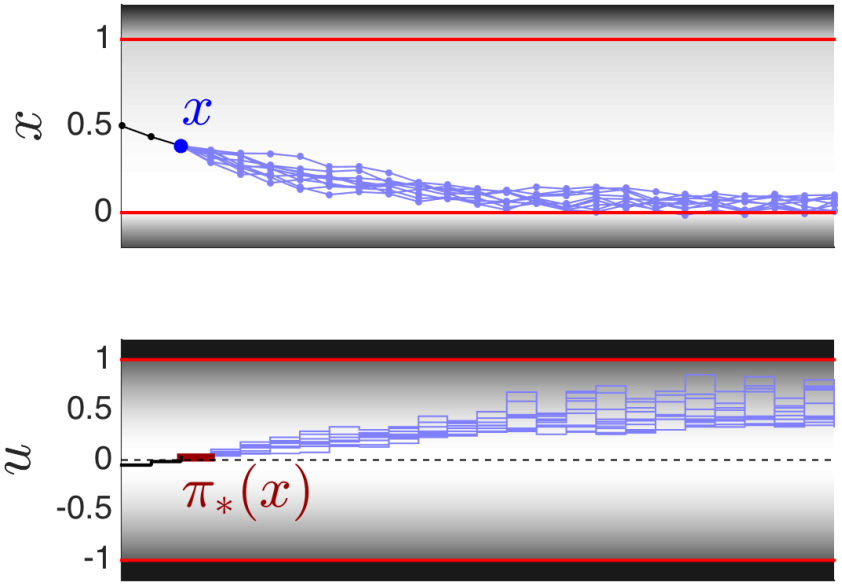
# Core principle - Function fitting

**Value function fitting:**

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{2}\mathbb{E}_{\tau_\pi}\left[\left(V_\pi(\mathbf{x}) - V_\theta(\mathbf{x})\right)^2\right]$$

**Optimal parameter $\hat{\theta}$ satisfies:**

$$\mathbb{E}_{\tau_\pi}\left[\left(V_\pi(\mathbf{x}) - V_\theta(\mathbf{x})\right)\nabla_\theta V_\theta(\mathbf{x})\right] = 0$$




**Where do we get $V_\pi$ from?**

$$V_\pi(\mathbf{x}) = \mathbb{E}\left[\sum_{k=0}^{\infty}\gamma^k L(\mathbf{x}_k, \pi(\mathbf{x}_k)) \,\middle|\, \mathbf{x}_0 = \mathbf{x}\right] = L(\mathbf{x}, \pi(\mathbf{x})) + \gamma\mathbb{E}\left[V_\pi(\mathbf{x}_+) \mid \mathbf{x}\right]$$

**MC evaluation**

$$V_\pi(\mathbf{x}) \approx \sum_{i=0}^{\infty}\gamma^i L(\mathbf{x}_i, \pi(\mathbf{x}_i))$$

(noisy)

**TD evaluation**

$$\begin{aligned} V_\pi(\mathbf{x}) &\approx L(\mathbf{x}, \pi(\mathbf{x})) + \gamma V_\pi(\mathbf{x}_+) \\ &\approx L(\mathbf{x}, \pi(\mathbf{x})) + \gamma V_\theta(\mathbf{x}_+) \end{aligned}$$

(biased)

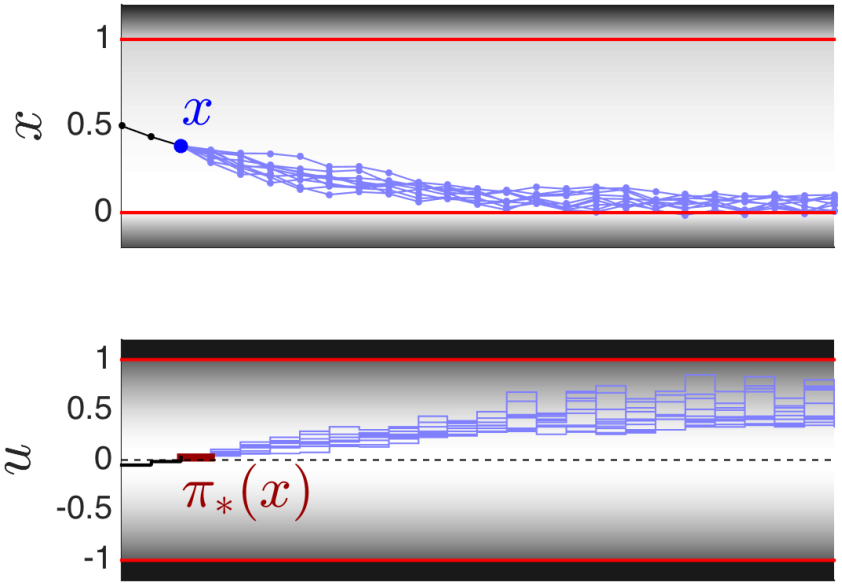
# Core principle - Function fitting

**Value function fitting:**

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{2}\mathbb{E}_{\tau_\pi}\left[\left(V_\pi(\mathbf{x}) - V_\theta(\mathbf{x})\right)^2\right]$$

**Optimal parameter $\hat{\theta}$ satisfies:**

$$\mathbb{E}_{\tau_\pi}\left[\left(V_\pi(\mathbf{x}) - V_\theta(\mathbf{x})\right)\nabla_\theta V_\theta(\mathbf{x})\right] = 0$$




**Where do we get $V_\pi$ from?**

Using $V_\pi(\mathbf{x}) \approx L(\mathbf{x}, \pi(\mathbf{x})) + \gamma V_\theta(\mathbf{x}_+)$

**Compute parameter $\hat{\theta}$ using:**

$$\mathbb{E}_{\tau_\pi}\left[\left(L(\mathbf{x}, \pi(\mathbf{x})) + \gamma V_\theta(\mathbf{x}_+) - V_\theta(\mathbf{x})\right)\nabla_\theta V_\theta(\mathbf{x})\right] = 0$$

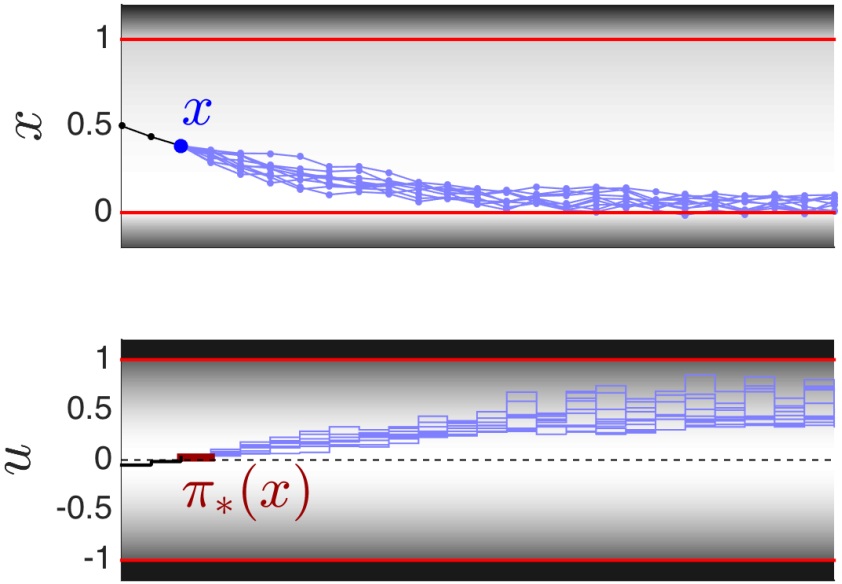
# Core principle - Function fitting

**Value function fitting:**

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{2}\mathbb{E}_{\tau_\pi}\left[\left(V_\pi(\mathbf{x}) - V_\theta(\mathbf{x})\right)^2\right]$$

**Optimal parameter $\hat{\theta}$ satisfies:**

$$\mathbb{E}_{\tau_\pi}\left[\left(V_\pi(\mathbf{x}) - V_\theta(\mathbf{x})\right)\nabla_\theta V_\theta(\mathbf{x})\right] = 0$$




**Where do we get $V_\pi$ from?**

Using $V_\pi(\mathbf{x}) \approx L(\mathbf{x}, \pi(\mathbf{x})) + \gamma V_\theta(\mathbf{x}_+)$

**Compute parameter $\hat{\theta}$ using:**

$$\mathbb{E}_{\tau_\pi}\left[\left(L(\mathbf{x}, \pi(\mathbf{x})) + \gamma V_\theta(\mathbf{x}_+) - V_\theta(\mathbf{x})\right)\nabla_\theta V_\theta(\mathbf{x})\right] = 0$$

equivalently:

$$\mathbb{E}_{\tau_\pi}\left[\delta \nabla_\theta V_\theta\right] = 0$$

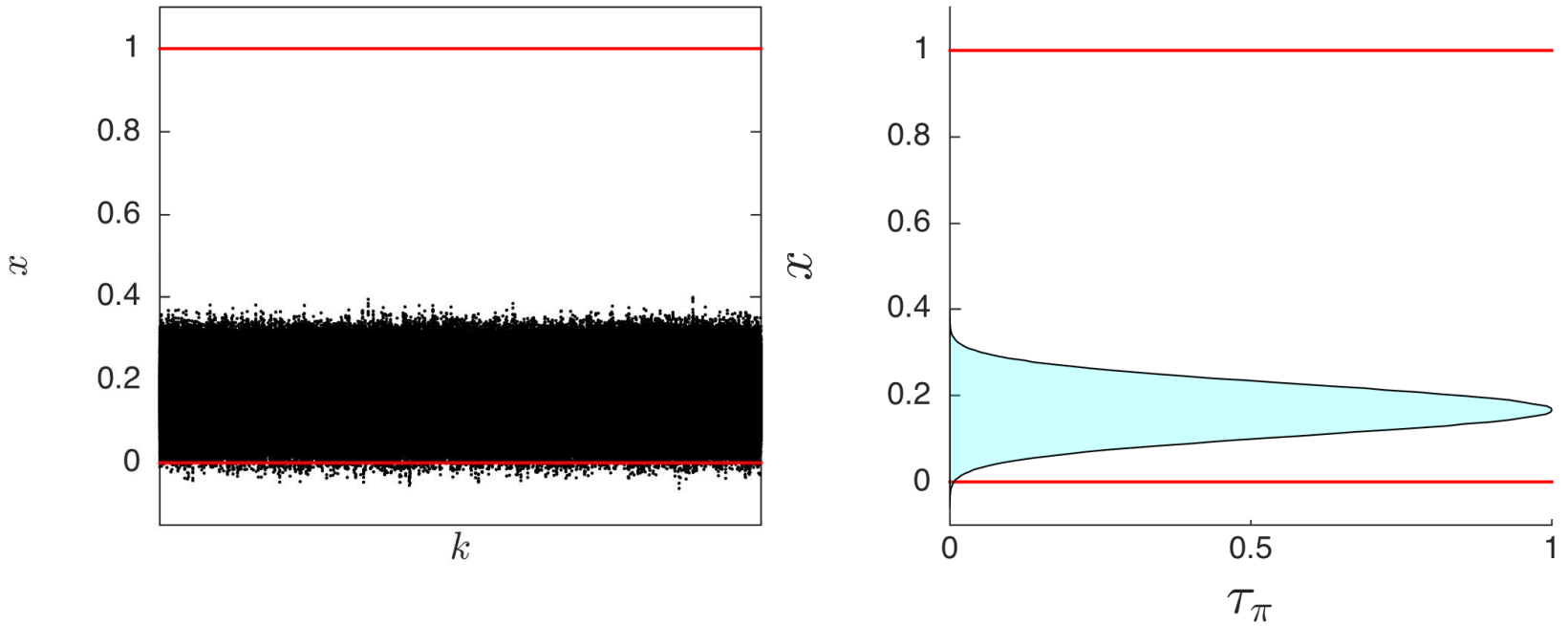
where $\quad \delta = L(\mathbf{x}, \pi(\mathbf{x})) + \gamma V_\theta(\mathbf{x}_+) - V_\theta(\mathbf{x})$

# Function fitting - Example

- Dynamics: $\mathbf{x}_+ = \mathbf{x} + 0.1\mathbf{u} + \mathbf{e}$ with $\mathbf{e} \sim \mathcal{U}([-0.1, 0])$
- Stage cost: $L(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^2 + \frac{1}{2}\mathbf{u}^2 +$ strong penalty for $\mathbf{x} \notin [0, 1]$
- Discount $\gamma = 0.9$
- Let us e.g. select $\boldsymbol{\pi}(\mathbf{x}) = -\frac{3}{2}\mathbf{x} + \frac{3}{4}$ clipped to $[-1, 1]$

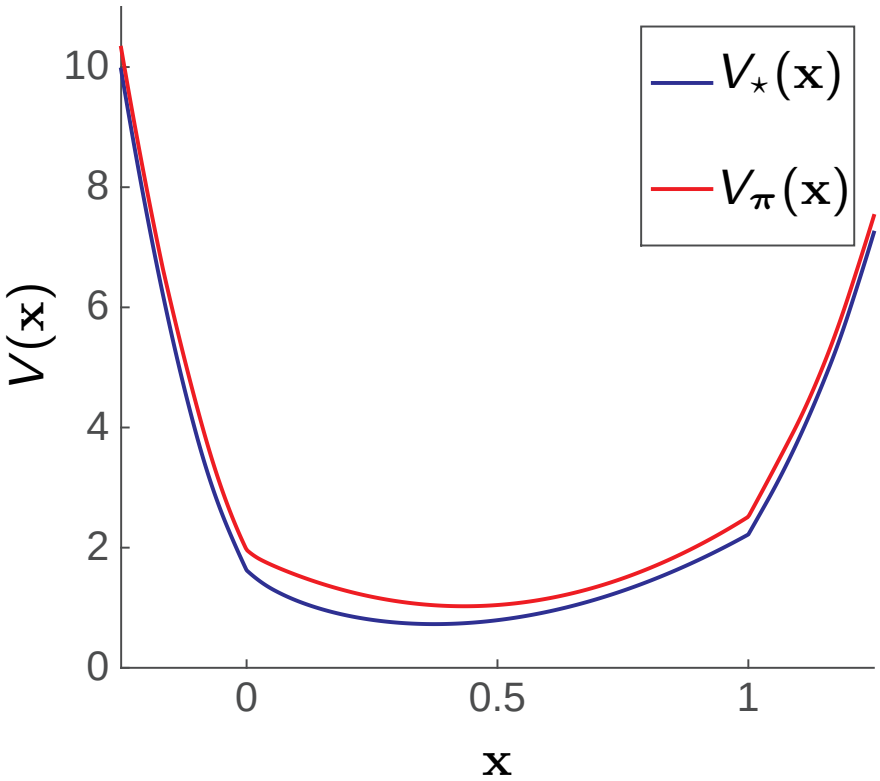
# Function fitting - Example

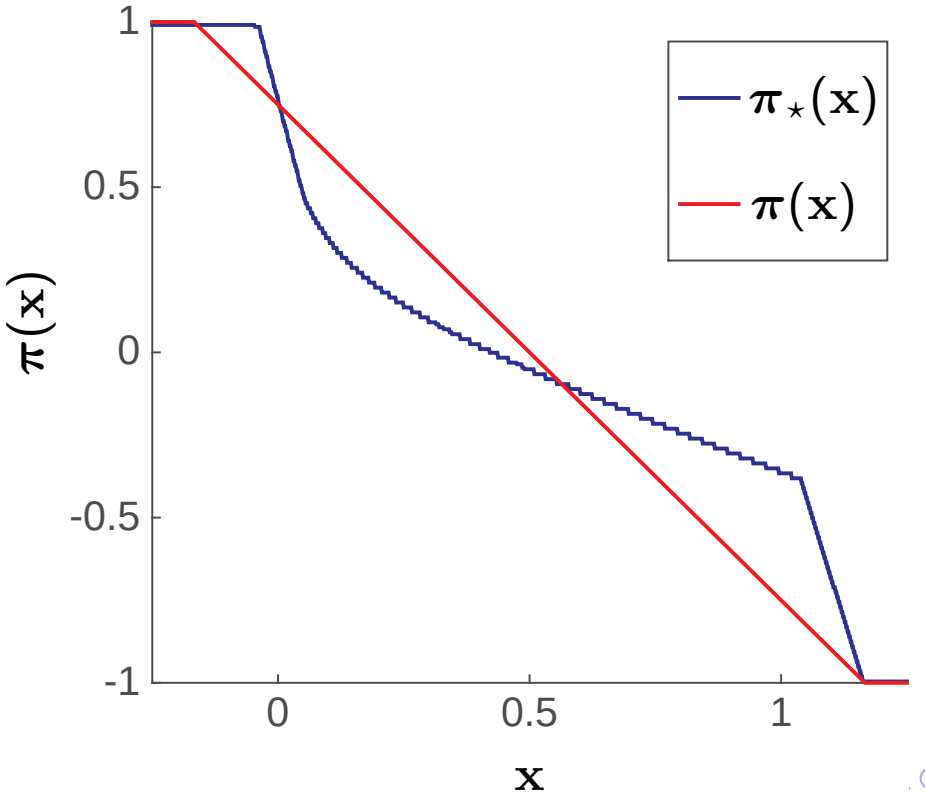
- Dynamics: $\mathbf{x}_+ = \mathbf{x} + 0.1\mathbf{u} + \mathbf{e}$ with $\mathbf{e} \sim \mathcal{U}([-0.1, 0])$
- Stage cost: $L(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^2 + \frac{1}{2}\mathbf{u}^2$ + strong penalty for $\mathbf{x} \notin [0, 1]$
- Discount $\gamma = 0.9$
- Let us e.g. select $\boldsymbol{\pi}(\mathbf{x}) = -\frac{3}{2}\mathbf{x} + \frac{3}{4}$ clipped to $[-1, 1]$
- Markov Chain trajectories $\tau_{\boldsymbol{\pi}}$



