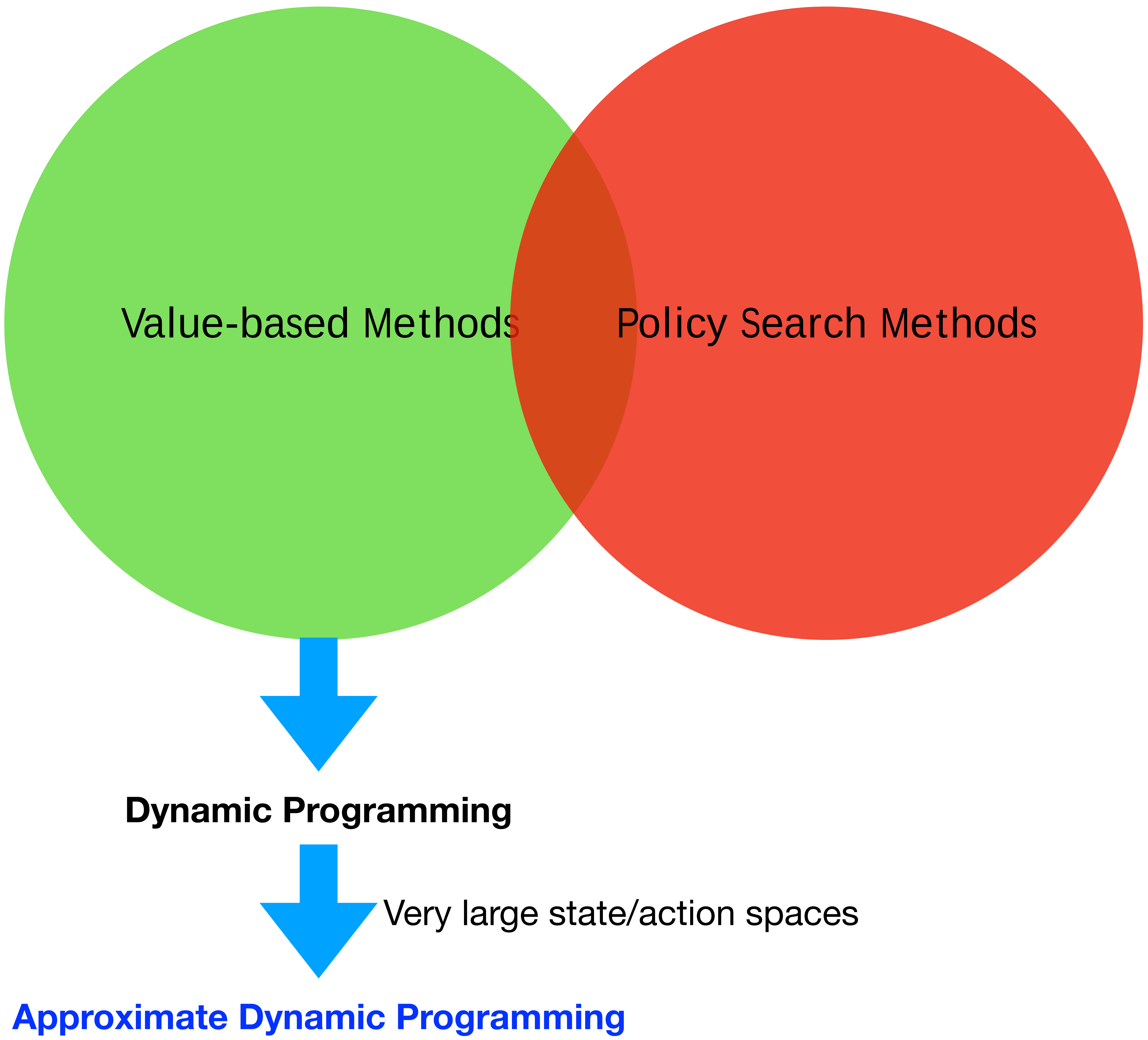
VECTOR INSTITUTE | INSTITUT VECTEUR

Approximate Dynamic Programming and Batch Reinforcement Learning

Amir-massoud Farahmand
http://academic.SoloGen.net
@sologen

CIFAR Deep Learning and Reinforcement Learning Summer School
July 25th-August 3rd, 2018

Version: 1.0


Value-based Methods
Policy Search Methods
Dynamic Programming
Very large state/action spaces
Approximate Dynamic Programming

Reinforcement Learning (RL)

An agent

observes the world

takes an action and its states changes

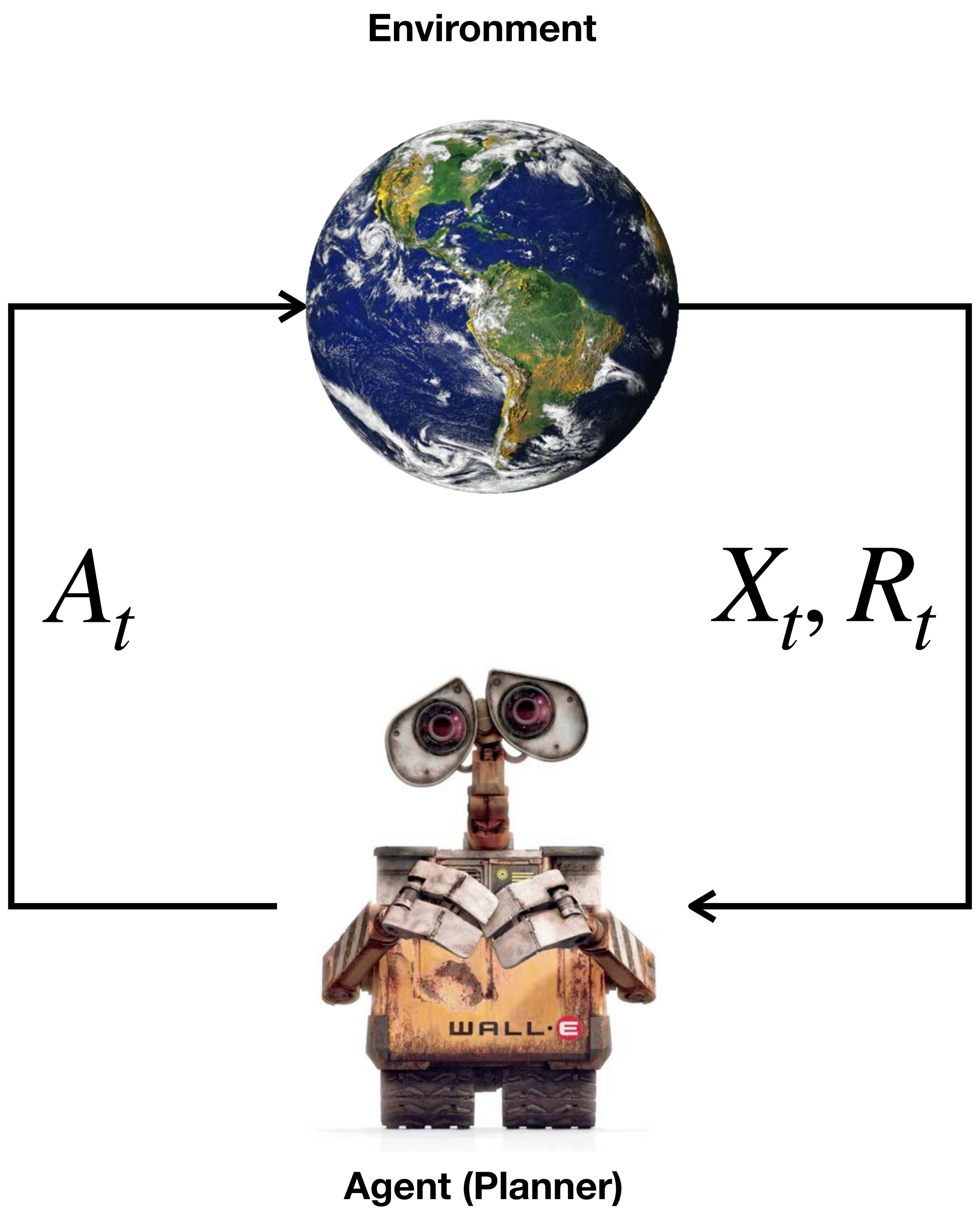
with the goal of achieving long-term rewards.

Reinforcement Learning Problem: An agent continually interacts with the environment. How should it choose its actions so that its long-term rewards are maximized?

Also might be called:

- Adaptive Controller for Stochastic Nonlinear Dynamical Systems
- Adaptive Situated Agent Design

RL Agent



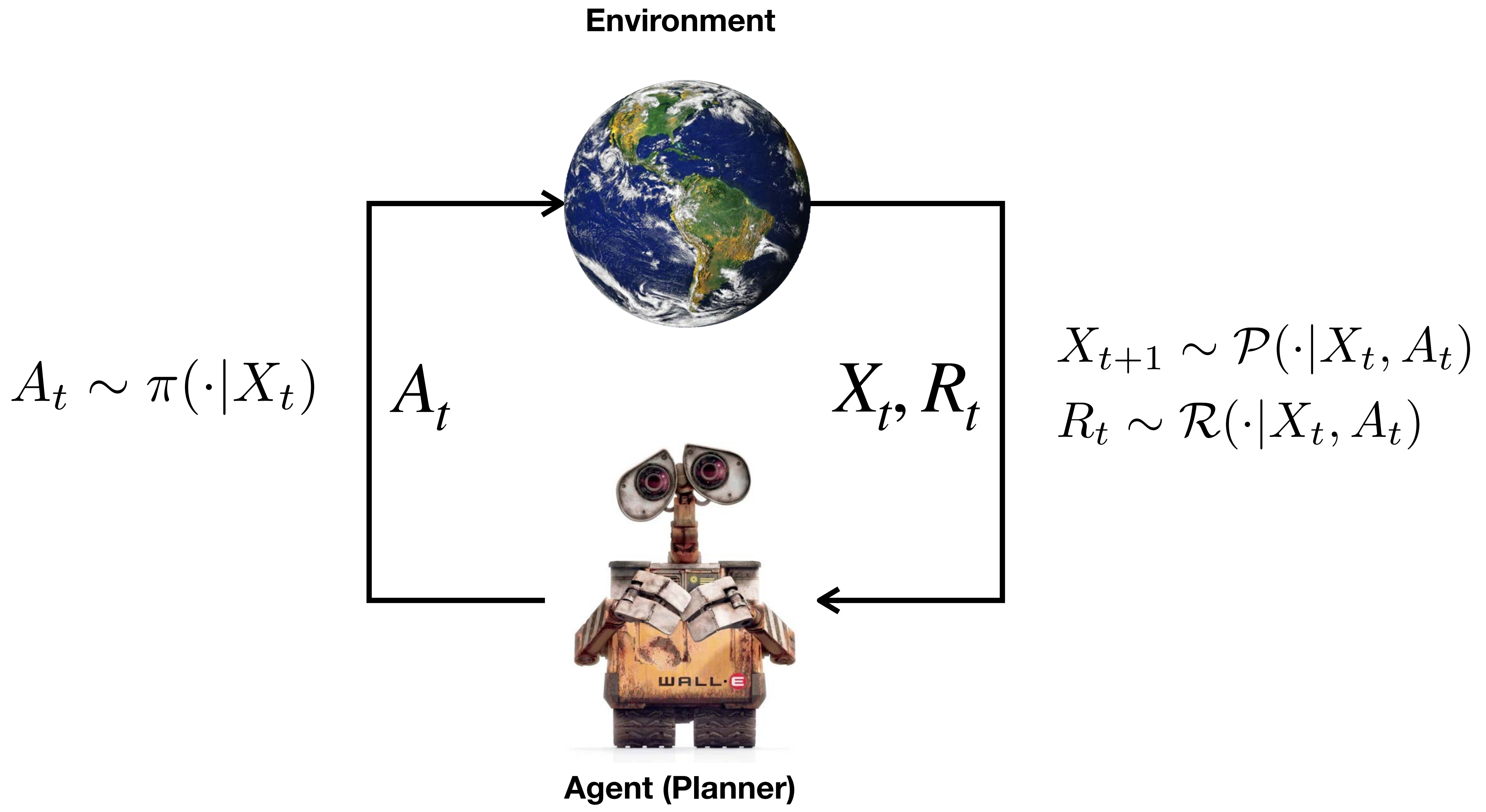
$$A_t \sim \pi(\cdot|X_t)$$

$$X_{t+1} \sim \mathcal{P}(\cdot|X_t, A_t)$$
$$R_t \sim \mathcal{R}(\cdot|X_t, A_t)$$

Markov Decision Process (MDP)

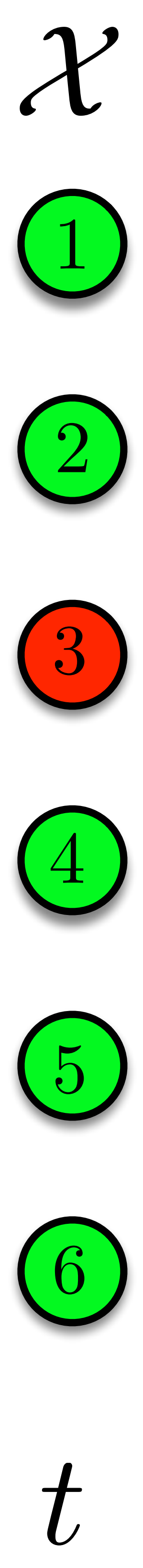
Discounted finite-action MDP $(\mathcal{X}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$:

- $\mathcal{X}$: State space
- $\mathcal{A} = \{a_1, \ldots, a_{|\mathcal{A}|}\}$: Action space (finite)
- $\mathcal{P} : \mathcal{X} \times \mathcal{A} \to \mathcal{M}(\mathcal{X})$: Transition probability kernel
- $\mathcal{R} : \mathcal{X} \times \mathcal{A} \to \mathcal{M}(\mathbb{R})$: Immediate reward distribution
- γ: Discount factor $(0 \leq \gamma < 1)$

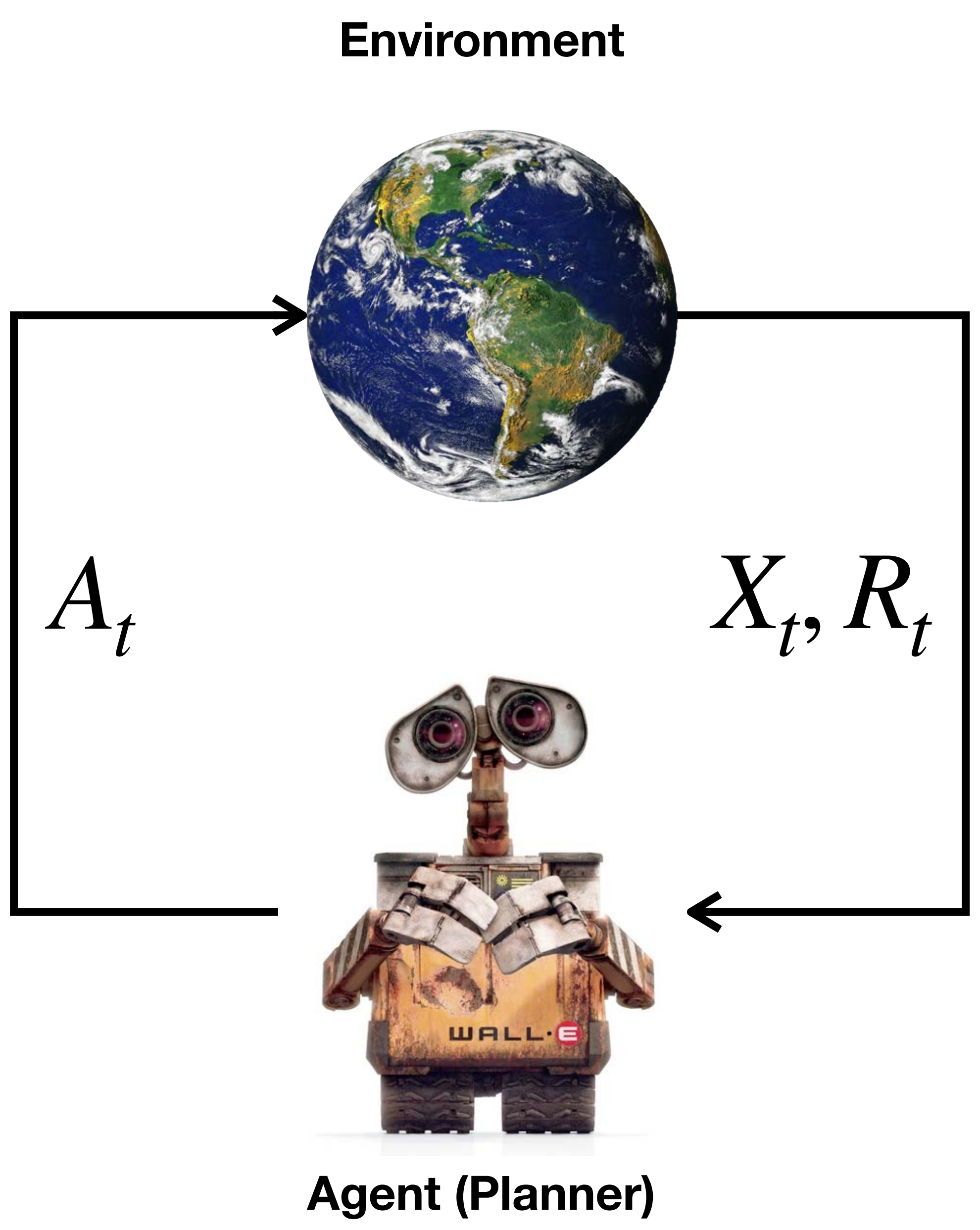


Other MDP models exist too: average reward, episodic, etc.

Markov Decision Process (MDP)



$A_t \sim \pi(\cdot|X_t)$

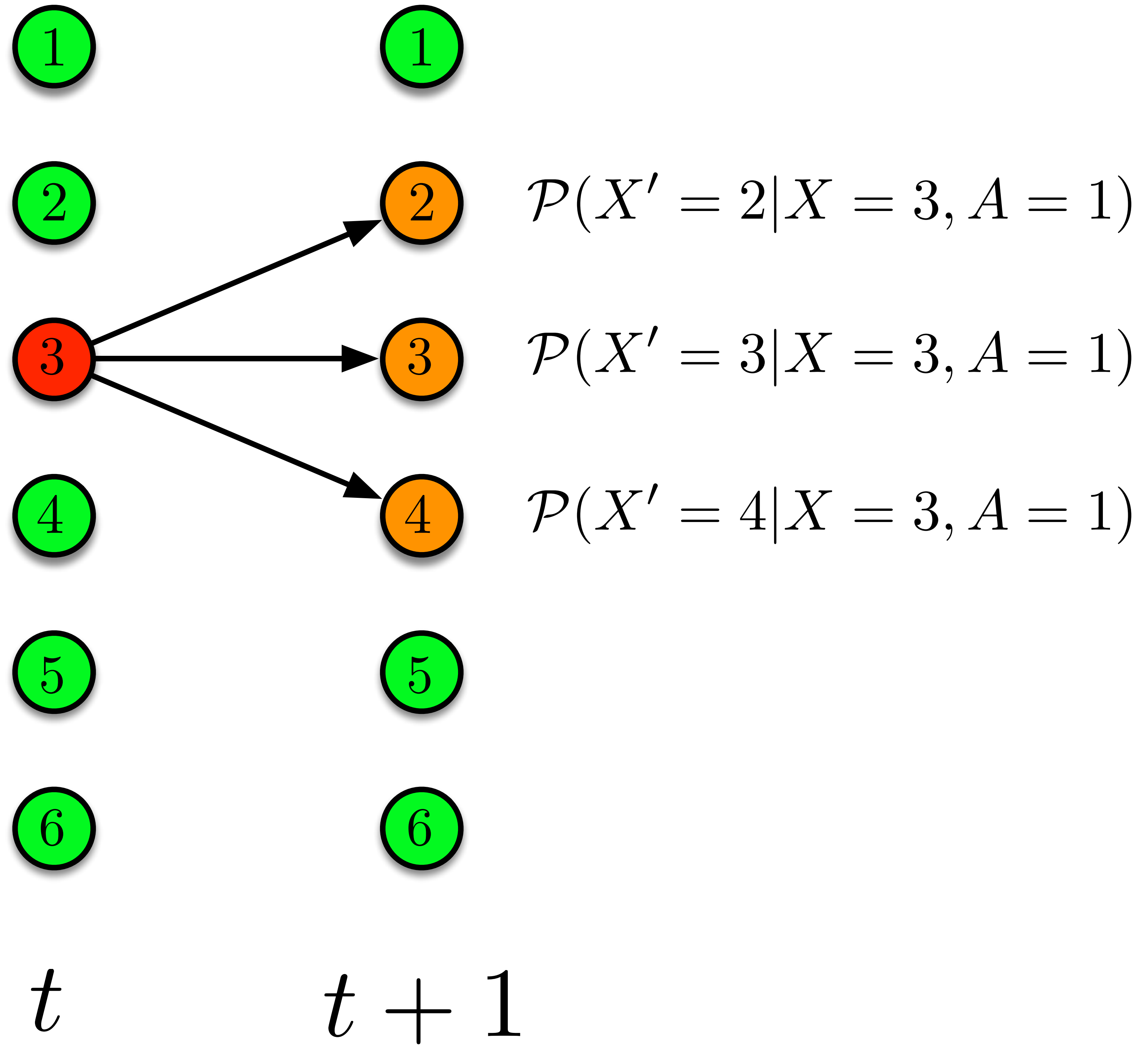


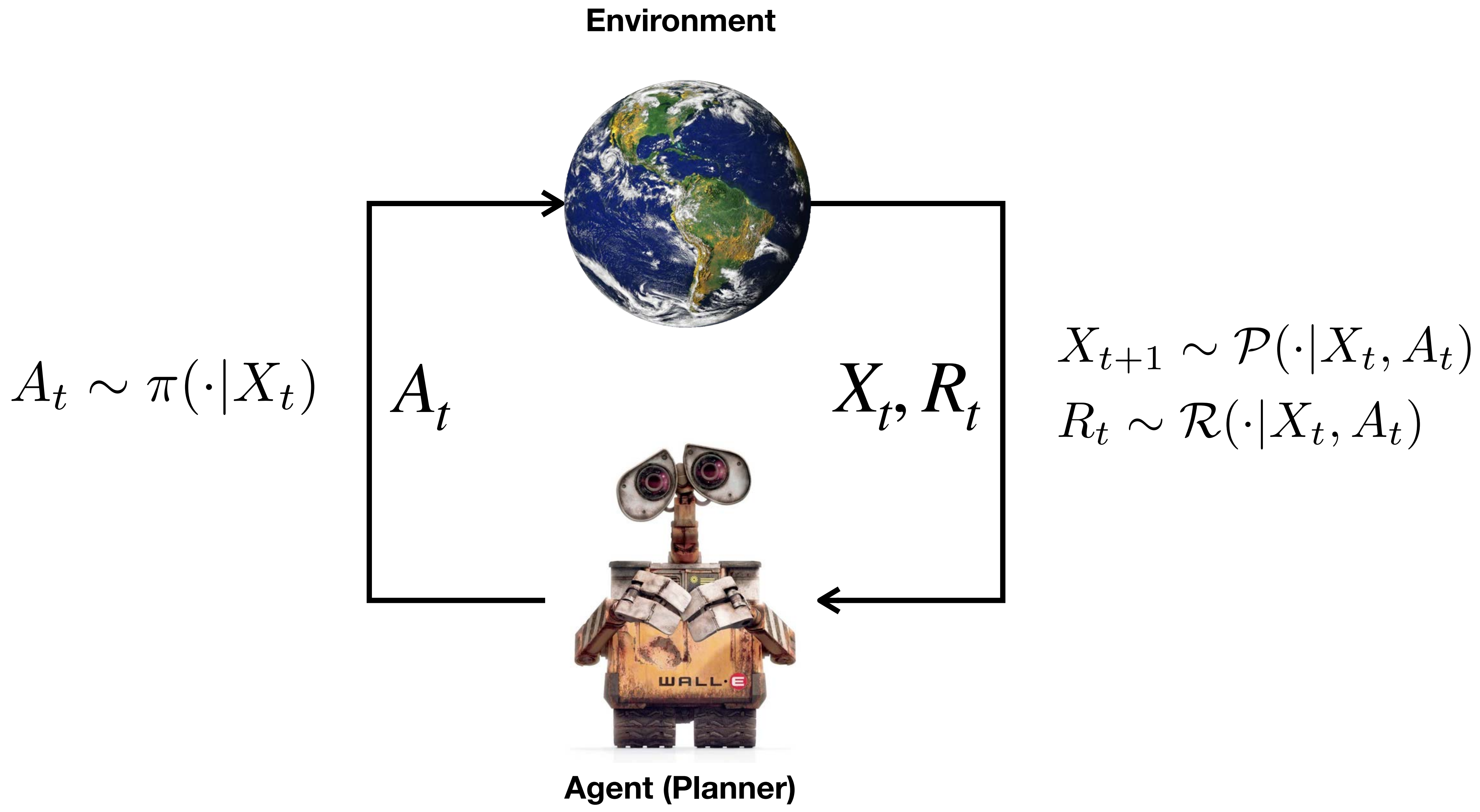
$X_{t+1} \sim \mathcal{P}(\cdot|X_t, A_t)$
$R_t \sim \mathcal{R}(\cdot|X_t, A_t)$

Discounted finite-action MDP $(\mathcal{X}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$:

- $\mathcal{X}$: State space
- $\mathcal{A} = \{a_1, \ldots, a_{|\mathcal{A}|}\}$: Action space (finite)
- $\mathcal{P} : \mathcal{X} \times \mathcal{A} \to \mathcal{M}(\mathcal{X})$: Transition probability kernel
- $\mathcal{R} : \mathcal{X} \times \mathcal{A} \to \mathcal{M}(\mathbb{R})$: Immediate reward distribution
- γ: Discount factor $(0 \leq \gamma < 1)$

Markov Decision Process (MDP)

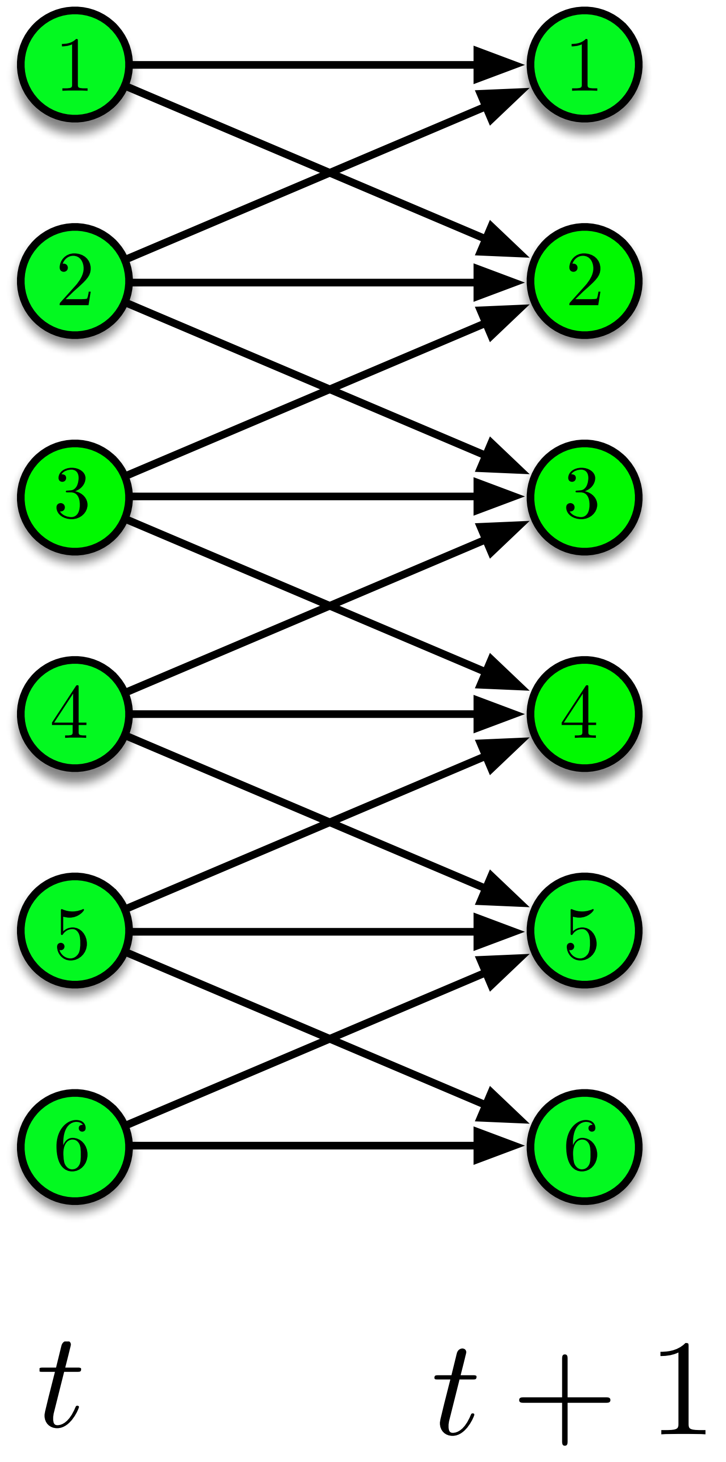


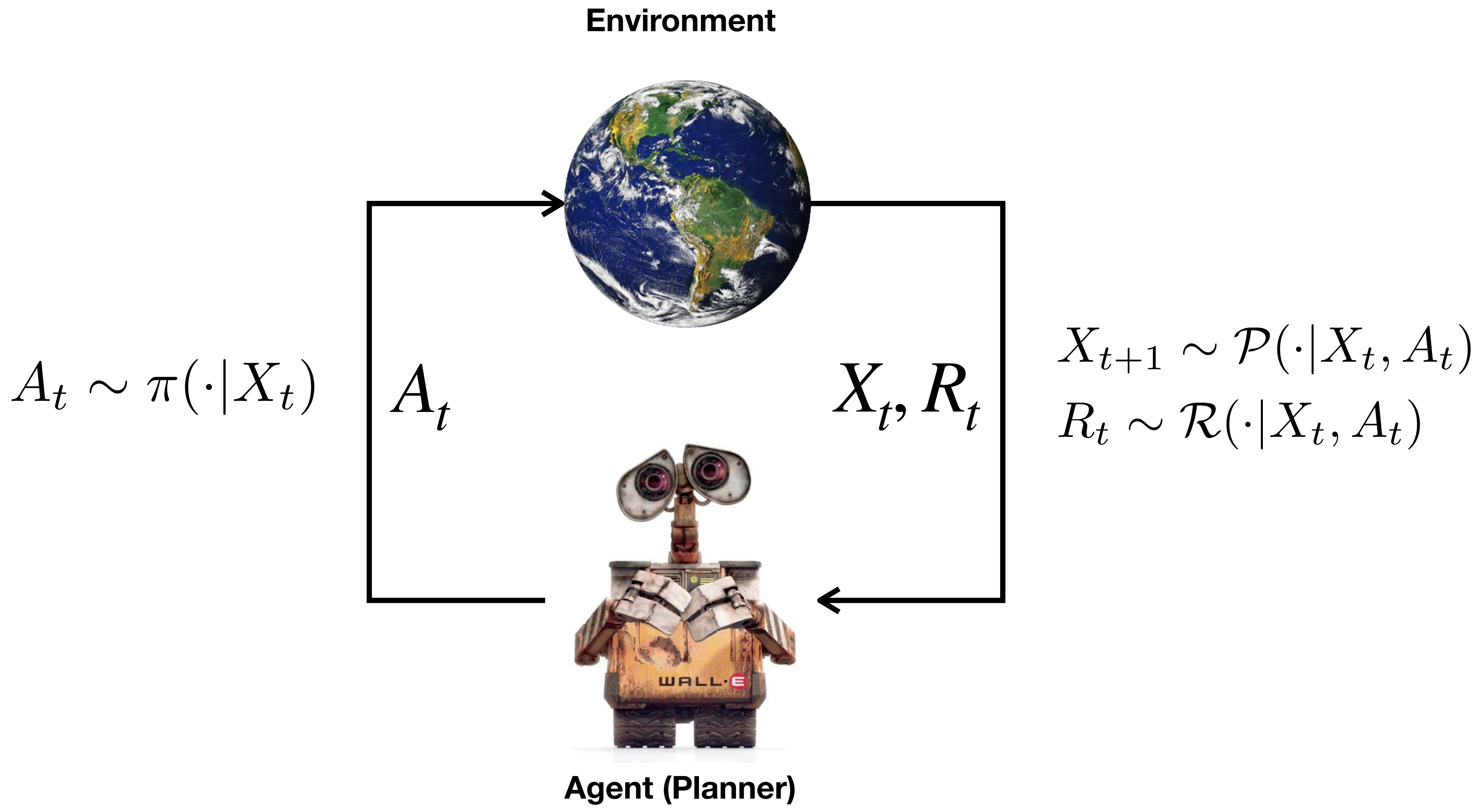


Discounted finite-action MDP $(\mathcal{X}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$:

- $\mathcal{X}$: State space
- $\mathcal{A} = \{a_1, \dots, a_{|\mathcal{A}|}\}$: Action space (finite)
- $\mathcal{P} : \mathcal{X} \times \mathcal{A} \to \mathcal{M}(\mathcal{X})$: Transition probability kernel
- $\mathcal{R} : \mathcal{X} \times \mathcal{A} \to \mathcal{M}(\mathbb{R})$: Immediate reward distribution
- γ: Discount factor $(0 \leq \gamma < 1)$

Markov Decision Process (MDP)

