CS 498 Reinforcement Learning

Nan Jiang

Logistics

All information about this course: http://nanjiang.cs.illinois.edu/cs498/

Logistics

All information about this course: http://nanjiang.cs.illinois.edu/cs498/

- piazza link (please **self enroll**!)

Logistics

All information about this course: http://nanjiang.cs.illinois.edu/cs498/

- piazza link (please **self enroll**!)
  - piazza will be the main platform for course-related communications

Logistics

All information about this course: http://nanjiang.cs.illinois.edu/cs498/

- piazza link (please **self enroll**!)
  - piazza will be the main platform for course-related communications
  - you can write private posts only visible to instructors, so post on piazza before requesting an individual meeting

Logistics

All information about this course: http://nanjiang.cs.illinois.edu/cs498/

- piazza link (please **self enroll**!)
  - piazza will be the main platform for course-related communications
  - you can write private posts only visible to instructors, so post on piazza before requesting an individual meeting
- Slides & schedule (updated every week)

Logistics

All information about this course: http://nanjiang.cs.illinois.edu/cs498/

- piazza link (please **self enroll**!)
  - piazza will be the main platform for course-related communications
  - you can write private posts only visible to instructors, so post on piazza before requesting an individual meeting
- Slides & schedule (updated every week)
- textbook links

Logistics

All information about this course: http://nanjiang.cs.illinois.edu/cs498/

- piazza link (please **self enroll**!)
  - piazza will be the main platform for course-related communications
  - you can write private posts only visible to instructors, so post on piazza before requesting an individual meeting
- Slides & schedule (updated every week)
- textbook links
- TA & office hours: 5-6pm Tue, 10-11am Fri (starting next week)

Logistics

All information about this course: http://nanjiang.cs.illinois.edu/cs498/

- piazza link (please **self enroll**!)
  - piazza will be the main platform for course-related communications
  - you can write private posts only visible to instructors, so post on piazza before requesting an individual meeting
- Slides & schedule (updated every week)
- textbook links
- TA & office hours: 5-6pm Tue, 10-11am Fri (starting next week)
- course policies

What’s this course about?

What’s this course about?

- A 400-level special topics course on RL

What’s this course about?

- A 400-level special topics course on RL
- Developed & taught for the first time — expect rough edges

What's this course about?

- A 400-level special topics course on RL
- Developed & taught for the first time — expect rough edges
- Less theoretical version of CS598 that I taught last year

What's this course about?

- A 400-level special topics course on RL
- Developed & taught for the first time — expect rough edges
- Less theoretical version of CS598 that I taught last year
- Will have some (minimum) programming components, but **still** lean heavily towards the mathematical side

What's this course about?

- A 400-level special topics course on RL
- Developed & taught for the first time — expect rough edges
- Less theoretical version of CS598 that I taught last year
- Will have some (minimum) programming components, but **still lean heavily towards the mathematical side**
- **Think of this mainly as a mathematical course;** the programming component is just to ensure that you have some exposure to (some of) the actual algorithms

What's this course about?

- A 400-level special topics course on RL
- Developed & taught for the first time — expect rough edges
- Less theoretical version of CS598 that I taught last year
- Will have some (minimum) programming components, but **still lean heavily towards the mathematical side**
- **Think of this mainly as a mathematical course;** the programming component is just to ensure that you have some exposure to (some of) the actual algorithms
- "I want to do cool applications" — we no **not** meet this demand (but hopefully will give you a good foundation for it)

What's this course about?

- A 400-level special topics course on RL
- Developed & taught for the first time — expect rough edges
- Less theoretical version of CS598 that I taught last year
- Will have some (minimum) programming components, but **still lean heavily towards the mathematical side**
- **Think of this mainly as a mathematical course;** the programming component is just to ensure that you have some exposure to (some of) the actual algorithms
- "I want to do cool applications" — we no **not** meet this demand (but hopefully will give you a good foundation for it)
- Intellectual curiosity is the only reason you should take this course (and I will tell you why honestly in the class…)

Prerequisites

- Maths

Prerequisites

- Maths
  - Linear algebra, probability & statistics, basic calculus

Prerequisites

- Maths
  - Linear algebra, probability & statistics, basic calculus
  - Optional: Markov chains, stochastic processes, numerical analysis

Prerequisites

- Maths
  - Linear algebra, probability & statistics, basic calculus
  - Optional: Markov chains, stochastic processes, numerical analysis
- Exposure to ML

Prerequisites

- Maths
  - Linear algebra, probability & statistics, basic calculus
  - Optional: Markov chains, stochastic processes, numerical analysis
- Exposure to ML
  - e.g., CS 446 Machine Learning

Prerequisites

- Maths
  - Linear algebra, probability & statistics, basic calculus
  - Optional: Markov chains, stochastic processes, numerical analysis
- Exposure to ML
  - e.g., CS 446 Machine Learning
- We will soon have a homework 0 (hopefully this Thursday) that helps you recall some of the math tools and check whether you are mathematically prepared for this course

Coursework & Grading

- Some readings after/before class

Coursework & Grading

- Some readings after/before class
- Homework (50%): ~5 assignments, with both programming & writing components

Coursework & Grading

- Some readings after/before class
- Homework (50%): ~5 assignments, with both programming & writing components
  - Will likely to be handled via compass—more on this later

Coursework & Grading

- Some readings after/before class
- Homework (50%): ~5 assignments, with both programming & writing components
  - Will likely to be handled via compass—more on this later
- Exam (35%): Either final or a late mid exam