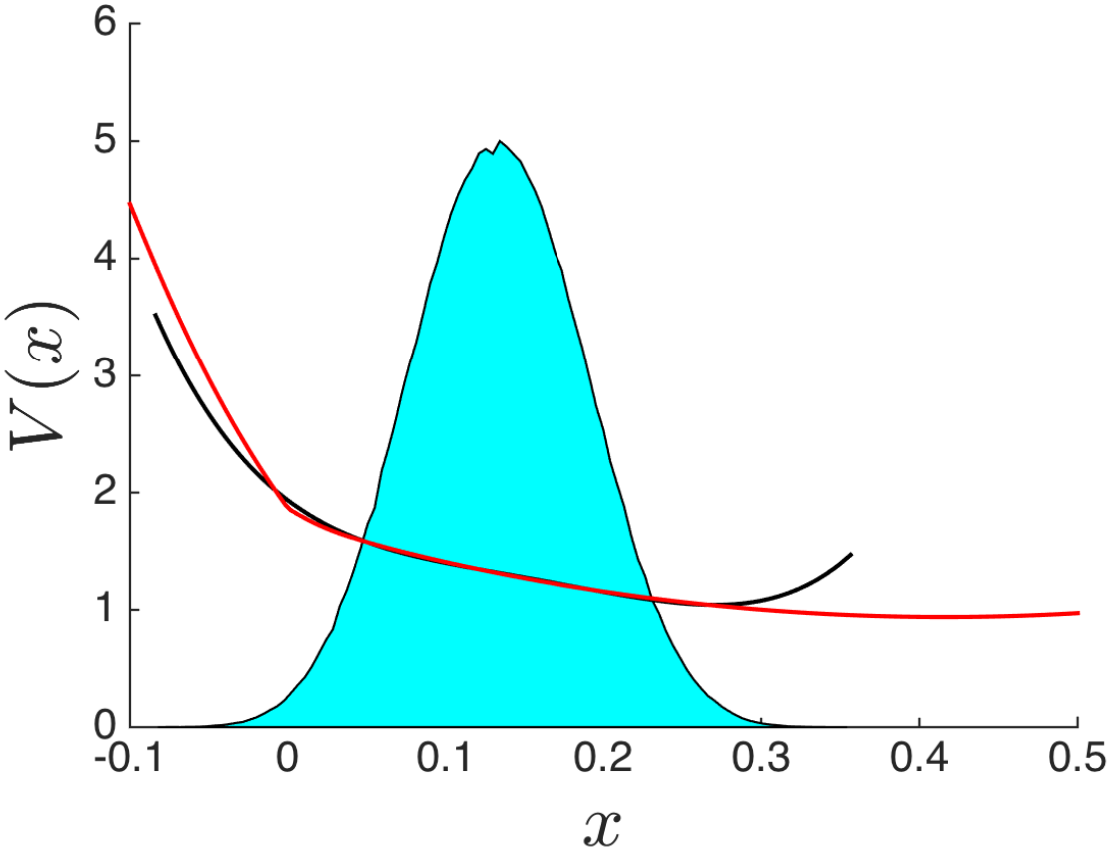
# Function fitting - Example

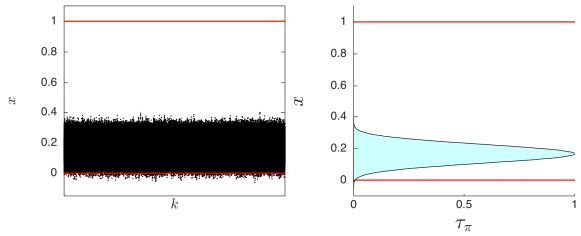
- Dynamics: $\mathbf{x}_+ = \mathbf{x} + 0.1\mathbf{u} + \mathbf{e}$ with $\mathbf{e} \sim \mathcal{U}([-0.1, 0])$
- Stage cost: $L(\mathbf{x}, \mathbf{u}) = \frac{1}{2}\mathbf{x}^2 + \frac{1}{2}\mathbf{u}^2 +$ strong penalty for $\mathbf{x} \notin [0, 1]$
- Discount $\gamma = 0.9$
- Let us e.g. select $\boldsymbol{\pi}(\mathbf{x}) = -\frac{3}{2}\mathbf{x} + \frac{3}{4}$ clipped to $[-1, 1]$
- Markov Chain trajectories $\tau_{\boldsymbol{\pi}}$
- Resulting value function $V_{\boldsymbol{\pi}}(\mathbf{x}) \geq V_\star(\mathbf{x})$ for all $\mathbf{x}$



# Function fitting - Example



We label

$$V_\pi(\mathbf{x}),\ \hat{V}(\mathbf{x})$$

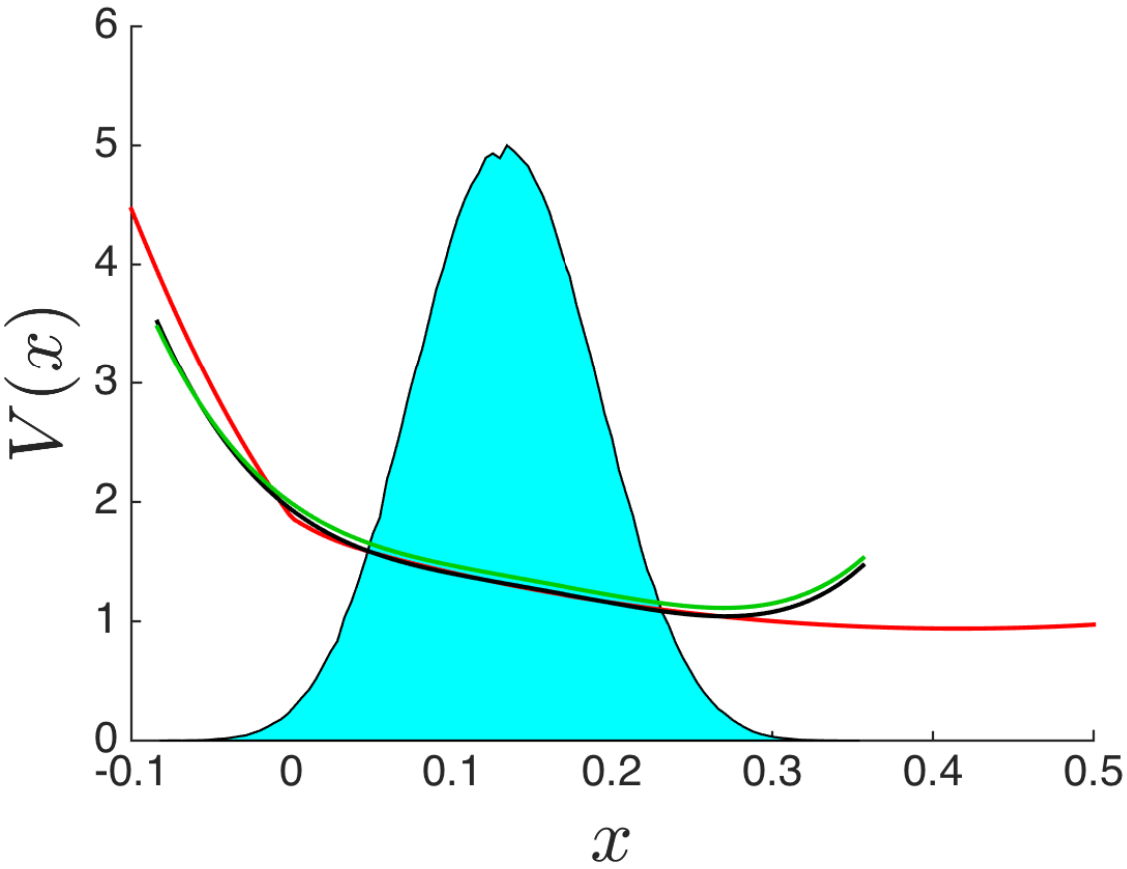
We choose e.g.:

$$\hat{V}(\mathbf{x}) = \sum_{k=0}^{3} \hat{\boldsymbol{\theta}}_k \mathbf{x}^k$$

**Value function fitting:**

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{2}\mathbb{E}_{\tau_\pi}\left[\left(V_\pi - \hat{V}\right)^2\right] \iff \mathbb{E}_{\tau_\pi}\left[\left(V_\pi - \hat{V}\right)\nabla_{\boldsymbol{\theta}}\hat{V}\right] = 0$$

# Function fitting using target - Example




We label

$$V_\pi(\mathbf{x}),\ \hat{V}(\mathbf{x}),\ V_{\mathrm{TD}}(\mathbf{x})$$

We choose e.g.:

$$\hat{V}(\mathbf{x}) = \sum_{k=0}^{4} \hat{\boldsymbol{\theta}}_k \mathbf{x}^k$$

**Value function fitting:**

$$\mathbb{E}_{\tau_\pi}\left[\left(V_\pi - \hat{V}\right)\nabla_{\boldsymbol{\theta}}\hat{V}\right] = 0$$

**TD-based fitting:** $(V_\pi \approx L + \gamma V_{\mathrm{TD}}^+)$

$$\mathbb{E}_{\tau_\pi}\left[\left(L + \gamma V_{\mathrm{TD}}^+ - V_{\mathrm{TD}}\right)\nabla_{\boldsymbol{\theta}} V_{\mathrm{TD}}\right] = 0$$

$$\hat{V} \neq V_{\mathrm{TD}} \text{ (bias)}$$

# Remark - Fitting towards fixed target?

**Value function fitting:**

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{2}\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[(V_{\boldsymbol{\pi}} - V_{\boldsymbol{\theta}})^2\right]$$

using TD approximation

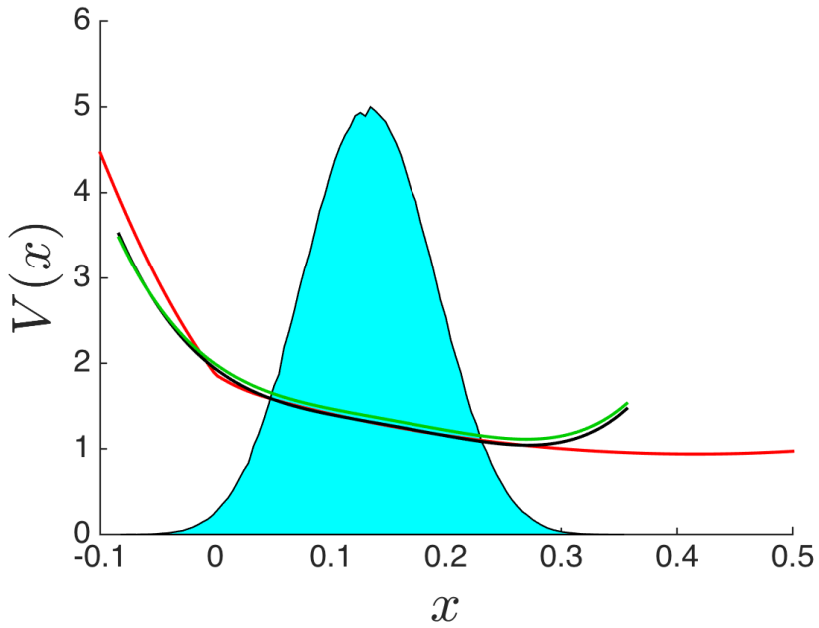
$$V_{\boldsymbol{\pi}}(\mathbf{x}) \approx L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma V_{\boldsymbol{\theta}}(\mathbf{x}_+)$$

**Compute parameter $\hat{\boldsymbol{\theta}}$ using:**

$$\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[\delta \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}\right] = 0$$
$$\delta = L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma V_{\boldsymbol{\theta}}(\mathbf{x}_+) - V_{\boldsymbol{\theta}}(\mathbf{x})$$

We label

$$V_\star(\mathbf{x}),\ V_{\boldsymbol{\pi}}(\mathbf{x}),\ \hat{V}(\mathbf{x}),\ V_{\mathrm{TD}}(\mathbf{x})$$



# Remark - Fitting towards fixed target?

**Value function fitting:**

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{2}\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[(V_{\boldsymbol{\pi}} - V_{\boldsymbol{\theta}})^2\right]$$

using TD approximation

$$V_{\boldsymbol{\pi}}(\mathbf{x}) \approx L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma V_{\boldsymbol{\theta}}(\mathbf{x}_+)$$

Note: $\delta \approx V_{\boldsymbol{\pi}} - V_{\boldsymbol{\theta}}$ **Do min of TD error?**

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{2}\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[\delta^2\right]$$

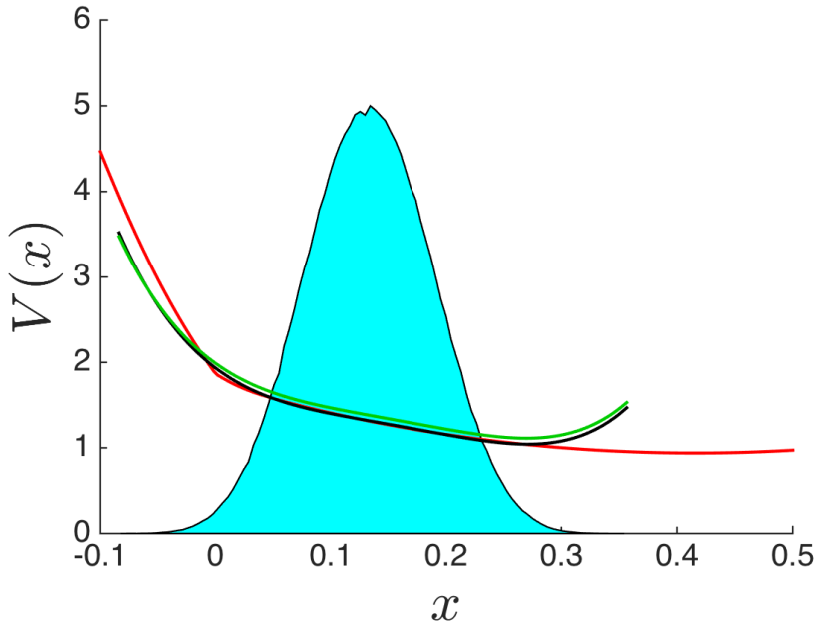
hence solve $\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[\delta \cdot \nabla_{\boldsymbol{\theta}}\delta\right] = 0$

**Compute parameter $\hat{\theta}$ using:**

$$\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[\delta \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}\right] = 0$$
$$\delta = L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma V_{\boldsymbol{\theta}}(\mathbf{x}_+) - V_{\boldsymbol{\theta}}(\mathbf{x})$$

We label

$V_\star(\mathbf{x})$, $V_{\boldsymbol{\pi}}(\mathbf{x})$, $\hat{V}(\mathbf{x})$, $V_{\mathrm{TD}}(\mathbf{x})$



# Remark - Fitting towards fixed target?

**Value function fitting:**

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{2}\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[(V_{\boldsymbol{\pi}} - V_{\boldsymbol{\theta}})^2\right]$$

using TD approximation

$$V_{\boldsymbol{\pi}}(\mathbf{x}) \approx L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma V_{\boldsymbol{\theta}}(\mathbf{x}_+)$$

Note: $\delta \approx V_{\boldsymbol{\pi}} - V_{\boldsymbol{\theta}}$ **Do min of TD error?**

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{2}\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[\delta^2\right]$$

hence solve $\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[\delta \cdot \nabla_{\boldsymbol{\theta}}\delta\right] = 0$

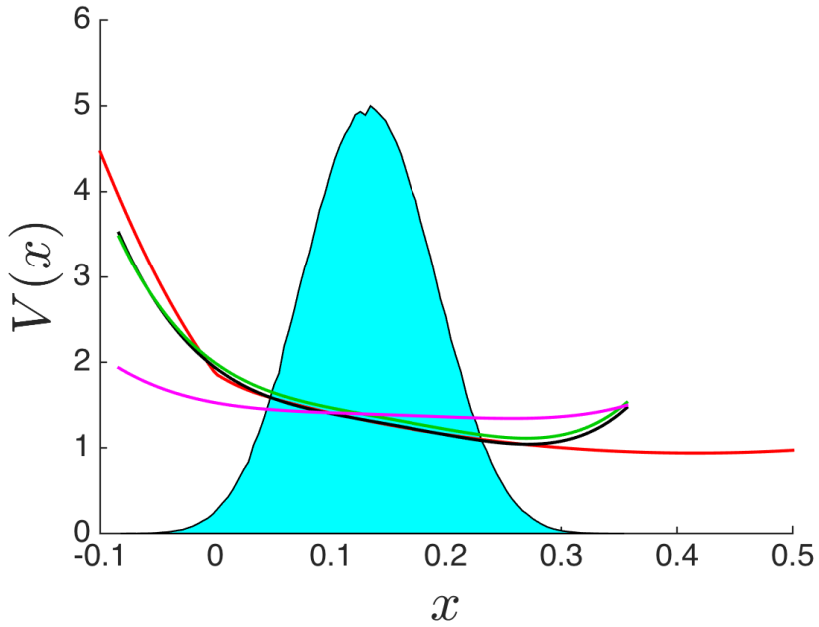
- generally does not work
- exception: deterministic system and "rich" parametrization (get $\delta = 0$)
- true $\delta$ is not necessarily RMS-minimum

**Compute parameter $\hat{\theta}$ using:**

$$\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[\delta \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}\right] = 0$$
$$\delta = L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma V_{\boldsymbol{\theta}}(\mathbf{x}_+) - V_{\boldsymbol{\theta}}(\mathbf{x})$$

We label

$V_\star(\mathbf{x})$, $V_{\boldsymbol{\pi}}(\mathbf{x})$, $\hat{V}(\mathbf{x})$, $V_{\mathrm{TD}}(\mathbf{x})$



# Outline

# Stochastic TD approach - Solving fitting $\min_{\boldsymbol{\theta}} \frac{1}{2}\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[(V_{\boldsymbol{\pi}} - V_{\boldsymbol{\theta}})^2\right]$

**Gradient** approach to solve

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \underbrace{\frac{1}{2}\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[(V_{\boldsymbol{\pi}} - V_{\boldsymbol{\theta}})^2\right]}_{:=\phi(\boldsymbol{\theta})}$$

via gradient steps:

$$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} - \alpha \nabla_{\boldsymbol{\theta}} \phi$$

# Stochastic TD approach - Solving fitting $\min_{\boldsymbol{\theta}} \frac{1}{2}\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[(V_{\boldsymbol{\pi}} - V_{\boldsymbol{\theta}})^2\right]$

**Gradient** approach to solve

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \underbrace{\frac{1}{2}\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[(V_{\boldsymbol{\pi}} - V_{\boldsymbol{\theta}})^2\right]}_{:=\phi(\boldsymbol{\theta})}$$

via gradient steps:

$$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} - \alpha \nabla_{\boldsymbol{\theta}} \phi$$

Iteration reads as:

$$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} + \alpha \mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[(V_{\boldsymbol{\pi}} - V_{\boldsymbol{\theta}})\, \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}\right]$$

**Stochastic gradient** approach:

$$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} + \alpha\, (V_{\boldsymbol{\pi}} - V_{\boldsymbol{\theta}})\, \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}$$

# Stochastic TD approach - Solving fitting $\min_{\boldsymbol{\theta}} \frac{1}{2}\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[(V_{\boldsymbol{\pi}} - V_{\boldsymbol{\theta}})^2\right]$

**Gradient** approach to solve

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \underbrace{\frac{1}{2}\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[(V_{\boldsymbol{\pi}} - V_{\boldsymbol{\theta}})^2\right]}_{:=\phi(\boldsymbol{\theta})}$$

via gradient steps:

$$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} - \alpha \nabla_{\boldsymbol{\theta}} \phi$$

Iteration reads as:

$$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} + \alpha \mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[(V_{\boldsymbol{\pi}} - V_{\boldsymbol{\theta}})\, \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}\right]$$

**Stochastic gradient** approach:

$$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} + \alpha\, (V_{\boldsymbol{\pi}} - V_{\boldsymbol{\theta}})\, \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}$$

**TD approximation**

$$V_{\boldsymbol{\pi}}(\mathbf{x}) \approx L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma V_{\boldsymbol{\theta}}(\mathbf{x}_+)$$

is used.