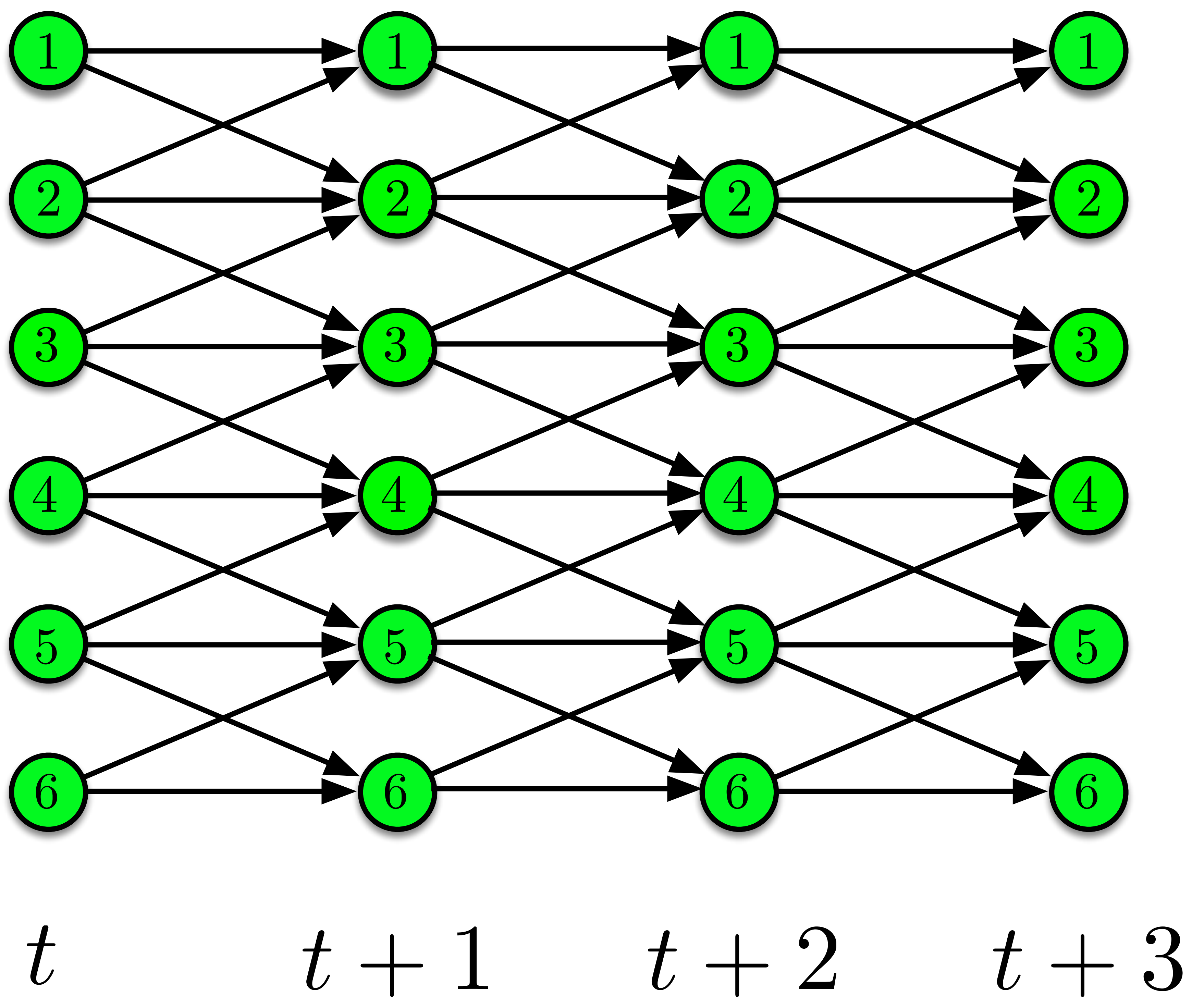


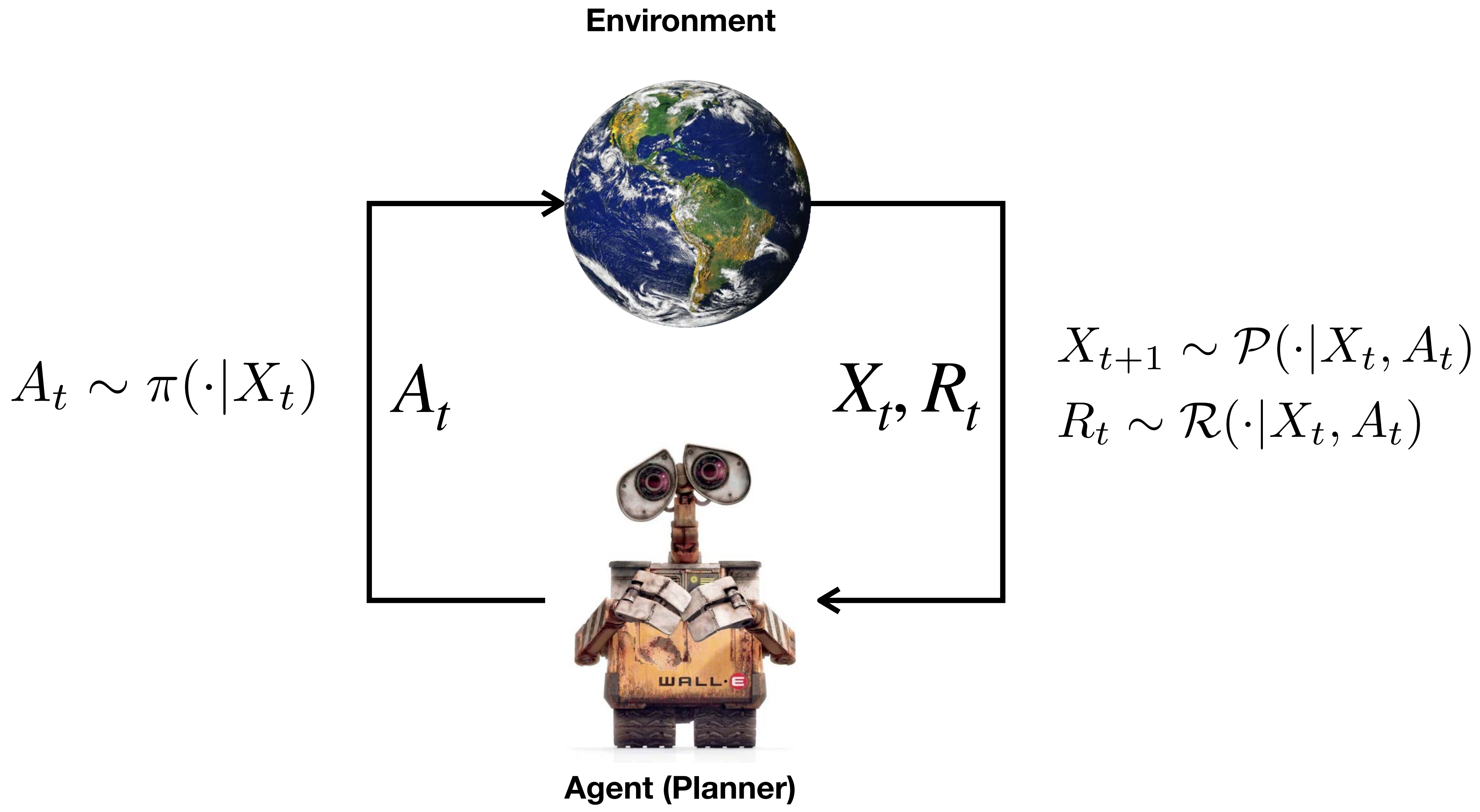


Discounted finite-action MDP $(\mathcal{X}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$:

- $\mathcal{X}$: State space
- $\mathcal{A} = \{a_1, \ldots, a_{|\mathcal{A}|}\}$: Action space (finite)
- $\mathcal{P} : \mathcal{X} \times \mathcal{A} \to \mathcal{M}(\mathcal{X})$: Transition probability kernel
- $\mathcal{R} : \mathcal{X} \times \mathcal{A} \to \mathcal{M}(\mathbb{R})$: Immediate reward distribution
- γ: Discount factor $(0 \leq \gamma < 1)$

Markov Decision Process (MDP)

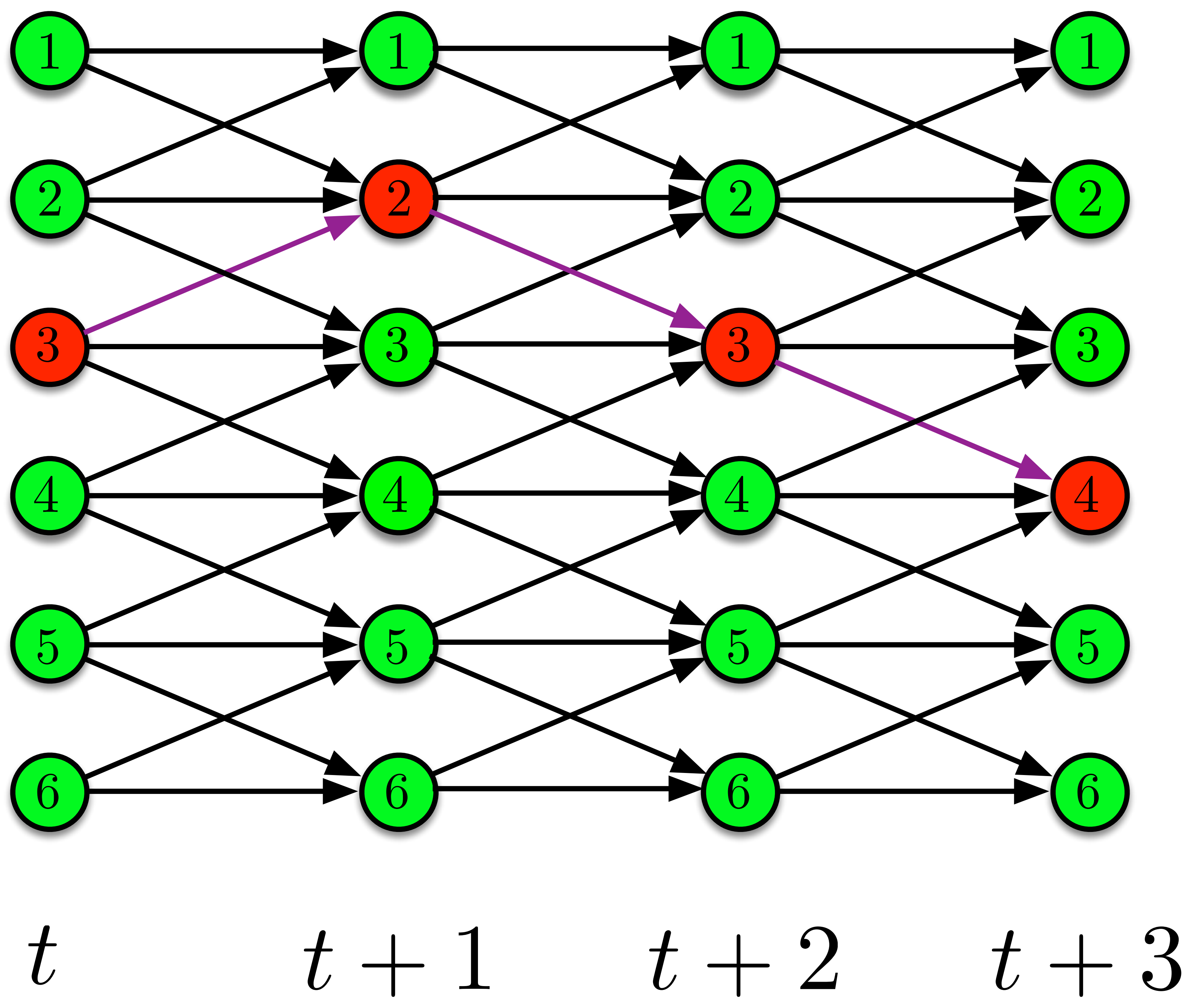


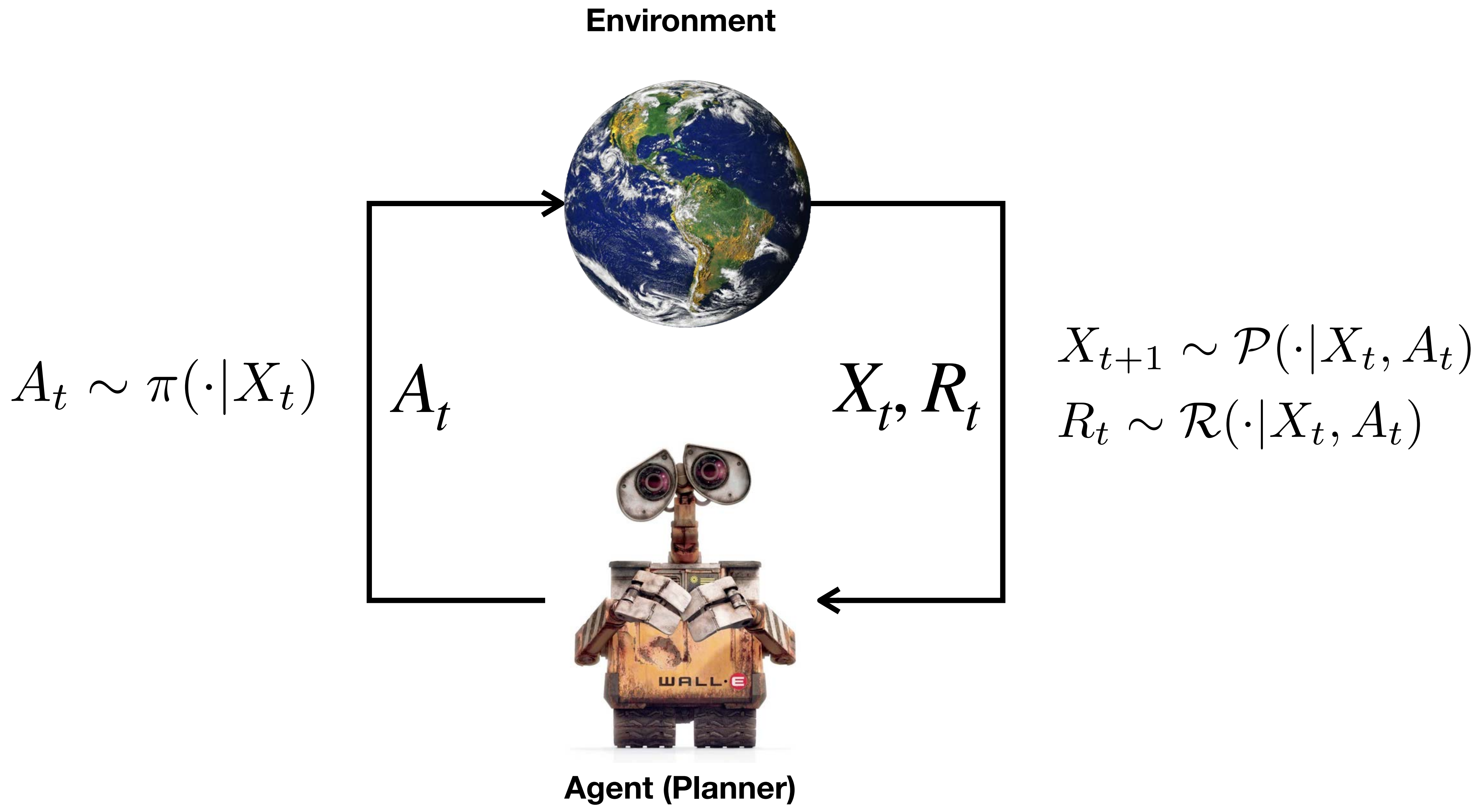


Discounted finite-action MDP $(\mathcal{X}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$:

- $\mathcal{X}$: State space
- $\mathcal{A} = \{a_1, \dots, a_{|\mathcal{A}|}\}$: Action space (finite)
- $\mathcal{P} : \mathcal{X} \times \mathcal{A} \to \mathcal{M}(\mathcal{X})$: Transition probability kernel
- $\mathcal{R} : \mathcal{X} \times \mathcal{A} \to \mathcal{M}(\mathbb{R})$: Immediate reward distribution
- γ: Discount factor $(0 \leq \gamma < 1)$

Markov Decision Process (MDP)

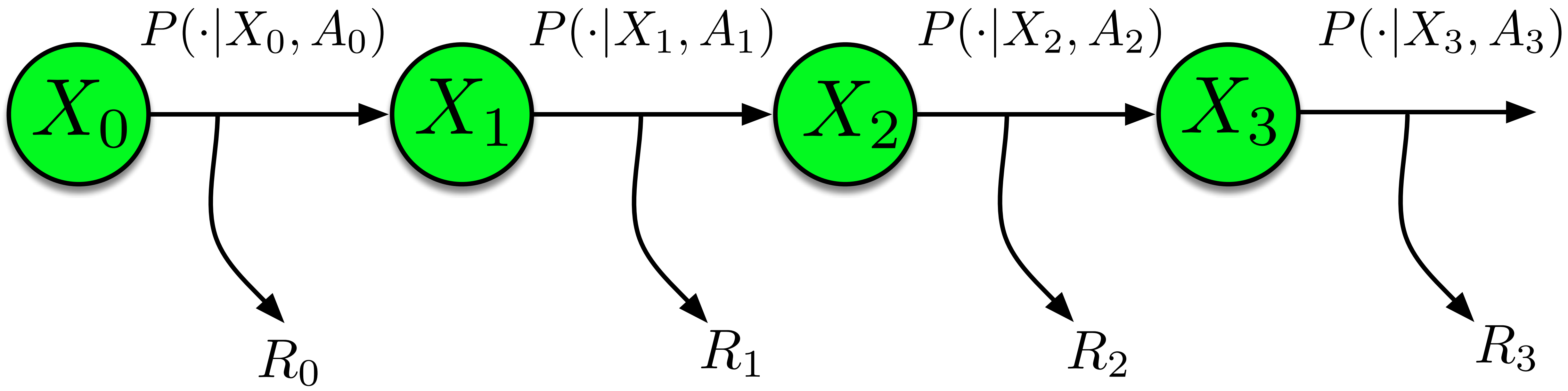


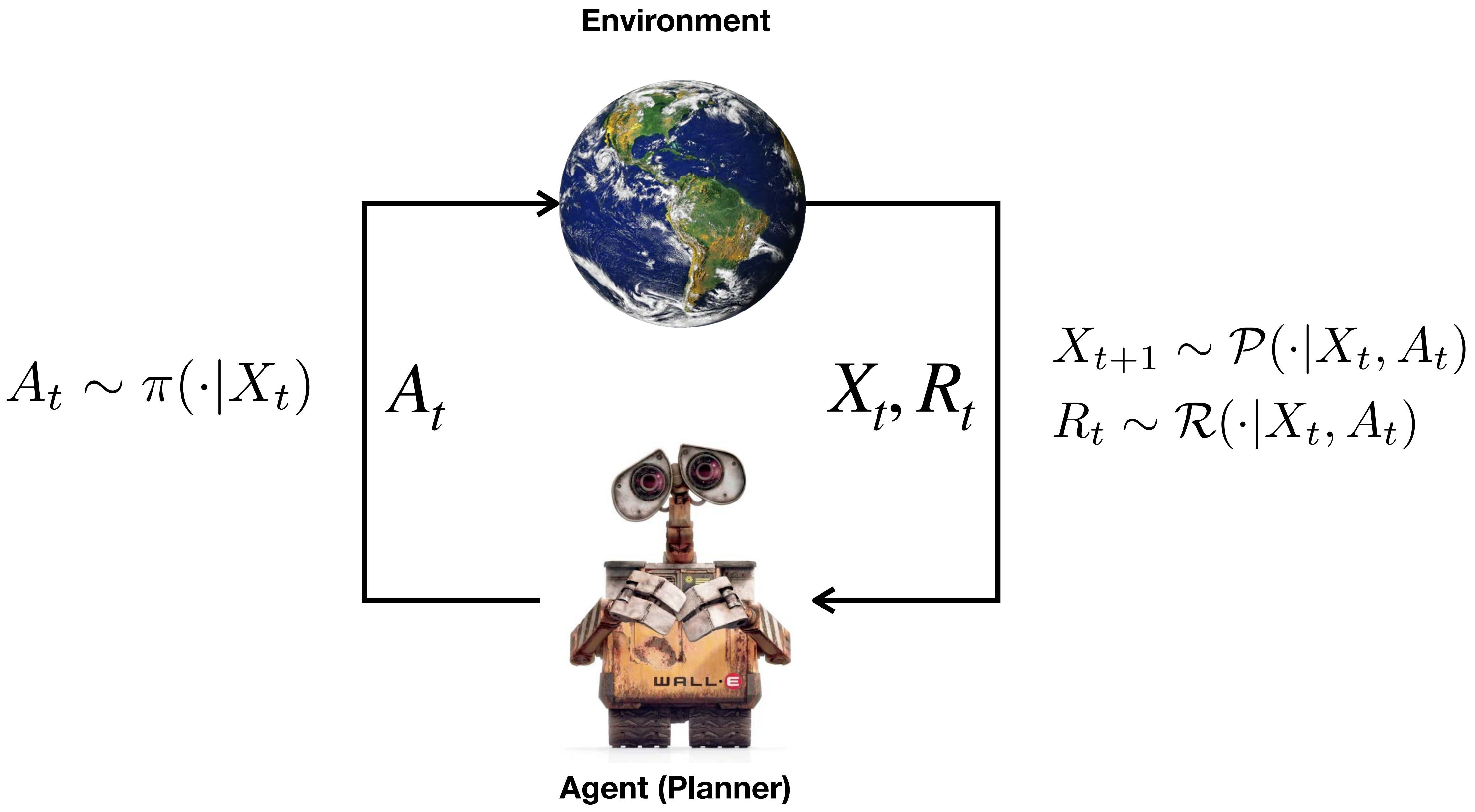


Discounted finite-action MDP $(\mathcal{X}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$:

- $\mathcal{X}$: State space
- $\mathcal{A} = \{a_1, \ldots, a_{|\mathcal{A}|}\}$: Action space (finite)
- $\mathcal{P} : \mathcal{X} \times \mathcal{A} \to \mathcal{M}(\mathcal{X})$: Transition probability kernel
- $\mathcal{R} : \mathcal{X} \times \mathcal{A} \to \mathcal{M}(\mathbb{R})$: Immediate reward distribution
- γ: Discount factor $(0 \leq \gamma < 1)$

Markov Decision Process (MDP)





Return: $\sum_{t=0}^{\infty} \gamma^t R_t$ Policy: $\pi : \mathcal{X} \to \mathcal{A}$

Value functions of a policy:

$$V^\pi(x) \triangleq \mathbb{E}_\pi \left[\sum_{t \geq 0} \gamma^t R_t \mid X_0 = x \right]$$

$$Q^\pi(x, a) \triangleq \mathbb{E}_\pi \left[\sum_{t \geq 0} \gamma^t R_t \mid X_0 = x, A_0 = a \right]$$

Discounted finite-action MDP $(\mathcal{X}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$:

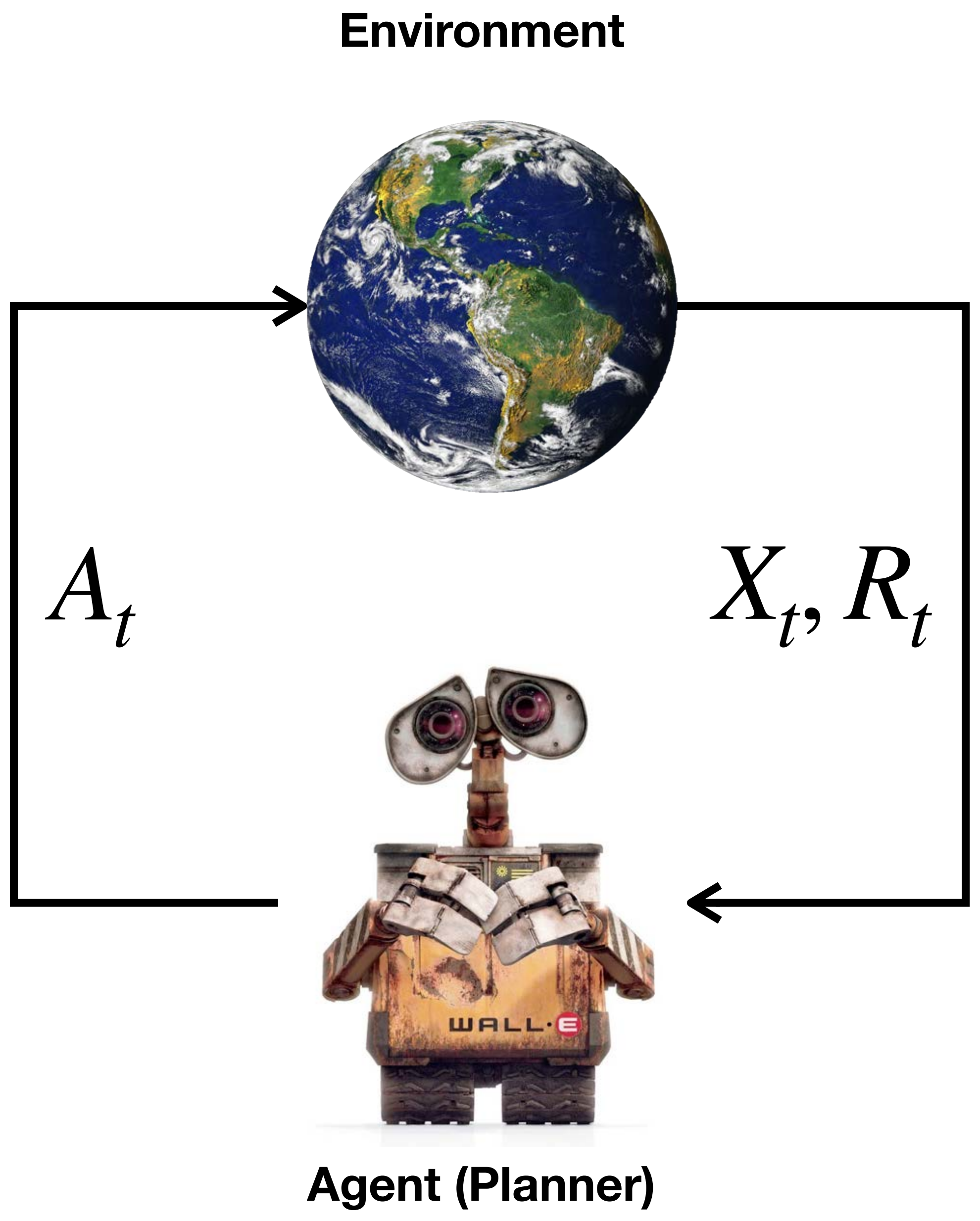
- $\mathcal{X}$: State space
- $\mathcal{A} = \{a_1, \ldots, a_{|\mathcal{A}|}\}$: Action space (finite)
- $\mathcal{P} : \mathcal{X} \times \mathcal{A} \to \mathcal{M}(\mathcal{X})$: Transition probability kernel
- $\mathcal{R} : \mathcal{X} \times \mathcal{A} \to \mathcal{M}(\mathbb{R})$: Immediate reward distribution
- γ: Discount factor $(0 \leq \gamma < 1)$

Markov Decision Process (MDP)

$$V^\pi(x) \triangleq \mathbb{E}_\pi \left[\sum_{t \geq 0} \gamma^t R_t \mid X_0 = x \right]$$

$$Q^\pi(x, a) \triangleq \mathbb{E}_\pi \left[\sum_{t \geq 0} \gamma^t R_t \mid X_0 = x, A_0 = a \right]$$

$A_t \sim \pi(\cdot|X_t)$



$X_{t+1} \sim \mathcal{P}(\cdot|X_t, A_t)$
$R_t \sim \mathcal{R}(\cdot|X_t, A_t)$

Optimal value function:

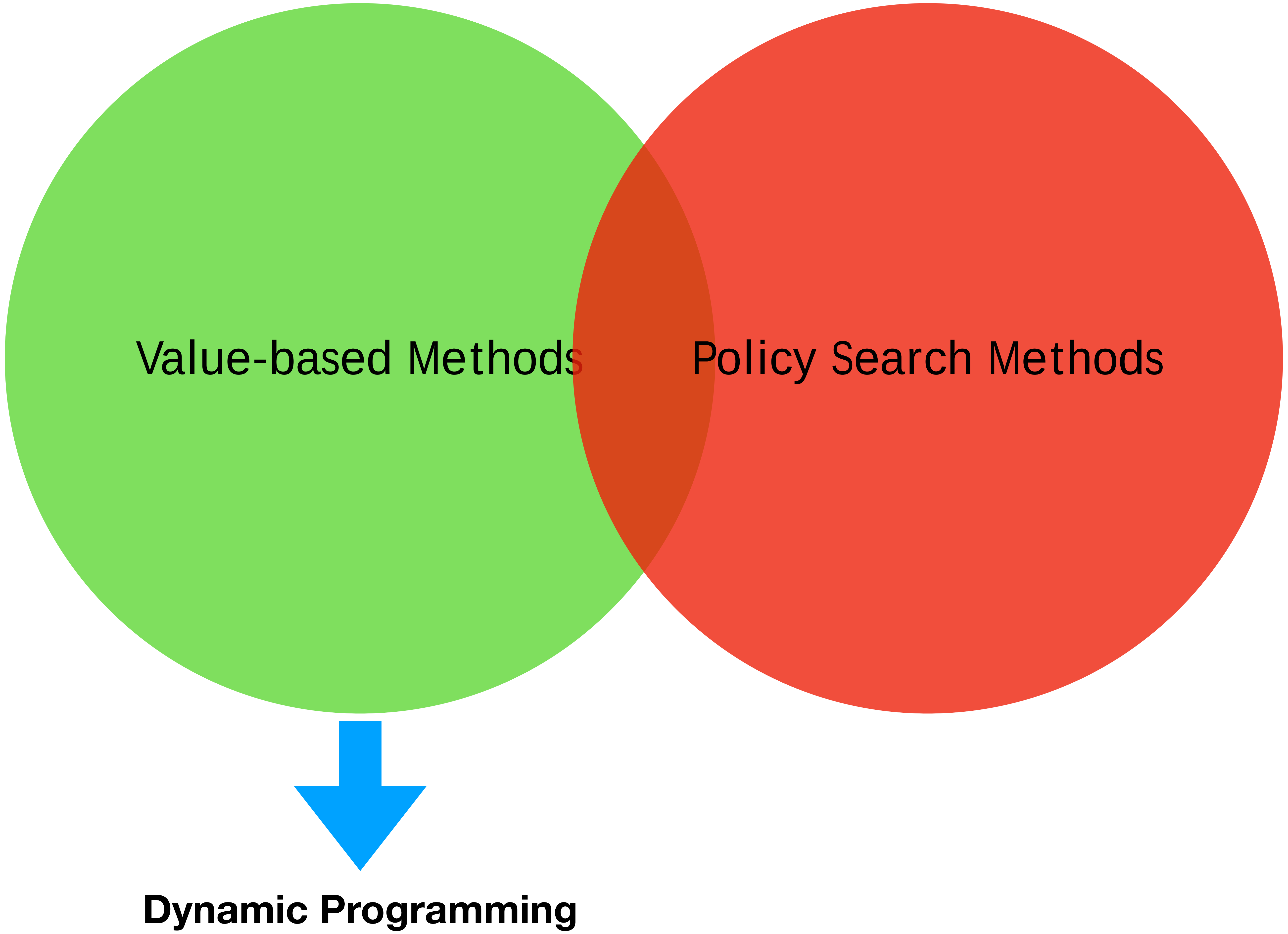
$$Q^*(x, a) = \sup_\pi Q^\pi(x, a)$$

Given Q^*, the optimal policy can be obtained as

$$\pi^*(x) \leftarrow \operatorname*{argmax}_a Q^*(x, a)$$

The goal of an RL agent is to find a policy π that is close to optimal, i.e., $Q^* \approx Q^*$.

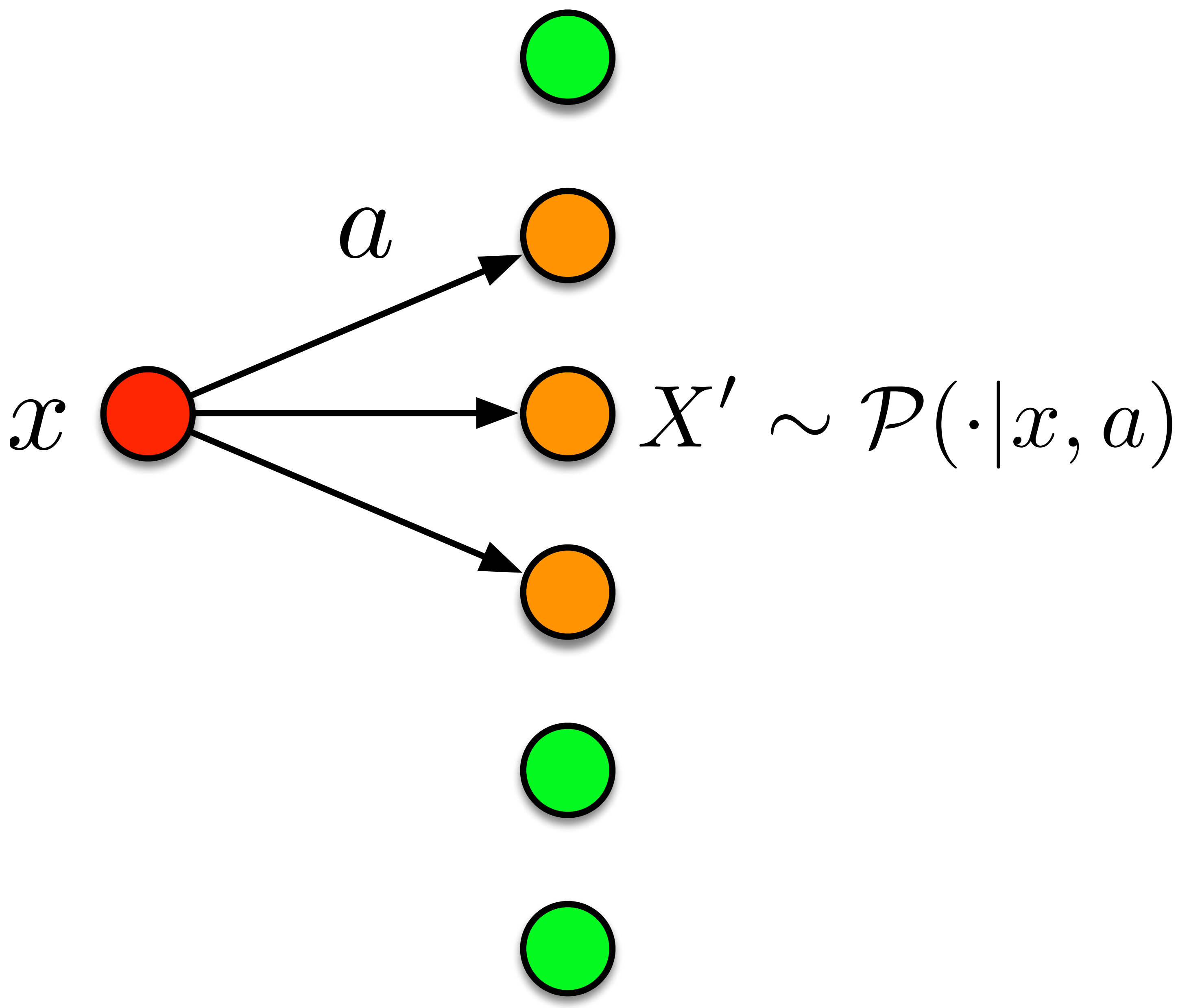
How to Solve MDP and RL Problems?



Bellman Equation

We have the following recursive relationship:

$$
\begin{aligned}
Q^\pi(x,a) &= \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t R_t | X_0 = x, A_0 = a\right] \\
&= \mathbb{E}\left[R(X_0,A_0) + \gamma\sum_{t=0}^{\infty}\gamma^t R_{t+1} | X_0 = x, A_0 = a\right] \\
&= \mathbb{E}\left[R(X_0,A_0) + \gamma Q^\pi(X_1, \pi(X_1)) | X_0 = x, A_0 = a\right] \\
&= \underbrace{r(x,a) + \gamma \int_{\mathcal{X}} \mathcal{P}(\mathrm{d}x'|x,a) Q^\pi(x', \pi(x'))}_{\triangleq (T^\pi Q^\pi)(x,a)}
\end{aligned}
$$



This is called the Bellman equation and $T^\pi : B(\mathcal{X}\times\mathcal{A}) \to B(\mathcal{X}\times\mathcal{A})$ is the Bellman operator. Similarly, we define the Bellman *optimality* operator:

$$
(T^*Q)(x,a) \triangleq r(x,a) + \gamma \int_{\mathcal{X}} \mathcal{P}(\mathrm{d}x'|x,a) \max_{a'\in\mathcal{A}} Q(x',a')
$$

Bellman Equation

Key observation:

$$Q^\pi = T^\pi Q^\pi$$
$$Q^* = T^* Q^*$$

Value-based approaches try to find a $\hat{Q}$ such that

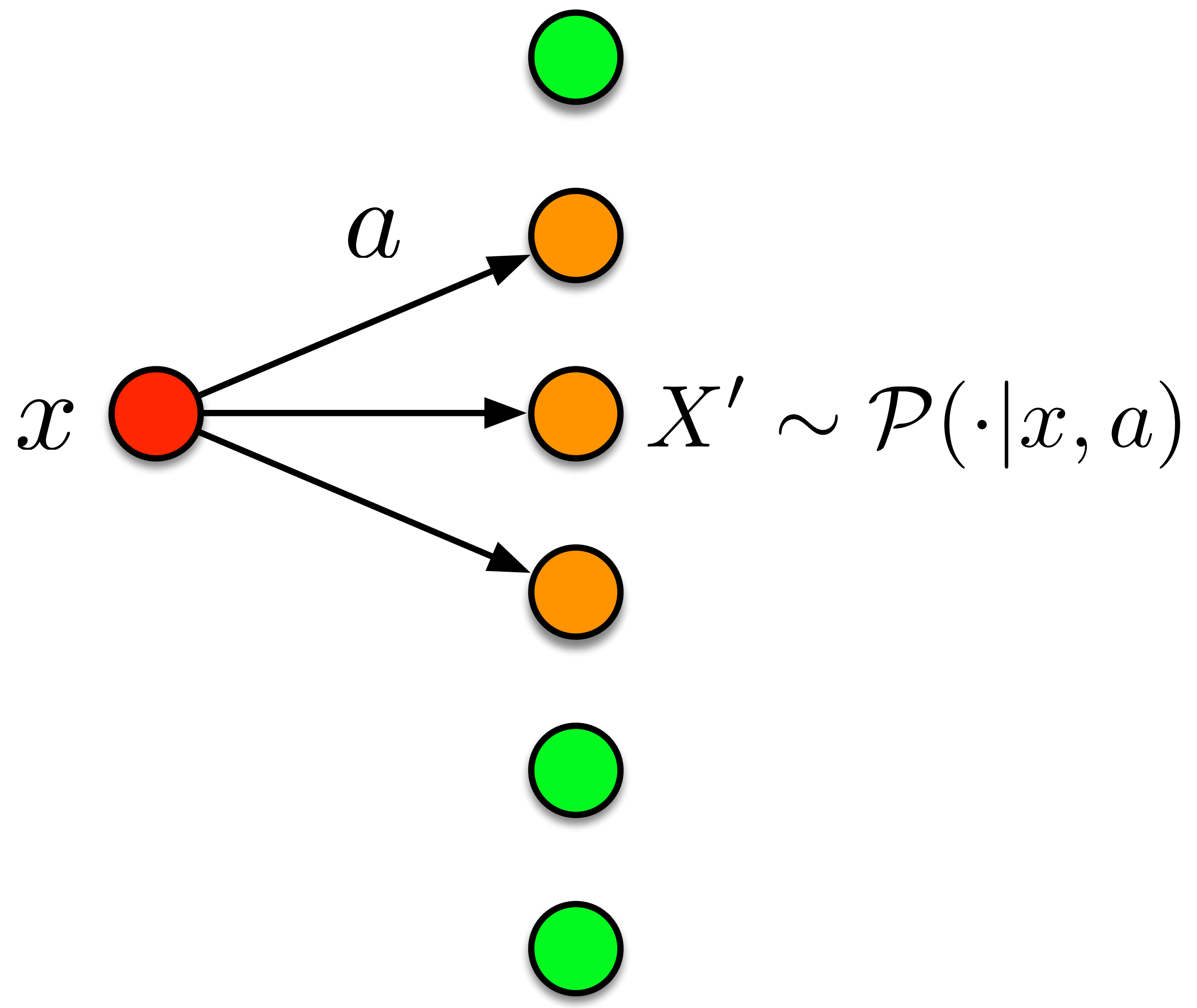
$$\hat{Q} \approx T^* \hat{Q}$$

The greedy policy of $\hat{Q}$ is close to the optimal policy:

$$Q^{\pi(x;\hat{Q})} \approx Q^{\pi^*} = Q^*$$

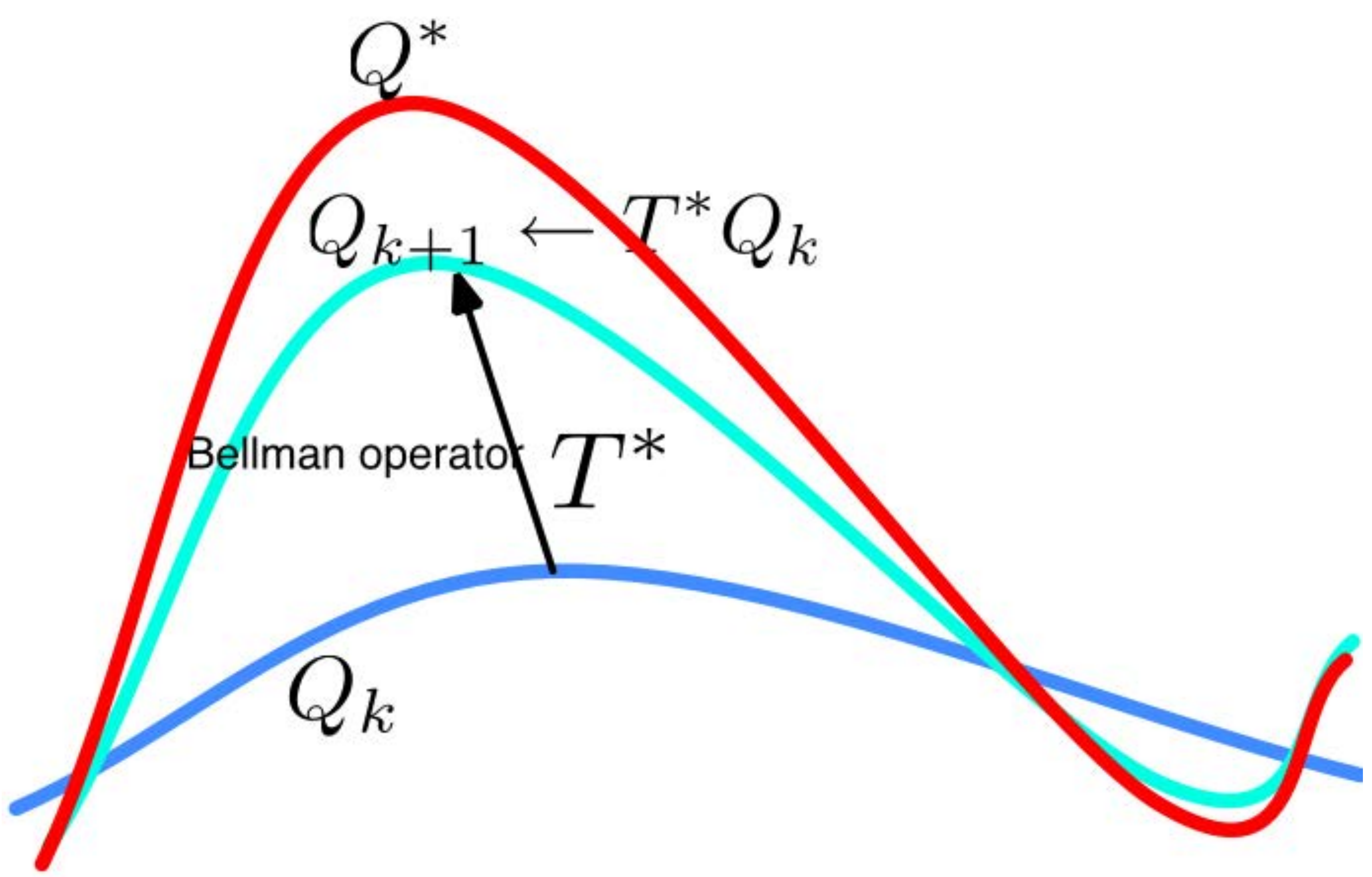
where the greedy policy is defined as

$$\pi(x;\hat{Q}) \leftarrow \operatorname*{argmax}_{a \in \mathcal{A}} \hat{Q}(x,a)$$

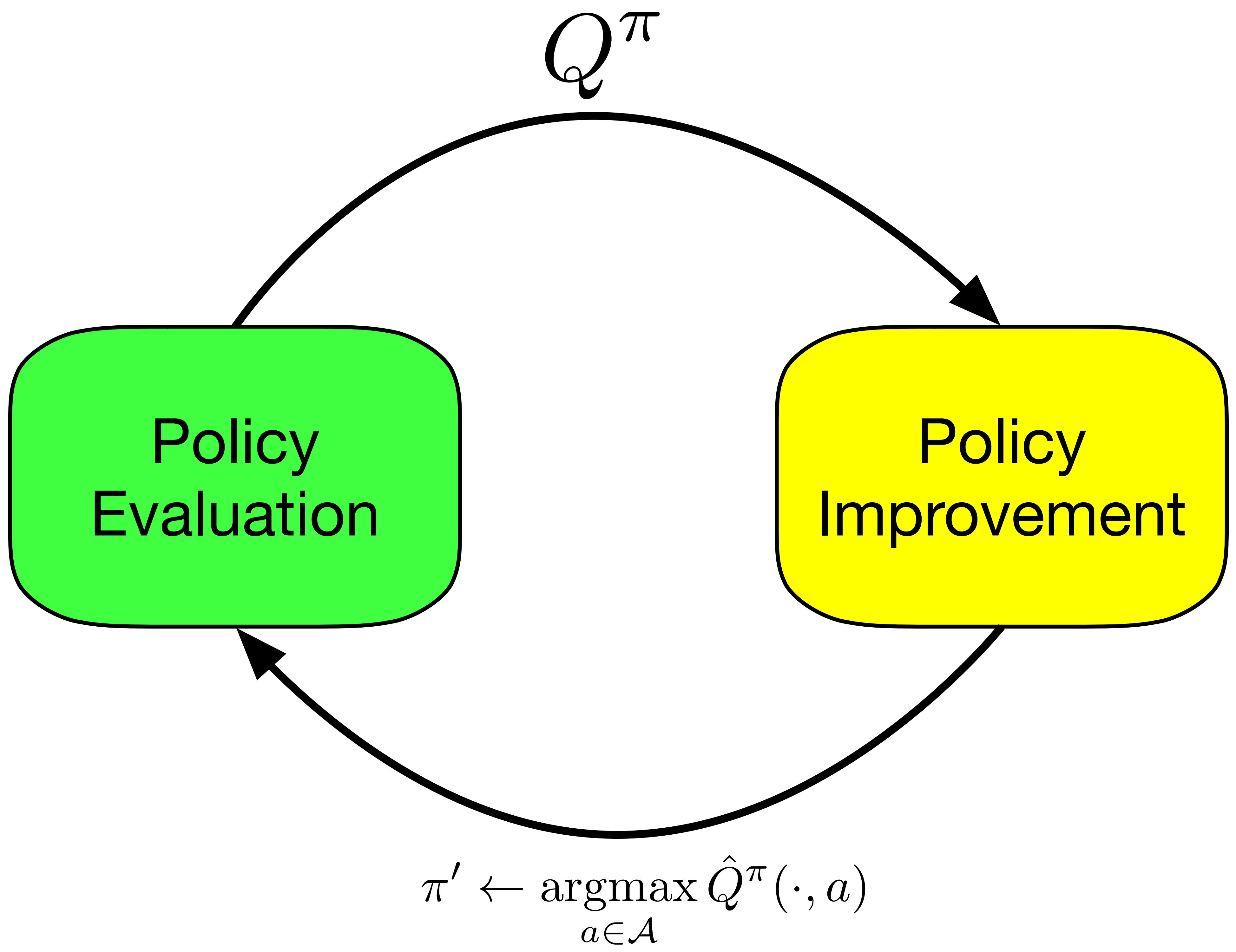


Value and Policy Iteration

Value Iteration



Policy Iteration



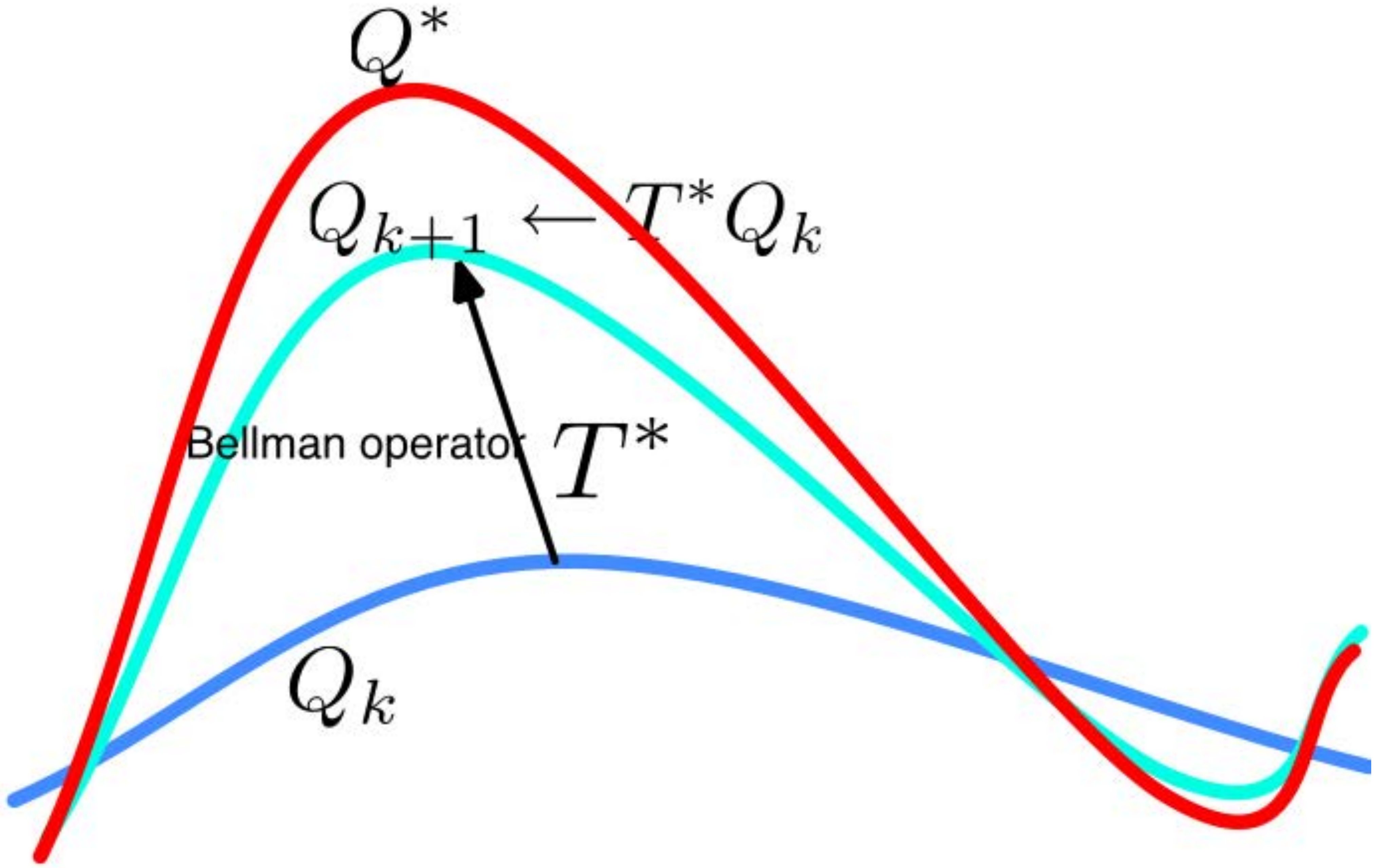
Value Iteration

Value Iteration

$$Q_{k+1} \leftarrow T^* Q_k$$

Value Iteration

$$Q_{k+1} \leftarrow T^* Q_k$$

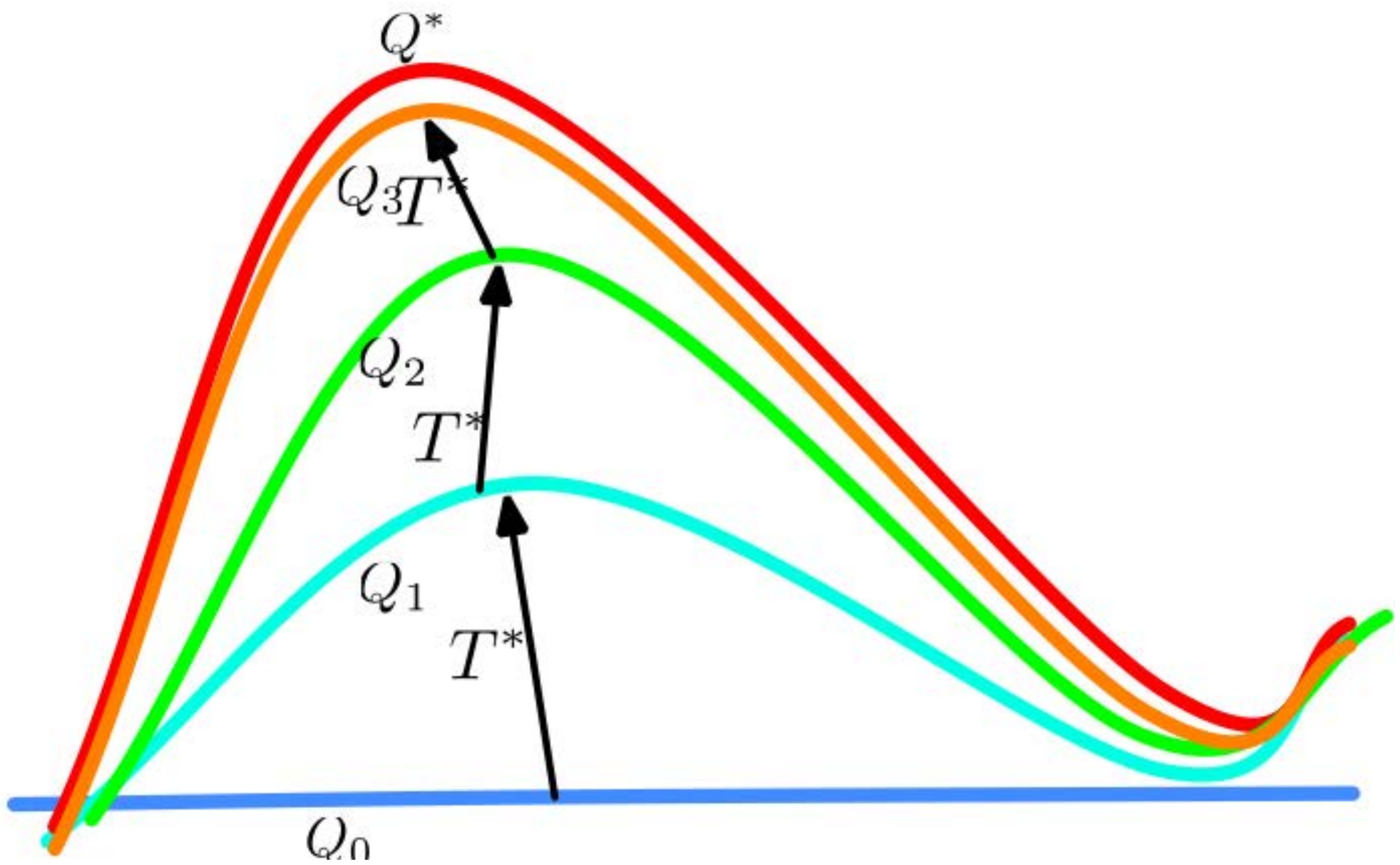


$$Q_{k+1}(x,a) \leftarrow r(x,a) + \gamma \int_{\mathcal{X}} \mathcal{P}(\mathrm{d}x'|x,a) \max_{a' \in \mathcal{A}} Q_k(x',a')$$

$$Q_{k+1}(x,a) \leftarrow r(x,a) + \gamma \sum_{x' \in \mathcal{X}} \mathcal{P}(x'|x,a) \max_{a' \in \mathcal{A}} Q_k(x',a')$$

Value Iteration

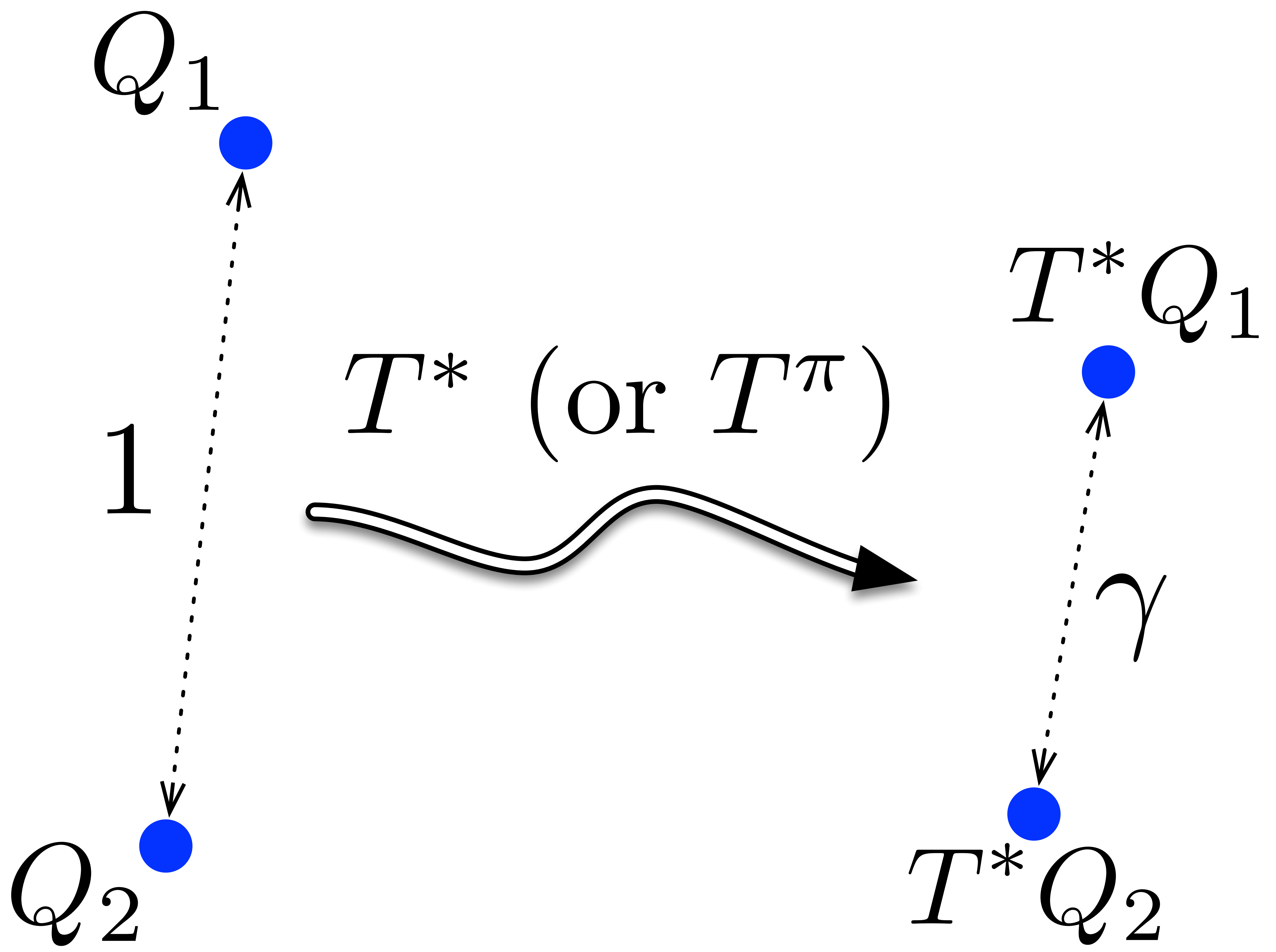
$$Q_{k+1} \leftarrow T^* Q_k$$



Convergence due to the **contraction** property of the Bellman operator

Contraction Property of the Bellman Operator

$$\|T^*Q_1 - T^*Q_2\|_\infty \leq \gamma \|Q_1 - Q_2\|_\infty .$$



Contraction Property of the Bellman Operator (Detail)

$$
\begin{aligned}
|(T^*Q_1)(x,a) - (T^*Q_2)(x,a)| =& \left| \left[r(x,a) + \gamma \int_{\mathcal{X}} \mathcal{P}(\mathrm{d}x'|x,a) \max_{a' \in \mathcal{A}} Q_1(x',a') \right] - \right. \\
& \left. \left[r(x,a) + \gamma \int_{\mathcal{X}} \mathcal{P}(\mathrm{d}x'|x,a) \max_{a' \in \mathcal{A}} Q_2(x',a') \right] \right| \\
=& \gamma \left| \int_{\mathcal{X}} \mathcal{P}(\mathrm{d}x'|x,a) \left[\max_{a' \in \mathcal{A}} Q_1(x',a') - \max_{a' \in \mathcal{A}} Q_2(x',a') \right] \right| \\
\leq & \gamma \int_{\mathcal{X}} \mathcal{P}(\mathrm{d}x'|x,a) \max_{a' \in \mathcal{A}} |Q_1(x',a') - Q_2(x',a')| \\
\leq & \gamma \max_{(x',a') \in \mathcal{X} \times \mathcal{A}} |Q_1(x',a') - Q_2(x',a')| \underbrace{\int_{\mathcal{X}} \mathcal{P}(\mathrm{d}x'|x,a)}_{=1}
\end{aligned}
$$

Therefore, we get that

$$
\sup_{(x,a) \in \mathcal{X} \times \mathcal{A}} |(T^*Q_1)(x,a) - (T^*Q_2)(x,a)| \leq \gamma \sup_{(x,a) \in \mathcal{X} \times \mathcal{A}} |Q_1(x,a) - Q_2(x,a)| .
$$

Or more succinctly,

$$
\|T^*Q_1 - T^*Q_2\|_\infty \leq \gamma \|Q_1 - Q_2\|_\infty .
$$

We also have a similar result for the Bellman operator of a policy π:

$$
\|T^\pi Q_1 - T^\pi Q_2\|_\infty \leq \gamma \|Q_1 - Q_2\|_\infty .
$$

Convergence of Value Iteration

Banach's fixed-point theorem can be used to show the existence and uniqueness of the (optimal) value function. Moreover, it shows that VI converges to that fixed point.

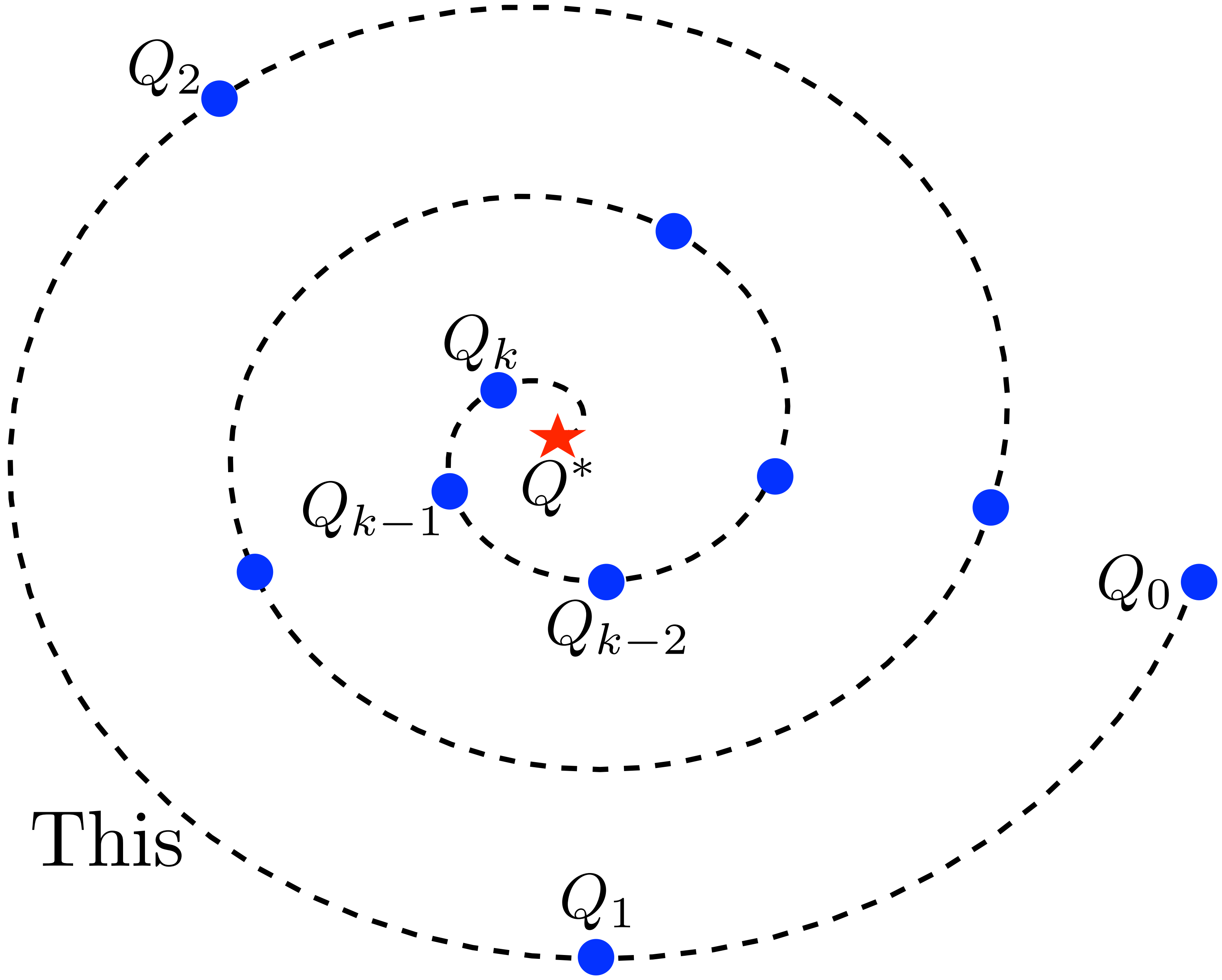
To see this (somehow non-rigorously): Note that $Q^* = T^*Q^*$. So

$$\|T^*Q_0 - Q^*\|_\infty = \|T^*Q_0 - T^*Q^*\|_\infty \le \gamma \|Q_0 - Q^*\|_\infty .$$

Therefore, for VI where $Q_{k+1} \leftarrow T^*Q_k$, we have

$$\|Q_k - Q^*\|_\infty = \left\|(T^*)^{(k)}Q_0 - T^*Q^*\right\|_\infty \le \gamma^k \|Q_0 - Q^*\|_\infty .$$

As $k \to \infty$, the RHS converges to zero, showing that $\|Q_k - Q^*\|_\infty \to 0$. This shows that $Q_k \to Q^*$.



To make it rigorous, we first need to show the existence of a limit (which requires the completeness of the space) and its uniqueness. Banach's fixed-point theorem takes care of this. We also haven't shown that the obtained optimal value function corresponds to any policy. But in fact, it is the case.

Challenges

- Large state space $\mathcal{X} \subset \mathbb{R}^d$
    - Exact representation of the value function is infeasible $Q(x, a) \text{ for all } (x, a) \in \mathcal{X} \times \mathcal{A}$
    - The exact integration in the Bellman operator is challenging

$$Q_{k+1}(x, a) \leftarrow r(x, a) + \gamma \int_{\mathcal{X}} \mathcal{P}(\mathrm{d}x'|x, a) \max_{a' \in \mathcal{A}} Q_k(x', a')$$

- Dynamics is not known (so is Bellman operator)

Is there any hope?