# Stochastic TD approach - Solving fitting $\min_{\boldsymbol{\theta}} \frac{1}{2}\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[(V_{\boldsymbol{\pi}} - V_{\boldsymbol{\theta}})^2\right]$

**Gradient** approach to solve

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \underbrace{\frac{1}{2}\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[(V_{\boldsymbol{\pi}} - V_{\boldsymbol{\theta}})^2\right]}_{:=\phi(\boldsymbol{\theta})}$$

via gradient steps:

$$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} - \alpha \nabla_{\boldsymbol{\theta}} \phi$$

Iteration reads as:

$$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} + \alpha \mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[(V_{\boldsymbol{\pi}} - V_{\boldsymbol{\theta}})\, \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}\right]$$

**Stochastic gradient** approach:

$$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} + \alpha \left(V_{\boldsymbol{\pi}} - V_{\boldsymbol{\theta}}\right) \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}$$

**TD approximation**

$$V_{\boldsymbol{\pi}}(\mathbf{x}) \approx L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma V_{\boldsymbol{\theta}}(\mathbf{x}_+)$$

is used. Yields:

$$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} + \alpha \left(L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma V_{\boldsymbol{\theta}}(\mathbf{x}_+) - V_{\boldsymbol{\theta}}(\mathbf{x})\right) \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}$$

# Stochastic TD approach - Solving fitting $\min_{\boldsymbol{\theta}} \frac{1}{2}\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[(V_{\boldsymbol{\pi}} - V_{\boldsymbol{\theta}})^2\right]$

**Gradient** approach to solve

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \underbrace{\frac{1}{2}\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[(V_{\boldsymbol{\pi}} - V_{\boldsymbol{\theta}})^2\right]}_{:=\phi(\boldsymbol{\theta})}$$

via gradient steps:

$$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} - \alpha \nabla_{\boldsymbol{\theta}} \phi$$

Iteration reads as:

$$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} + \alpha \mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[(V_{\boldsymbol{\pi}} - V_{\boldsymbol{\theta}})\nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}\right]$$

**Stochastic gradient** approach:

$$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} + \alpha (V_{\boldsymbol{\pi}} - V_{\boldsymbol{\theta}})\nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}$$

**TD approximation**

$$V_{\boldsymbol{\pi}}(\mathbf{x}) \approx L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma V_{\boldsymbol{\theta}}(\mathbf{x}_+)$$

is used. Yields:

$$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} + \alpha \left(L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma V_{\boldsymbol{\theta}}(\mathbf{x}_+) - V_{\boldsymbol{\theta}}(\mathbf{x})\right) \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}$$

or equivalently

$$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} + \alpha\delta \cdot \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}$$
$$\delta = L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma V_{\boldsymbol{\theta}}(\mathbf{x}_+) - V_{\boldsymbol{\theta}}(\mathbf{x})$$

# Stochastic TD approach - Solving fitting $\min_{\theta} \frac{1}{2}\mathbb{E}_{\tau_\pi}\left[(V_\pi - V_\theta)^2\right]$

**Gradient** approach to solve

$$\hat{\theta} = \arg\min_{\theta} \underbrace{\frac{1}{2}\mathbb{E}_{\tau_\pi}\left[(V_\pi - V_\theta)^2\right]}_{:=\phi(\theta)}$$

via gradient steps:

$$\hat{\theta} \leftarrow \hat{\theta} - \alpha \nabla_\theta \phi$$

Iteration reads as:

$$\hat{\theta} \leftarrow \hat{\theta} + \alpha \mathbb{E}_{\tau_\pi}\left[(V_\pi - V_\theta)\nabla_\theta V_\theta\right]$$

**Stochastic gradient** approach:

$$\hat{\theta} \leftarrow \hat{\theta} + \alpha (V_\pi - V_\theta)\nabla_\theta V_\theta$$

**TD approximation**

$$V_\pi(\mathbf{x}) \approx L(\mathbf{x}, \pi(\mathbf{x})) + \gamma V_\theta(\mathbf{x}_+)$$

is used. Yields:

$$\hat{\theta} \leftarrow \hat{\theta} + \alpha \left(L(\mathbf{x}, \pi(\mathbf{x})) + \gamma V_\theta(\mathbf{x}_+) - V_\theta(\mathbf{x})\right)\nabla_\theta V_\theta$$

or equivalently

$$\hat{\theta} \leftarrow \hat{\theta} + \alpha\delta \cdot \nabla_\theta V_\theta$$
$$\delta = L(\mathbf{x}, \pi(\mathbf{x})) + \gamma V_\theta(\mathbf{x}_+) - V_\theta(\mathbf{x})$$

*note that $\delta \nabla_\theta V_\theta$ is not the gradient of anything...*

# Stochastic TD approach - Solving fitting $\min_{\theta} \frac{1}{2}\mathbb{E}_{\tau_\pi}\left[(V_\pi - V_\theta)^2\right]$

**Gradient** approach to solve

$$\hat{\theta} = \arg\min_{\theta} \underbrace{\frac{1}{2}\mathbb{E}_{\tau_\pi}\left[(V_\pi - V_\theta)^2\right]}_{:=\phi(\theta)}$$

via gradient steps:

$$\hat{\theta} \leftarrow \hat{\theta} - \alpha \nabla_\theta \phi$$

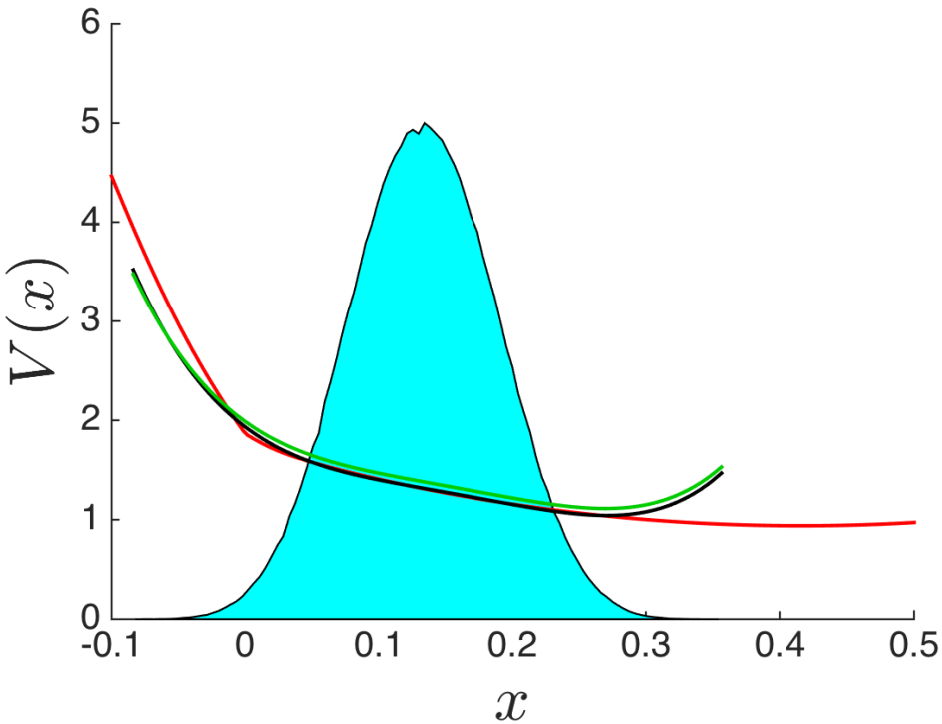
Iteration reads as:

$$\hat{\theta} \leftarrow \hat{\theta} + \alpha \mathbb{E}_{\tau_\pi}\left[(V_\pi - V_\theta)\nabla_\theta V_\theta\right]$$

**Stochastic gradient** approach:

$$\hat{\theta} \leftarrow \hat{\theta} + \alpha\,(V_\pi - V_\theta)\nabla_\theta V_\theta$$

$$\hat{\theta} \leftarrow \hat{\theta} + \alpha\delta \cdot \nabla_\theta V_\theta$$
$$\delta = L(\mathbf{x}, \pi(\mathbf{x})) + \gamma V_\theta(\mathbf{x}_+) - V_\theta(\mathbf{x})$$




$V_\pi(\mathbf{x})$, $\hat{V}(\mathbf{x})$, $V_{\mathrm{TD}}(\mathbf{x})$

# Stochastic TD approach - Solving fitting $\min_{\theta} \frac{1}{2}\mathbb{E}_{\tau_\pi}\left[(V_\pi - V_\theta)^2\right]$

**Gradient** approach to solve

$$\hat{\theta} = \arg\min_{\theta} \underbrace{\frac{1}{2}\mathbb{E}_{\tau_\pi}\left[(V_\pi - V_\theta)^2\right]}_{:=\phi(\theta)}$$

via gradient steps:

$$\hat{\theta} \leftarrow \hat{\theta} - \alpha \nabla_\theta \phi$$

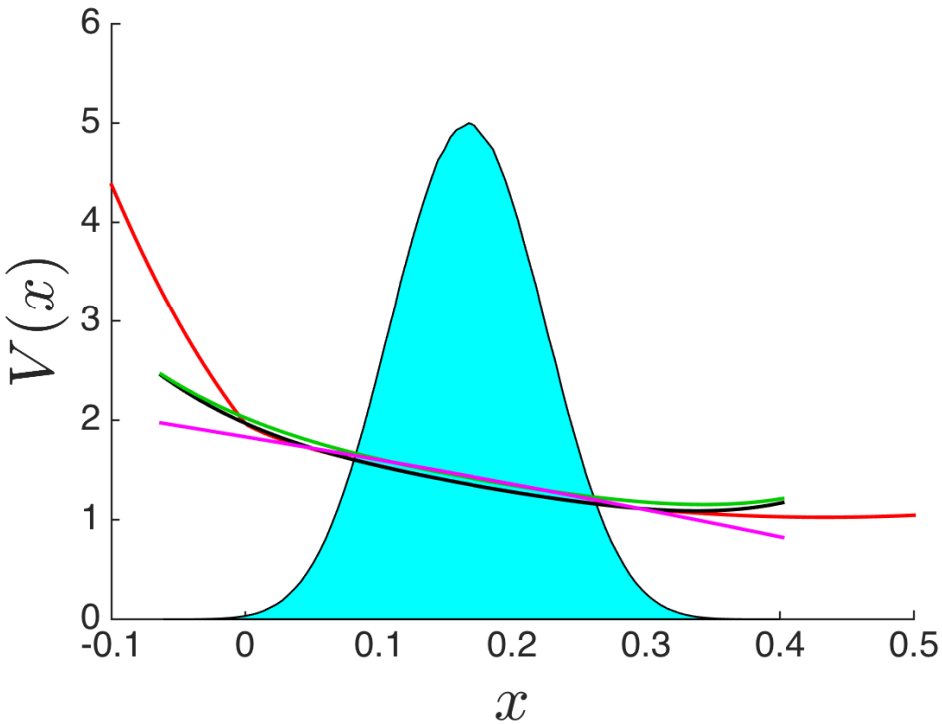
Iteration reads as:

$$\hat{\theta} \leftarrow \hat{\theta} + \alpha \mathbb{E}_{\tau_\pi}\left[(V_\pi - V_\theta)\nabla_\theta V_\theta\right]$$

**Stochastic gradient** approach:

$$\hat{\theta} \leftarrow \hat{\theta} + \alpha\,(V_\pi - V_\theta)\nabla_\theta V_\theta$$

$$\hat{\theta} \leftarrow \hat{\theta} + \alpha\delta \cdot \nabla_\theta V_\theta$$
$$\delta = L(\mathbf{x}, \pi(\mathbf{x})) + \gamma V_\theta(\mathbf{x}_+) - V_\theta(\mathbf{x})$$




$V_\pi(\mathbf{x})$, $\hat{V}(\mathbf{x})$, $V_{\mathrm{TD}}(\mathbf{x})$, $V_{\mathrm{Gradient}}(\mathbf{x})$

# TD($\lambda$) approach

**Stochastic gradient with TD**:

$$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} + \alpha\delta\nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}$$
$$\delta = L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma V_{\boldsymbol{\theta}}(\mathbf{x}_+) - V_{\boldsymbol{\theta}}(\mathbf{x})$$

**Stochastic gradient with TD($\lambda$)**:

$$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} + \alpha\delta E$$
$$E \leftarrow \gamma\lambda E + \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}$$
$$\delta = L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma V_{\boldsymbol{\theta}}(\mathbf{x}_+) - V_{\boldsymbol{\theta}}(\mathbf{x})$$

# TD($\lambda$) approach

**Stochastic gradient with TD**:

$$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} + \alpha\delta\nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}$$

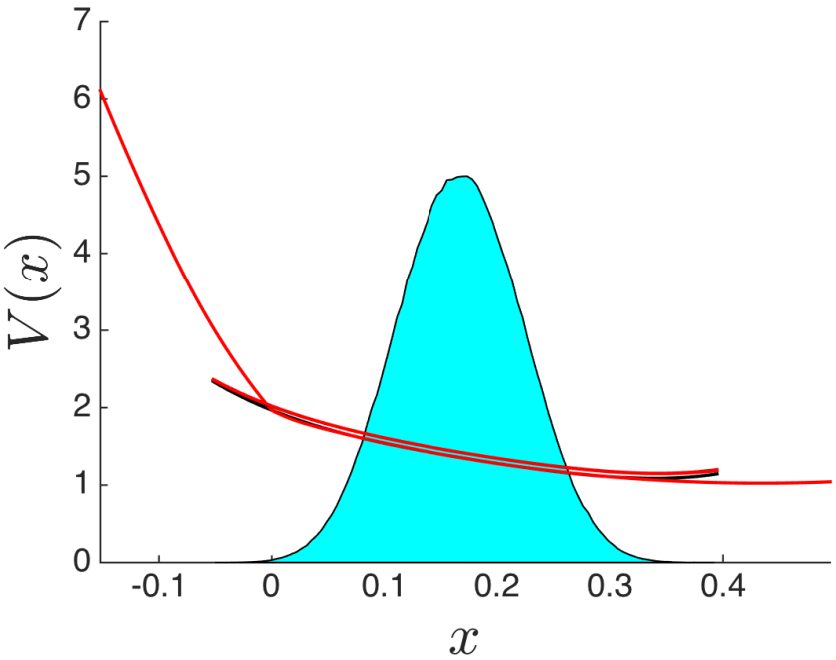
$$\delta = L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma V_{\boldsymbol{\theta}}(\mathbf{x}_+) - V_{\boldsymbol{\theta}}(\mathbf{x})$$

**Stochastic gradient with TD($\lambda$)**:

$$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} + \alpha\delta E$$

$$E \leftarrow \gamma\lambda E + \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}$$

$$\delta = L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma V_{\boldsymbol{\theta}}(\mathbf{x}_+) - V_{\boldsymbol{\theta}}(\mathbf{x})$$



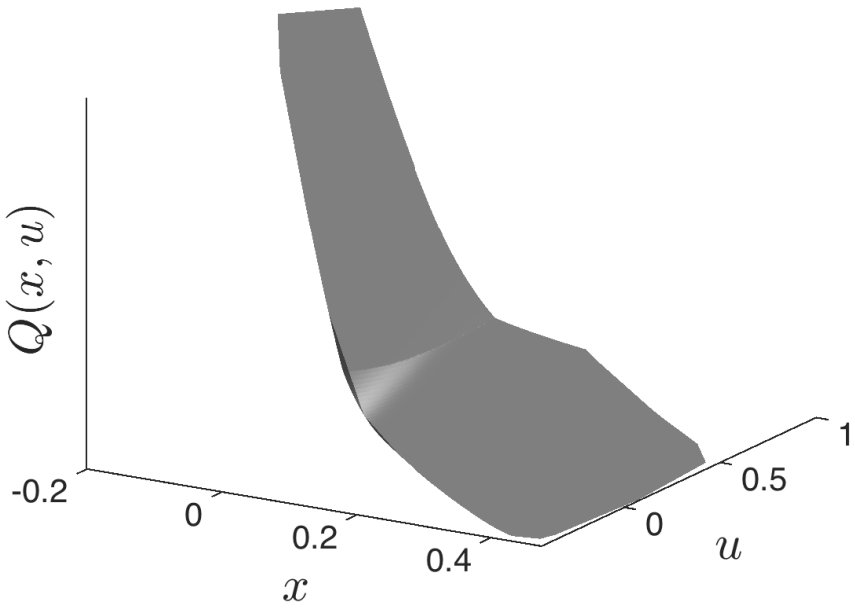
# Outline

# Function fitting for $Q_\theta$ - Temporal Difference

**Value function fitting:**

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{2} \mathbb{E}_{\tau_{\boldsymbol{\pi}}^{\mathrm{d}}} \left[ \left( Q_{\boldsymbol{\pi}}(\mathbf{x}, \mathbf{u}) - Q_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u}) \right)^2 \right]$$