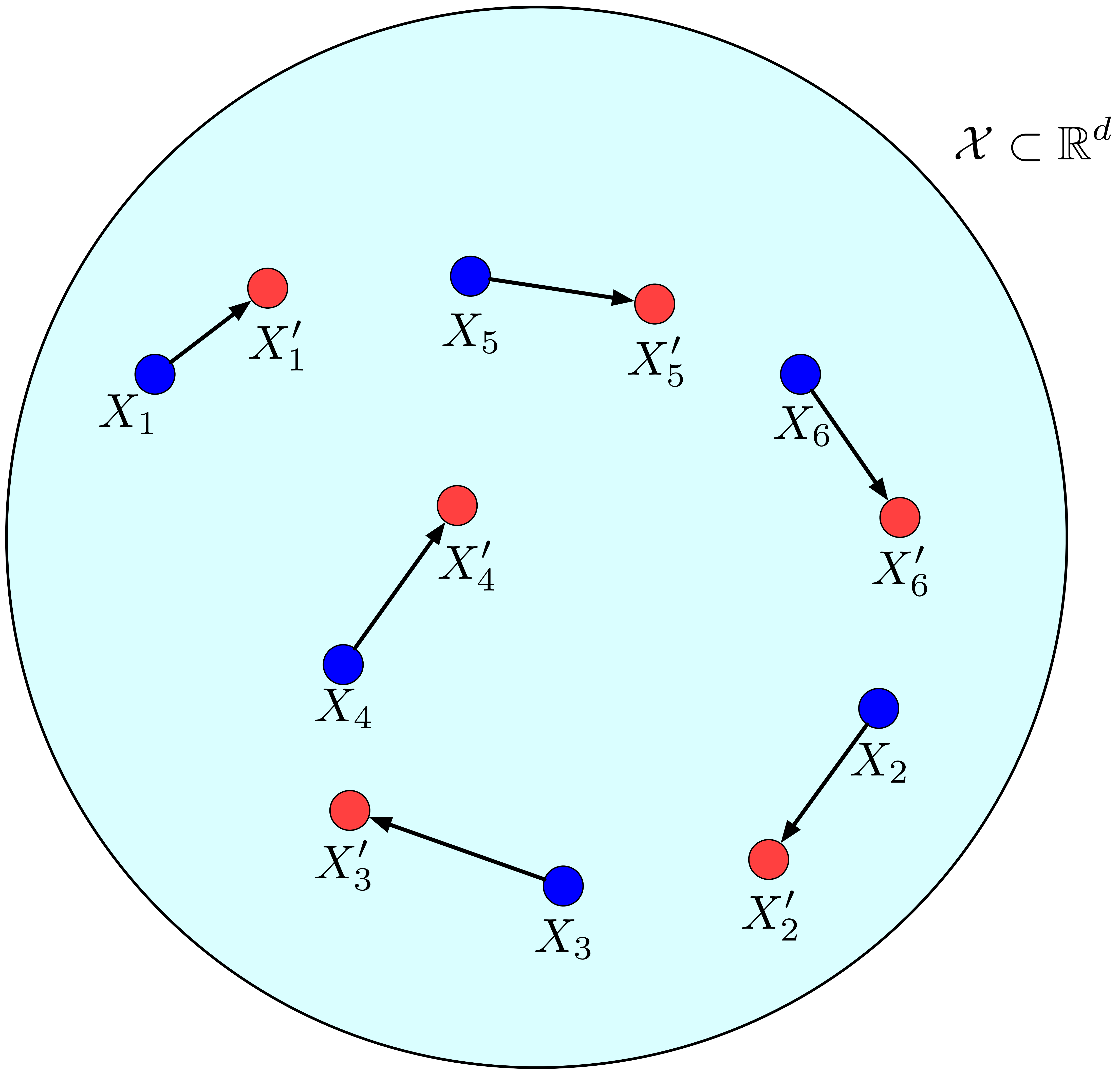
(Batch) RL and Approximate Dynamic Programming

$$\mathcal{D}_n = \{(X_i, A_i, R_i, X_i')\}_{i=1}^{n}$$

$$(X_i, A_i) \sim \nu$$

$$X_i' \sim \mathcal{P}(\cdot | X_i, A_i)$$

$$R_i \sim \mathcal{R}(\cdot | X_i, A_i)$$



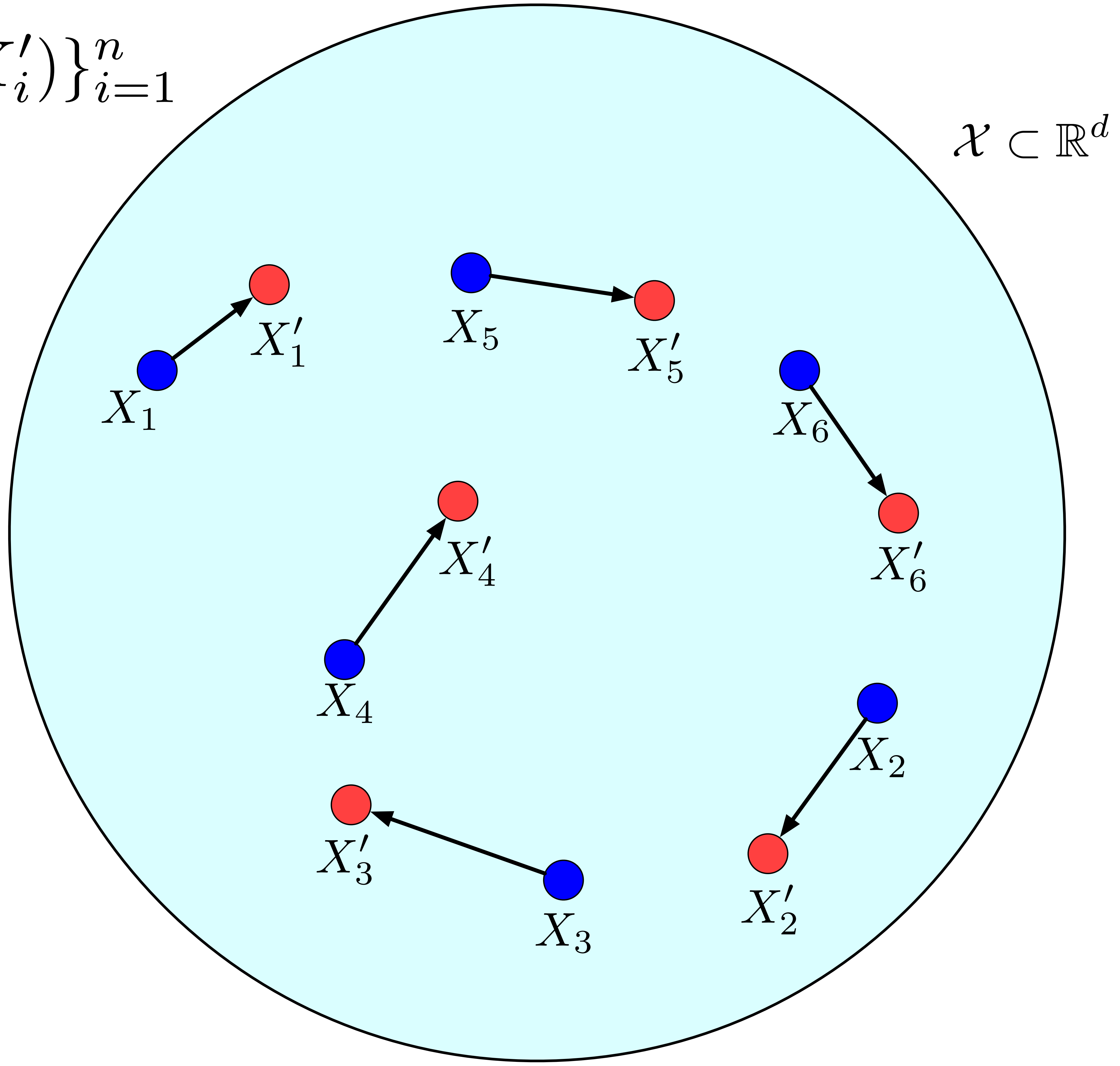
(Batch) RL and Approximate Dynamic Programming

$$\mathcal{D}_n = \{(X_i, A_i, R_i, X'_i)\}_{i=1}^n$$

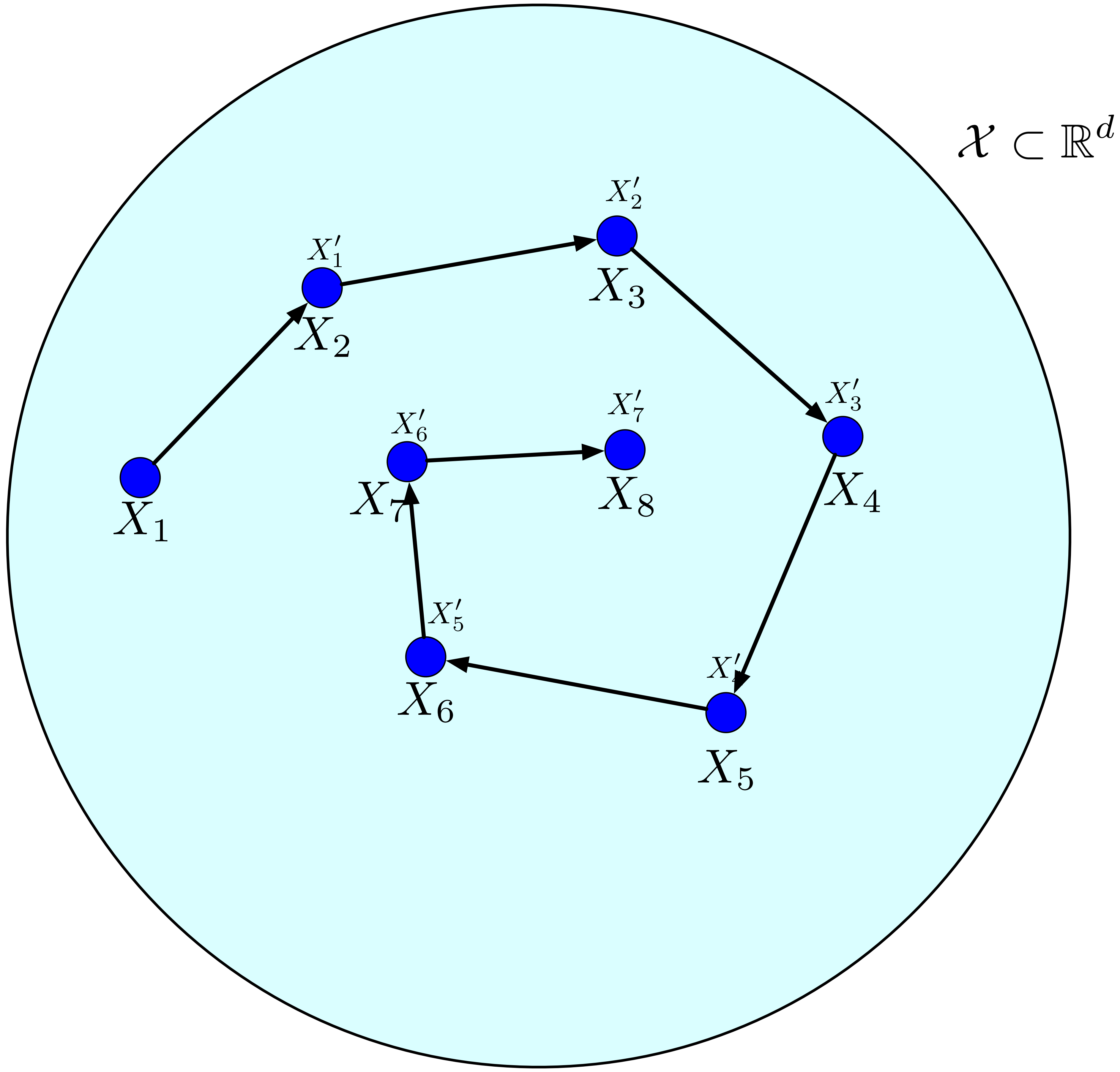
$$(X_i, A_i) \sim \nu$$

$$X'_i \sim \mathcal{P}(\cdot|X_i, A_i)$$

$$R_i \sim \mathcal{R}(\cdot|X_i, A_i)$$



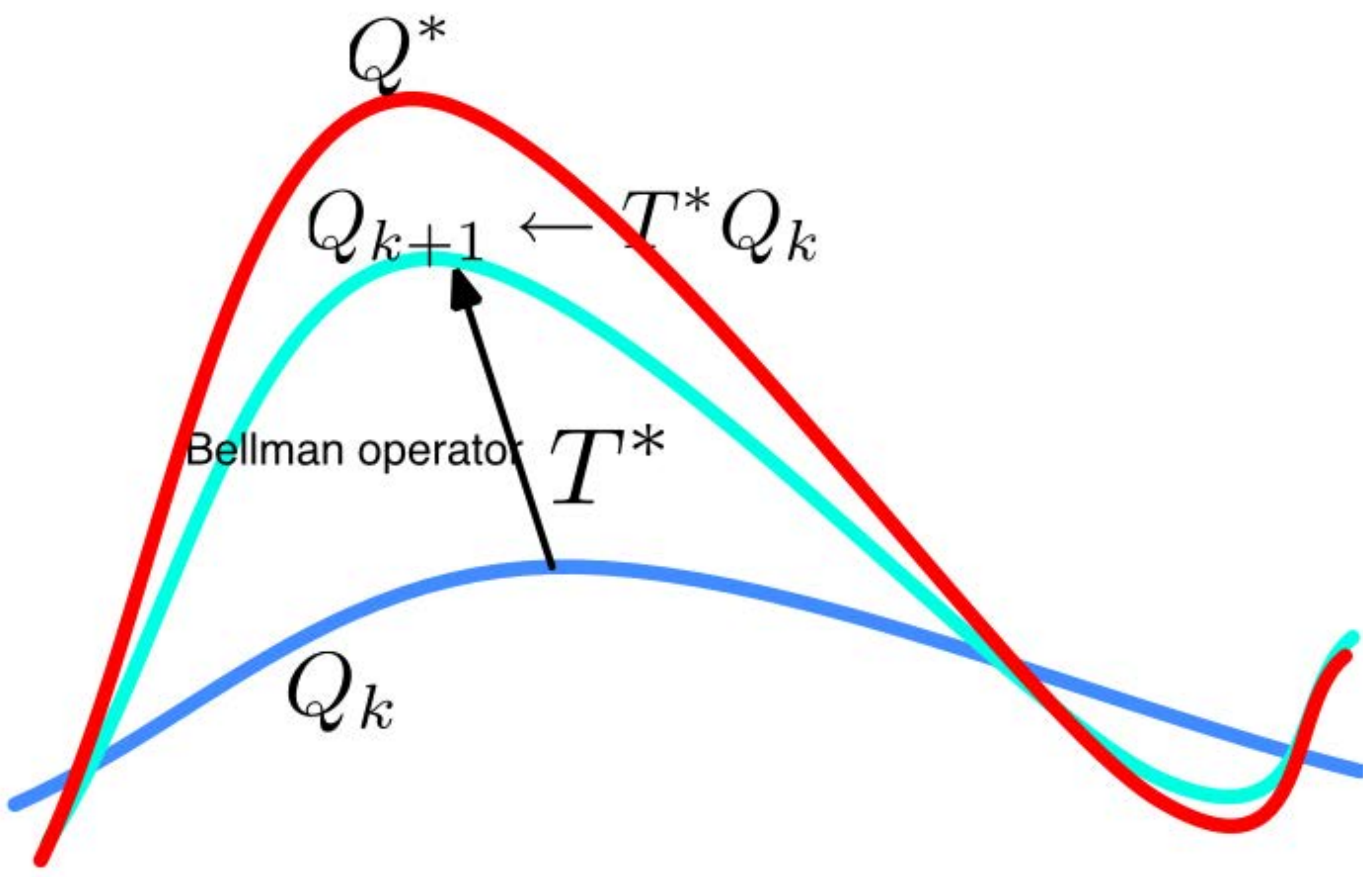
Independent samples

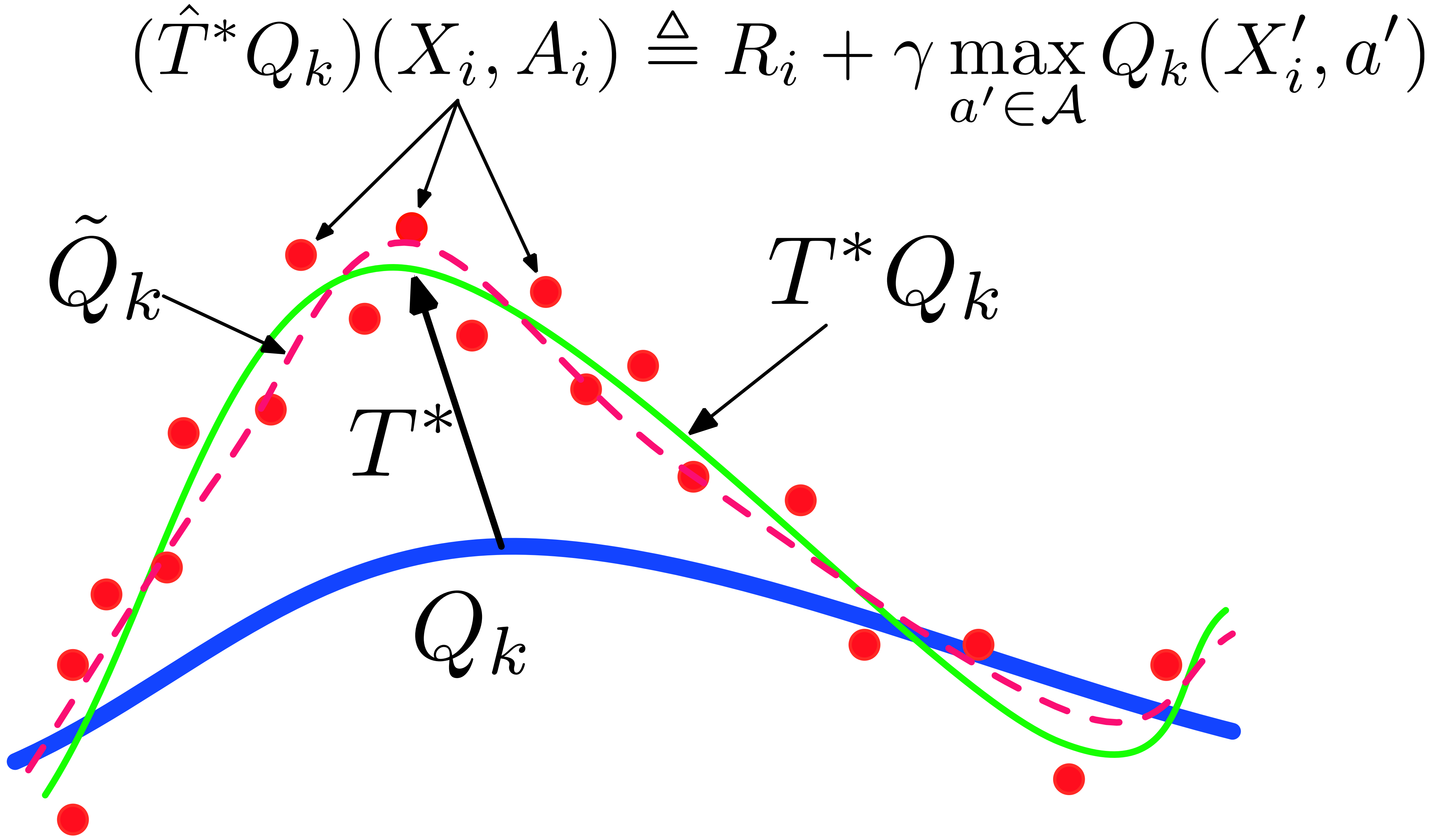


Dependent samples (single trajectory)

Approximate Value Iteration (AVI)

Approximate Value Iteration





$$\mathbb{E}\left[R(x,a) + \gamma \max_{a' \in \mathcal{A}} Q(X', a') \middle| X = x, A = a\right] = (T^*Q)(x,a)$$

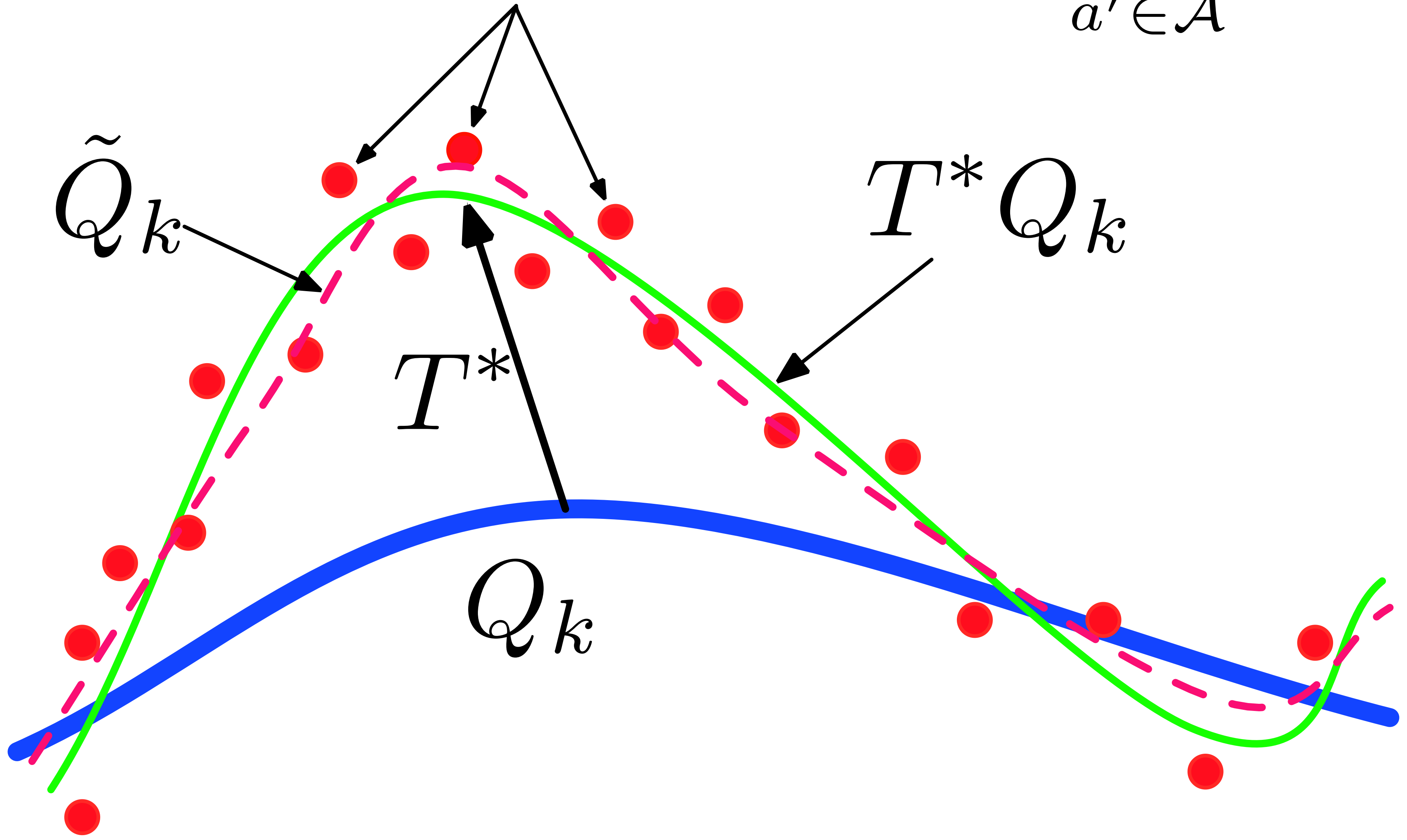
Regression problem

Approximate Value Iteration

$$\mathbb{E}\left[R(x,a) + \gamma \max_{a' \in \mathcal{A}} Q(X', a') \,\middle|\, X = x, A = a\right] = (T^* Q)(x,a)$$

Regression problem

$$(\hat{T}^* Q_k)(X_i, A_i) \triangleq R_i + \gamma \max_{a' \in \mathcal{A}} Q_k(X_i', a')$$



Solve the following estimation problem at each iteration:

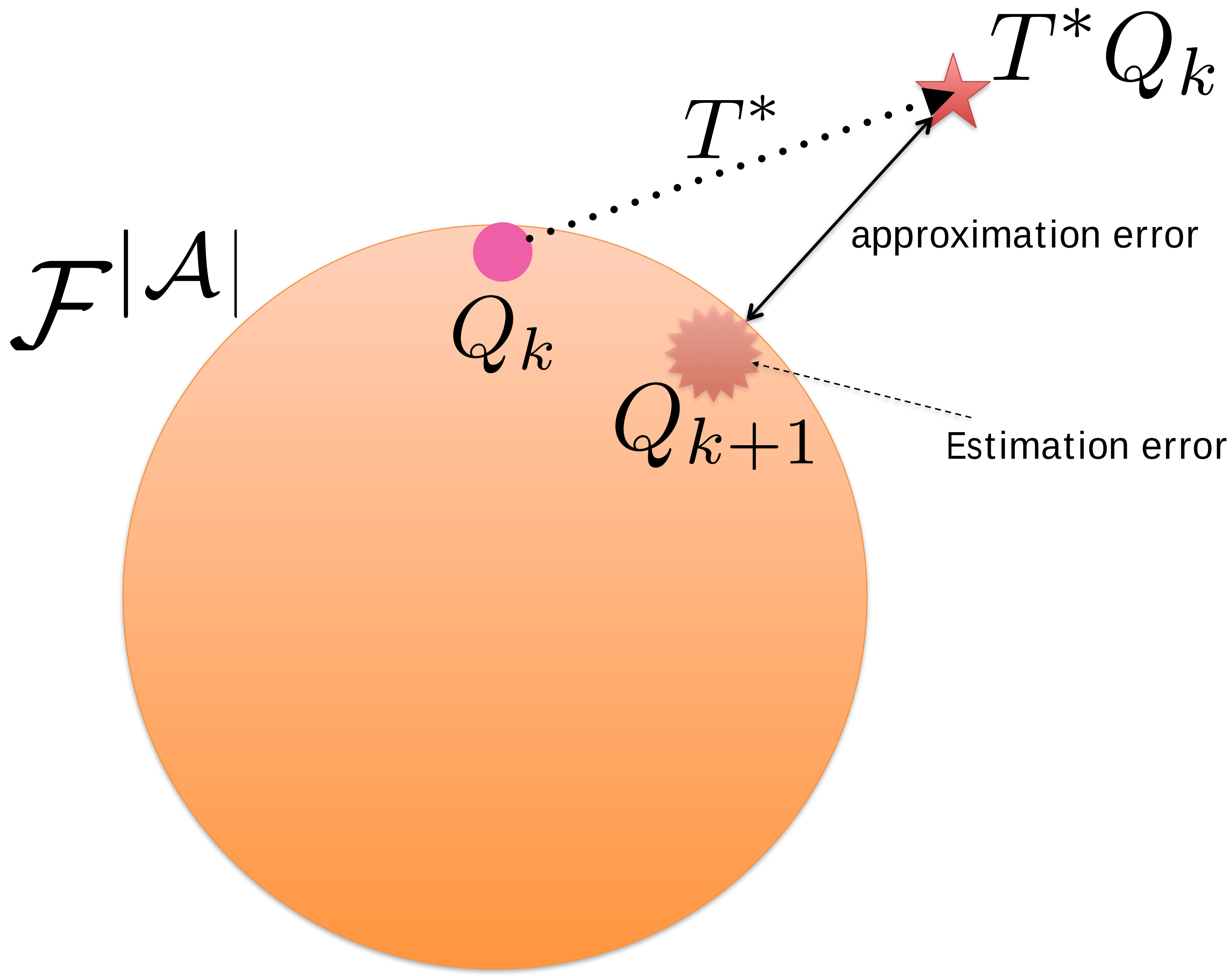
$$Q_{k+1} \leftarrow \underset{Q \in \mathcal{F}^{|\mathcal{A}|}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \left| Q(X_i, A_i) - (\hat{T}^* Q_k)(X_i, A_i) \right|^2$$

AVI is also known as Fitted Value Iteration (FVI) or Fitted Q-Iteration (FQI).

Common Choices of Function Space

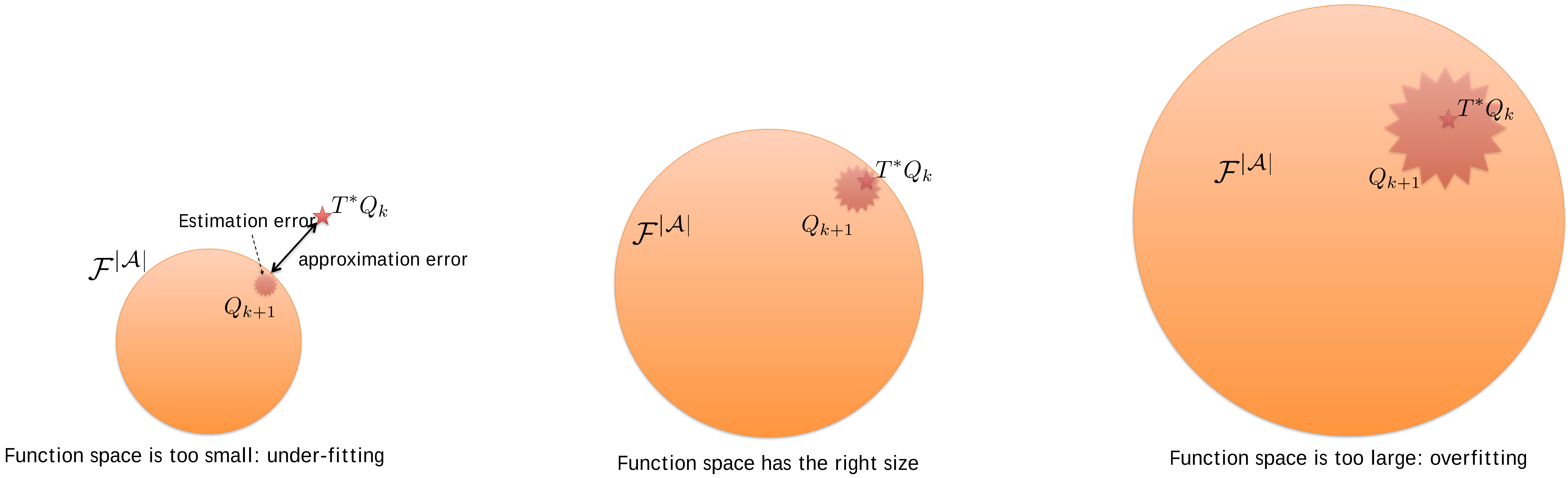
- Linear function space with features ϕ: $\mathcal{F}^{|\mathcal{A}|} = \left\{ Q(x,a) = \phi^\top(x,a)w \;:\; w \in \mathbb{R}^p \right\}$
- Trees, Randomized Trees, etc.
- Reproducing Kernel Hilbert Spaces (RKHS)
- (Deep) Neural Networks

Choice of the Function Space?



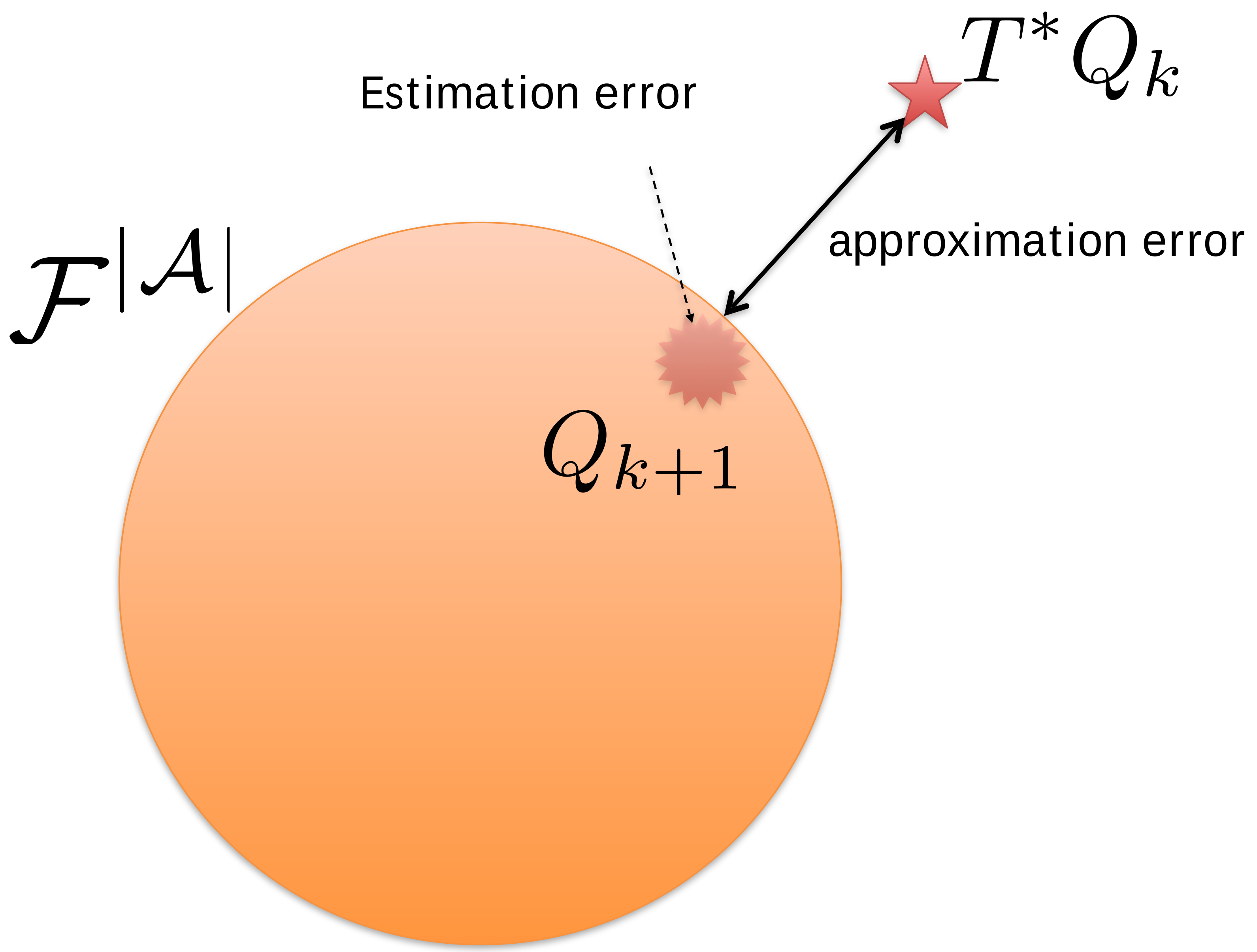
The analysis of the approximation/estimation errors of an estimator is a subject of statistical learning theory. It has studied mostly in the supervised learning context. There are corresponding results for the RL/ADP context too.

Choice of the Function Space?



Regularized Fitted Q-Iteration

Main Idea: Start from a large function space (e.g., dense in the space of continuous functions) and control its complexity using regularization.



Solve the following estimation problem at each iteration:

$$Q_{k+1} \leftarrow \underset{Q \in \mathcal{F}^{|\mathcal{A}|}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \left| Q(X_i, A_i) - (\hat{T}^* Q_k)(X_i, A_i) \right|^2 + \lambda_{Q,n} J^2(Q)$$

Reproducing Kernel Hilbert Space (RKHS)

$$Q_{k+1} \leftarrow \underset{Q \in \mathcal{H}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \left| Q(X_i, A_i) - (\hat{T}^* Q_k)(X_i, A_i) \right|^2 + \lambda_{Q,n} \|Q\|_{\mathcal{H}}^2$$

Regularized Fitted Q-Iteration

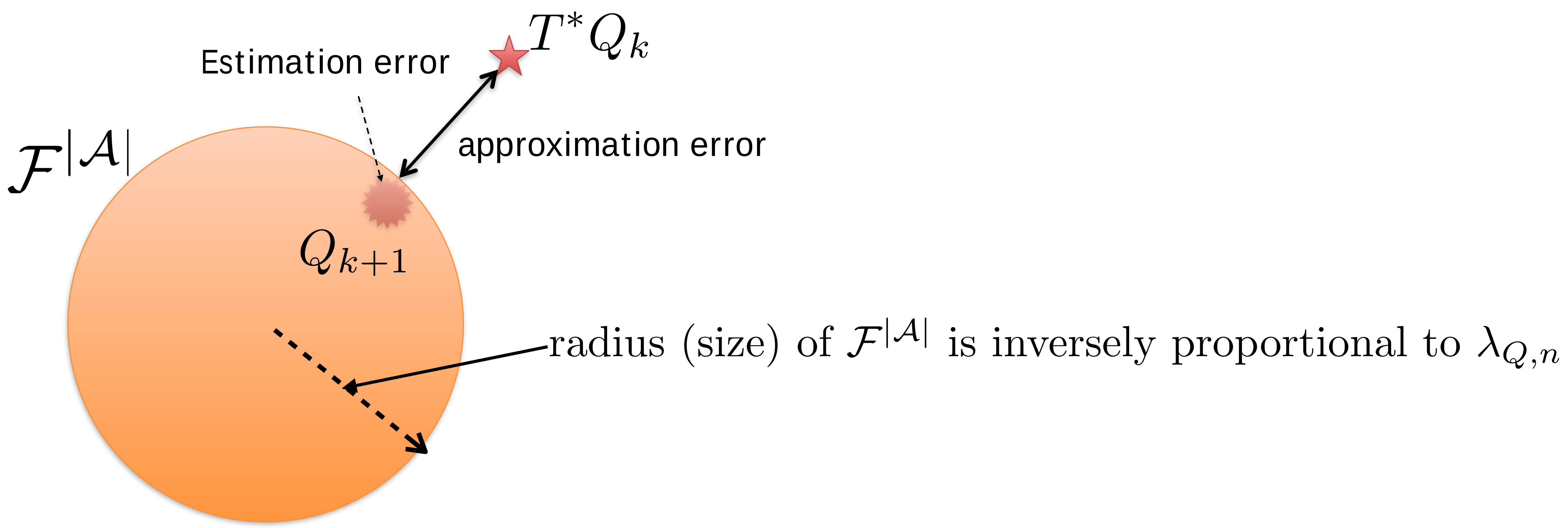
$$Q_{k+1} \leftarrow \operatorname*{argmin}_{Q \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \left| Q(X_i, A_i) - (\hat{T}^* Q_k)(X_i, A_i) \right|^2 + \lambda_{Q,n} \|Q\|_{\mathcal{H}}^2$$

Proof (rough upper bound)

$$\frac{1}{n} \sum_{i=1}^{n} \left| \hat{Q}(X_i, A_i) - (\hat{T}^* Q_k)(X_i, A_i) \right|^2 + \lambda_{Q,n} \left\| \hat{Q} \right\|_{\mathcal{H}}^2 \leq$$

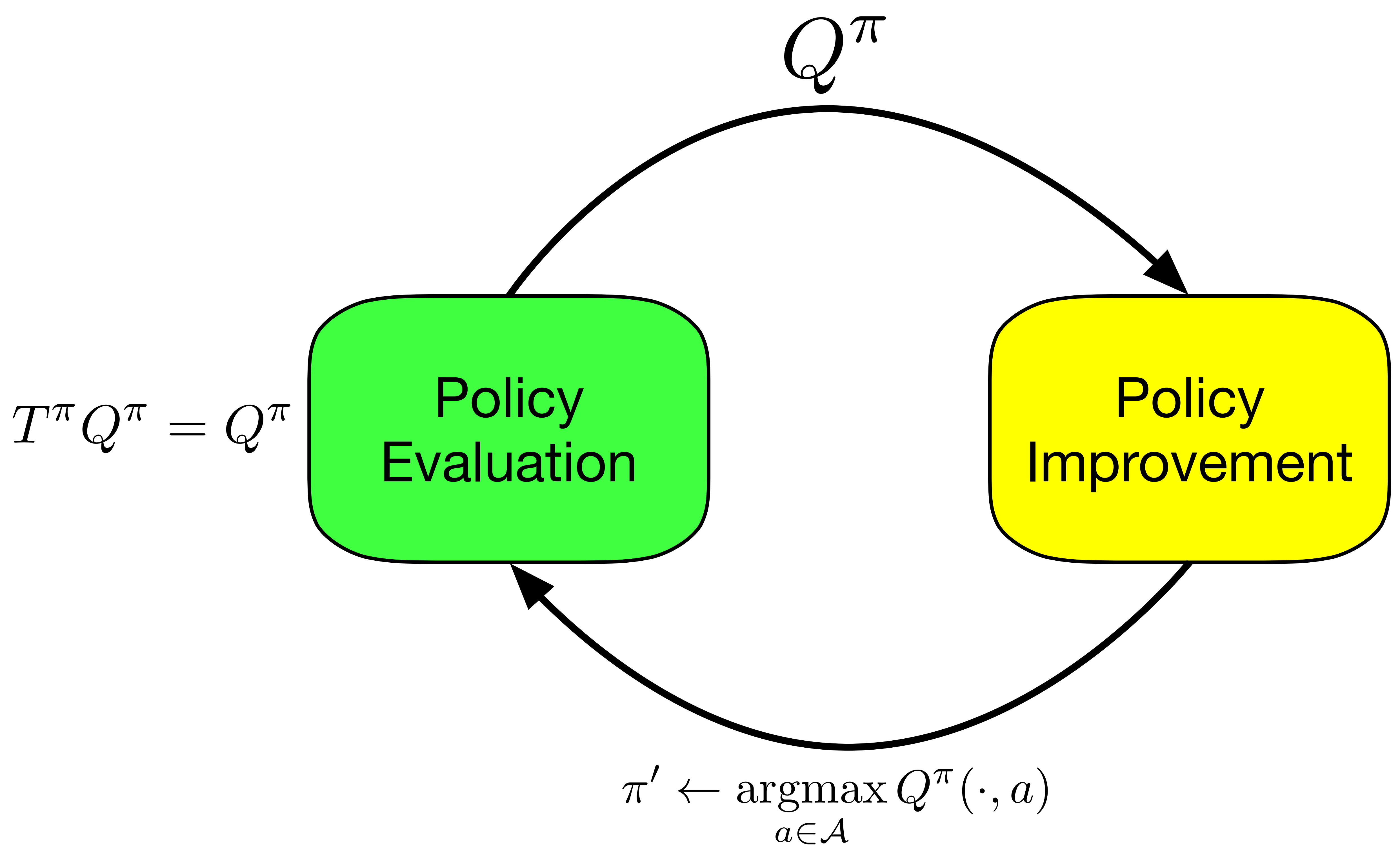
$$\frac{1}{n} \sum_{i=1}^{n} \left| 0 - (\hat{T}^* Q_k)(X_i, A_i) \right|^2 + \lambda_{Q,n} \|0\|_{\mathcal{H}}^2 \leq Q_{\max}^2$$

$$\Rightarrow \left\| \hat{Q} \right\|_{\mathcal{H}} \leq \frac{Q_{\max}}{\sqrt{\lambda_{Q,n}}}$$



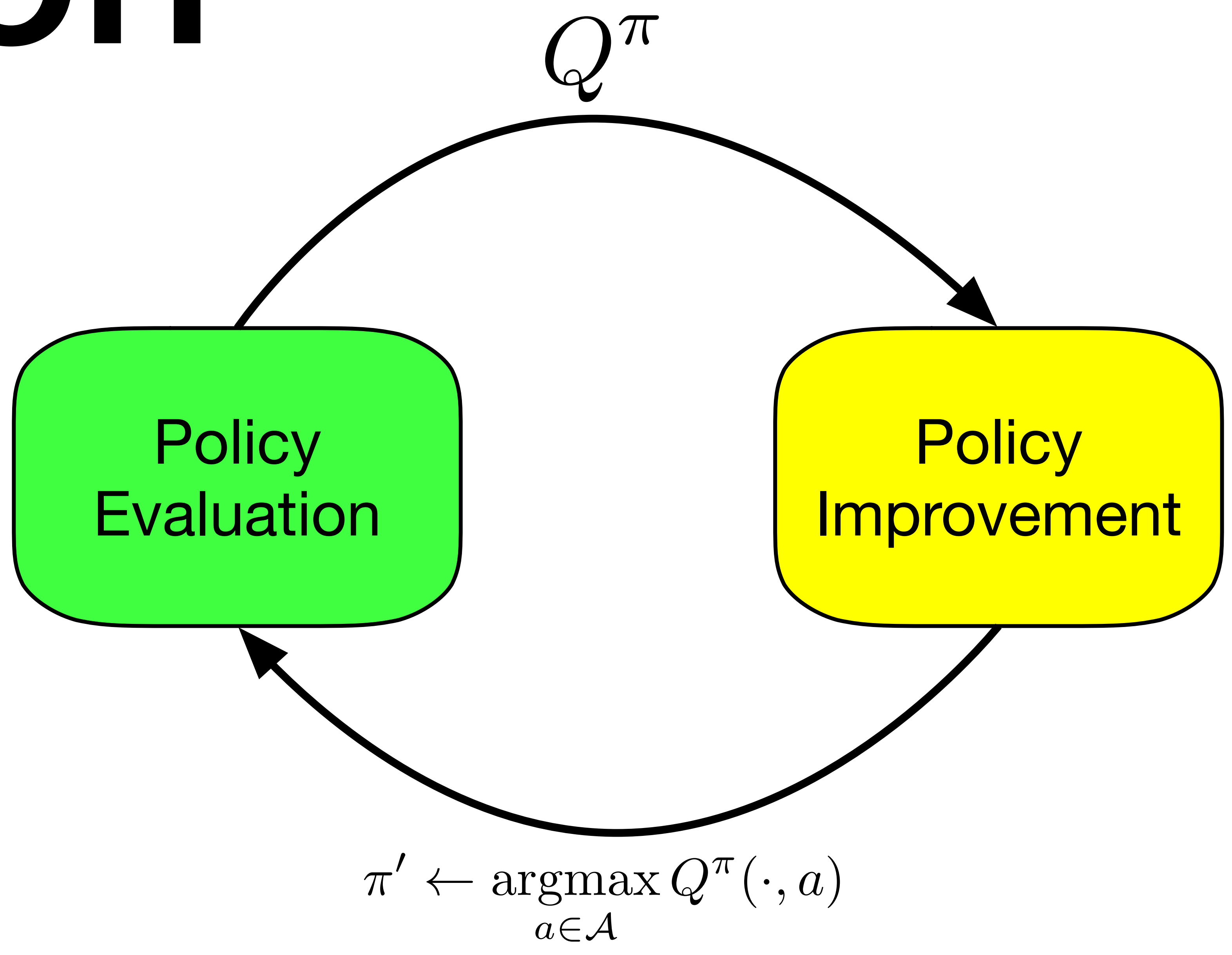
Approximate Policy Iteration (API)

Policy Iteration



If we find a Q such that $T^\pi Q = Q$, then $Q = Q^\pi$.

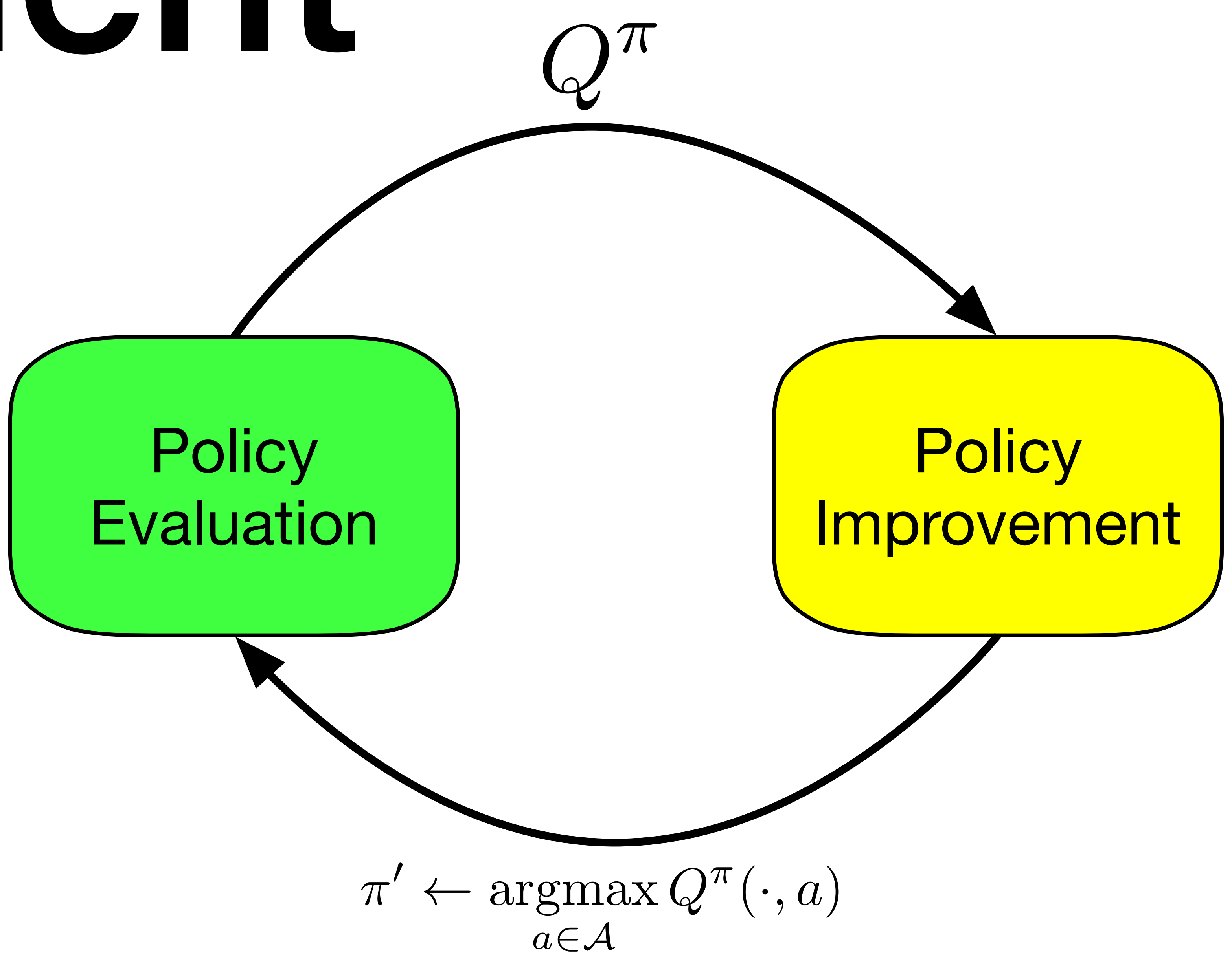
Policy Evaluation



If we find a Q such that $T^\pi Q = Q$, then $Q = Q^\pi$. Assuming that $\mathcal{P}$ and r are known, we have some possibilities:

- Linear System of Equation: Solve the linear system of equations: $Q(x,a) = r(x,a) + \gamma \sum_{x'\in\mathcal{X}} \mathcal{P}(x'|x,a)Q(x',\pi(x'))$ (for all $(x,a) \in \mathcal{X} \times \mathcal{A}$)
- Value Iteration: Iteratively perform $Q_{k+1} \leftarrow T^\pi Q_k$. As T^π is a contraction operator, we will have $Q_k \to Q^\pi$.
- Bellman Error Minimization: Solve $\min_{Q\in\mathcal{F}^{|\mathcal{A}|}} \|Q - T^\pi Q\|$ over the space of $\mathcal{F}^{|\mathcal{A}|} = \mathbb{R}^{\mathcal{X}\times\mathcal{A}}$.

Policy Improvement



Proposition: Suppose that the improved policy π' is the greedy policy w.r.t. Q^π, i.e., $T^{\pi'} Q^\pi = T^* Q^\pi$. We have $Q^{\pi'} \geq Q^\pi$.

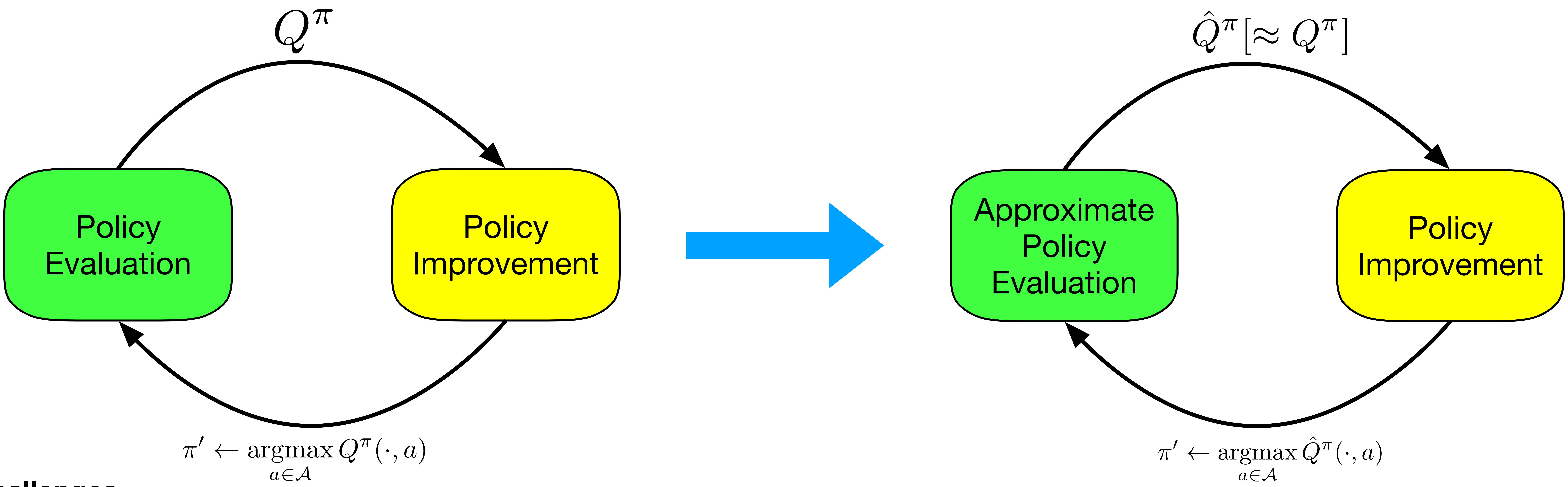
Proof. As $T^{\pi'} Q^\pi = T^* Q^\pi \geq Q^\pi$, for $k \geq 1$, we have

$$(T^{\pi'})^{(k)} Q^\pi = (T^{\pi'})^{(k-1)} (\underbrace{T^{\pi'} Q^\pi}_{= T^* Q^\pi}) \geq (T^{\pi'})^{(k-1)} Q^\pi \geq \cdots \geq Q^\pi.$$

Take the limit of $k \to \infty$. The LHS converges to $Q^{\pi'}$. So $Q^{\pi'} \geq Q^\pi$. □

This is a simplified result. We haven't shown that this converges to the optimal policy.

Approximate Policy Iteration



Challenges

- Large state space
  - Exact representation of the value function is infeasible
  - The exact integration in the Bellman operator is challenging
- Dynamics is not known

Approximate Policy Evaluation?

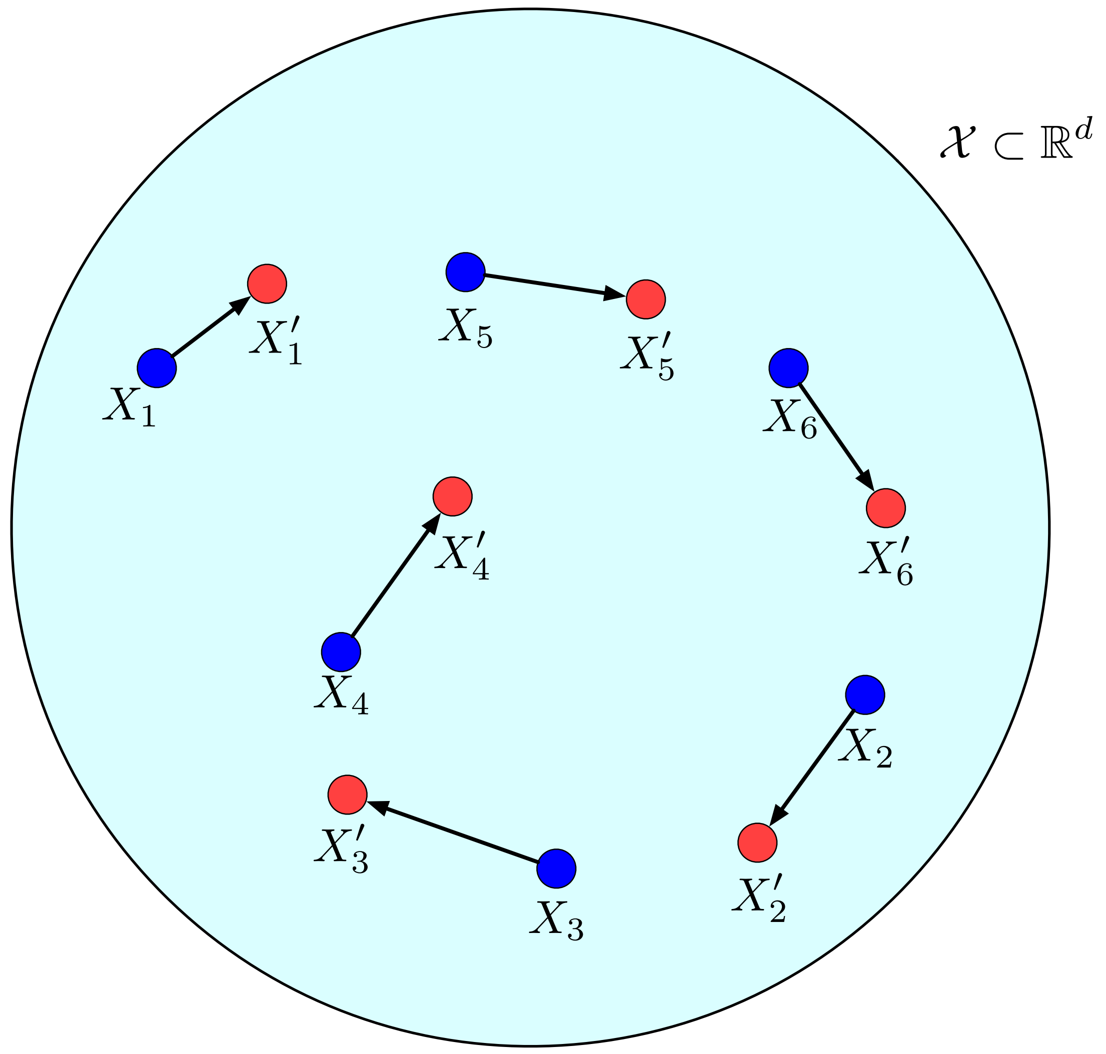
$$Q^{\pi} = T^{\pi} Q^{\pi}$$

$$\Downarrow$$

$$\hat{Q} \approx T^{\pi} \hat{Q}$$

Approximate Policy Evaluation with a Data Batch

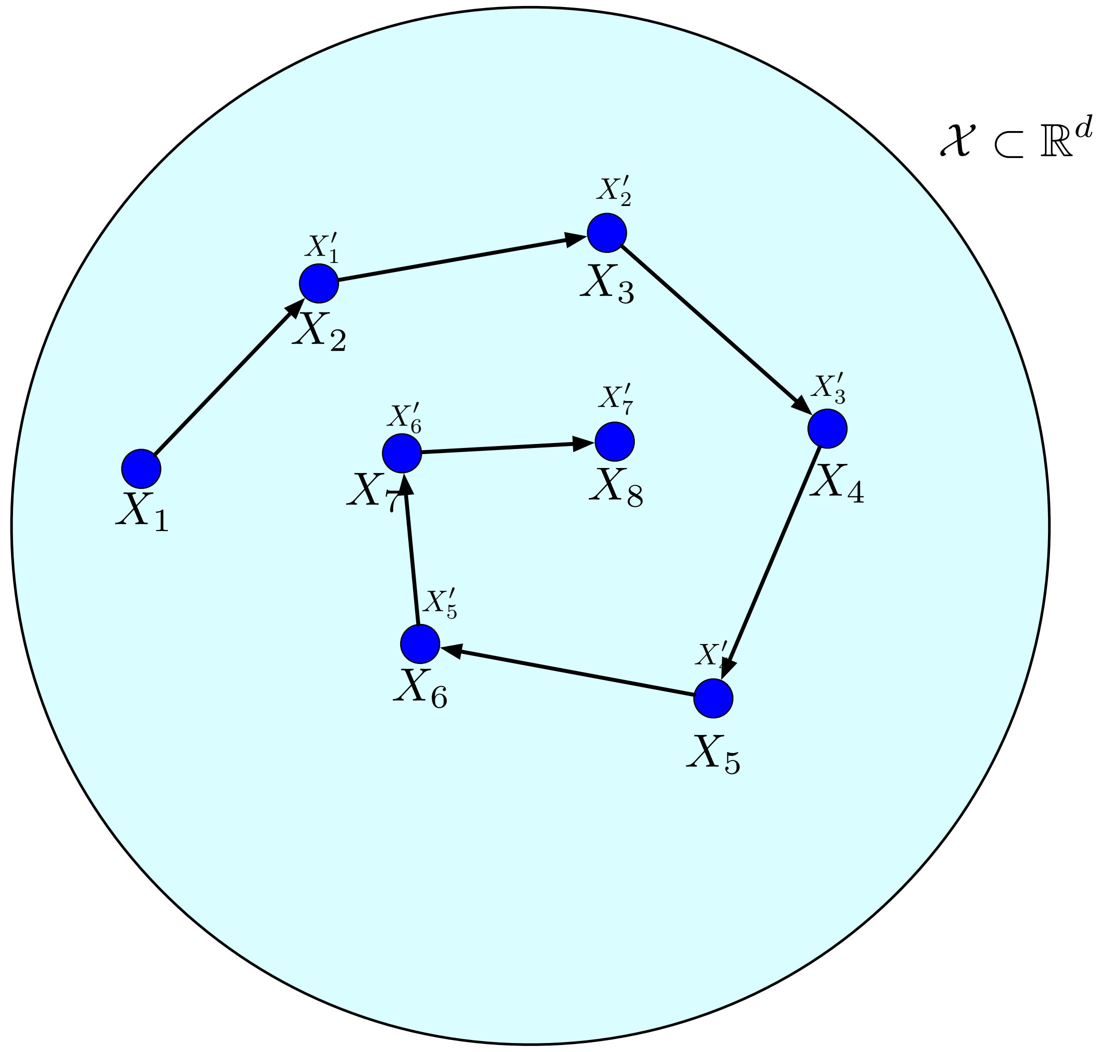
$$\mathcal{D}_n = \{(X_i, A_i, R_i, X'_i)\}_{i=1}^n$$
$$(X_i, A_i) \sim \nu$$
$$X'_i \sim \mathcal{P}(\cdot|X_i, A_i)$$
$$R_i \sim \mathcal{R}(\cdot|X_i, A_i)$$



Recipe for (exact) Policy Evaluation

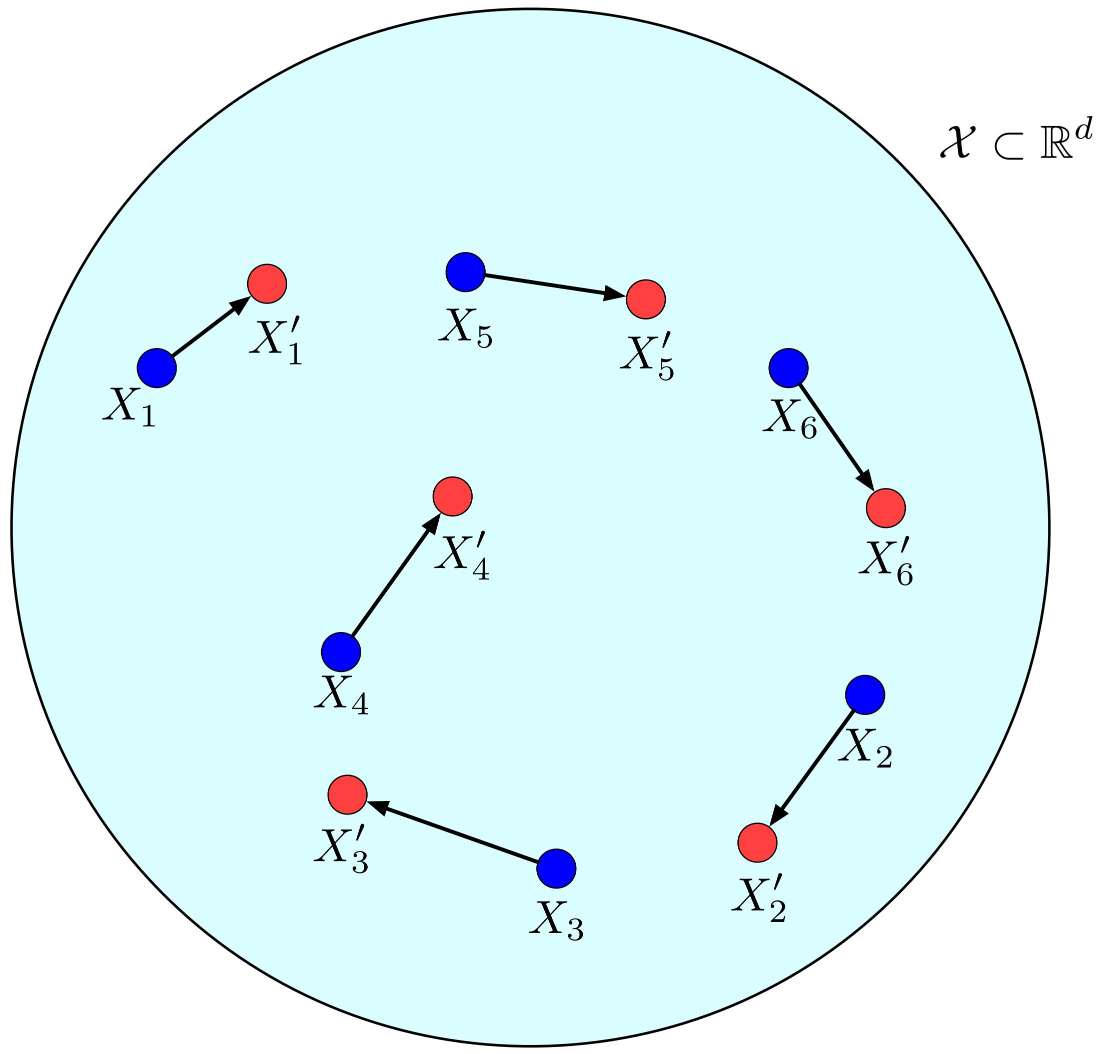
If we find a Q such that $T^\pi Q = Q$, then $Q = Q^\pi$. Assuming that $\mathcal{P}$ and r are known, we have some possibilities:

- Linear System of Equation: Solve the linear system of equations: $Q(x, a) = r(x, a) + \gamma \sum_{x' \in \mathcal{X}} \mathcal{P}(x'|x, a) Q(x', \pi(x'))$ (for all $(x, a) \in \mathcal{X} \times \mathcal{A}$)
- Value Iteration: Iteratively perform $Q_{k+1} \leftarrow T^\pi Q_k$. As T^π is a contraction operator, we will have $Q_k \to Q^\pi$.
- Bellman Error Minimization: Solve $\min_{Q \in \mathcal{F}^{|\mathcal{A}|}} \|Q - T^\pi Q\|$ over the space of $\mathcal{F}^{|\mathcal{A}|} = \mathbb{R}^{\mathcal{X} \times \mathcal{A}}$.