Where $\tau_{\boldsymbol{\pi}}^{\mathrm{d}}$ is the Markov chain distribution under disturbed policy $\boldsymbol{\pi} + \mathbf{d}$



# Function fitting for $Q_\theta$ - Temporal Difference

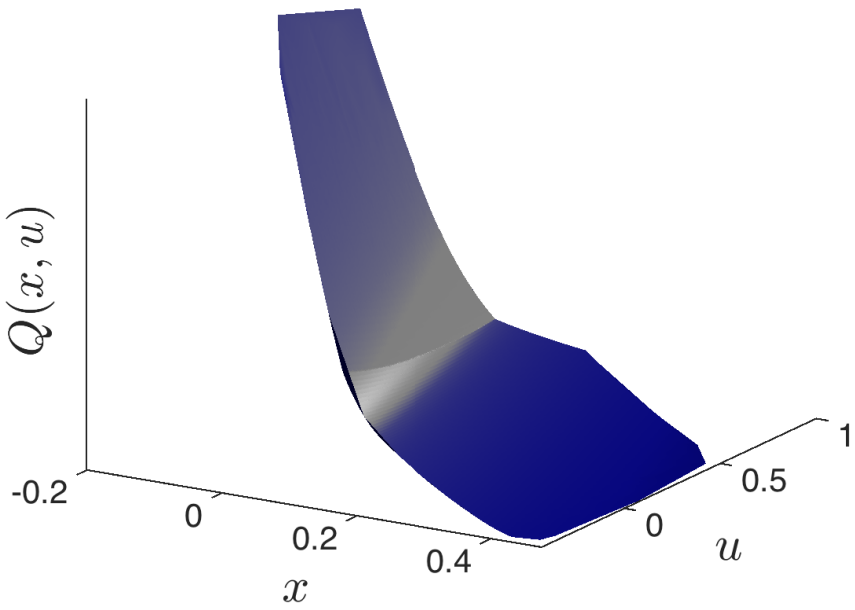
**Value function fitting:**

$$\hat{\theta} = \mathrm{a}\min_{\theta} \frac{1}{2}\mathbb{E}_{\tau_\pi^{\mathrm{d}}}\left[\left(Q_\pi(\mathbf{x},\mathbf{u}) - Q_\theta(\mathbf{x},\mathbf{u})\right)^2\right]$$

**Optimal parameter $\hat{\theta}$ satisfies:**

$$\mathbb{E}_{\tau_\pi^{\mathrm{d}}}\left[\left(Q_\pi - Q_\theta\right)\nabla_\theta Q_\theta\right] = 0$$

Where $\tau_\pi^{\mathrm{d}}$ is the Markov chain distribution under disturbed policy $\boldsymbol{\pi} + \mathbf{d}$



# Function fitting for $Q_\theta$ - Temporal Difference

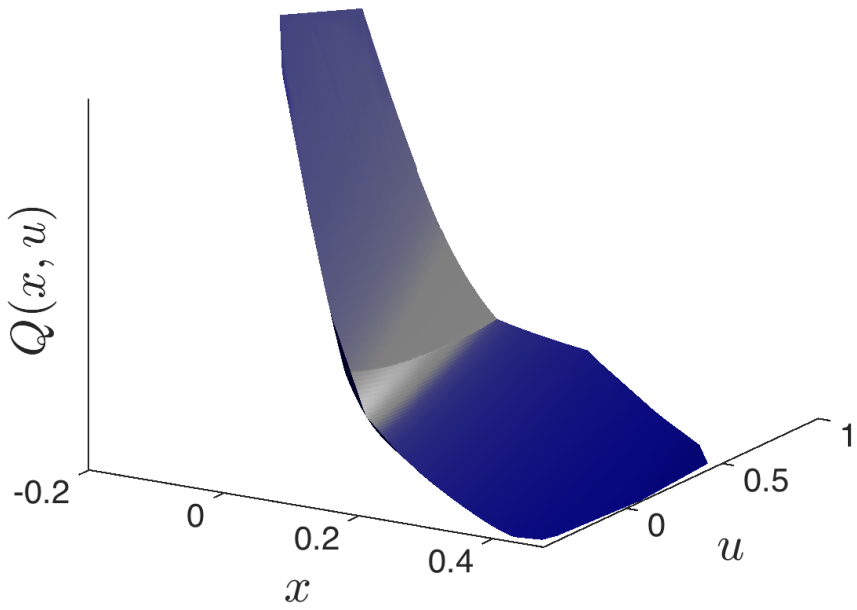
**Value function fitting:**

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{2}\mathbb{E}_{\tau_\pi^{\mathrm{d}}}\left[\left(Q_\pi(\mathbf{x},\mathbf{u}) - Q_\theta(\mathbf{x},\mathbf{u})\right)^2\right]$$

**Optimal parameter $\hat{\theta}$ satisfies:**

$$\mathbb{E}_{\tau_\pi^{\mathrm{d}}}\left[\left(Q_\pi - Q_\theta\right)\nabla_\theta Q_\theta\right] = 0$$

Where $\tau_\pi^{\mathrm{d}}$ is the Markov chain distribution under disturbed policy $\pi + \mathbf{d}$




**Where do we get $Q_\pi$ from?** Recall that

$$Q_\pi(\mathbf{x},\mathbf{u}) = L(\mathbf{x},\mathbf{u}) + \gamma\mathbb{E}\left[Q_\pi(\mathbf{x}_+,\pi(\mathbf{x}_+))|\mathbf{x},\mathbf{u}\right]$$

**TD evaluation**

$$\begin{aligned} Q_\pi(\mathbf{x},\mathbf{u}) &\approx L(\mathbf{x},\mathbf{u}) + \gamma Q_\pi(\mathbf{x}_+,\pi(\mathbf{x})) \\ &\approx L(\mathbf{x},\mathbf{u}) + \gamma Q_\theta(\mathbf{x}_+,\pi(\mathbf{x})) \end{aligned}$$

# Function fitting for $Q_\theta$ - Temporal Difference

**Value function fitting:**

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{2} \mathbb{E}_{\tau_\pi^{\mathrm{d}}} \left[ \left( Q_\pi(\mathbf{x}, \mathbf{u}) - Q_\theta(\mathbf{x}, \mathbf{u}) \right)^2 \right]$$

**Optimal parameter $\hat{\theta}$ satisfies:**

$$\mathbb{E}_{\tau_\pi^{\mathrm{d}}} \left[ (Q_\pi - Q_\theta) \nabla_\theta Q_\theta \right] = 0$$

Where $\tau_\pi^{\mathrm{d}}$ is the Markov chain distribution under disturbed policy $\pi + \mathbf{d}$

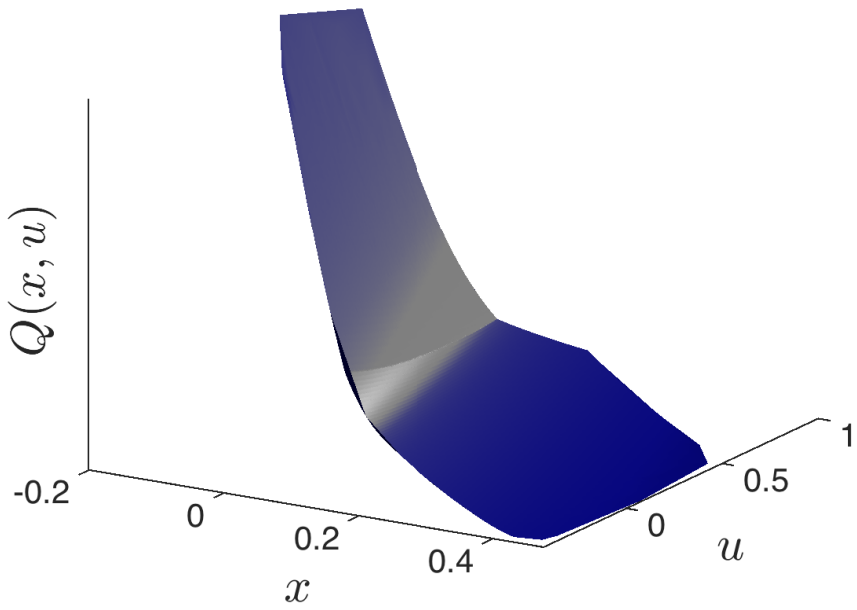
Using

$$Q_\pi(\mathbf{x}, \mathbf{u}) \approx L(\mathbf{x}, \mathbf{u}) + \gamma Q_\theta(\mathbf{x}_+, \pi(\mathbf{x}_+))$$

**Compute parameter $\hat{\theta}$ using:**

$$\mathbb{E}_{\tau_\pi^{\mathrm{d}}} \left[ \delta_Q \nabla_\theta Q_\theta \right] = 0$$

$$\delta_Q = L(\mathbf{x}, \mathbf{u}) + \gamma Q_\theta(\mathbf{x}_+, \pi(\mathbf{x}_+)) - Q_\theta(\mathbf{x}, \mathbf{u})$$



# Function fitting for $Q_\theta$ - Temporal Difference

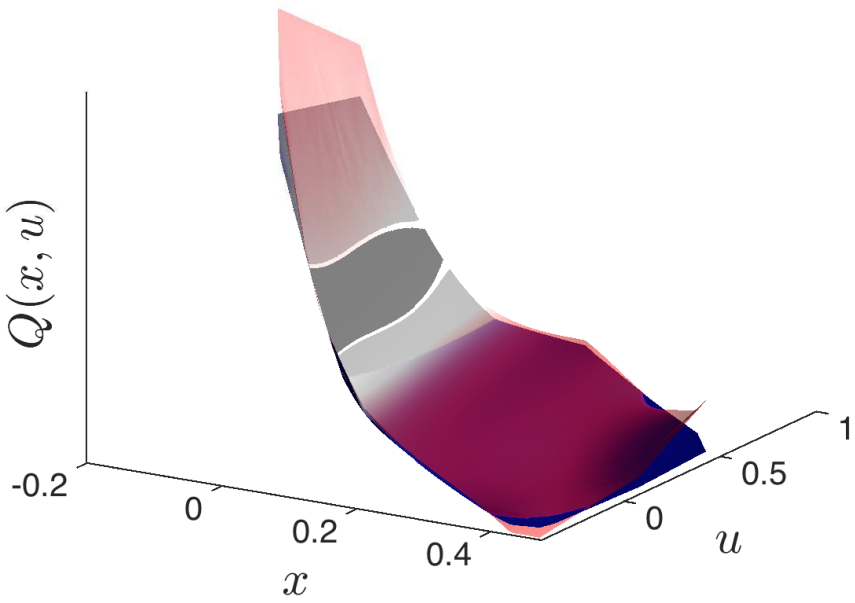
**Value function fitting:**

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{2}\mathbb{E}_{\tau_\pi^{\mathrm{d}}}\left[(Q_\pi(\mathbf{x},\mathbf{u}) - Q_\theta(\mathbf{x},\mathbf{u}))^2\right]$$

**Optimal parameter $\hat{\theta}$ satisfies:**

$$\mathbb{E}_{\tau_\pi^{\mathrm{d}}}\left[(Q_\pi - Q_\theta)\nabla_\theta Q_\theta\right] = 0$$

Where $\tau_\pi^{\mathrm{d}}$ is the Markov chain distribution under disturbed policy $\pi + \mathbf{d}$




Using

$$Q_\pi(\mathbf{x},\mathbf{u}) \approx L(\mathbf{x},\mathbf{u}) + \gamma Q_\theta(\mathbf{x}_+, \pi(\mathbf{x}_+))$$

**Compute parameter $\hat{\theta}$ using:**

$$\mathbb{E}_{\tau_\pi^{\mathrm{d}}}\left[\delta_Q \nabla_\theta Q_\theta\right] = 0$$

$$\delta_Q = L(\mathbf{x},\mathbf{u}) + \gamma Q_\theta(\mathbf{x}_+, \pi(\mathbf{x}_+)) - Q_\theta(\mathbf{x},\mathbf{u})$$

# Function fitting for $Q_\theta$ - Temporal Difference

**Value function fitting:**

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{2}\mathbb{E}_{\tau_\pi^{\mathrm{d}}}\left[(Q_\pi(\mathbf{x},\mathbf{u}) - Q_\theta(\mathbf{x},\mathbf{u}))^2\right]$$

**Optimal parameter $\hat{\theta}$ satisfies:**

$$\mathbb{E}_{\tau_\pi^{\mathrm{d}}}\left[(Q_\pi - Q_\theta)\nabla_\theta Q_\theta\right] = 0$$

Where $\tau_\pi^{\mathrm{d}}$ is the Markov chain distribution under disturbed policy $\pi + \mathbf{d}$

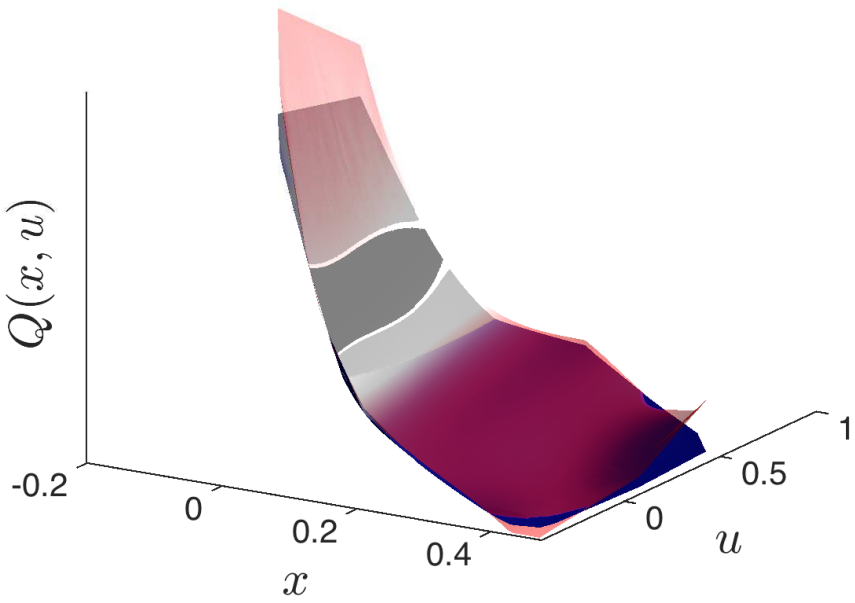
Using

$$Q_\pi(\mathbf{x},\mathbf{u}) \approx L(\mathbf{x},\mathbf{u}) + \gamma Q_\theta(\mathbf{x}_+, \pi(\mathbf{x}_+))$$

**Compute parameter $\hat{\theta}$ using:**

$$\mathbb{E}_{\tau_\pi^{\mathrm{d}}}\left[\delta_Q \nabla_\theta Q_\theta\right] = 0$$

$$\delta_Q = L(\mathbf{x},\mathbf{u}) + \gamma Q_\theta(\mathbf{x}_+, \pi(\mathbf{x}_+)) - Q_\theta(\mathbf{x},\mathbf{u})$$




**Stochastic gradient**:

$$\hat{\theta} \leftarrow \hat{\theta} + \alpha\delta_Q \nabla_\theta Q_\theta$$

# Function fitting for $Q_\theta$ - Temporal Difference

**Value function fitting:**

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{2}\mathbb{E}_{\tau_\pi^{\mathrm{d}}}\left[\left(Q_\pi(\mathbf{x},\mathbf{u}) - Q_\theta(\mathbf{x},\mathbf{u})\right)^2\right]$$

**Optimal parameter $\hat{\theta}$ satisfies:**

$$\mathbb{E}_{\tau_\pi^{\mathrm{d}}}\left[\left(Q_\pi - Q_\theta\right)\nabla_\theta Q_\theta\right] = 0$$

Using

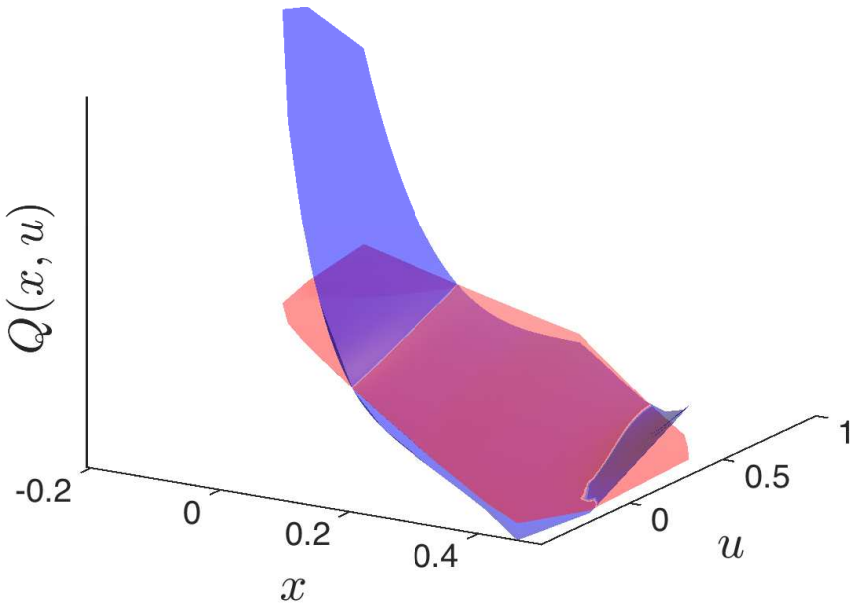
$$Q_\pi(\mathbf{x},\mathbf{u}) \approx L(\mathbf{x},\mathbf{u}) + \gamma Q_\theta(\mathbf{x}_+, \pi(\mathbf{x}_+))$$