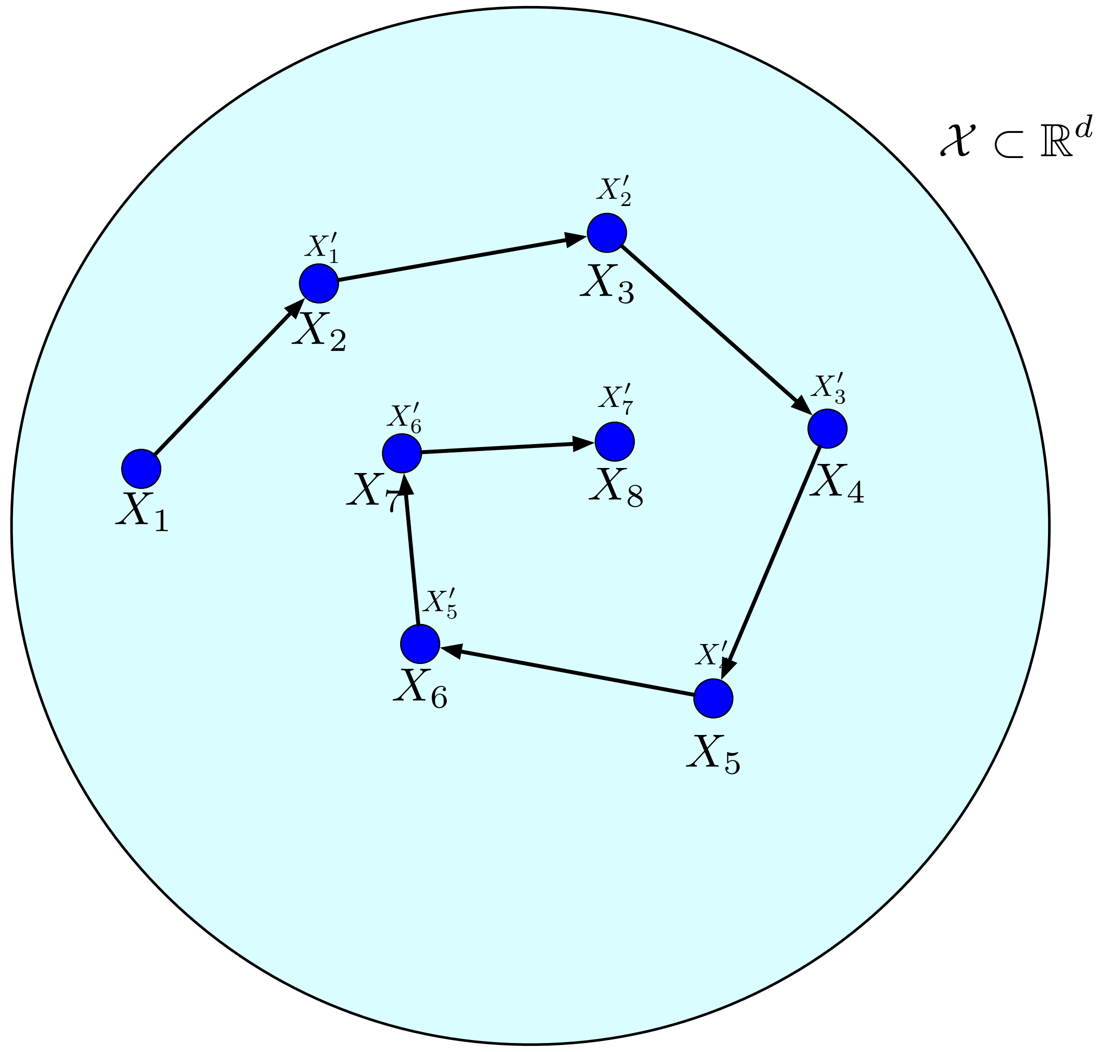
Approximate Policy Evaluation with a Data Batch

$$\mathcal{D}_n = \{(X_i, A_i, R_i, X'_i)\}_{i=1}^n$$
$$(X_i, A_i) \sim \nu$$
$$X'_i \sim \mathcal{P}(\cdot|X_i, A_i)$$
$$R_i \sim \mathcal{R}(\cdot|X_i, A_i)$$



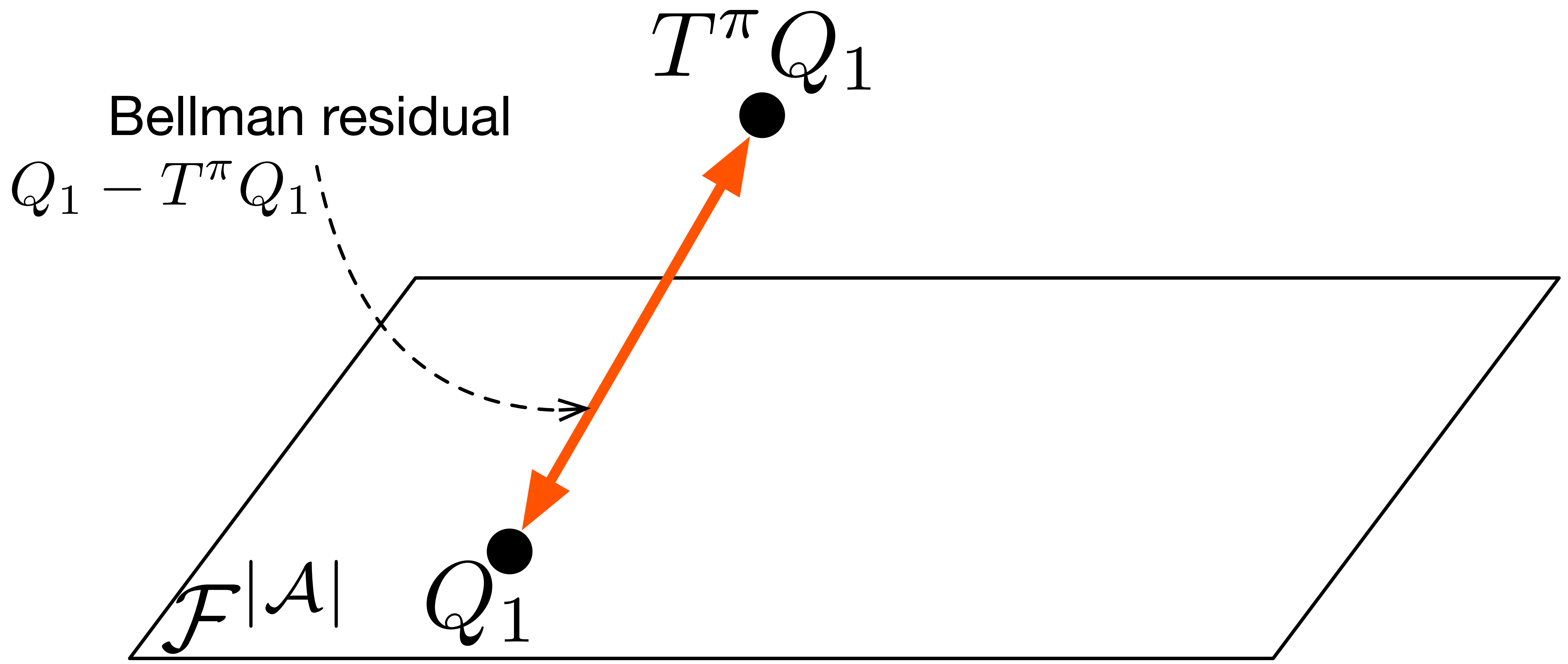
Recipe for Approximate Policy Evaluation

If we find a Q such that $T^\pi Q \approx Q$, then $Q \approx Q^\pi$. Assuming that $\mathcal{P}$ and r are known, we have some possibilities:



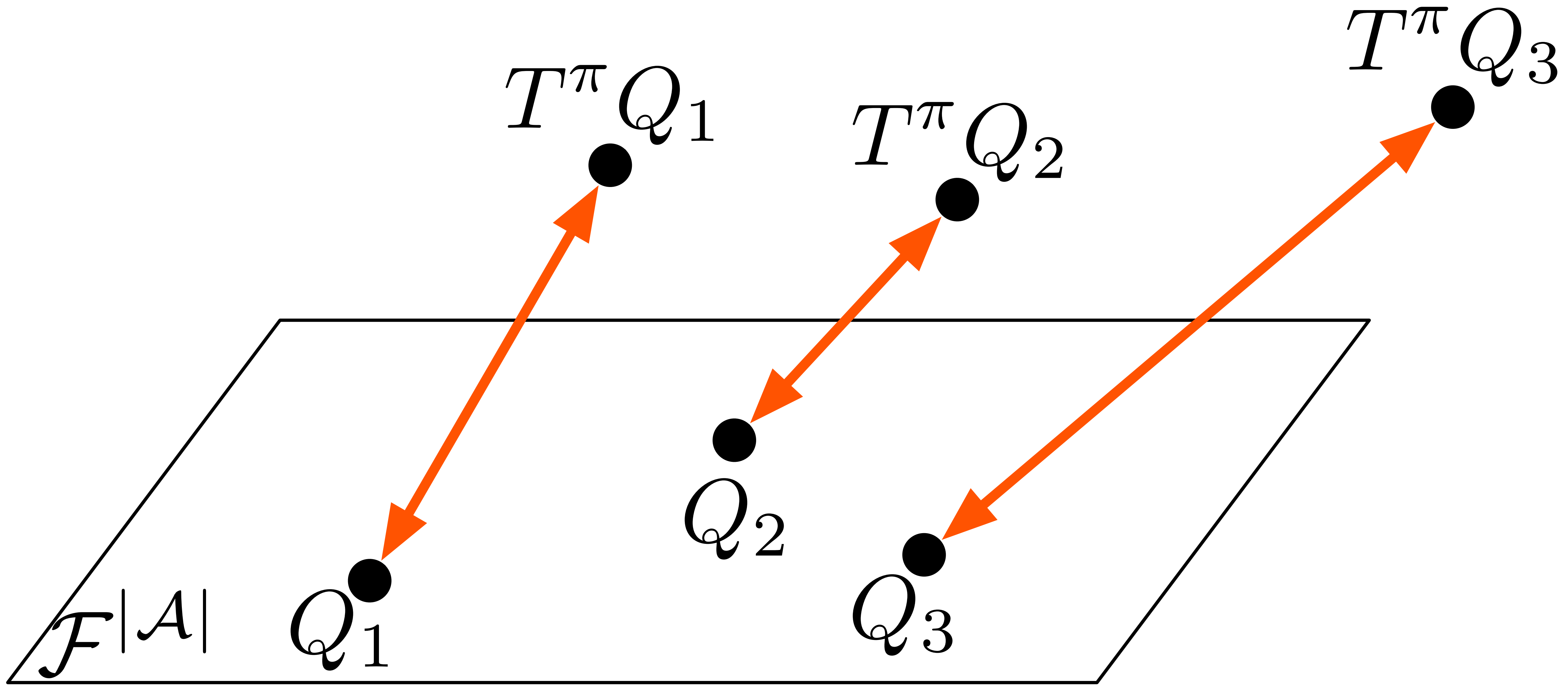
- Approximate Value Iteration: Iteratively perform $Q_{k+1} \approx T^\pi Q_k$.
- Bellman Error Minimization: Solve $\min_{Q \in \mathcal{F}^{|\mathcal{A}|}} \|Q - T^\pi Q\|$ over a representative enough function space $\mathcal{F}^{|\mathcal{A}|}$.
- Least Squares Temporal Difference (LSTD)

Bellman Residual/Error Minimization



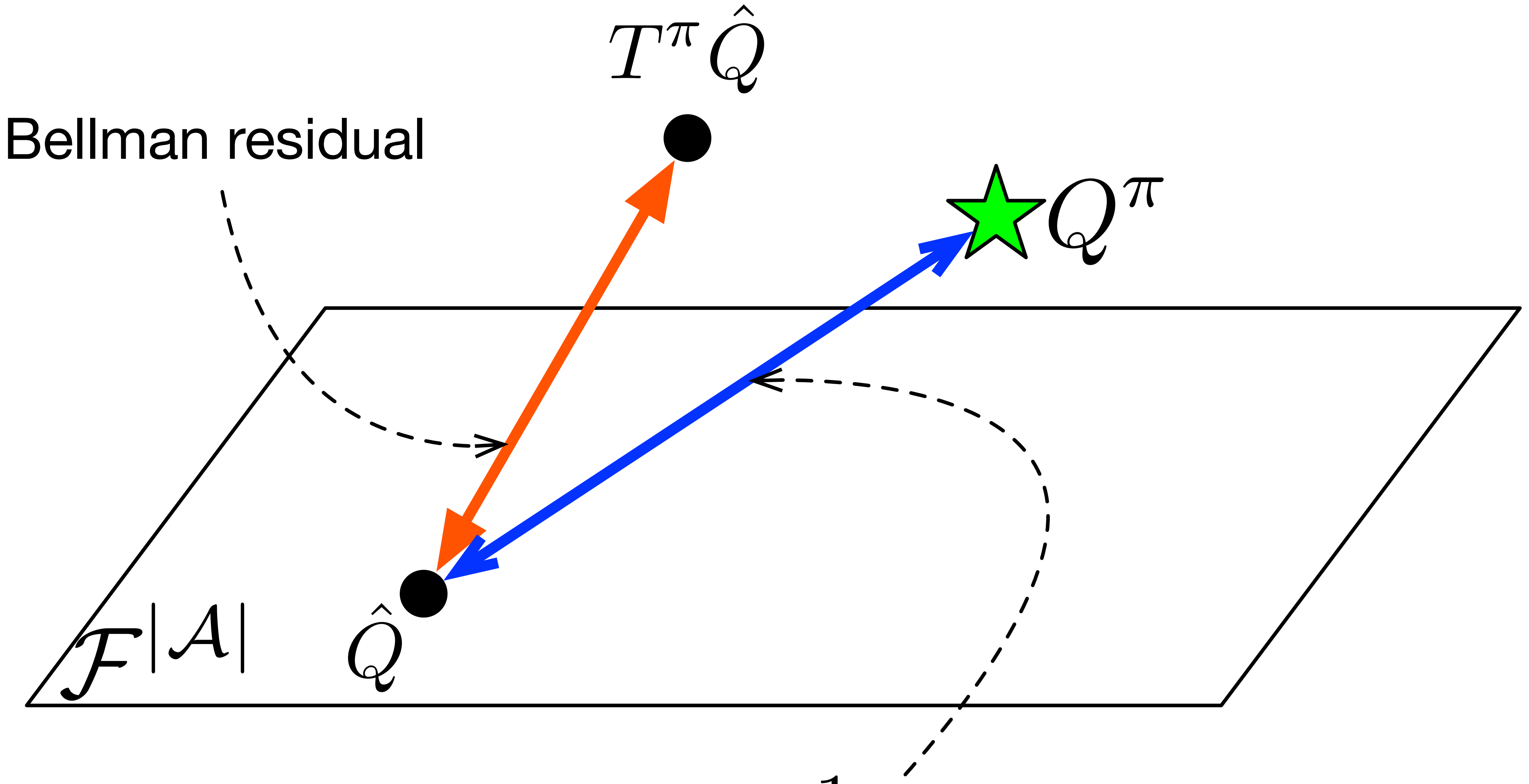
We represent the value function Q within the function space $\mathcal{F}^{|\mathcal{A}|}$. For example,
$\mathcal{F}^{|\mathcal{A}|} = \left\{ Q(x,a) = \phi^\top(x,a)w \,:\, w \in \mathbb{R}^p \right\}$.
The effect of applying T^π on a $Q \in \mathcal{F}^{|\mathcal{A}|}$ might be outside $\mathcal{F}^{|\mathcal{A}|}$.

Bellman Residual/Error Minimization



We represent the value function Q within the function space $\mathcal{F}^{|\mathcal{A}|}$. For example,
$\mathcal{F}^{|\mathcal{A}|} = \{ Q(x,a) = \phi^\top(x,a)w \,:\, w \in \mathbb{R}^p \}$.
The effect of applying T^π on a $Q \in \mathcal{F}^{|\mathcal{A}|}$ might be outside $\mathcal{F}^{|\mathcal{A}|}$.

Bellman Residual/Error Minimization (Detail)



$$\left\| Q^\pi - \hat{Q} \right\|_\infty \le \frac{1}{1-\gamma} \left\| T^\pi \hat{Q} - \hat{Q} \right\|_\infty$$

Proof

$$Q^\pi - \hat{Q} = T^\pi Q^\pi - T^\pi \hat{Q} + T^\pi \hat{Q} - \hat{Q} = \gamma \mathcal{P}^\pi \left(Q^\pi - \hat{Q} \right) + T^\pi \hat{Q} - \hat{Q}$$

$$\Rightarrow (\mathbf{I} - \gamma \mathcal{P}^\pi)(Q^\pi - \hat{Q}) = T^\pi \hat{Q} - \hat{Q}$$

$$\Rightarrow Q^\pi - \hat{Q} = (\mathbf{I} - \gamma \mathcal{P}^\pi)^{-1} \left(T^\pi \hat{Q} - \hat{Q} \right)$$

$$\Rightarrow \left\| Q^\pi - \hat{Q} \right\|_\infty \le \frac{1}{1-\gamma} \left\| T^\pi \hat{Q} - \hat{Q} \right\|_\infty$$

Bellman Residual/Error Minimization

If we find a Q such that $Q = T^{\pi} Q$, we have $Q = Q^{\pi}$. We define a loss function:

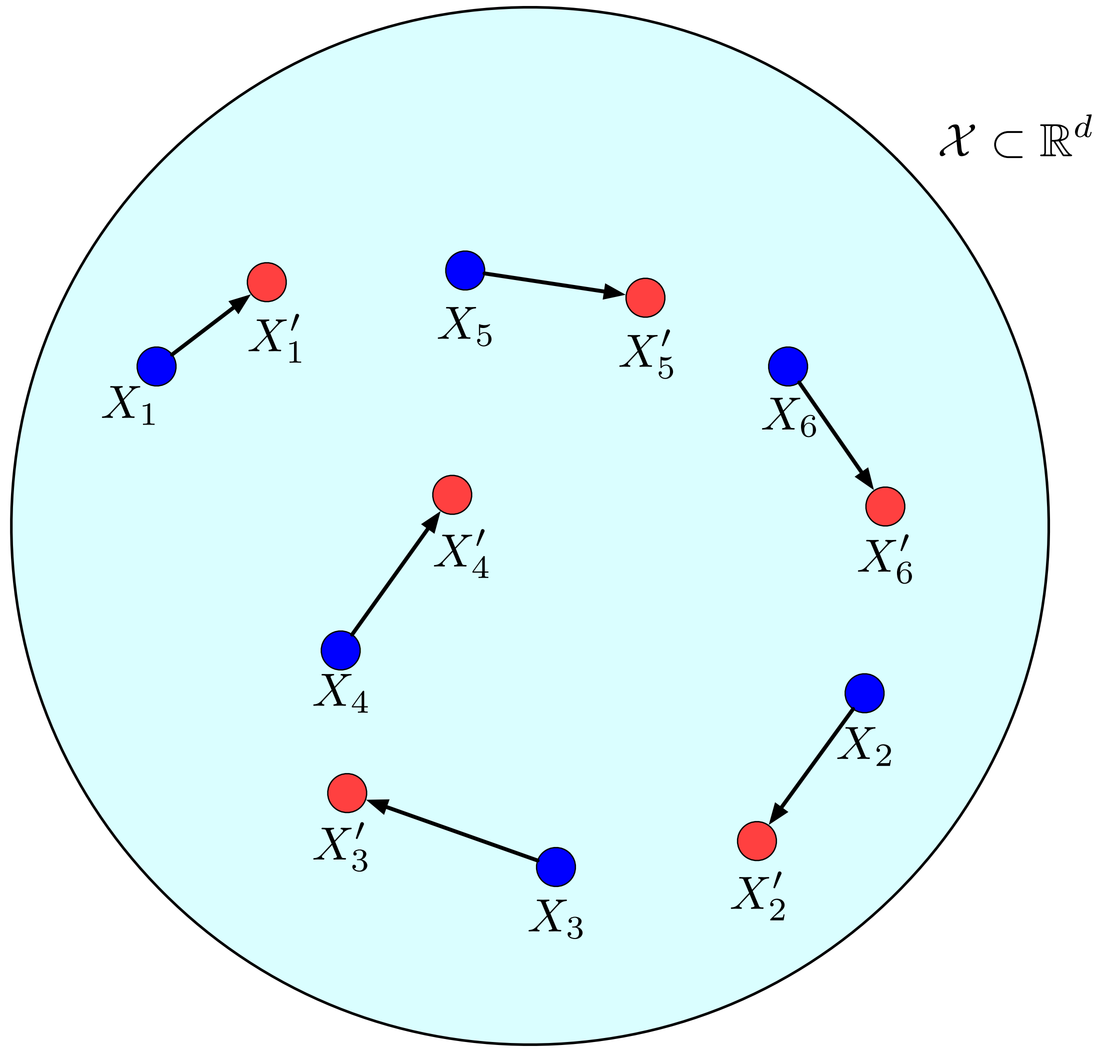
$$L_{BRM}(Q; \pi) \triangleq \|Q - T^{\pi} Q\|_{\nu}^{2} .$$

Choose a function space $\mathcal{F}^{|\mathcal{A}|}$ and solve:

$$\hat{Q} \leftarrow \operatorname*{argmin}_{Q \in \mathcal{F}^{|\mathcal{A}|}} L_{BRM}(Q; \pi).$$

Since we only have access to data $\mathcal{D}_n = \{(X_i, A_i, R_i, X_i')\}_{i=1}^{n}$ with $(X_i, A_i) \sim \nu$ and $X_i' \sim \mathcal{P}(\cdot|X_i, A_i)$ $R_i \sim \mathcal{R}(\cdot|X_i, A_i)$, it may seem reasonable to minimize the empirical loss:

$$\hat{L}_{BRM}(Q; \pi, n) \triangleq \left\| Q - \hat{T}^{\pi} Q \right\|_{\mathcal{D}_n}^{2} = \frac{1}{n} \sum_{i=1}^{n} \left[Q(X_i, A_i) - \Big(R_i + \gamma Q\big(X_i', \pi(X_i')\big) \Big) \right]^{2} .$$



Empirical Bellman Error is Biased

$$\mathbb{E}\left[\left\|Q - \hat{T}^\pi Q\right\|_{\mathcal{D}_n}^2\right] = \|Q - T^\pi Q\|_{2,\nu} + \mathbb{E}\left[|T^\pi Q - \hat{T}^\pi Q|^2\right]$$
$$\neq \|Q - T^\pi Q\|_{2,\nu}$$

- Minimizing the empirical Bellman error minimizes an objective different from the Bellman error.
- The extra bias depends on the value function. This is different from the supervised learning where tl extra bias is independent of the estimator (it would be related to the variance of the target).
- If the dynamics is deterministic, this is fine. **Otherwise, not.**
- Related to the double sampling issue.

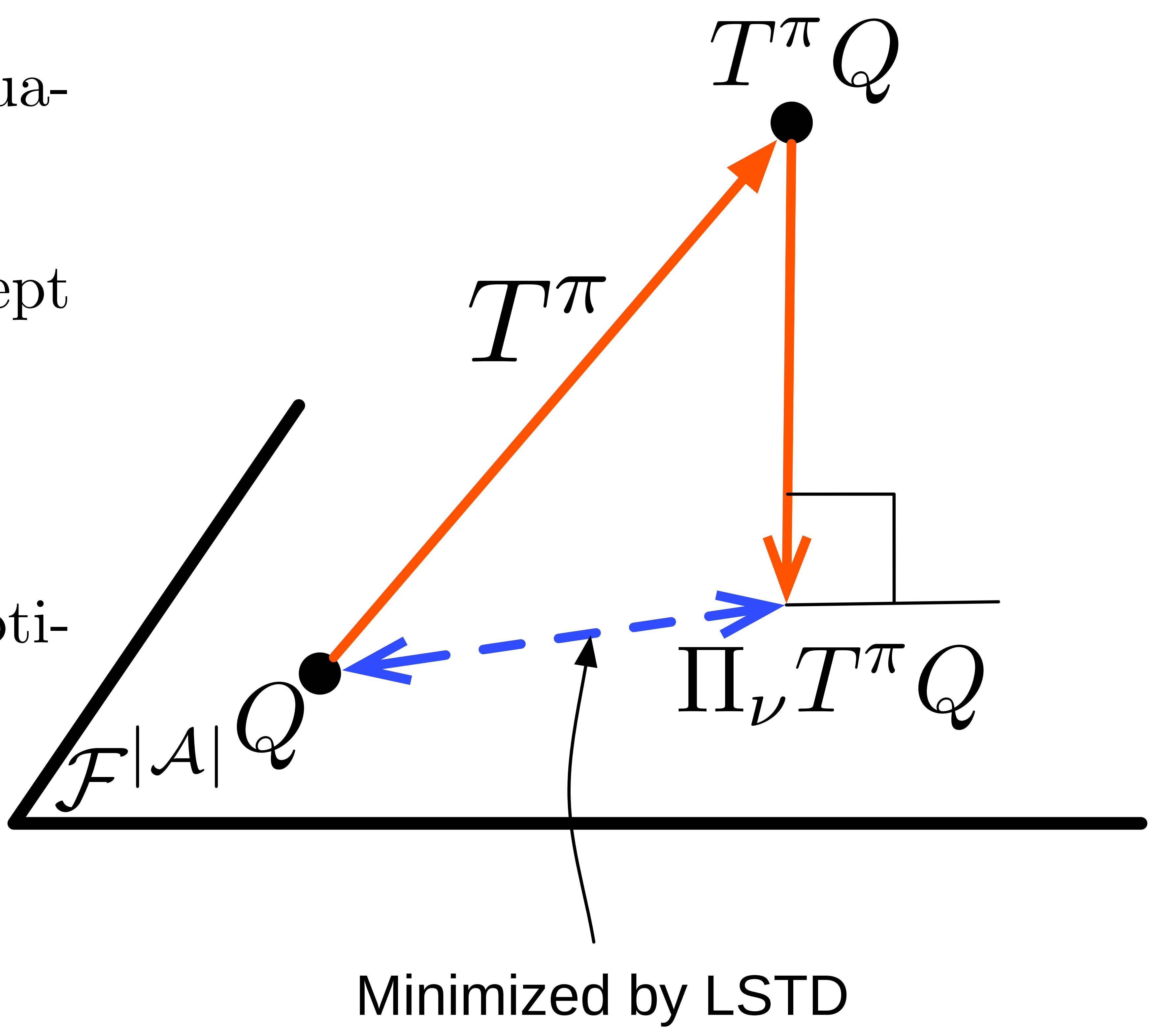
Empirical Bellman Error is Biased (Detail)

$$\begin{aligned}
\mathbb{E}\left[\left|Q(X,A) - (R + \gamma Q(X', \pi(X')))\right|^2\right] =& \mathbb{E}\left[|Q(X,A) - T^\pi Q(X,A) + T^\pi Q(X,A) - (R + \gamma Q(X', \pi(X')))|^2\right] \\
=& \mathbb{E}\left[|Q(X,A) - T^\pi Q(X,A)|^2\right] + \\
& \mathbb{E}\left[\left|T^\pi Q(X,A) - (R + \gamma Q(X', \pi(X')))\right|^2\right] + \\
& 2\mathbb{E}\left[\left(Q(X,A) - T^\pi Q(X,A)\right)\left(T^\pi Q(X,A) - (R + \gamma Q(X', \pi(X')))\right)\right] \\
=& \|Q - T^\pi Q\|_{2,\nu}^2 + \mathbb{E}\left[\left|T^\pi Q(X,A) - (R + \gamma Q(X', \pi(X')))\right|^2\right] \\
\neq& \|Q - T^\pi Q\|_{2,\nu}^2
\end{aligned}$$

$$\begin{aligned}
&\mathbb{E}\left[\left(Q(X,A) - T^\pi Q(X,A)\right)\left(T^\pi Q(X,A) - (R + \gamma Q(X', \pi(X')))\right)\right] = \\
&\mathbb{E}_{X,A}\left[\mathbb{E}_{X'}\left[\left(Q(X,A) - T^\pi Q(X,A)\right)\left(T^\pi Q(X,A) - (R + \gamma Q(X', \pi(X')))\right)\middle| X, A\right]\right] = \\
&\mathbb{E}_{X,A}\left[\left(Q(X,A) - T^\pi Q(X,A)\right)\mathbb{E}_{X'}\left[T^\pi Q(X,A) - (R + \gamma Q(X', \pi(X')))\middle| X, A\right]\right] = \\
&\mathbb{E}_{X,A}\left[\left(Q(X,A) - T^\pi Q(X,A)\right)\left(T^\pi Q(X,A) - \underbrace{\mathbb{E}_{X'}\left[R + \gamma Q(X', \pi(X'))\middle| X, A\right]}_{=T^\pi Q(X,A)}\right)\right] = 0.
\end{aligned}$$

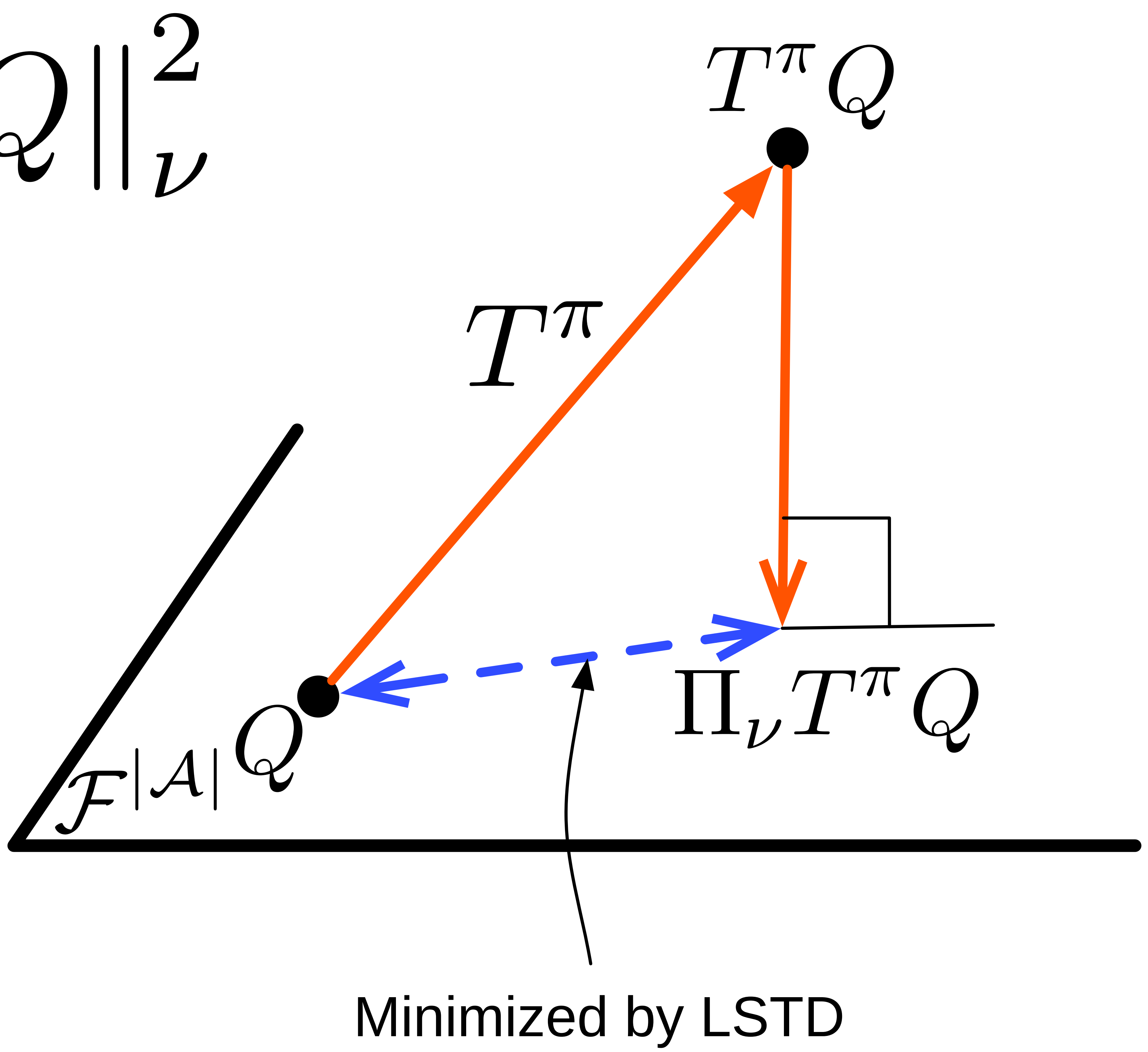
Least Squares Temporal Difference (LSTD)

- The original formulation of LSTD finds a solution to the fixed-point equation $Q = \Pi_\nu T^\pi Q$, where $\Pi_\nu Q = \Pi_{\nu,\mathcal{F}^{|\mathcal{A}|}} Q \operatorname{argmin}_{h \in \mathcal{F}^{|\mathcal{A}|}} \|h - Q\|_\nu^2$.
- The operator $\Pi_\nu T^\pi$ is not a contraction for arbitrary choice of ν (except when ν is the stationary distribution induced by π).
- Instead: Find the minimizer of $\|Q - \Pi_\nu T^\pi Q\|_\nu^2$.
- Whenever ν is the stationary distribution of π, the solution of this optimization problem is the same as the fixed-point of $Q = \Pi_\nu T^\pi Q$.



Least Squares Temporal Difference (LSTD)

$$\min_{Q \in \mathcal{F}^{|\mathcal{A}|}} \left\| Q - \Pi_\nu T^\pi Q \right\|_\nu^2$$



Or equivalently:

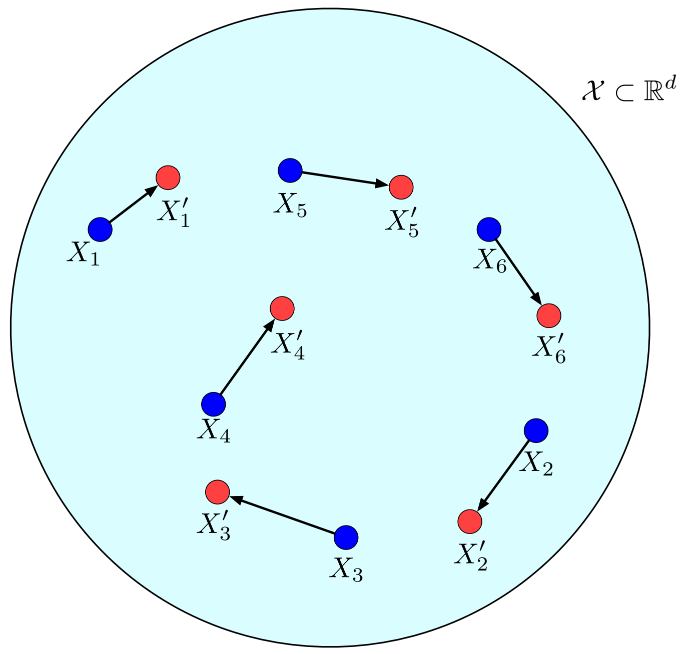
$$h(\cdot; Q) = \operatorname*{argmin}_{h' \in \mathcal{F}^{|\mathcal{A}|}} \left\| h' - T^\pi Q \right\|_\nu^2 ,$$

$$Q_{LSTD} = \operatorname*{argmin}_{Q \in \mathcal{F}^{|\mathcal{A}|}} \left\| Q - h(\cdot; Q) \right\|_\nu^2 ,$$

Empirical Version:

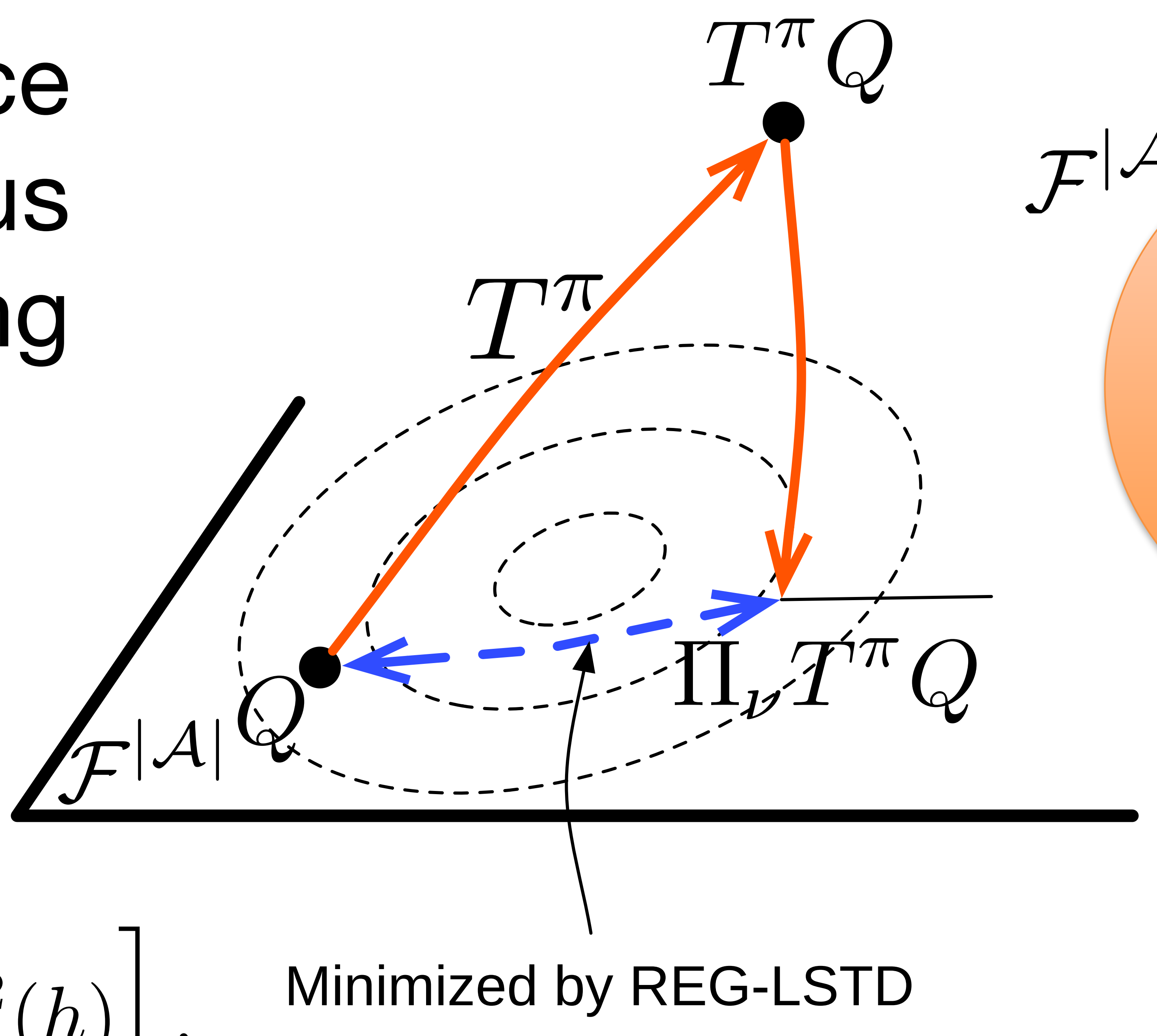
$$\hat{h}_n(\cdot; Q) = \operatorname*{argmin}_{h \in \mathcal{F}^{|\mathcal{A}|}} \left\| h - \hat{T}^\pi Q \right\|_{\mathcal{D}_n}^2 = \operatorname*{argmin}_{h \in \mathcal{F}^{|\mathcal{A}|}} \frac{1}{n} \sum_{i=1}^{n} \left| h(X_i, A_i) - (R_i + \gamma Q(X_i', \pi(X_i'))) \right|^2$$

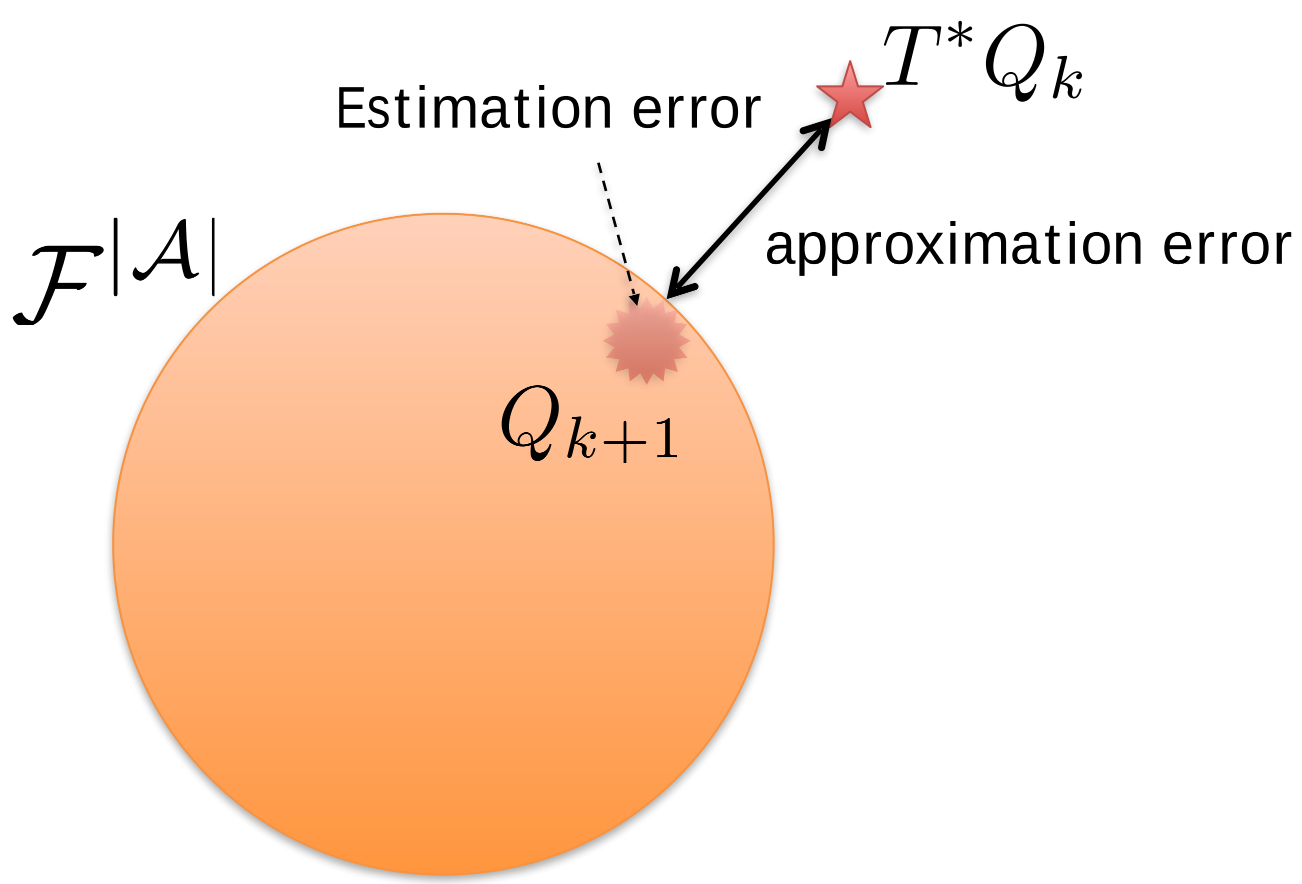
$$\hat{Q}_{LSTD} = \operatorname*{argmin}_{Q \in \mathcal{F}^{|\mathcal{A}|}} \left\| Q - \hat{h}_n(\cdot; Q) \right\|_{\mathcal{D}_n}^2 = \operatorname*{argmin}_{Q \in \mathcal{F}^{|\mathcal{A}|}} \frac{1}{n} \sum_{i=1}^{n} \left| Q(X_i, A_i) - \hat{h}_n(X_i, A_i; Q) \right|^2$$



Regularized LSTD

Main Idea: Start from a large function space (e.g., dense in the space of continuous functions) and control its complexity using regularization.



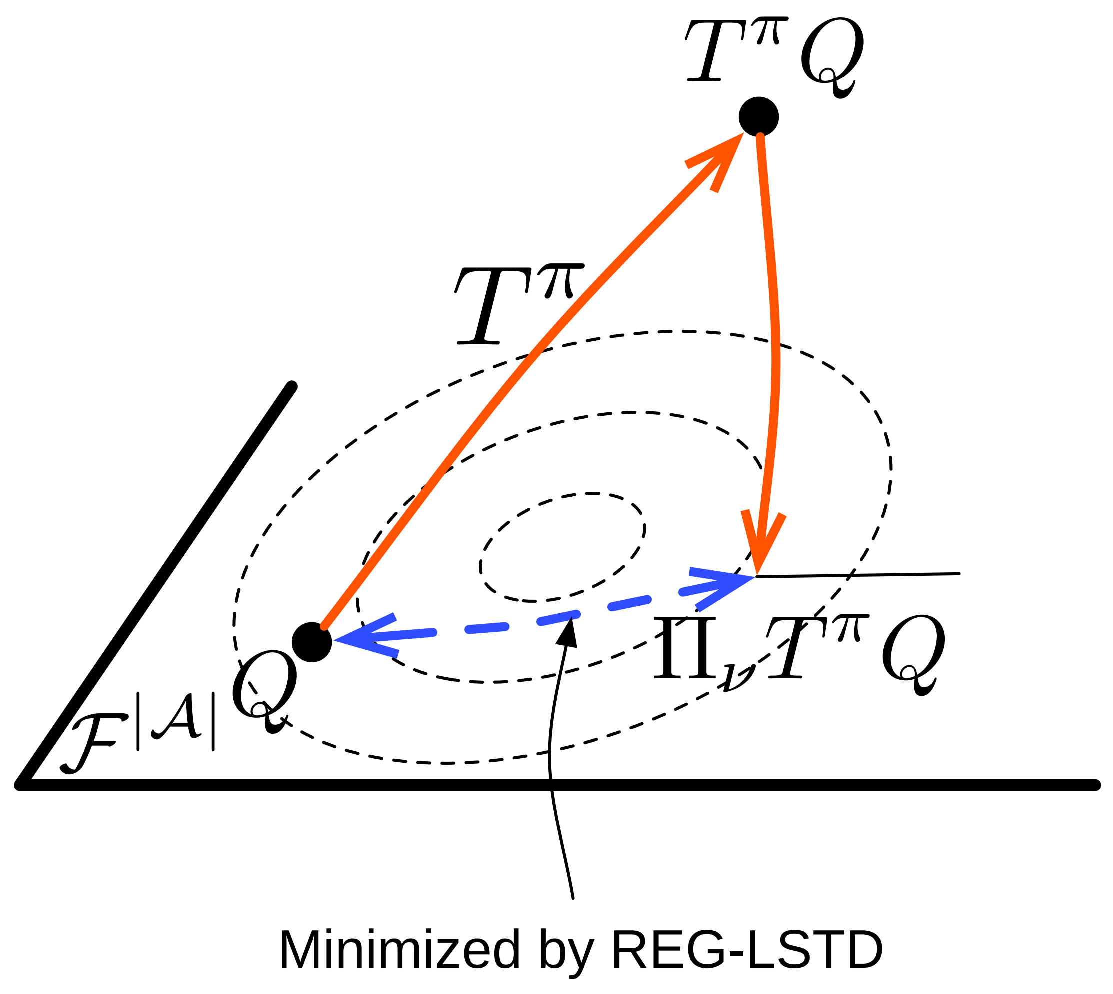


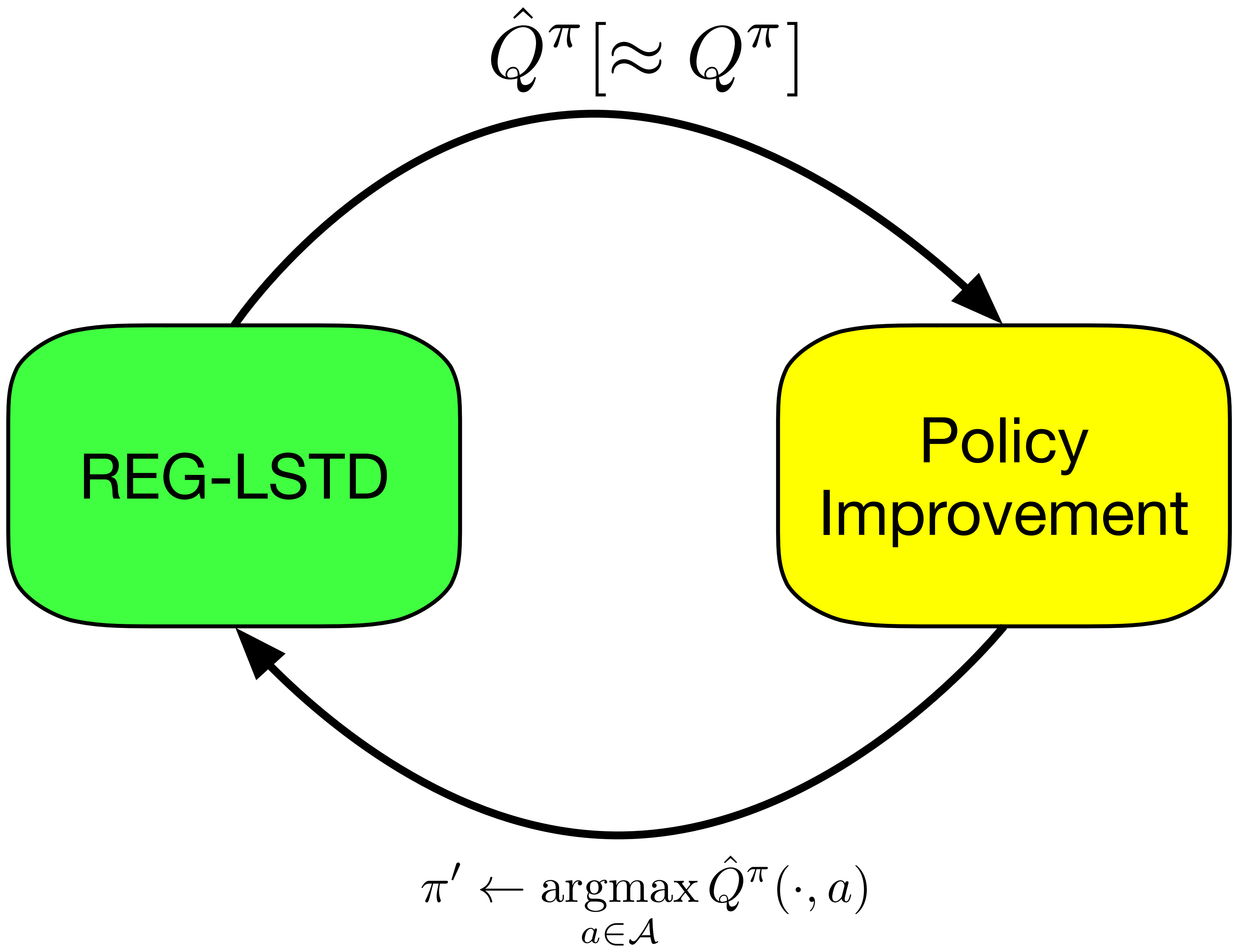
$$\hat{h}_n(\cdot; Q) = \operatorname*{argmin}_{h \in \mathcal{F}^{|\mathcal{A}|}} \left[\left\| h - \hat{T}^{\pi_k} Q \right\|_{\mathcal{D}_n}^2 + \lambda_{h,n}^{(k)} J^2(h) \right],$$

$$\hat{Q}^{(k)} = \operatorname*{argmin}_{Q \in \mathcal{F}^{|\mathcal{A}|}} \left[\left\| Q - \hat{h}_n(\cdot; Q) \right\|_{\mathcal{D}_n}^2 + \lambda_{Q,n}^{(k)} J^2(Q) \right].$$

where $J : \mathcal{F}^{|\mathcal{A}|} \to \mathbb{R}$ is the regularizer and $\lambda_{h,n}^{(k)}, \lambda_{Q,n}^{(k)} > 0$ are regularization coefficients. The regularizer can be any pseudo-norm defined on $\mathcal{F}^{|\mathcal{A}|}$, e.g., $J^2(\cdot) = \|\cdot\|_{\mathcal{H}}$ for an RKHS $\mathcal{F}^{|\mathcal{A}|} = \mathcal{H}$.

(Regularized) Least Squares Policy Iteration (LSPI)





Theoretical Analysis

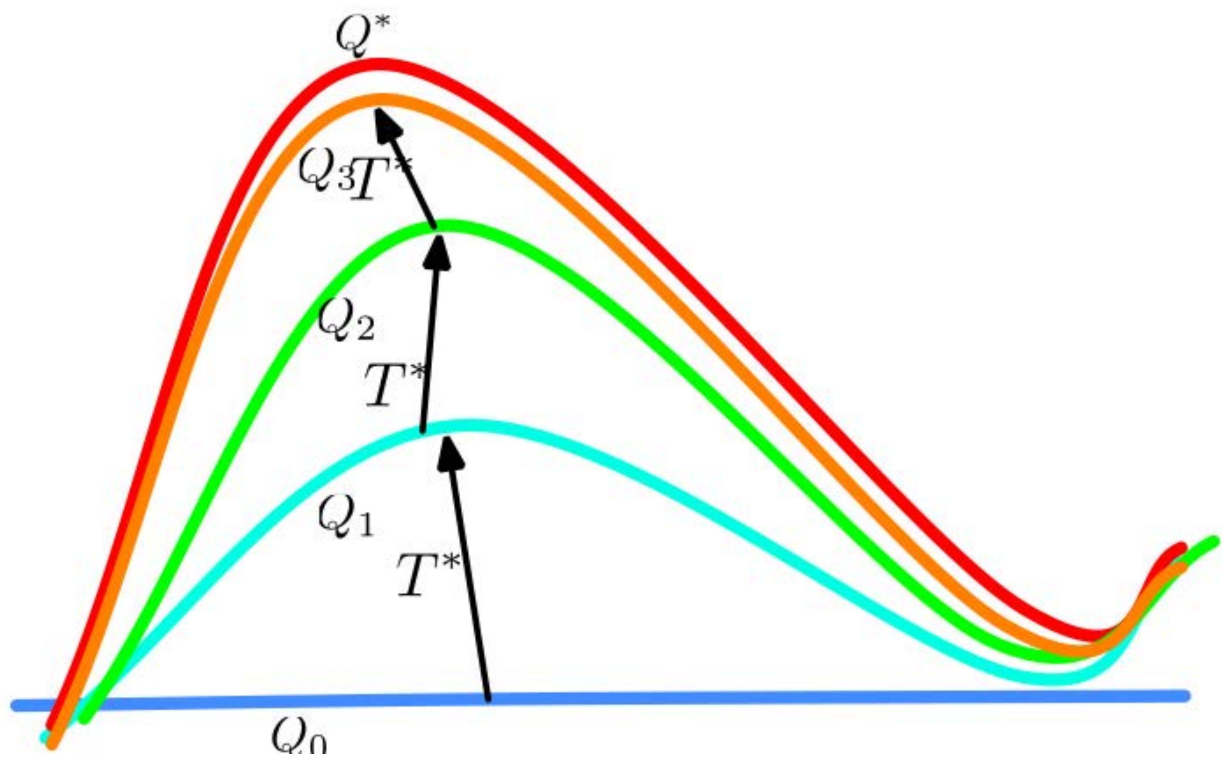
Why Theory?

- Soundness of algorithm
  - Does it even work after feeding it a lot of data? Conditions when it works?
- Efficiency of algorithm
  - How fast can it learn? Does it learn faster if the problem is easier? Is it adaptive to the difficulty of the problem?
- The limit of what/how fast can be learned
  - Can we learn any function if we have enough data?
- Design of new algorithms

Two-Part Analysis

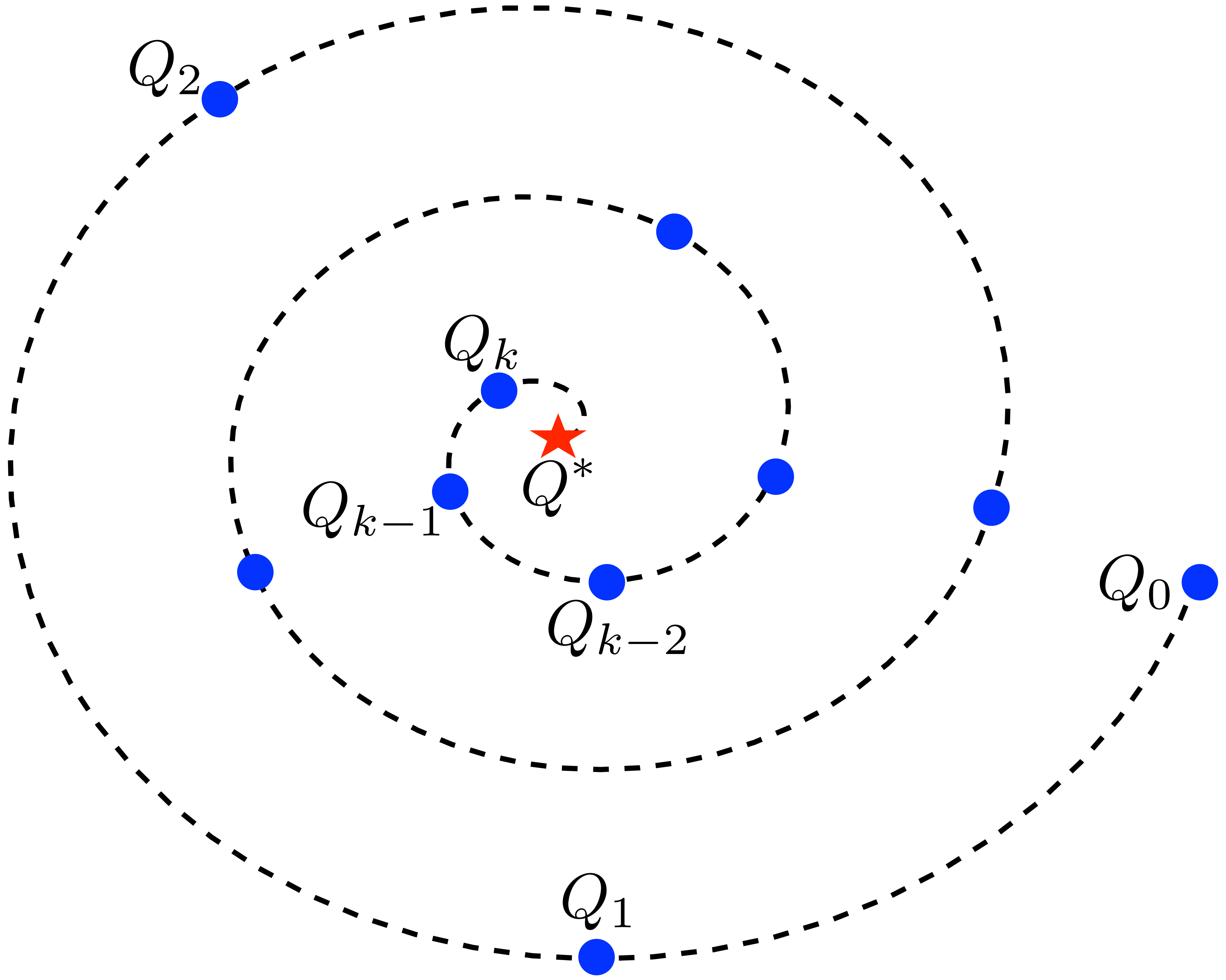
- [ ] Statistical analysis of each step
- [ ] Error Propagation

Brief Analysis of AVI



$$Q_{k+1} \leftarrow T^* Q_k$$

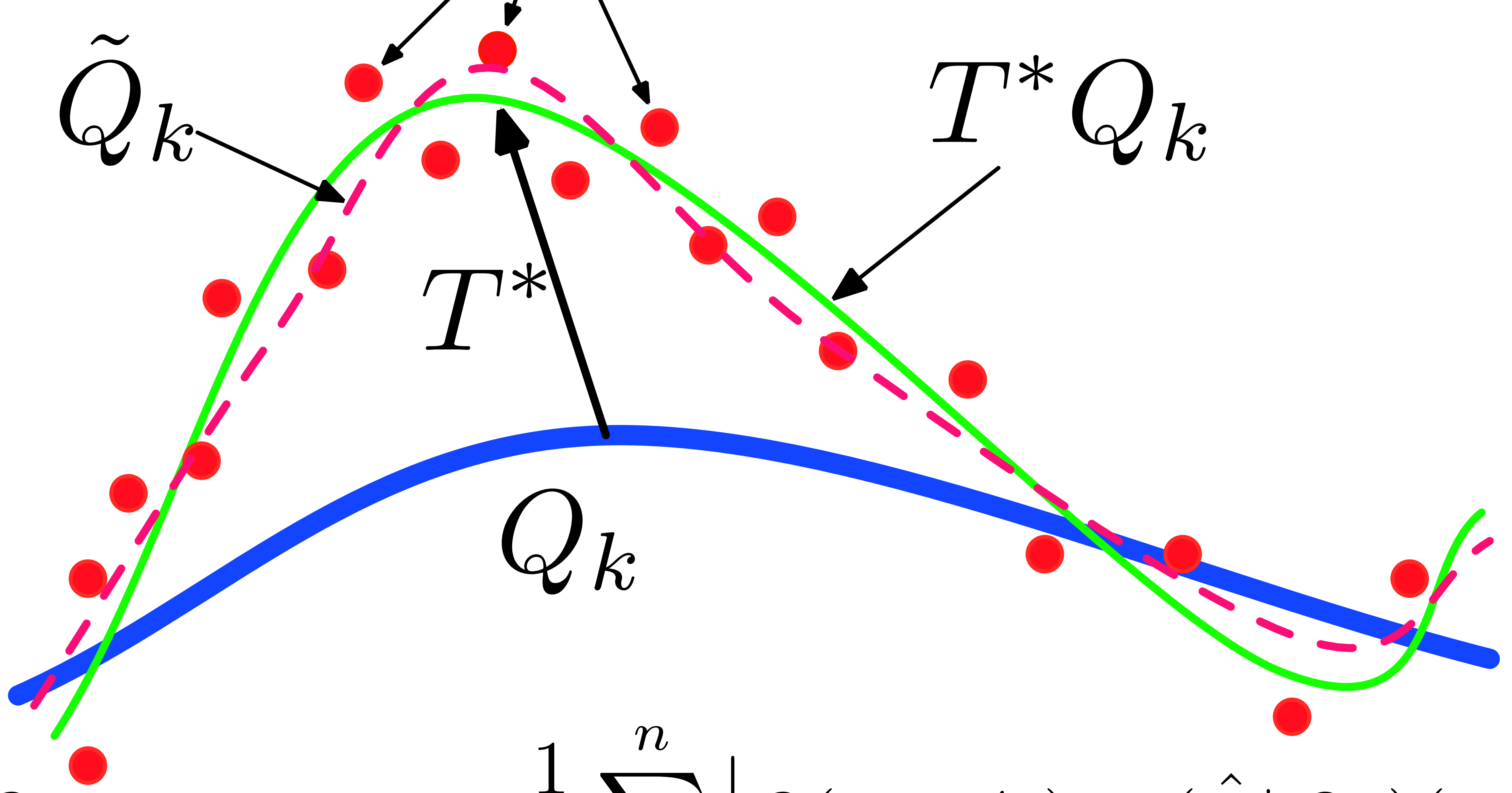
Contraction Property



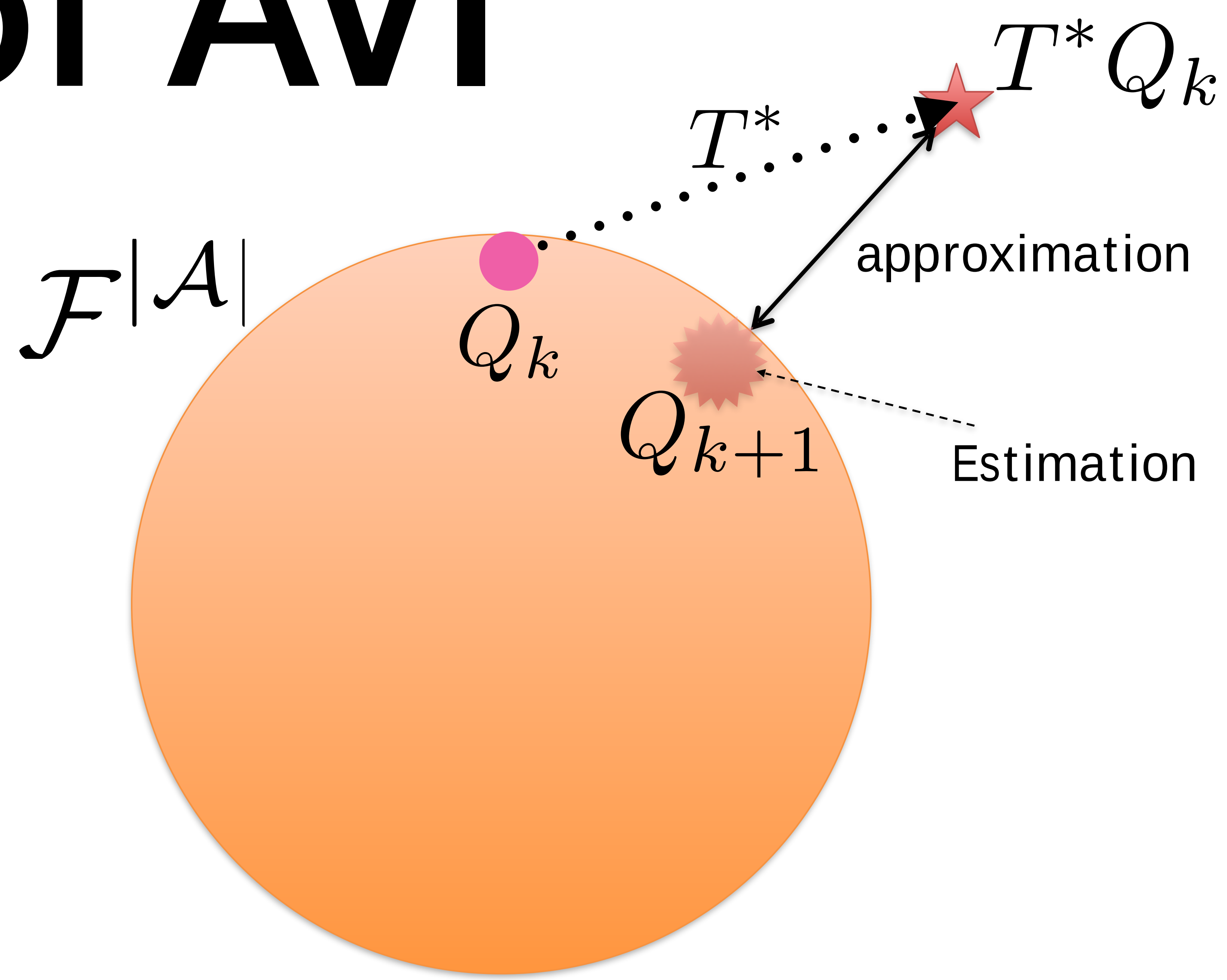
$$\|Q_k - Q^*\|_\infty \leq \gamma^k \|Q_0 - Q^*\|_\infty .$$

Brief Analysis of AVI

$$(\hat{T}^* Q_k)(X_i, A_i) \triangleq R_i + \gamma \max_{a' \in \mathcal{A}} Q_k(X_i', a')$$



$$Q_{k+1} \leftarrow \operatorname*{argmin}_{Q \in \mathcal{F}^{|\mathcal{A}|}} \frac{1}{n} \sum_{i=1}^{n} \left| Q(X_i, A_i) - (\hat{T}^* Q_k)(X_i, A_i) \right|^2$$



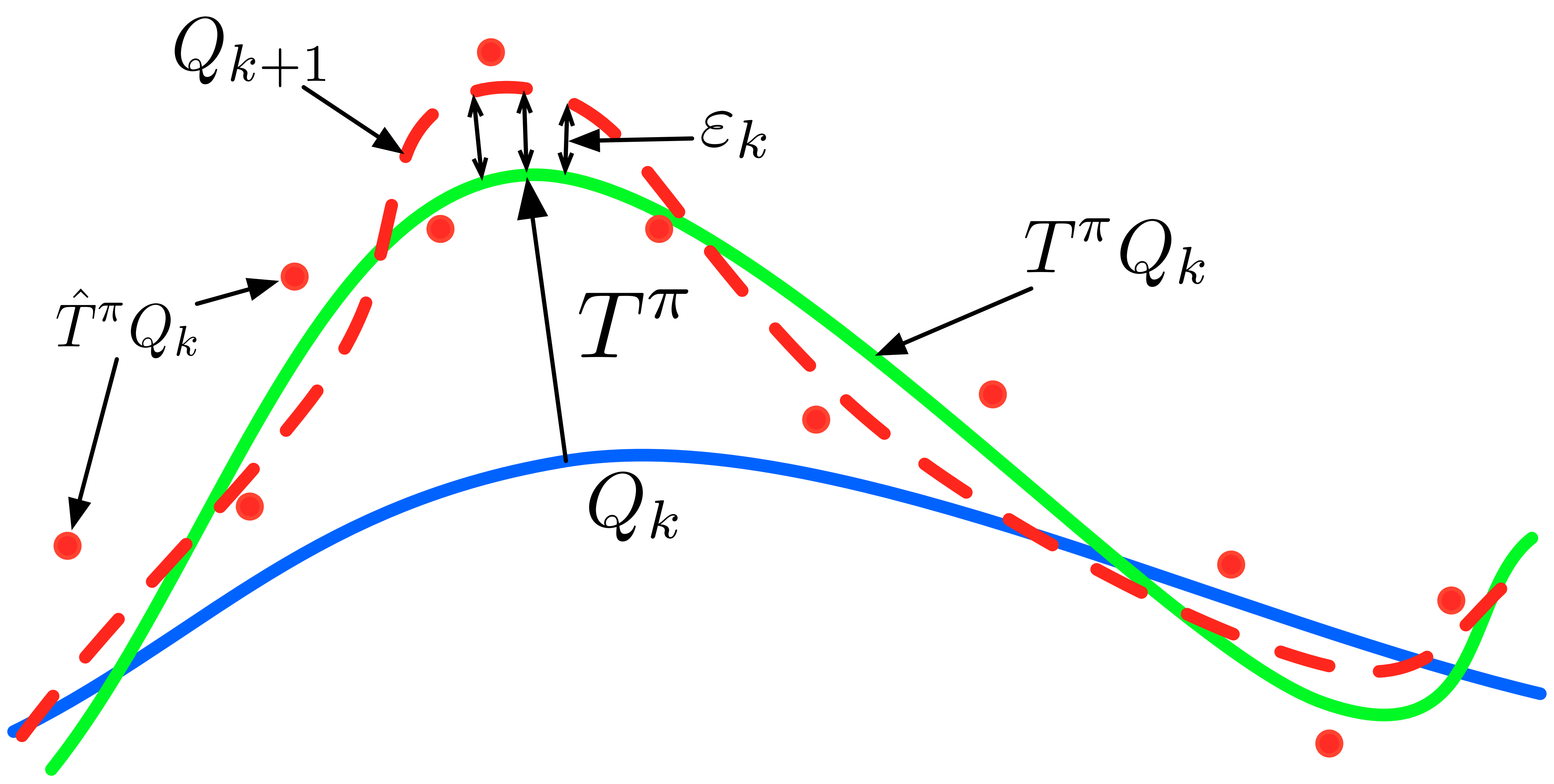
$$\begin{aligned} Q_{k+1} &\leftarrow \operatorname*{argmin}_{Q \in \mathcal{F}^{|\mathcal{A}|}} \frac{1}{n} \sum_{i=1}^{n} \left| Q(X_i, A_i) - (\hat{T}^* Q_k)(X_i, A_i) \right|^2 \\ &= \Pi_{\nu_n, \mathcal{F}^{|\mathcal{A}|}} \hat{T}^* Q_k \\ &\approx \Pi_{\nu, \mathcal{F}^{|\mathcal{A}|}} T^* Q_k \end{aligned}$$