**Compute parameter $\hat{\theta}$ using:**

$$\mathbb{E}_{\tau_\pi^{\mathrm{d}}}\left[\delta_Q \nabla_\theta Q_\theta\right] = 0$$

$$\delta_Q = L(\mathbf{x},\mathbf{u}) + \gamma Q_\theta(\mathbf{x}_+, \pi(\mathbf{x}_+)) - Q_\theta(\mathbf{x},\mathbf{u})$$

Where $\tau_\pi^{\mathrm{d}}$ is the Markov chain distribution under disturbed policy $\pi + \mathbf{d}$




**Stochastic gradient**:

$$\hat{\theta} \leftarrow \hat{\theta} + \alpha\delta_Q \nabla_\theta Q_\theta$$

# Off/On-policy $Q$-Learning - SARSA($\lambda$)

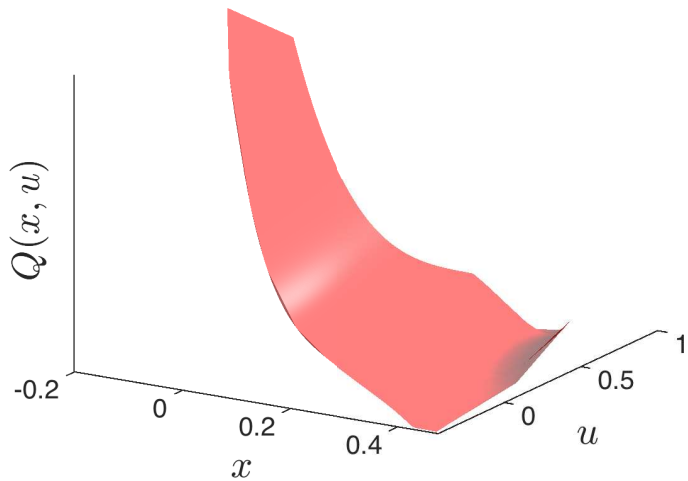
**Off-policy TD error**

$$\delta_Q^{\text{off}} = L(\mathbf{x}_k, \mathbf{u}_k) + \gamma Q_{\boldsymbol{\theta}}(\mathbf{x}_{k+1}, \boldsymbol{\pi}(\mathbf{x}_{k+1})) - Q_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k)$$

**Stochastic gradient**

$$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} + \alpha \delta_Q^{\text{off}} \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}$$

where $\mathbf{u}_k$ can be arbitrary




$$\mathbb{E}_{\tau_{\boldsymbol{\pi}}^{\text{d}}}\left[\delta_Q^{\text{on/off}} \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}\right] = 0$$

# Off/On-policy $Q$-Learning - SARSA($\lambda$)

**Off-policy TD error**

$$\delta_Q^{\text{off}} = L(\mathbf{x}_k, \mathbf{u}_k) + \gamma Q_{\boldsymbol{\theta}}(\mathbf{x}_{k+1}, \boldsymbol{\pi}(\mathbf{x}_{k+1})) - Q_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k)$$

**Stochastic gradient**

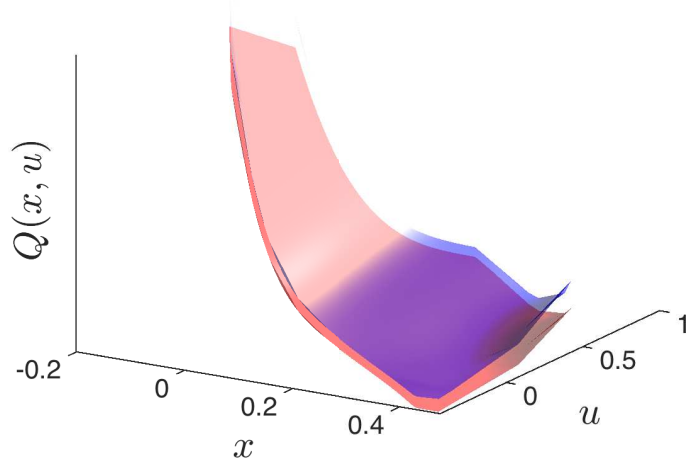
$$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} + \alpha \delta_Q^{\text{off}} \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}$$

where $\mathbf{u}_k$ can be arbitrary

**On-policy TD error**

$$\delta_Q^{\text{on}} = L(\mathbf{x}_k, \mathbf{u}_k) + \gamma Q_{\boldsymbol{\theta}}(\mathbf{x}_{k+1}, \mathbf{u}_{k+1}) - Q_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k)$$

where $\mathbf{u}_k$ is drawn from $\boldsymbol{\pi}(\mathbf{x}_k)$ + disturbance




$$\mathbb{E}_{\tau_{\boldsymbol{\pi}}^{\text{d}}} \left[ \delta_Q^{\text{on/off}} \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}} \right] = 0$$

# Off/On-policy $Q$-Learning - SARSA($\lambda$)

**Off-policy TD error**

$$\delta_Q^{\text{off}} = L(\mathbf{x}_k, \mathbf{u}_k) + \gamma Q_{\boldsymbol{\theta}}(\mathbf{x}_{k+1}, \boldsymbol{\pi}(\mathbf{x}_{k+1})) - Q_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k)$$

**Stochastic gradient**

$$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} + \alpha \delta_Q^{\text{off}} \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}$$

where $\mathbf{u}_k$ can be arbitrary

**On-policy TD error**

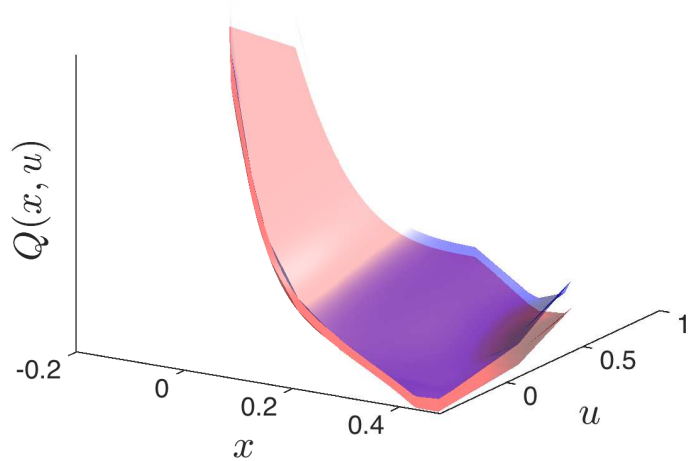
$$\delta_Q^{\text{on}} = L(\mathbf{x}_k, \mathbf{u}_k) + \gamma Q_{\boldsymbol{\theta}}(\mathbf{x}_{k+1}, \mathbf{u}_{k+1}) - Q_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k)$$

**SARSA($\lambda$)**:

$$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} + \alpha \delta_Q^{\text{on}} E$$

$$E \leftarrow \gamma \lambda E + \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}$$

where $\mathbf{u}_k$ is drawn from $\boldsymbol{\pi}(\mathbf{x}_k)$ + disturbance




$$\mathbb{E}_{\tau_{\boldsymbol{\pi}}^{\text{d}}} \left[ \delta_Q^{\text{on/off}} \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}} \right] = 0$$

# Off/On-policy $Q$-Learning - SARSA($\lambda$)

**Off-policy TD error**

$$\delta_Q^{\text{off}} = L(\mathbf{x}_k, \mathbf{u}_k) + \gamma Q_{\boldsymbol{\theta}}(\mathbf{x}_{k+1}, \boldsymbol{\pi}(\mathbf{x}_{k+1})) - Q_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k)$$

**Stochastic gradient**

$$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} + \alpha \delta_Q^{\text{off}} \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}$$

where $\mathbf{u}_k$ can be arbitrary

**On-policy TD error**

$$\delta_Q^{\text{on}} = L(\mathbf{x}_k, \mathbf{u}_k) + \gamma Q_{\boldsymbol{\theta}}(\mathbf{x}_{k+1}, \mathbf{u}_{k+1}) - Q_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k)$$

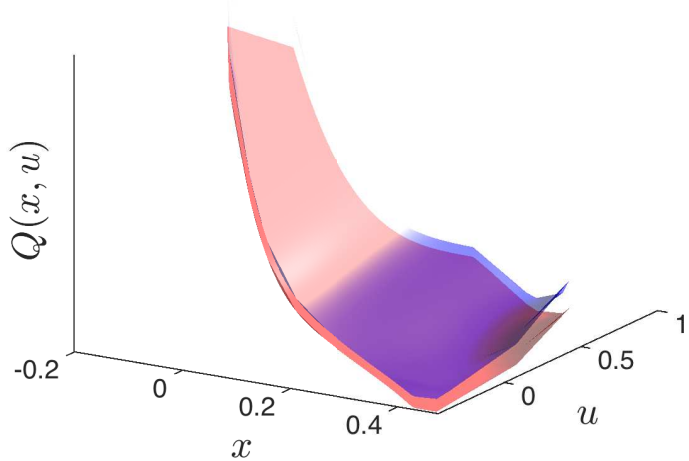
**SARSA($\lambda$)**:

$$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} + \alpha \delta_Q^{\text{on}} E$$

$$E \leftarrow \gamma \lambda E + \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}$$

where $\mathbf{u}_k$ is drawn from $\boldsymbol{\pi}(\mathbf{x}_k)$ + disturbance

**SARSA($\lambda$)** gets $Q_{\boldsymbol{\theta}} \approx Q_{\boldsymbol{\pi}}$ if disturbance vanishes over time, while exploring the state-input state "$\infty$"-many times




$$\mathbb{E}_{\tau_{\boldsymbol{\pi}}^{\text{d}}} \left[ \delta_Q^{\text{on/off}} \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}} \right] = 0$$

# Outline

# LSTD - Efficient evaluation of $V$

Recall that we want to solve

$$\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[\delta \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}\right] = 0$$
$$\delta = L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma V_{\boldsymbol{\theta}}(\mathbf{x}_+) - V_{\boldsymbol{\theta}}(\mathbf{x})$$

for $\boldsymbol{\theta}$

**Stochastic gradient**:

$$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} + \alpha \delta \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}$$

gets us there (or close enough...)

- $+$ trivial to code
- $+$ computationally cheap (for $\boldsymbol{\theta}$ large)
- $-$ inefficient (many many iterations)

# LSTD - Efficient evaluation of $V$

Recall that we want to solve

$$\mathbb{E}_{\tau_\pi}\left[\delta \nabla_\theta V_\theta\right] = 0$$
$$\delta = L(\mathbf{x}, \pi(\mathbf{x})) + \gamma V_\theta(\mathbf{x}_+) - V_\theta(\mathbf{x})$$

for $\theta$

Given trajectory, **LSTD** solves:

$$\frac{1}{N}\sum_{k=0}^{N-1} \delta_k(\mathbf{x}_k, \mathbf{x}_{k+1}) \nabla_\theta V_\theta(\mathbf{x}_k) = 0$$
$$\delta_k = L(\mathbf{x}_k, \pi(\mathbf{x}_k)) + \gamma V_\theta(\mathbf{x}_{k+1}) - V_\theta(\mathbf{x}_k)$$

for observed $\mathbf{x}_{0,\ldots,N}$

**Stochastic gradient**:

$$\hat{\theta} \leftarrow \hat{\theta} + \alpha \delta \nabla_\theta V_\theta$$

gets us there (or close enough...)

- $+$ trivial to code
- $+$ computationally cheap (for $\theta$ large)
- $-$ inefficient (many many iterations)

# LSTD - Efficient evaluation of $V$

Recall that we want to solve

$$\mathbb{E}_{\tau_\pi}\left[\delta \nabla_\theta V_\theta\right] = 0$$
$$\delta = L(\mathbf{x}, \pi(\mathbf{x})) + \gamma V_\theta(\mathbf{x}_+) - V_\theta(\mathbf{x})$$

for $\theta$

Given trajectory, **LSTD** solves:

$$\frac{1}{N}\sum_{k=0}^{N-1} \delta_k(\mathbf{x}_k, \mathbf{x}_{k+1}) \nabla_\theta V_\theta(\mathbf{x}_k) = 0$$
$$\delta_k = L(\mathbf{x}_k, \pi(\mathbf{x}_k)) + \gamma V_\theta(\mathbf{x}_{k+1}) - V_\theta(\mathbf{x}_k)$$

for observed $\mathbf{x}_{0,\ldots,N}$

- If $V_\theta$ is linear in $\theta$ then solution is explicit, global solution
- If nonlinear, Newton required
- Nonlinear $\rightarrow$ possibly local solution / initial guess matters

**Stochastic gradient**:

$$\hat{\theta} \leftarrow \hat{\theta} + \alpha \delta \nabla_\theta V_\theta$$

gets us there (or close enough...)

- $+$ trivial to code
- $+$ computationally cheap (for $\theta$ large)
- $-$ inefficient (many many iterations)

# LSTD - Efficient evaluation of $V$

Recall that we want to solve

$$\mathbb{E}_{\tau_\pi}\left[\delta \nabla_\theta V_\theta\right] = 0$$
$$\delta = L(\mathbf{x}, \pi(\mathbf{x})) + \gamma V_\theta(\mathbf{x}_+) - V_\theta(\mathbf{x})$$

for $\theta$

Given trajectory, **LSTD** solves:

$$\frac{1}{N}\sum_{k=0}^{N-1} \delta_k(\mathbf{x}_k, \mathbf{x}_{k+1}) \nabla_\theta V_\theta(\mathbf{x}_k) = 0$$
$$\delta_k = L(\mathbf{x}_k, \pi(\mathbf{x}_k)) + \gamma V_\theta(\mathbf{x}_{k+1}) - V_\theta(\mathbf{x}_k)$$

for observed $\mathbf{x}_{0,\ldots,N}$

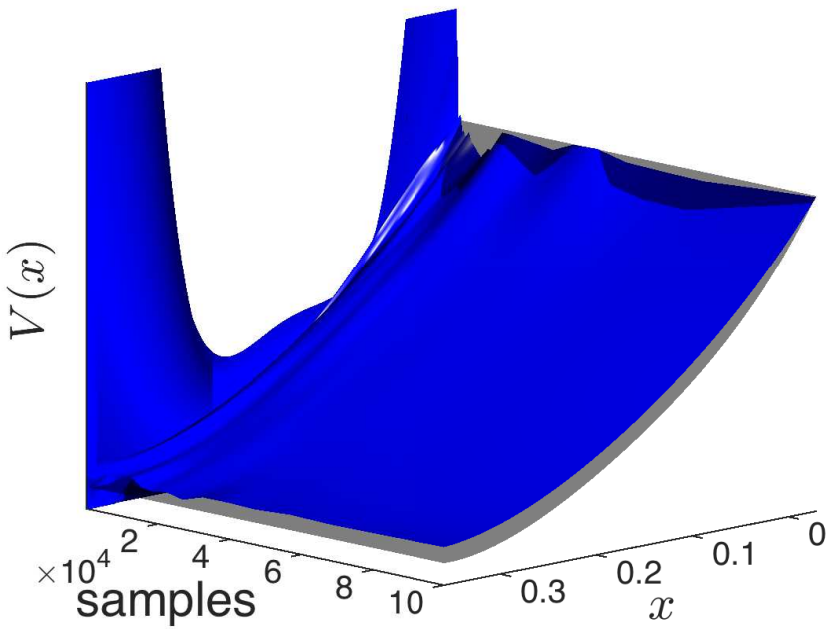
- If $V_\theta$ is linear in $\theta$ then solution is explicit, global solution
- If nonlinear, Newton required
- Nonlinear $\rightarrow$ possibly local solution / initial guess matters

**Stochastic gradient**:

$$\hat{\theta} \leftarrow \hat{\theta} + \alpha\delta\nabla_\theta V_\theta$$

gets us there (or close enough...)

- $+$ trivial to code
- $+$ computationally cheap (for $\theta$ large)
- $-$ inefficient (many many iterations)



# LSTDQ - Efficient evaluation of $Q_\pi$

Recall that we want to solve

$$\mathbb{E}_{\tau^{\mathrm{d}}_{\boldsymbol{\pi}}}\left[\delta \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}\right] = 0$$

$$\delta_Q = L(\mathbf{x}, \mathbf{u}) + \gamma Q_{\boldsymbol{\theta}}(\mathbf{x}_+, \boldsymbol{\pi}_+) - Q_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u})$$

for $\boldsymbol{\theta}$

**Stochastic gradient**:

$$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} + \alpha \delta_Q \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}$$

gets us there.

- $+$ trivial to code
- $+$ computationally cheap (for $\boldsymbol{\theta}$ large)
- $-$ inefficient (many many iterations)

# LSTDQ - Efficient evaluation of $Q_\pi$

Recall that we want to solve

$$\mathbb{E}_{\tau^{\mathrm{d}}_{\pi}}\left[\delta \nabla_{\theta} Q_{\theta}\right] = 0$$

$$\delta_Q = L(\mathbf{x}, \mathbf{u}) + \gamma Q_{\theta}(\mathbf{x}_+, \boldsymbol{\pi}_+) - Q_{\theta}(\mathbf{x}, \mathbf{u})$$

for $\theta$

**LSTDQ** solves:

$$\sum_{k=0}^{N-1} \delta_k \nabla_{\theta} Q_{\theta}\left(\mathbf{x}_k, \mathbf{u}_k\right) = 0$$

$$\delta_k = L(\mathbf{x}_k, \mathbf{u}_k) + \gamma Q_{\theta}(\mathbf{x}_{k+1}, \boldsymbol{\pi}(\mathbf{x}_{k+1})) - Q_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

for observed trajectory $(\mathbf{x}_0, \mathbf{u}_0), \ldots, (\mathbf{x}_N, \mathbf{u}_N)$

- If $Q_\theta$ is linear in $\theta$ then solution is explicit, global solution
- If nonlinear, Newton required
- Nonlinear $\rightarrow$ possibly local solution / initial guess matters

**Stochastic gradient**:

$$\hat{\theta} \leftarrow \hat{\theta} + \alpha \delta_Q \nabla_{\theta} Q_{\theta}$$

gets us there.