Not a contraction operator

Brief Analysis of AVI - Error Propagation (Policy Evaluation)

$$\varepsilon_k \triangleq T^\pi Q_k - Q_{k+1}$$

Error in approximation



$$Q_{k+1} \leftarrow \operatorname*{argmin}_{Q \in \mathcal{F}^{|\mathcal{A}|}} \frac{1}{n} \sum_{i=1}^{n} \left| Q(X_i, A_i) - (\hat{T}^* Q_k)(X_i, A_i) \right|^2$$

$$(\hat{T}^\pi Q_k)(X_i, A_i) \triangleq R_i + \gamma Q_k(X_i', \pi(X_i'))$$

$$\begin{aligned} Q^\pi - Q_{k+1} &= T^\pi Q^\pi - T^\pi Q_k + \varepsilon_k \\ &= \gamma \mathcal{P}^\pi (Q^\pi - Q_k) + \varepsilon_k \end{aligned}$$

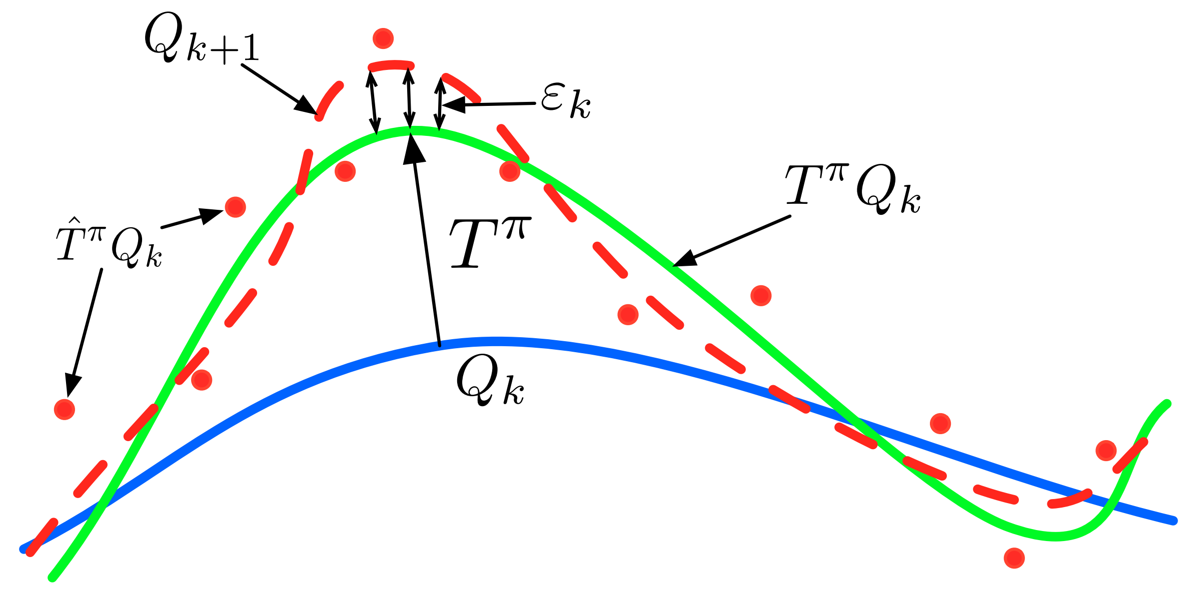
$$\cdots \Rightarrow Q^\pi - Q_K = \sum_{k=0}^{K-1} (\gamma \mathcal{P}^\pi)^{K-1-k} \varepsilon_k + (\gamma \mathcal{P}^\pi)^K (Q^\pi - Q_0)$$

Brief Analysis of AVI (Policy Evaluation)

$$\varepsilon_k \triangleq T^\pi Q_k - Q_{k+1}$$

$$\begin{aligned} Q^\pi - Q_{k+1} &= T^\pi Q^\pi - T^\pi Q_k + \varepsilon_k \\ &= \gamma \mathcal{P}^\pi (Q^\pi - Q_k) + \varepsilon_k \end{aligned}$$

$$\cdots \Rightarrow Q^\pi - Q_K = \sum_{k=0}^{K-1} (\gamma \mathcal{P}^\pi)^{K-1-k} \varepsilon_k + (\gamma \mathcal{P}^\pi)^K (Q^\pi - Q_0)$$



$$Q_{k+1} \leftarrow \operatorname*{argmin}_{Q \in \mathcal{F}^{|\mathcal{A}|}} \frac{1}{n} \sum_{i=1}^{n} \left| Q(X_i, A_i) - (\hat{T}^* Q_k)(X_i, A_i) \right|^2$$

$$(\hat{T}^\pi Q_k)(X_i, A_i) \triangleq R_i + \gamma Q_k(X_i', \pi(X_i'))$$

- This equality relates the errors ε_k at each iteration of AVI (Policy Evaluation case) to the quality of policy evaluation, i.e., how close Q_K to Q^π is.
- If we make no errors ($\varepsilon_k = 0$), $Q^\pi - Q_K = \gamma^K (\mathcal{P}^\pi)^K (Q^\pi - Q_0)$. The error decays as fast as γ^K. Taking the supremum norm of both sides, we get

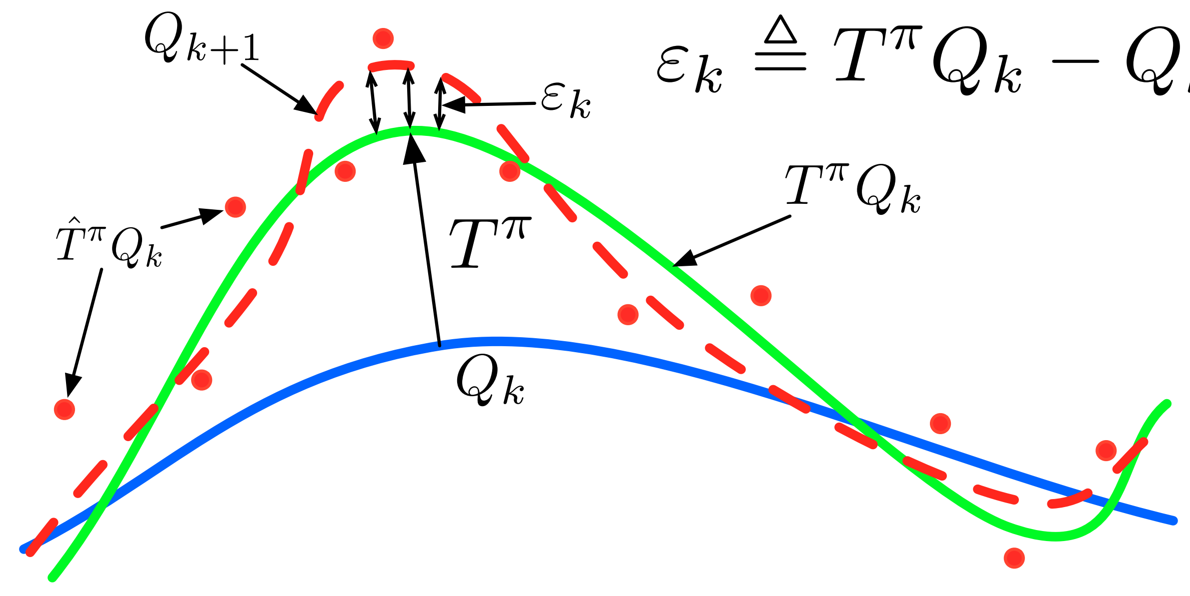
$$\|Q^\pi - Q_K\|_\infty \leq \gamma^K \|Q^\pi - Q_0\|_\infty .$$

- The supremum norm is conservative. Machine learning methods do not often minimize it; they minimize the L_2-norm or alike.
- Can we have a tighter L_p-based upper bound?

Brief Analysis of AVI (Policy Evaluation)

Consider a distribution ρ. We have:

$$\rho\left|Q^\pi - Q_K\right| \leq \rho\left|\sum_{k=0}^{K-1}\left(\gamma\mathcal{P}^\pi\right)^{K-1-k}\varepsilon_k\right| + \rho\left|\left(\gamma\mathcal{P}^\pi\right)^K\left(Q^\pi - Q_0\right)\right|$$
$$\leq \sum_{k=0}^{K-1}\gamma^{K-1-k}\rho\left(\mathcal{P}^\pi\right)^{K-1-k}\left|\varepsilon_k\right| + \rho\left(\gamma\mathcal{P}^\pi\right)^K\left|Q^\pi - Q_0\right|.$$



$$\varepsilon_k \triangleq T^\pi Q_k - Q_{k+1}$$

$$Q_{k+1} \leftarrow \underset{Q\in\mathcal{F}^{|\mathcal{A}|}}{\operatorname{argmin}}\frac{1}{n}\sum_{i=1}^{n}\left|Q(X_i, A_i) - (\hat{T}^* Q_k)(X_i, A_i)\right|^2$$

$$(\hat{T}^\pi Q_k)(X_i, A_i) \triangleq R_i + \gamma Q_k(X_i', \pi(X_i'))$$

Suppose that $\rho = \rho^\pi$ is the stationary distribution of policy π, i.e., $\rho^\pi\mathcal{P}^\pi = \rho^\pi$. This implies that $\rho^\pi(\mathcal{P}^\pi)^k = \rho^\pi$. So we can simplify

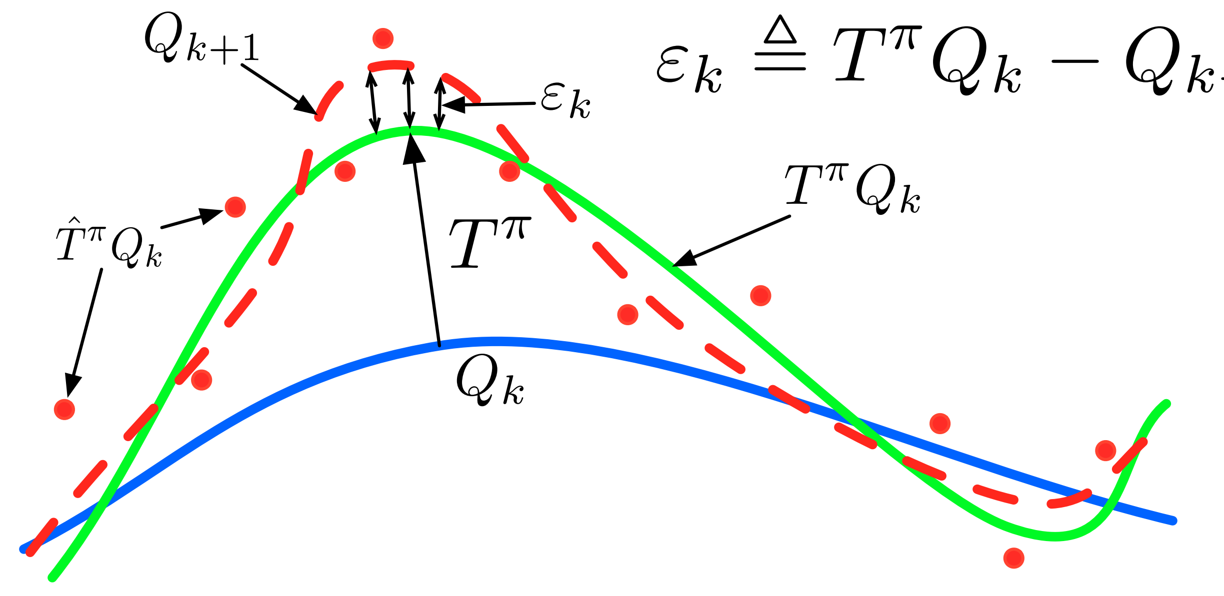
$$\begin{aligned} Q^\pi - Q_{k+1} &= T^\pi Q^\pi - T^\pi Q_k + \varepsilon_k \\ &= \gamma\mathcal{P}^\pi(Q^\pi - Q_k) + \varepsilon_k \\ \cdots \Rightarrow Q^\pi - Q_K &= \sum_{k=0}^{K-1}\left(\gamma\mathcal{P}^\pi\right)^{K-1-k}\varepsilon_k + \left(\gamma\mathcal{P}^\pi\right)^K\left(Q^\pi - Q_0\right) \end{aligned}$$

$$\rho^\pi\left|Q^\pi - Q_K\right| \leq \sum_{k=0}^{K-1}\gamma^{K-1-k}\rho^\pi\left|\varepsilon_k\right| + \gamma^K\rho^\pi\left|Q^\pi - Q_0\right|.$$

Denote $\|f(X)\|_{p,\rho} = [\int f^p(x)\mathrm{d}\rho]^{1/p}$. Using the fact that $\|f(X)\|_{1,\rho} \leq \|f(X)\|_{2,\rho}$ (from Jensen's inequality), we get

$$\left\|Q^\pi - Q_K\right\|_{1,\rho^\pi} \leq \sum_{k=0}^{K-1}\gamma^{K-1-k}\left\|\varepsilon_k\right\|_{2,\rho^\pi} + \gamma^K\left\|Q^\pi - Q_0\right\|_{2,\rho^\pi}.$$

Brief Analysis of AVI (Policy Evaluation)



$$\varepsilon_k \triangleq T^\pi Q_k - Q_{k+1}$$

$$Q_{k+1} \leftarrow \underset{Q \in \mathcal{F}^{|\mathcal{A}|}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \left| Q(X_i, A_i) - (\hat{T}^* Q_k)(X_i, A_i) \right|^2$$

$$(\hat{T}^\pi Q_k)(X_i, A_i) \triangleq R_i + \gamma Q_k(X_i', \pi(X_i'))$$

$$\|Q^\pi - Q_K\|_{1,\rho^\pi} \leq \sum_{k=0}^{K-1} \gamma^{K-1-k} \|\varepsilon_k\|_{2,\rho^\pi} + \gamma^K \|Q^\pi - Q_0\|_{2,\rho^\pi}$$

Geometric decay **Regression error**

- If the quality of policy evaluation is weighed according to the stationary distribution ρ^π, the regression error is also measured according to the same distribution. This is the on-policy sampling scenario.
- Not all iterations are the same. Earlier errors will be forgotten.
- It remains to provide an upper bound on $\|\varepsilon_k\|_{2,\rho^\pi}$. What is the relation between the number of samples n, the properties of MDP, and this norm? This is the statistical analysis.

Statistical Analysis of (Batch) RL

- Differences with the usual supervised learning setting
  - Moving target (for AVI)
  - Not a standard objective function (API - Policy Evaluation using BRM or LSTD)
  - Dependent data
  - Off-policy and distribution mismatch
- One can still obtain fast convergence rates, e.g., for REG-LSTD and RFQI

Statistical Guarantee for REG-LSTD

Theorem (REG-LSTD). For any fixed policy π, let $\hat{Q}$ be the solution to the REG-LSTD optimization problem with the choice of $\lambda_{h,n} = \lambda_{Q,n} = \left[\frac{1}{n\, J^2(Q^\pi)}\right]^{\frac{1}{1+\alpha}}$. Under certain conditions, there exists $c(\delta) > 0$ such that for any $n \in \mathbb{N}$ and $0 < \delta < 1$, we have

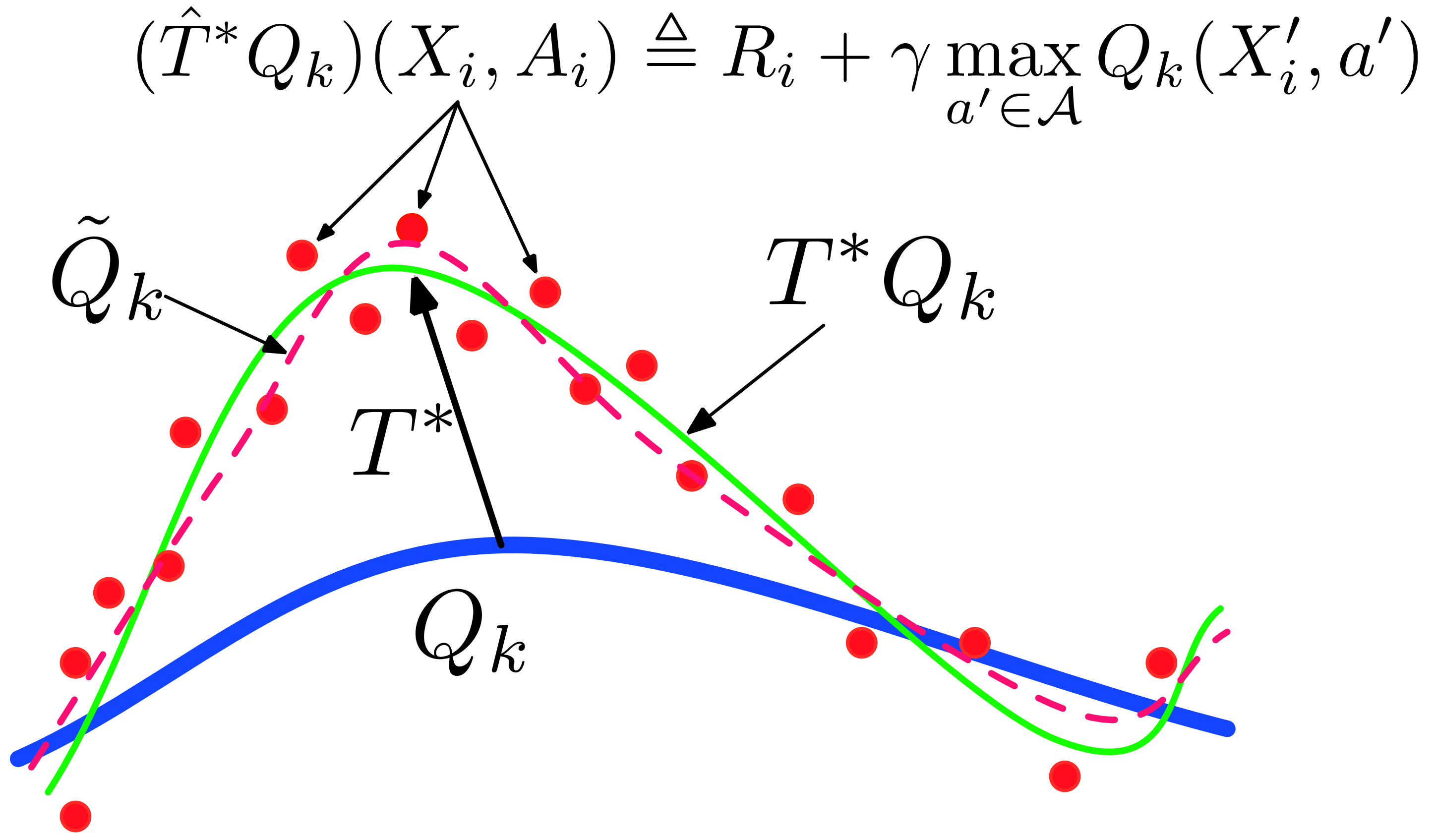
$$\left\| \hat{Q} - T^\pi \hat{Q} \right\|_\nu^2 \leq c(\delta)\, n^{-\frac{1}{1+\alpha}},$$

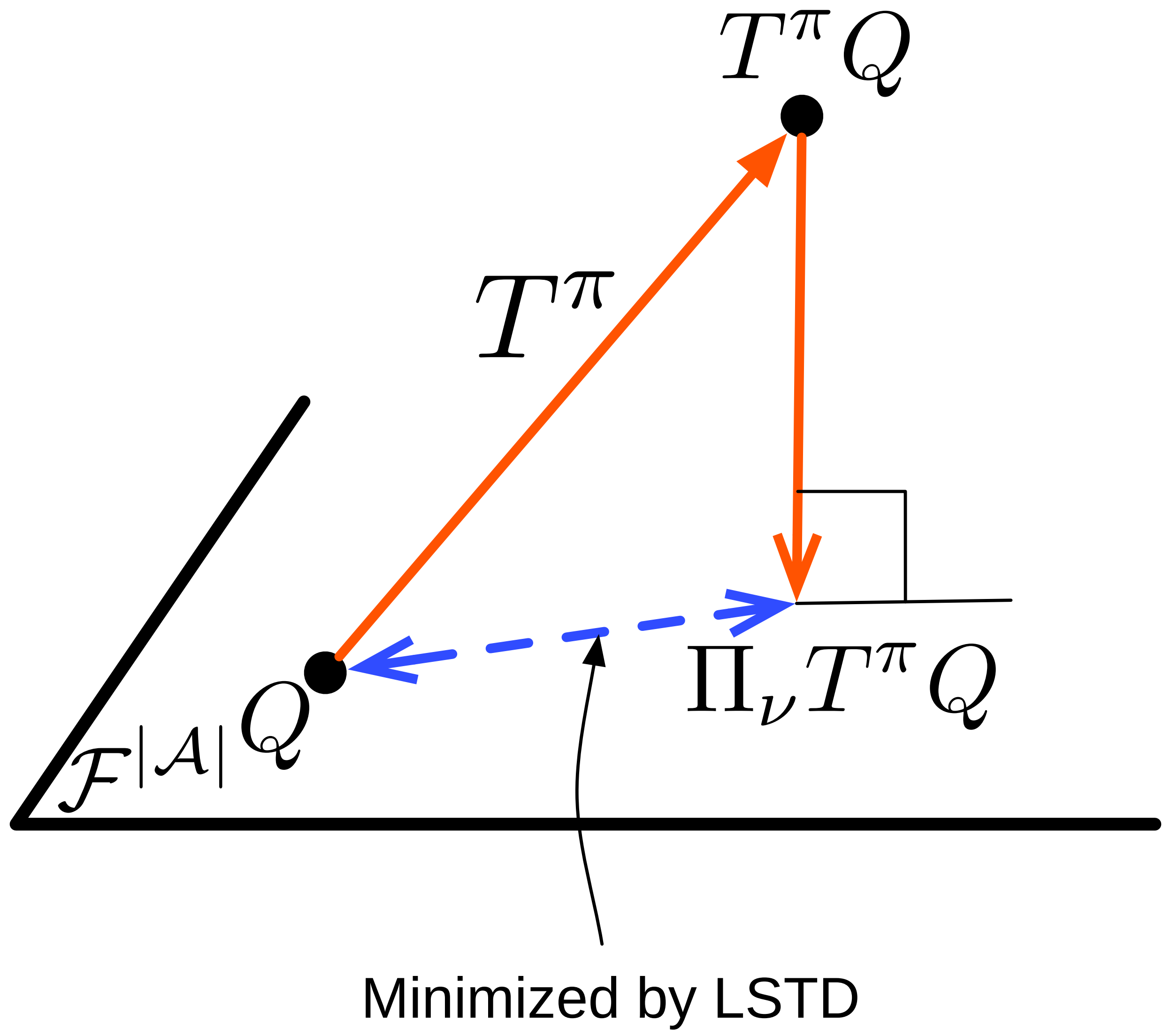
with probability at least $1 - \delta$. Here $0 < \alpha < 1$ is a measure of the complexity of the function space $\mathcal{F}^{|\mathcal{A}|}$.

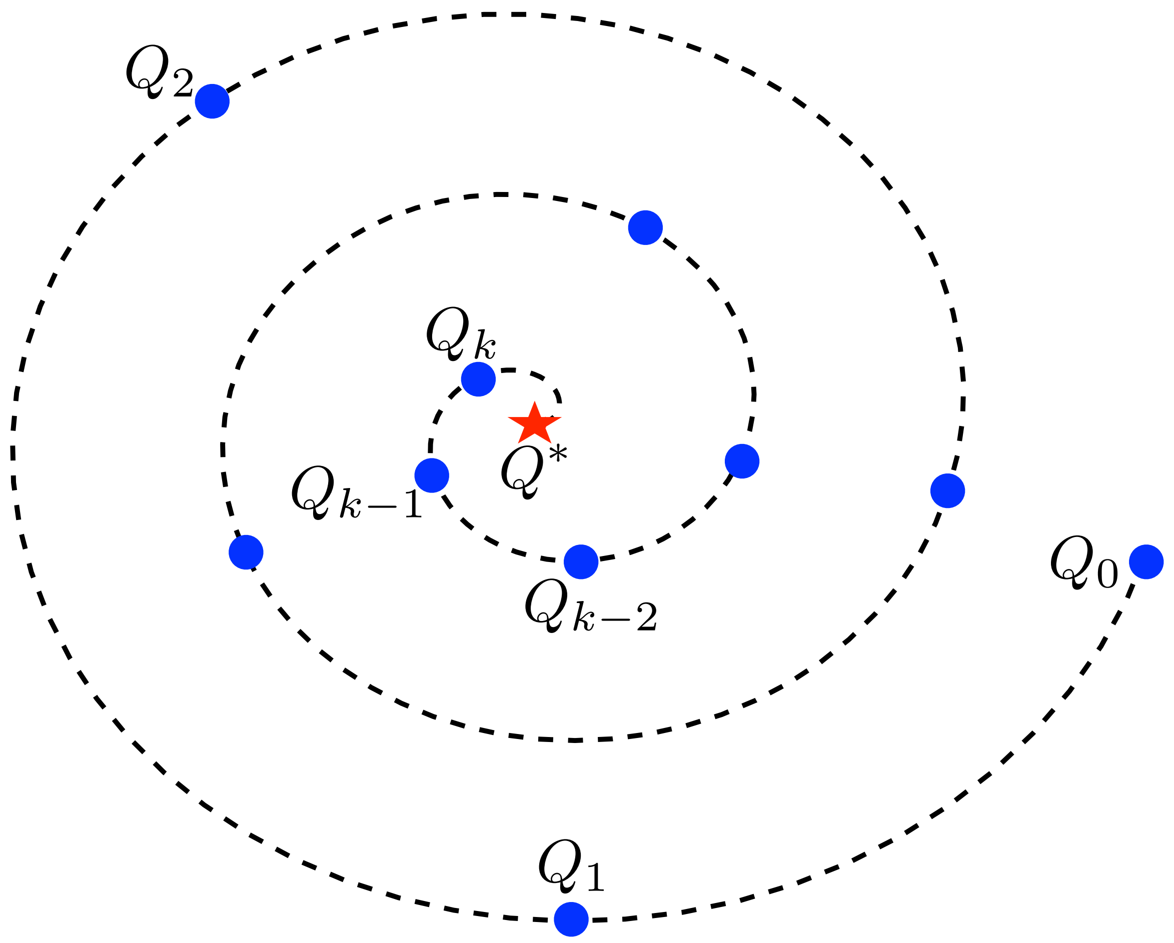
Conclusion

What has been Covered?

- ☑ VI and PI
- ☑ From VI to AVI
- ☑ From PI to API
  - ☑ Bellman Residual Minimization
  - ☑ LSTD
- ☑ Theoretical Analysis
  - ☑ Error Propagation







What has not been Covered?

- Statistical analysis
- Other Ideas from Dynamic Programming
  - Modified Policy Iteration (between VI and PI)
  - FA for Policy Space
    - Classification-based Approximate Policy Iteration (and Actor-Critic)
- How to choose FA?
  - Feature Generation, Nonparametric, DNN
- Computational concerns
  - From Batch to Online
- How to collect data efficiently?

References

MDP and RL

- Dimitri P. Bertsekas and Steven E. Shreve, Stochastic Optimal Control: The Discrete-Time Case, 1978.
- Dimitri P. Bertsekas and John N. Tsitsiklis, Neuro-Dynamic Programming, 1996.
- Lucian Busoniu, Robert Babuska, Bart De Schutter, and Damien Ernst, Reinforcement Learning and Dynamic Programming Using Function Approximators, 2010.
- Richard S. Sutton and Andrew G. Barto, Reinforcement Learning: An Introduction, 2nd edition, 2018.
- Csaba Szepesvari, Algorithms for Reinforcement Learning, 2010.

Note:
1) Categorization is approximate. Many papers belong to several categories.
2) This is not a comprehensive reference list. Many good papers are not mentioned.
3) Some of the entry-point papers in each category are underlined.

References

Batch RL/ADP Monographs and Surveys

- Lucian Busoniu, Robert Babuska, Bart De Schutter, and Damien Ernst, Reinforcement Learning and Dynamic Programming Using Function Approximators, 2010.
- Amir-massoud Farahmand, Regularization in Reinforcement Learning, PhD Dissertation, University of Alberta, 2011. [Similar style to these slides.]
- Mohammad Ghavamzadeh, Sample Complexity in Sequential Decision-Making, HDR Dissertation, Université Lille 1 - Sciences et Technologies, France, 2014.
- Christoph Dann, Gerhard Neumann, and Jan Peters, “Policy evaluation with temporal differences: A survey and comparison,” JMLR, 2014.

References

AVI

- Andras Antos, Remi Munos, Csaba Szepesvari, “Fitted Q-iteration in continuous action-space MDPs,” NIPS, 2008.
- Andre M.S. Barreto, Doina Precup, and Joelle Pineau, “Reinforcement learning using kernel-based stochastic factorization,” NIPS, 2011.
- Damien Ernst, Pierre Geurts, and Louis Wehenkel, “Tree-based batch mode reinforcement learning,” Journal of Machine Learning Research (JMLR), 2005.
- Amir-massoud Farahmand and Doina Precup, “Value pursuit iteration,” NIPS, 2012.
- Amir-massoud Farahmand, Mohammad Ghavamzadeh, Csaba Szepesvari, and Shie Mannor, “Regularized fitted Q-iteration for planning in continuous-space Markovian Decision Problems,” In Proceedings of American Control Conference (ACC), 2009.
- Remi Munos and Csaba Szepesvari, “Finite-Time Bounds for Fitted Value Iteration,” JMLR, 2008.
- Dirk Ormoneit and Saunak Sen, “Kernel-based reinforcement learning,” Machine Learning, 2002.

References

API (LSPI, BRM, CAPI)

LSTD & LSPI

- Bernardo Avila Pires and Csaba Szepesvari, “Statistical linear estimation with penalized estimators: an application to reinforcement learning,” ICML, 2012.
- Dimitri P. Bertsekas, “Approximate policy iteration: A survey and some new methods,” Journal of Control Theory and Applications, 2011.
- Steven J. Bradtke and Andrew G. Barto, “Linear least-squares algorithms for temporal difference learning,” Machine Learning, 1996.
- Amir-massoud Farahmand, Mohammad Ghavamzadeh, Csaba Szepesvari, and Shie Mannor, “Regularized policy iteration,” NIPS, 2008.
- Amir-massoud Farahmand, Mohammad Ghavamzadeh, Csaba Szepesvari, and She Mannor, “Regularized Policy Iteration with Nonparametric Function Spaces,”JMLR, 2016.
- Mohammad Ghavamzadeh, Alessandro Lazaric, Remi Munos, and Matthew Hoffman, “Finite-sample analysis of Lasso-TD,” ICML, 2011.

References

API (LSPI, BRM, CAPI)

LSTD & LSPI

- Matthew Hoffman, Alessandro Lazaric, Mohammad Ghavamzadeh, and Remi Munos, "Regularized least squares temporal difference learning with nested l1 and l2 penalization," Recent Advances in Reinforcement Learning, 2012.
- J. Zico Kolter and Andrew Ng, "Regularization and feature selection in least-squares temporal difference learning," ICML, 2009.
- Michail Lagoudakis and Ronald Parr, "Least-squares policy iteration," JMLR, 2003.
- Alessandro Lazaric, Mohammad Ghavamzadeh, and Remi Munos, "Finite-sample analysis of least-squares policy iteration," JMLR, 2012.
- Christopher Painter-Wakefield and Ronald Parr, "Greedy algorithms for sparse reinforcement learning," ICML, 2012.
- Gavin Taylor and Ronald Parr, "Kernelized value function approximation for reinforcement learning," ICML, 2009.
- Xin Xu, Dewen Hu, and Xicheng Lu, "Kernel-based least squares policy iteration for reinforcement learning," IEEE Transactions on Neural Networks, 2007.

References

API (LSPI, BRM, CAPI)

BRM

- Andras Antos, Csaba Szepesvari, and Remi Munos, "Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path," Machine Learning, 2008.
- Leemon Baird, "Residual algorithms: Reinforcement learning with function approximation," ICML, 1995.
- Matthieu Geist and Bruno Scherrer, "l1-penalized projected Bellman residual," Recent Advances in Reinforcement Learning, 2012.
- Tobias Jung and Daniel Polani, "Least squares SVM for least squares TD learning," European Conference on Artificial Intelligence (ECAI), 2006.
- Odalric Maillard, Remi Munos, Alessandro Lazaric, and Mohammad Ghavamzadeh, "Finite-Sample Analysis of Bellman Residual Minimization," Asian Conference on Machine Learning (ACML), 2010.

References

API (LSPI, BRM, CAPI)

Classification-based API

- Amir-massoud Farahmand, Doina Precup, Mohammad Ghavamzadeh, Andre M.S. Barreto, "Classification-based Approximate Policy Iteration," IEEE Transactions on Automatic Control, 2015.
- Alessandro Lazaric, Mohammad Ghavamzadeh, and Remi Munos, "Analysis of Classification-based Policy Iteration Algorithms," JMLR, 2016.

Error Propagation Analysis (AVI and API)

- Amir-massoud Farahmand, "Action-gap phenomenon in reinforcement learning," NIPS, 2011.
- Amir-massoud Farahmand, Remi Munos, and Csaba Szepesvari, "Error propagation for approximate policy and value iteration," NIPS, 2010.
- Remi Munos, "Performance bounds in Lp norm for approximate value iteration," SIAM Journal on Control and Optimization, 2007.
- Remi Munos, "Error bounds for approximate policy iteration," ICML, 2003.

References

Feature Generation

- Sridhar Mahadevan and Bo Liu, “Basis construction from power series expansions of value functions,” NIPS, 2010.
- Sridhar Mahadevan and Mauro Maggioni, “Proto-value functions: A Laplacian framework for learning representation and control in Markov decision processes,” JMLR, 2007.
- Mahdi Milani Fard, Yuri Grinberg, Amir-massoud Farahmand, Joelle Pineau, and Doina Precup, “Bellman error based feature generation using random projections on sparse spaces,” NIPS, 2013.
- Ronald Parr, Christopher Painter-Wakefield, Lihong Li, and Michael Littman, “Analyzing feature generation for value-function approximation,” ICML, 2007.
- Marek Petrik, “An analysis of Laplacian methods for value function approximation in MDPs,” IJCAI, 2007.

Model Selection in RL

- Amir-massoud Farahmand and Csaba Szepesvari, “Model selection in reinforcement learning,” Machine Learning, 2011.