- $+$ trivial to code
- $+$ computationally cheap (for $\theta$ large)
- $-$ inefficient (many many iterations)

# LSTDQ - Efficient evaluation of $Q_\pi$

Recall that we want to solve

$$\mathbb{E}_{\tau^{\mathrm{d}}_{\pi}}\left[\delta \nabla_{\theta} Q_{\theta}\right] = 0$$

$$\delta_Q = L(\mathbf{x}, \mathbf{u}) + \gamma Q_{\theta}(\mathbf{x}_+, \boldsymbol{\pi}_+) - Q_{\theta}(\mathbf{x}, \mathbf{u})$$

for $\theta$

**Algorithm:** LSTDQ (prototype)

**Input**: Initial $Q$ parameters $\theta$

Collect $\mathcal{D} = \{(\mathbf{x}_k, \mathbf{u}_k, \mathbf{x}_{k+1}),\ k = 0, ..., N\}$

**while** $\|\zeta_{\theta}\| > \mathrm{Tol}$ **do**

Form

$$\zeta_{\theta} = \sum_{\mathcal{D}} \delta_k \nabla_{\theta} Q_{\theta}(\mathbf{x}_k, \mathbf{u}_k)$$

and gradient

Newton step: $\theta \leftarrow \theta + \alpha \nabla_{\theta} \zeta_{\theta}^{-1} \zeta_{\theta}$

**Stochastic gradient**:

$$\hat{\theta} \leftarrow \hat{\theta} + \alpha \delta_Q \nabla_{\theta} Q_{\theta}$$

gets us there.

- $+$ trivial to code
- $+$ computationally cheap (for $\theta$ large)
- $-$ inefficient (many many iterations)

# LSTDQ - Efficient evaluation of $Q_\pi$

Recall that we want to solve

$$\mathbb{E}_{\tau_\pi^{\mathrm{d}}}\left[\delta \nabla_\theta Q_\theta\right] = 0$$

$$\delta_Q = L(\mathbf{x}, \mathbf{u}) + \gamma Q_\theta(\mathbf{x}_+, \boldsymbol{\pi}_+) - Q_\theta(\mathbf{x}, \mathbf{u})$$

for $\theta$

**Stochastic gradient**:

$$\hat{\theta} \leftarrow \hat{\theta} + \alpha \delta_Q \nabla_\theta Q_\theta$$

gets us there.

- $+$ trivial to code
- $+$ computationally cheap (for $\theta$ large)
- $-$ inefficient (many many iterations)

**Algorithm:** LSTDQ (prototype)

**Input**: Initial $Q$ parameters $\theta$

Collect $\mathcal{D} = \{(\mathbf{x}_k, \mathbf{u}_k, \mathbf{x}_{k+1}),\ k = 0, ..., N\}$

**while** $\|\zeta_\theta\| > \mathrm{Tol}$ **do**

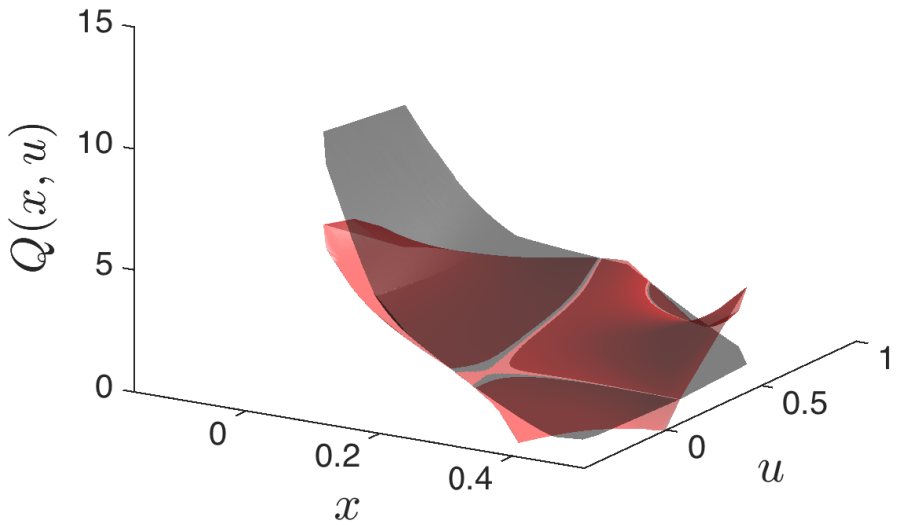
Form

$$\zeta_\theta = \sum_{\mathcal{D}} \delta_k \nabla_\theta Q_\theta(\mathbf{x}_k, \mathbf{u}_k)$$

and gradient

Newton step: $\theta \leftarrow \theta + \alpha \nabla_\theta \zeta_\theta^{-1} \zeta_\theta$

**# samples = 10**



# LSTDQ - Efficient evaluation of $Q_\pi$

Recall that we want to solve

$$\mathbb{E}_{\tau_\pi^{\mathrm{d}}}\left[\delta \nabla_\theta Q_\theta\right] = 0$$

$$\delta_Q = L(\mathbf{x}, \mathbf{u}) + \gamma Q_\theta(\mathbf{x}_+, \boldsymbol{\pi}_+) - Q_\theta(\mathbf{x}, \mathbf{u})$$

for $\theta$

**Stochastic gradient**:

$$\hat{\theta} \leftarrow \hat{\theta} + \alpha \delta_Q \nabla_\theta Q_\theta$$

gets us there.

- $+$ trivial to code
- $+$ computationally cheap (for $\theta$ large)
- $-$ inefficient (many many iterations)

**Algorithm:** LSTDQ (prototype)

**Input**: Initial $Q$ parameters $\theta$

Collect $\mathcal{D} = \{(\mathbf{x}_k, \mathbf{u}_k, \mathbf{x}_{k+1}),\ k = 0, ..., N\}$

**while** $\|\zeta_\theta\| > \mathrm{Tol}$ **do**

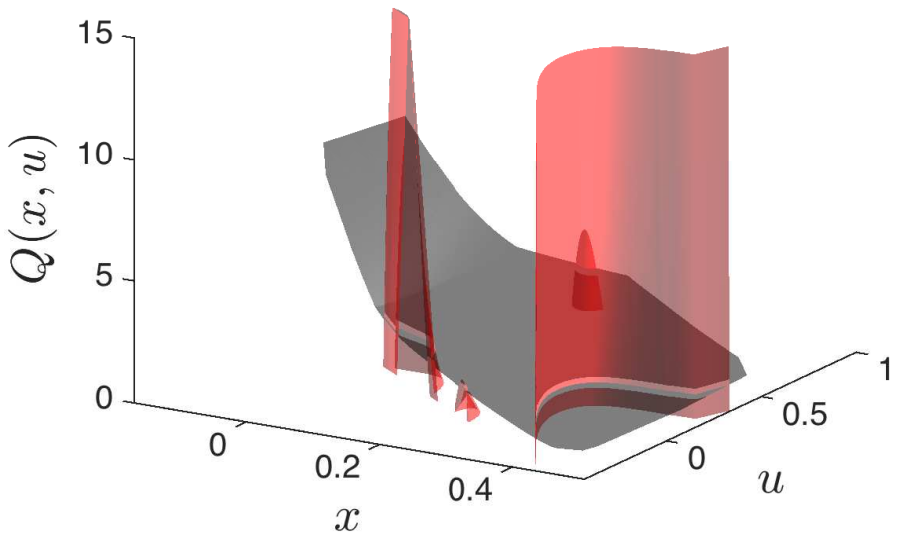
Form

$$\zeta_\theta = \sum_{\mathcal{D}} \delta_k \nabla_\theta Q_\theta(\mathbf{x}_k, \mathbf{u}_k)$$

and gradient

Newton step: $\theta \leftarrow \theta + \alpha \nabla_\theta \zeta_\theta^{-1} \zeta_\theta$

**# samples = 35**



# LSTDQ - Efficient evaluation of $Q_\pi$

Recall that we want to solve

$$\mathbb{E}_{\tau_\pi^{\mathrm{d}}}\left[\delta \nabla_\theta Q_\theta\right] = 0$$

$$\delta_Q = L(\mathbf{x},\mathbf{u}) + \gamma Q_\theta(\mathbf{x}_+,\boldsymbol{\pi}_+) - Q_\theta(\mathbf{x},\mathbf{u})$$

for $\theta$

**Stochastic gradient**:

$$\hat{\theta} \leftarrow \hat{\theta} + \alpha \delta_Q \nabla_\theta Q_\theta$$

gets us there.

- $+$ trivial to code
- $+$ computationally cheap (for $\theta$ large)
- $-$ inefficient (many many iterations)

**Algorithm:** LSTDQ (prototype)

**Input**: Initial $Q$ parameters $\theta$

Collect $\mathcal{D} = \{(\mathbf{x}_k, \mathbf{u}_k, \mathbf{x}_{k+1}),\ k = 0, ..., N\}$

**while** $\|\zeta_\theta\| > \mathrm{Tol}$ **do**

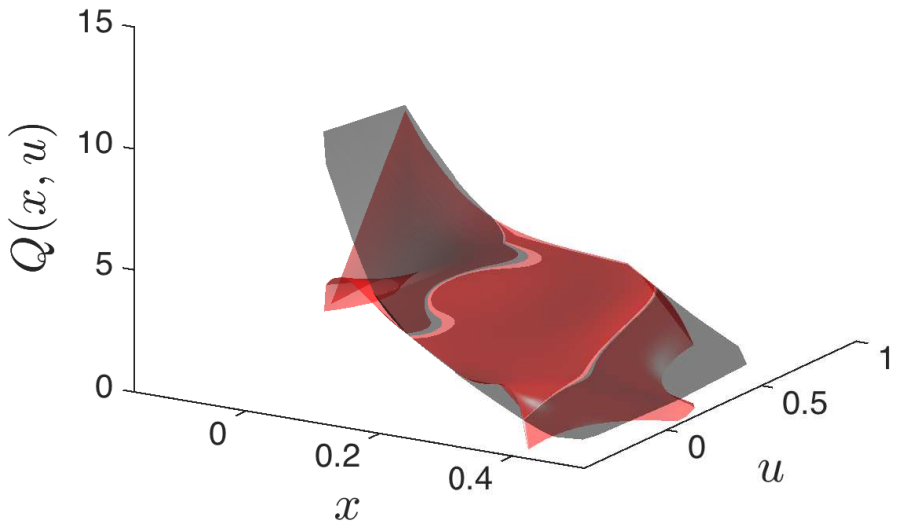
Form

$$\zeta_\theta = \sum_{\mathcal{D}} \delta_k \nabla_\theta Q_\theta(\mathbf{x}_k, \mathbf{u}_k)$$

and gradient

Newton step: $\theta \leftarrow \theta + \alpha \nabla_\theta \zeta_\theta^{-1} \zeta_\theta$

**# samples = 129**



# LSTDQ - Efficient evaluation of $Q_\pi$

Recall that we want to solve

$$\mathbb{E}_{\tau_\pi^{\mathrm{d}}}\left[\delta \nabla_\theta Q_\theta\right] = 0$$

$$\delta_Q = L(\mathbf{x},\mathbf{u}) + \gamma Q_\theta(\mathbf{x}_+,\boldsymbol{\pi}_+) - Q_\theta(\mathbf{x},\mathbf{u})$$

for $\theta$

**Stochastic gradient**:

$$\hat{\theta} \leftarrow \hat{\theta} + \alpha \delta_Q \nabla_\theta Q_\theta$$

gets us there.

- $+$ trivial to code
- $+$ computationally cheap (for $\theta$ large)
- $-$ inefficient (many many iterations)

**Algorithm:** LSTDQ (prototype)

**Input**: Initial $Q$ parameters $\theta$

Collect $\mathcal{D} = \{(\mathbf{x}_k, \mathbf{u}_k, \mathbf{x}_{k+1}),\ k = 0, ..., N\}$

**while** $\|\zeta_\theta\| > \mathrm{Tol}$ **do**

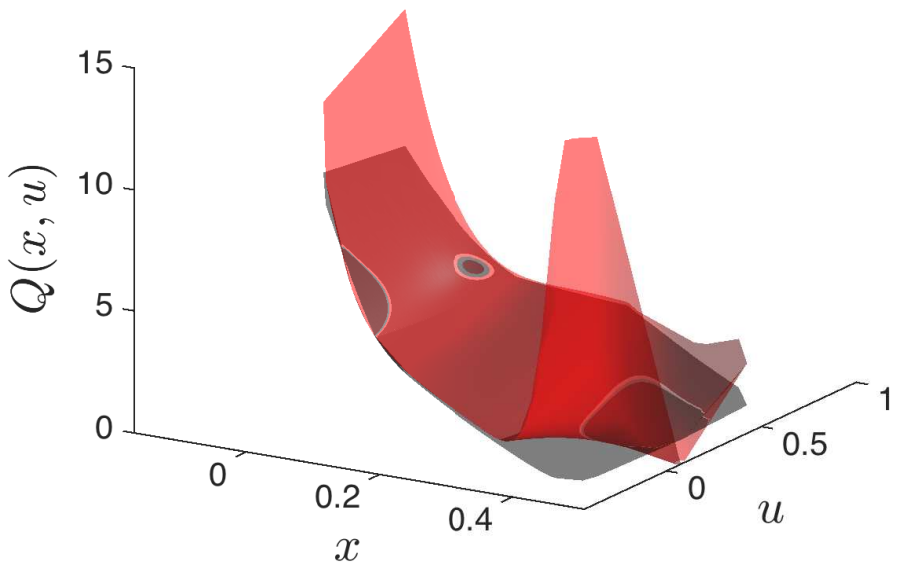
Form

$$\zeta_\theta = \sum_{\mathcal{D}} \delta_k \nabla_\theta Q_\theta(\mathbf{x}_k, \mathbf{u}_k)$$

and gradient

Newton step: $\theta \leftarrow \theta + \alpha \nabla_\theta \zeta_\theta^{-1} \zeta_\theta$

**# samples = 464**



# LSTDQ - Efficient evaluation of $Q_\pi$

Recall that we want to solve

$$\mathbb{E}_{\tau_\pi^{\mathrm{d}}}\left[\delta \nabla_\theta Q_\theta\right] = 0$$

$$\delta_Q = L(\mathbf{x},\mathbf{u}) + \gamma Q_\theta(\mathbf{x}_+,\boldsymbol{\pi}_+) - Q_\theta(\mathbf{x},\mathbf{u})$$

for $\theta$

**Stochastic gradient**:

$$\hat{\theta} \leftarrow \hat{\theta} + \alpha \delta_Q \nabla_\theta Q_\theta$$

gets us there.

- $+$ trivial to code
- $+$ computationally cheap (for $\theta$ large)
- $-$ inefficient (many many iterations)

**Algorithm:** LSTDQ (prototype)

**Input**: Initial $Q$ parameters $\theta$

Collect $\mathcal{D} = \{(\mathbf{x}_k, \mathbf{u}_k, \mathbf{x}_{k+1}),\ k = 0, ..., N\}$

**while** $\|\zeta_\theta\| > \mathrm{Tol}$ **do**

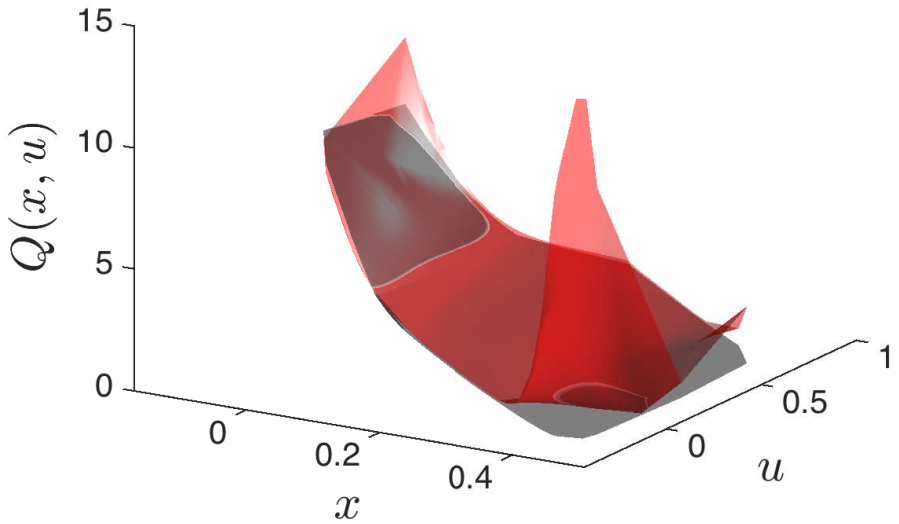
Form

$$\zeta_\theta = \sum_{\mathcal{D}} \delta_k \nabla_\theta Q_\theta(\mathbf{x}_k, \mathbf{u}_k)$$

and gradient

Newton step: $\theta \leftarrow \theta + \alpha \nabla_\theta \zeta_\theta^{-1} \zeta_\theta$

**# samples = 1668**



# LSTDQ - Efficient evaluation of $Q_\pi$

Recall that we want to solve

$$\mathbb{E}_{\tau_\pi^{\mathrm{d}}}\left[\delta \nabla_\theta Q_\theta\right] = 0$$

$$\delta_Q = L(\mathbf{x},\mathbf{u}) + \gamma Q_\theta(\mathbf{x}_+,\boldsymbol{\pi}_+) - Q_\theta(\mathbf{x},\mathbf{u})$$

for $\theta$

**Stochastic gradient**:

$$\hat{\theta} \leftarrow \hat{\theta} + \alpha \delta_Q \nabla_\theta Q_\theta$$

gets us there.

- $+$ trivial to code
- $+$ computationally cheap (for $\theta$ large)
- $-$ inefficient (many many iterations)

**Algorithm:** LSTDQ (prototype)

**Input**: Initial $Q$ parameters $\theta$

Collect $\mathcal{D} = \{(\mathbf{x}_k, \mathbf{u}_k, \mathbf{x}_{k+1}),\ k = 0, ..., N\}$

**while** $\|\zeta_\theta\| > \mathrm{Tol}$ **do**

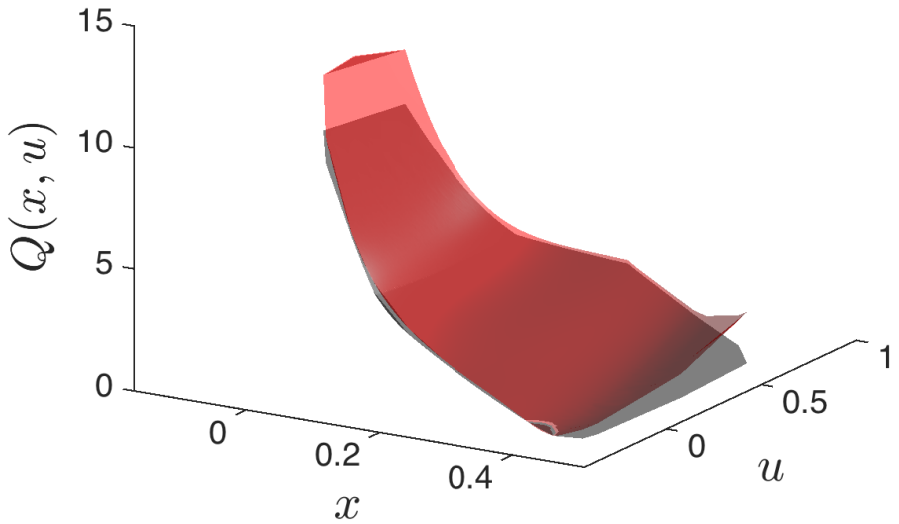
Form

$$\zeta_\theta = \sum_{\mathcal{D}} \delta_k \nabla_\theta Q_\theta(\mathbf{x}_k, \mathbf{u}_k)$$

and gradient

Newton step: $\theta \leftarrow \theta + \alpha \nabla_\theta \zeta_\theta^{-1} \zeta_\theta$

**# samples = 5994**



# LSTDQ - Efficient evaluation of $Q_\pi$

Recall that we want to solve

$$\mathbb{E}_{\tau_\pi^{\mathrm{d}}}\left[\delta \nabla_\theta Q_\theta\right] = 0$$

$$\delta_Q = L(\mathbf{x}, \mathbf{u}) + \gamma Q_\theta(\mathbf{x}_+, \boldsymbol{\pi}_+) - Q_\theta(\mathbf{x}, \mathbf{u})$$

for $\theta$

**Stochastic gradient**:

$$\hat{\theta} \leftarrow \hat{\theta} + \alpha \delta_Q \nabla_\theta Q_\theta$$

gets us there.

- $+$ trivial to code
- $+$ computationally cheap (for $\theta$ large)
- $-$ inefficient (many many iterations)

**Algorithm:** LSTDQ (prototype)

**Input**: Initial $Q$ parameters $\theta$

Collect $\mathcal{D} = \{(\mathbf{x}_k, \mathbf{u}_k, \mathbf{x}_{k+1}),\ k = 0, ..., N\}$

**while** $\|\zeta_\theta\| > \mathrm{Tol}$ **do**

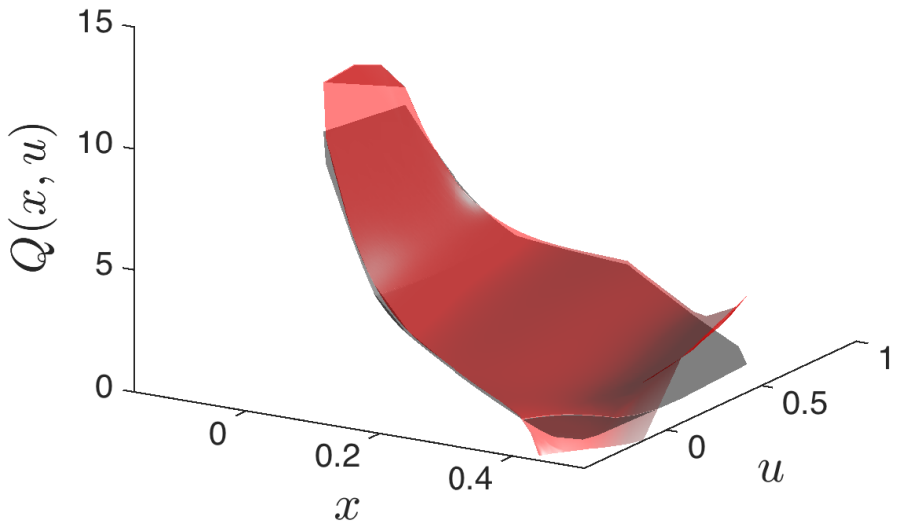
Form

$$\zeta_\theta = \sum_{\mathcal{D}} \delta_k \nabla_\theta Q_\theta(\mathbf{x}_k, \mathbf{u}_k)$$

and gradient

Newton step: $\theta \leftarrow \theta + \alpha \nabla_\theta \zeta_\theta^{-1} \zeta_\theta$

**# samples = 21544**



# LSTDQ - Efficient evaluation of $Q_\pi$

Recall that we want to solve

$$\mathbb{E}_{\tau_\pi^{\mathrm{d}}}\left[\delta \nabla_\theta Q_\theta\right] = 0$$

$$\delta_Q = L(\mathbf{x},\mathbf{u}) + \gamma Q_\theta(\mathbf{x}_+,\boldsymbol{\pi}_+) - Q_\theta(\mathbf{x},\mathbf{u})$$

for $\theta$

**Stochastic gradient**:

$$\hat{\theta} \leftarrow \hat{\theta} + \alpha \delta_Q \nabla_\theta Q_\theta$$

gets us there.

- $+$ trivial to code
- $+$ computationally cheap (for $\theta$ large)
- $-$ inefficient (many many iterations)

**Algorithm:** LSTDQ (prototype)

**Input**: Initial $Q$ parameters $\theta$

Collect $\mathcal{D} = \{(\mathbf{x}_k, \mathbf{u}_k, \mathbf{x}_{k+1}),\ k = 0, ..., N\}$

**while** $\|\zeta_\theta\| > \mathrm{Tol}$ **do**

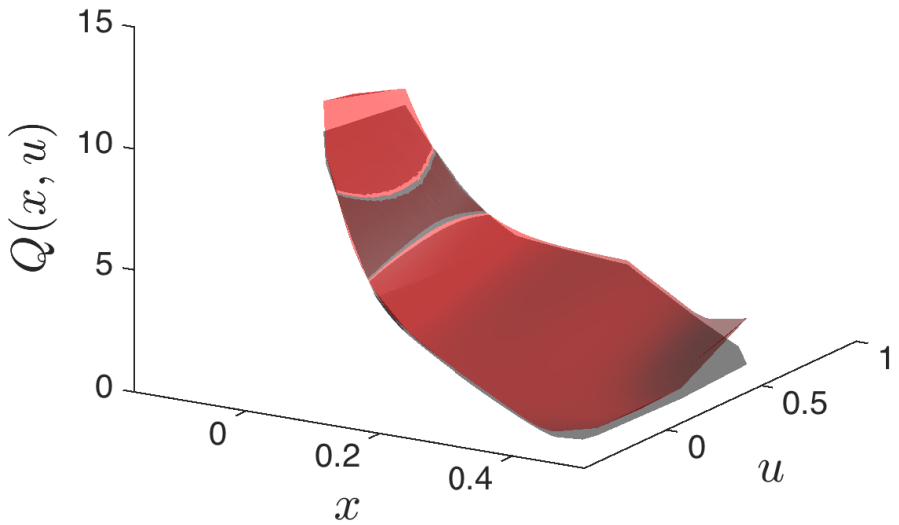
Form

$$\zeta_\theta = \sum_{\mathcal{D}} \delta_k \nabla_\theta Q_\theta(\mathbf{x}_k, \mathbf{u}_k)$$

and gradient

Newton step: $\theta \leftarrow \theta + \alpha \nabla_\theta \zeta_\theta^{-1} \zeta_\theta$

**# samples = 77426**



# LSTDQ - Efficient evaluation of $Q_\pi$

Recall that we want to solve

$$\mathbb{E}_{\tau_\pi^{\mathrm{d}}}\left[\delta \nabla_\theta Q_\theta\right] = 0$$

$$\delta_Q = L(\mathbf{x}, \mathbf{u}) + \gamma Q_\theta(\mathbf{x}_+, \boldsymbol{\pi}_+) - Q_\theta(\mathbf{x}, \mathbf{u})$$

for $\theta$

**Stochastic gradient**:

$$\hat{\theta} \leftarrow \hat{\theta} + \alpha \delta_Q \nabla_\theta Q_\theta$$

gets us there.

- $+$ trivial to code
- $+$ computationally cheap (for $\theta$ large)
- $-$ inefficient (many many iterations)

**Algorithm:** LSTDQ (prototype)

**Input**: Initial $Q$ parameters $\theta$

Collect $\mathcal{D} = \{(\mathbf{x}_k, \mathbf{u}_k, \mathbf{x}_{k+1}),\ k = 0, ..., N\}$

**while** $\|\zeta_\theta\| > \mathrm{Tol}$ **do**

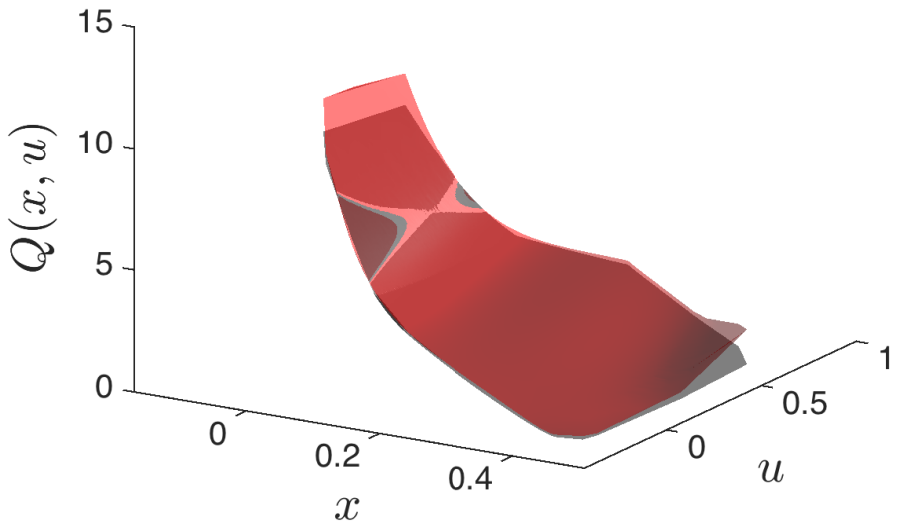
Form

$$\zeta_\theta = \sum_{\mathcal{D}} \delta_k \nabla_\theta Q_\theta(\mathbf{x}_k, \mathbf{u}_k)$$

and gradient

Newton step: $\theta \leftarrow \theta + \alpha \nabla_\theta \zeta_\theta^{-1} \zeta_\theta$

**# samples = 278255**



# LSTDQ - Efficient evaluation of $Q_\pi$

Recall that we want to solve

$$\mathbb{E}_{\tau_\pi^{\mathrm{d}}}\left[\delta \nabla_\theta Q_\theta\right] = 0$$

$$\delta_Q = L(\mathbf{x},\mathbf{u}) + \gamma Q_\theta(\mathbf{x}_+,\boldsymbol{\pi}_+) - Q_\theta(\mathbf{x},\mathbf{u})$$

for $\theta$

**Stochastic gradient**:

$$\hat{\theta} \leftarrow \hat{\theta} + \alpha \delta_Q \nabla_\theta Q_\theta$$

gets us there.

- $+$ trivial to code
- $+$ computationally cheap (for $\theta$ large)
- $-$ inefficient (many many iterations)

**Algorithm:** LSTDQ (prototype)

**Input**: Initial $Q$ parameters $\theta$

Collect $\mathcal{D} = \{(\mathbf{x}_k, \mathbf{u}_k, \mathbf{x}_{k+1}),\ k = 0, ..., N\}$

**while** $\|\zeta_\theta\| > \mathrm{Tol}$ **do**

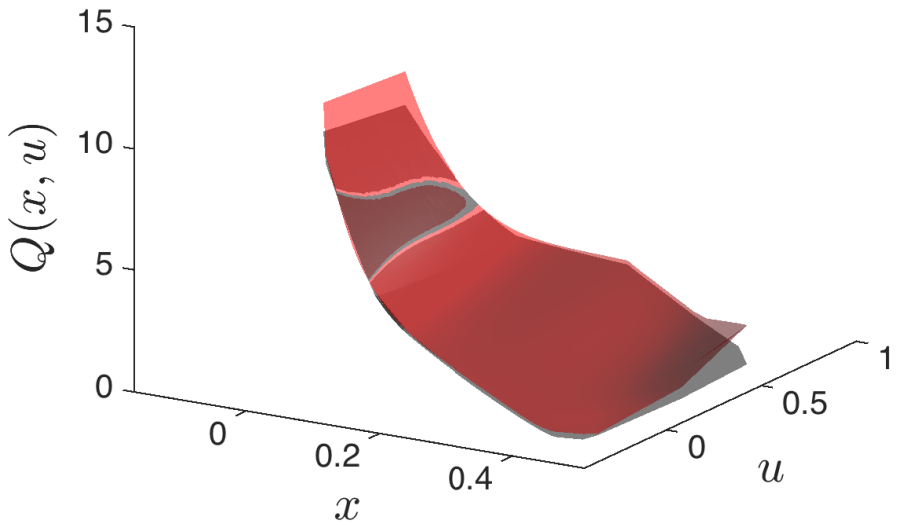
Form

$$\zeta_\theta = \sum_{\mathcal{D}} \delta_k \nabla_\theta Q_\theta(\mathbf{x}_k, \mathbf{u}_k)$$

and gradient

Newton step: $\theta \leftarrow \theta + \alpha \nabla_\theta \zeta_\theta^{-1} \zeta_\theta$

**# samples = 1000000**



# LSPI - Find the optimal policy

Recall

$$\delta_k = L(\mathbf{x}_k, \mathbf{u}_k) + \gamma Q_{\boldsymbol{\theta}}(\mathbf{x}_{k+1}, \boldsymbol{\pi}(\mathbf{x}_{k+1})) - Q_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k)$$

**Algorithm:** LSPI (prototype)

**Input**: Initial $Q$ parameters $\boldsymbol{\theta}$

**for** $i = 1 \rightarrow \infty$ **do**

- Set policy $\boldsymbol{\pi}(\mathbf{x}) = \arg\min_{\mathbf{u}} Q_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u})$
- Collect $\mathcal{D} = \{(\mathbf{x}_k, \mathbf{u}_k, \mathbf{x}_{k+1}),\ k = 1, ..., N\}$
- **while** $\|\zeta_{\boldsymbol{\theta}}\| > \mathrm{Tol}$ **do**
  - Form $$\zeta_{\boldsymbol{\theta}} = \sum_{\mathcal{D}} \delta_k \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k)$$ and gradient
  - Newton step: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \nabla_{\boldsymbol{\theta}} \zeta_{\boldsymbol{\theta}}^{-1} \zeta_{\boldsymbol{\theta}}$

**return** $Q$ parameters $\boldsymbol{\theta}$ for $\mathcal{D}$

1. Set policy $\boldsymbol{\pi}$ from $Q_{\boldsymbol{\theta}}$
2. Collect data ($N$ transitions)
3. Update $Q_{\boldsymbol{\theta}}$ using LSTDQ
4. Back to 1

# LSPI - Find the optimal policy

Recall

$$\delta_k = L(\mathbf{x}_k, \mathbf{u}_k) + \gamma Q_{\boldsymbol{\theta}}(\mathbf{x}_{k+1}, \boldsymbol{\pi}(\mathbf{x}_{k+1})) - Q_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k)$$

**Algorithm:** LSPI on-the-fly

**Input**: Initial $Q$ parameters $\boldsymbol{\theta}$

**for** $k = 1 \to \infty$ **do**

Set $\mathbf{u}_k = \arg\min_{\mathbf{u}} Q_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u})$

Observe transition $(\mathbf{x}_k, \mathbf{u}_k, \mathbf{x}_{k+1})$, build

$$\delta_k = L(\mathbf{x}_k, \mathbf{u}_k) + \gamma \min_{\mathbf{u}} Q_{\boldsymbol{\theta}}(\mathbf{x}_{k+1}, \mathbf{u}) - Q_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k)$$

Form (for some forgetting factor $\eta < 1$)

$$\zeta_{\boldsymbol{\theta}} = \sum_{i=0}^{k} \eta^i \delta_i \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{u}_i)$$

and gradient

Newton step: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \nabla_{\boldsymbol{\theta}} \zeta_{\boldsymbol{\theta}}^{-1} \zeta_{\boldsymbol{\theta}}$

1. Set policy $\boldsymbol{\pi}$ from $Q_{\boldsymbol{\theta}}$
2. Observe 1 transition, add to data
3. Update $Q_{\boldsymbol{\theta}}$ using LSTDQ (possibly discounted)
4. Back to 1

# LSPI - Find the optimal policy

Recall

$$\delta_k = L(\mathbf{x}_k, \mathbf{u}_k) + \gamma Q_{\boldsymbol{\theta}}(\mathbf{x}_{k+1}, \boldsymbol{\pi}(\mathbf{x}_{k+1})) - Q_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k)$$

**Algorithm:** LSPI on-the-fly

**Input**: Initial $Q$ parameters $\boldsymbol{\theta}$

**for** $k = 1 \to \infty$ **do**

Set $\mathbf{u}_k = \arg\min_{\mathbf{u}} Q_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u})$

Observe transition $(\mathbf{x}_k, \mathbf{u}_k, \mathbf{x}_{k+1})$, build

$\delta_k =$

$$L(\mathbf{x}_k, \mathbf{u}_k) + \gamma \min_{\mathbf{u}} Q_{\boldsymbol{\theta}}(\mathbf{x}_{k+1}, \mathbf{u}) - Q_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k)$$

Form (for some forgetting factor $\eta < 1$)

$$\zeta_{\boldsymbol{\theta}} = \sum_{i=0}^{k} \eta^i \delta_i \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{u}_i)$$

and gradient

Newton step: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \nabla_{\boldsymbol{\theta}} \zeta_{\boldsymbol{\theta}}^{-1} \zeta_{\boldsymbol{\theta}}$

1. Set policy $\boldsymbol{\pi}$ from $Q_{\boldsymbol{\theta}}$
2. Observe 1 transition, add to data
3. Update $Q_{\boldsymbol{\theta}}$ using LSTDQ (possibly discounted)
4. Back to 1

**Old samples should be progressively ignored, as they are generated from an old MDP distribution $\tau_{\boldsymbol{\pi}}$ (policy has changed!)**

## Experience replay

Consider the (possibly huge) data set (state transitions):

$$\mathcal{D} = \{(\mathbf{x}_1, \mathbf{u}_1 \rightarrow \mathbf{x}_2), \ldots (\mathbf{x}_N, \mathbf{u}_N \rightarrow \mathbf{x}_{N+1})\}$$

Full LSTD update

$$\zeta_{\boldsymbol{\theta}} = \sum_{\mathcal{D}} \delta_k \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k)$$

with

$$\delta_k = L(\mathbf{x}_k, \mathbf{u}_k) + \gamma Q_{\boldsymbol{\theta}}(\mathbf{x}_{k+1}, \boldsymbol{\pi}(\mathbf{x}_{k+1})) - Q_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k)$$

## Experience replay

Consider the (possibly huge) data set (state transitions):

$$\mathcal{D} = \{(\mathbf{x}_1, \mathbf{u}_1 \rightarrow \mathbf{x}_2), \ldots (\mathbf{x}_N, \mathbf{u}_N \rightarrow \mathbf{x}_{N+1})\}$$

Full LSTD update

$$\zeta_{\boldsymbol{\theta}} = \sum_{\mathcal{D}} \delta_k \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k)$$

with

$$\delta_k = L(\mathbf{x}_k, \mathbf{u}_k) + \gamma Q_{\boldsymbol{\theta}}(\mathbf{x}_{k+1}, \boldsymbol{\pi}(\mathbf{x}_{k+1})) - Q_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k)$$

May not be good:

- we need to re-evaluate $\delta_k$, $Q_{\boldsymbol{\theta}}(\mathbf{x}_{k+1}, \boldsymbol{\pi}(\mathbf{x}_{k+1}))$ and $Q_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k)$ for all $k$ whenever we change $\boldsymbol{\theta}$, becomes very expensive for $N$ large

## Experience replay

Consider the (possibly huge) data set (state transitions):

$$\mathcal{D} = \{(\mathbf{x}_1, \mathbf{u}_1 \to \mathbf{x}_2), \ldots (\mathbf{x}_N, \mathbf{u}_N \to \mathbf{x}_{N+1})\}$$

Full LSTD update

$$\zeta_{\boldsymbol{\theta}} = \sum_{\mathcal{D}} \delta_k \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k)$$

with

$$\delta_k = L(\mathbf{x}_k, \mathbf{u}_k) + \gamma Q_{\boldsymbol{\theta}}(\mathbf{x}_{k+1}, \boldsymbol{\pi}(\mathbf{x}_{k+1})) - Q_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k)$$

May not be good:

- we need to re-evaluate $\delta_k$, $Q_{\boldsymbol{\theta}}(\mathbf{x}_{k+1}, \boldsymbol{\pi}(\mathbf{x}_{k+1}))$ and $Q_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k)$ for all $k$ whenever we change $\boldsymbol{\theta}$, becomes very expensive for $N$ large

**Experience replay**: form $\zeta_{\boldsymbol{\theta}}$ on a sub-data set $\mathcal{D}_{\mathrm{S}} \subset \mathcal{D}$ of limited cardinality

- how to select the samples $\mathcal{D}_{\mathrm{S}}$? In principle random, but should match MDP distribution under new policy

## Experience replay

Consider the (possibly huge) data set (state transitions):

$$\mathcal{D} = \{(\mathbf{x}_1, \mathbf{u}_1 \to \mathbf{x}_2), \ldots (\mathbf{x}_N, \mathbf{u}_N \to \mathbf{x}_{N+1})\}$$

Full LSTD update

$$\zeta_{\boldsymbol{\theta}} = \sum_{\mathcal{D}} \delta_k \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k)$$

with

$$\delta_k = L(\mathbf{x}_k, \mathbf{u}_k) + \gamma Q_{\boldsymbol{\theta}}(\mathbf{x}_{k+1}, \boldsymbol{\pi}(\mathbf{x}_{k+1})) - Q_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k)$$

May not be good:

- we need to re-evaluate $\delta_k$, $Q_{\boldsymbol{\theta}}(\mathbf{x}_{k+1}, \boldsymbol{\pi}(\mathbf{x}_{k+1}))$ and $Q_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{u}_k)$ for all $k$ whenever we change $\boldsymbol{\theta}$, becomes very expensive for $N$ large

**Experience replay**: form $\zeta_{\boldsymbol{\theta}}$ on a sub-data set $\mathcal{D}_{\mathrm{S}} \subset \mathcal{D}$ of limited cardinality

- how to select the samples $\mathcal{D}_{\mathrm{S}}$? In principle random, but should match MDP distribution under new policy
- many tricks to make a "smart" subset selection

# Outline

# Is LSTD(Q) right?

**LSTD**

$$\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[\delta \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}\right] = 0$$
$$\delta = L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma V_{\boldsymbol{\theta}}(\mathbf{x}_+) - V_{\boldsymbol{\theta}}(\mathbf{x})$$

**LSTDQ**

$$\mathbb{E}_{\tau^{\mathrm{d}}_{\boldsymbol{\pi}}}\left[\delta_Q \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}\right] = 0$$
$$\delta_Q = L(\mathbf{x}, \mathbf{u}) + \gamma Q_{\boldsymbol{\theta}}(\mathbf{x}_+, \boldsymbol{\pi}(\mathbf{x})) - Q_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u})$$

**Remark**: if there is a $\boldsymbol{\theta}_\star$ such that $V_{\boldsymbol{\theta}_\star} = V_{\boldsymbol{\pi}}$ for all $\mathbf{x}$ then

# Is LSTD(Q) right?

**LSTD**

$$\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[\delta \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}\right] = 0$$
$$\delta = L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma V_{\boldsymbol{\theta}}(\mathbf{x}_+) - V_{\boldsymbol{\theta}}(\mathbf{x})$$

**LSTDQ**

$$\mathbb{E}_{\tau^{\mathrm{d}}_{\boldsymbol{\pi}}}\left[\delta_Q \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}\right] = 0$$
$$\delta_Q = L(\mathbf{x}, \mathbf{u}) + \gamma Q_{\boldsymbol{\theta}}(\mathbf{x}_+, \boldsymbol{\pi}(\mathbf{x})) - Q_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u})$$

**Remark**: if there is a $\boldsymbol{\theta}_\star$ such that $V_{\boldsymbol{\theta}_\star} = V_{\boldsymbol{\pi}}$ for all $\mathbf{x}$ then

$$V_{\boldsymbol{\theta}_\star}(\mathbf{x}) = L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma \mathbb{E}\left[V_{\boldsymbol{\theta}_\star}(\mathbf{x}_+) \,|\, \mathbf{x}\right] \quad \text{holds} \quad \forall\, \mathbf{x}$$

# Is LSTD(Q) right?

**LSTD**

$$\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[\delta \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}\right] = 0$$
$$\delta = L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma V_{\boldsymbol{\theta}}(\mathbf{x}_+) - V_{\boldsymbol{\theta}}(\mathbf{x})$$

**LSTDQ**

$$\mathbb{E}_{\tau^{\mathrm{d}}_{\boldsymbol{\pi}}}\left[\delta_Q \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}\right] = 0$$
$$\delta_Q = L(\mathbf{x}, \mathbf{u}) + \gamma Q_{\boldsymbol{\theta}}(\mathbf{x}_+, \boldsymbol{\pi}(\mathbf{x})) - Q_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u})$$

**Remark**: if there is a $\boldsymbol{\theta}_\star$ such that $V_{\boldsymbol{\theta}_\star} = V_{\boldsymbol{\pi}}$ for all $\mathbf{x}$ then

$$L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma \mathbb{E}\left[V_{\boldsymbol{\theta}_\star}(\mathbf{x}_+) \,|\, \mathbf{x}\right] - V_{\boldsymbol{\theta}_\star}(\mathbf{x}) = 0 \quad \text{holds} \quad \forall\, \mathbf{x}$$

# Is LSTD(Q) right?

**LSTD**

$$\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[\delta \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}\right] = 0$$

$$\delta = L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma V_{\boldsymbol{\theta}}(\mathbf{x}_+) - V_{\boldsymbol{\theta}}(\mathbf{x})$$

**LSTDQ**

$$\mathbb{E}_{\tau^{\mathrm{d}}_{\boldsymbol{\pi}}}\left[\delta_Q \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}\right] = 0$$

$$\delta_Q = L(\mathbf{x}, \mathbf{u}) + \gamma Q_{\boldsymbol{\theta}}(\mathbf{x}_+, \boldsymbol{\pi}(\mathbf{x})) - Q_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u})$$

**Remark**: if there is a $\boldsymbol{\theta}_\star$ such that $V_{\boldsymbol{\theta}_\star} = V_{\boldsymbol{\pi}}$ for all $\mathbf{x}$ then

$$\mathbb{E}\left[L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma V_{\boldsymbol{\theta}_\star}(\mathbf{x}_+) - V_{\boldsymbol{\theta}_\star}(\mathbf{x}) \,|\, \mathbf{x}\right] = 0 \quad \text{holds} \quad \forall\, \mathbf{x}$$

# Is LSTD(Q) right?

**LSTD**

$$\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[\delta \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}\right] = 0$$
$$\delta = L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma V_{\boldsymbol{\theta}}(\mathbf{x}_+) - V_{\boldsymbol{\theta}}(\mathbf{x})$$

**LSTDQ**

$$\mathbb{E}_{\tau^{\mathrm{d}}_{\boldsymbol{\pi}}}\left[\delta_Q \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}\right] = 0$$
$$\delta_Q = L(\mathbf{x}, \mathbf{u}) + \gamma Q_{\boldsymbol{\theta}}(\mathbf{x}_+, \boldsymbol{\pi}(\mathbf{x})) - Q_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u})$$

**Remark**: if there is a $\boldsymbol{\theta}_\star$ such that $V_{\boldsymbol{\theta}_\star} = V_{\boldsymbol{\pi}}$ for all $\mathbf{x}$ then

$$\bar{\delta}(\mathbf{x}) := \mathbb{E}\left[L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma V_{\boldsymbol{\theta}_\star}(\mathbf{x}_+) - V_{\boldsymbol{\theta}_\star}(\mathbf{x}) \,|\, \mathbf{x}\right] = 0 \quad \text{holds} \quad \forall\, \mathbf{x}$$

# Is LSTD(Q) right?

**LSTD**

$$\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[\delta \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}\right] = 0$$
$$\delta = L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma V_{\boldsymbol{\theta}}(\mathbf{x}_+) - V_{\boldsymbol{\theta}}(\mathbf{x})$$

**LSTDQ**

$$\mathbb{E}_{\tau^{\mathrm{d}}_{\boldsymbol{\pi}}}\left[\delta_Q \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}\right] = 0$$
$$\delta_Q = L(\mathbf{x}, \mathbf{u}) + \gamma Q_{\boldsymbol{\theta}}(\mathbf{x}_+, \boldsymbol{\pi}(\mathbf{x})) - Q_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u})$$

**Remark**: if there is a $\boldsymbol{\theta}_\star$ such that $V_{\boldsymbol{\theta}_\star} = V_{\boldsymbol{\pi}}$ for all $\mathbf{x}$ then

$$\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[\bar{\delta} \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}_\star}\right] = 0 \quad \text{with} \quad \bar{\delta}(\mathbf{x}) = \mathbb{E}\left[\delta(\mathbf{x}, \mathbf{x}_+) \,|\, \mathbf{x}\right]$$

*how does this relate to LSTD?*

# Is LSTD(Q) right?

**LSTD**

$$\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[\delta \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}\right] = 0$$
$$\delta = L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma V_{\boldsymbol{\theta}}(\mathbf{x}_+) - V_{\boldsymbol{\theta}}(\mathbf{x})$$

**LSTDQ**

$$\mathbb{E}_{\tau^{\mathrm{d}}_{\boldsymbol{\pi}}}\left[\delta_Q \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}\right] = 0$$
$$\delta_Q = L(\mathbf{x}, \mathbf{u}) + \gamma Q_{\boldsymbol{\theta}}(\mathbf{x}_+, \boldsymbol{\pi}(\mathbf{x})) - Q_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u})$$

**Remark**: if there is a $\boldsymbol{\theta}_\star$ such that $V_{\boldsymbol{\theta}_\star} = V_{\boldsymbol{\pi}}$ for all $\mathbf{x}$ then

$$\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[\bar{\delta} \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}_\star}\right] = 0 \quad \text{with} \quad \bar{\delta}(\mathbf{x}) = \mathbb{E}\left[\delta(\mathbf{x}, \mathbf{x}_+) \,|\, \mathbf{x}\right]$$

*how does this relate to LSTD?*

If $\mathbb{E}[\zeta(\mathbf{x}, \mathbf{x}_+) \,|\, \mathbf{x}] = 0$ for all $\mathbf{x}$, then $\mathbb{E}_{\tau_{\boldsymbol{\pi}}}[\zeta(\mathbf{x}, \mathbf{x}_+)\xi(\mathbf{x})] = 0$ for any function $\xi$

*Note:* $\mathbb{E}[. \,|\, \mathbf{x}]$ *is an expectation over state transitions* $\mathbf{x}, \boldsymbol{\pi}(\mathbf{x}) \to \mathbf{x}_+$, *i.e. it matches* $\tau_{\boldsymbol{\pi}}$

# Is LSTD(Q) right?

**LSTD**

$$\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[\delta \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}\right] = 0$$
$$\delta = L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma V_{\boldsymbol{\theta}}(\mathbf{x}_+) - V_{\boldsymbol{\theta}}(\mathbf{x})$$

**LSTDQ**

$$\mathbb{E}_{\tau^{\mathrm{d}}_{\boldsymbol{\pi}}}\left[\delta_Q \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}\right] = 0$$
$$\delta_Q = L(\mathbf{x}, \mathbf{u}) + \gamma Q_{\boldsymbol{\theta}}(\mathbf{x}_+, \boldsymbol{\pi}(\mathbf{x})) - Q_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u})$$

**Remark**: if there is a $\boldsymbol{\theta}_\star$ such that $V_{\boldsymbol{\theta}_\star} = V_{\boldsymbol{\pi}}$ for all $\mathbf{x}$ then

$$\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[\bar{\delta} \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}_\star}\right] = 0 \quad \text{with} \quad \bar{\delta}(\mathbf{x}) = \mathbb{E}\left[\delta(\mathbf{x}, \mathbf{x}_+) \,|\, \mathbf{x}\right]$$

*how does this relate to LSTD?*

If $\mathbb{E}[\zeta(\mathbf{x}, \mathbf{x}_+) \,|\, \mathbf{x}] = 0$ for all $\mathbf{x}$, then $\mathbb{E}_{\tau_{\boldsymbol{\pi}}}[\zeta(\mathbf{x}, \mathbf{x}_+)\xi(\mathbf{x})] = 0$ for any function $\xi$

*Note:* $\mathbb{E}[. \,|\, \mathbf{x}]$ *is an expectation over state transitions* $\mathbf{x}, \boldsymbol{\pi}(\mathbf{x}) \to \mathbf{x}_+$, *i.e. it matches* $\tau_{\boldsymbol{\pi}}$

Consequence: $\bar{\delta}(\mathbf{x}) = 0$ for all $\mathbf{x}$ implies $\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[\delta \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}_\star}\right] = 0$

**Hence LSTD holds if $V_{\boldsymbol{\theta}} = V_{\boldsymbol{\pi}}$**

**Similarly LSTDQ holds if $Q_{\boldsymbol{\theta}} = Q_{\boldsymbol{\pi}}$**

# What does not hold?

**LSTD**

$$\mathbb{E}_{\tau_{\boldsymbol{\pi}}}\left[\delta \nabla_{\boldsymbol{\theta}} V_{\boldsymbol{\theta}}\right] = 0$$
$$\delta = L(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x})) + \gamma V_{\boldsymbol{\theta}}(\mathbf{x}_+) - V_{\boldsymbol{\theta}}(\mathbf{x})$$

**LSTDQ**

$$\mathbb{E}_{\tau^{\mathrm{d}}_{\boldsymbol{\pi}}}\left[\delta \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}\right] = 0$$
$$\delta_Q = L(\mathbf{x}, \mathbf{u}) + \gamma Q_{\boldsymbol{\theta}}(\mathbf{x}_+, \boldsymbol{\pi}(\mathbf{x})) - Q_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{u})$$

**LSTD and LSTDQ do not necessarily deliver**

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{2}\mathbb{E}_{\tau^{\mathrm{d}}_{\boldsymbol{\pi}}}\left[(V_{\boldsymbol{\pi}} - V_{\boldsymbol{\theta}})^2\right] \qquad \hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{2}\mathbb{E}_{\tau^{\mathrm{d}}_{\boldsymbol{\pi}}}\left[(Q_{\boldsymbol{\pi}} - Q_{\boldsymbol{\theta}})^2\right]$$

if there is no $\boldsymbol{\theta}$ such that $V_{\boldsymbol{\pi}} = V_{\boldsymbol{\theta}}$ holds

**Bottom line: a "rich" parametrization of the value functions / policy is important!**

*MC-like targets (deeper backup) alleviate this problem...*