Coursework & Grading

- Some readings after/before class
- Homework (50%): ~5 assignments, with both programming & writing components
  - Will likely to be handled via compass—more on this later
- Exam (35%): Either final or a late mid exam
- Participation (15%): *might* take the form of in-class pop quizzes (without announcements)

Coursework & Grading

- Some readings after/before class
- Homework (50%): ~5 assignments, with both programming & writing components
  - Will likely to be handled via compass—more on this later
- Exam (35%): Either final or a late mid exam
- Participation (15%): *might* take the form of in-class pop quizzes (without announcements)
- For 4 credit students: an additional course project (reproducing part of an empirical or theory paper)

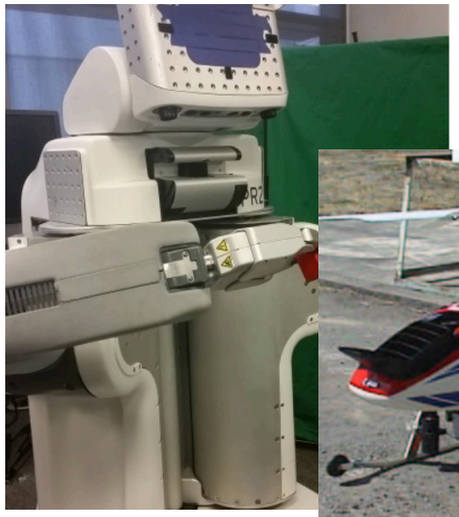
Introduction to MDPs and RL

Reinforcement Learning (RL) Applications

Reinforcement Learning (RL) Applications

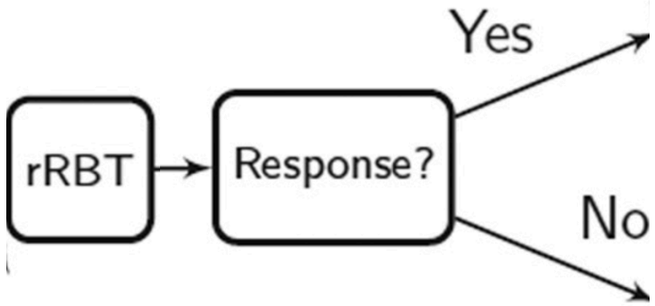
[Levine et al’16] [Ng et al’03]

Reinforcement Learning (RL) Applications



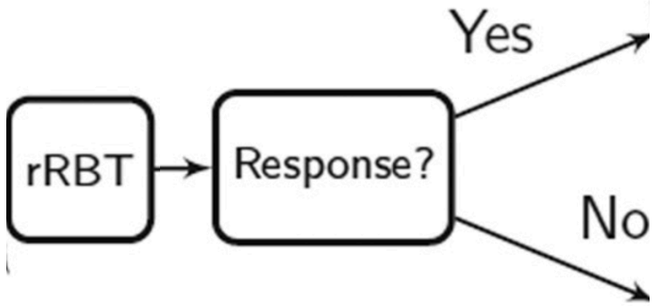
[Levine et al'16] [Ng et al'03] [Singh et al'02]

Reinforcement Learning (RL) Applications



[Levine et al'16] [Ng et al'03] [Singh et al'02] [Lei et al'12]

Reinforcement Learning (RL) Applications

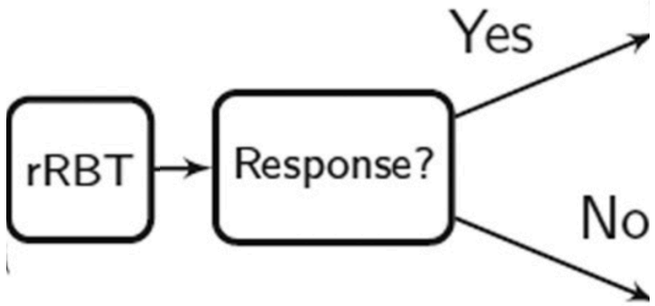


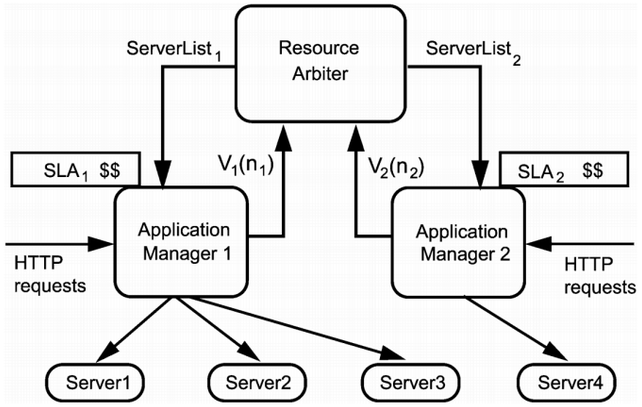
[Levine et al'16] [Ng et al'03] [Mandel et al'16]

[Singh et al'02]

[Lei et al'12]

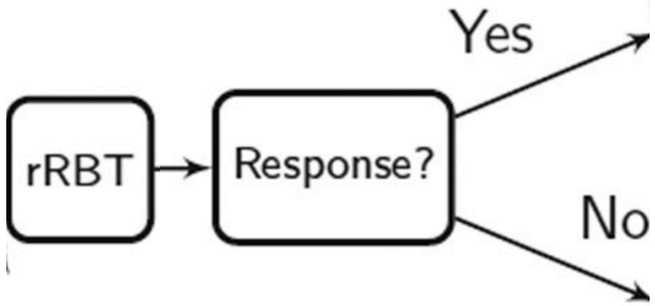
Reinforcement Learning (RL) Applications

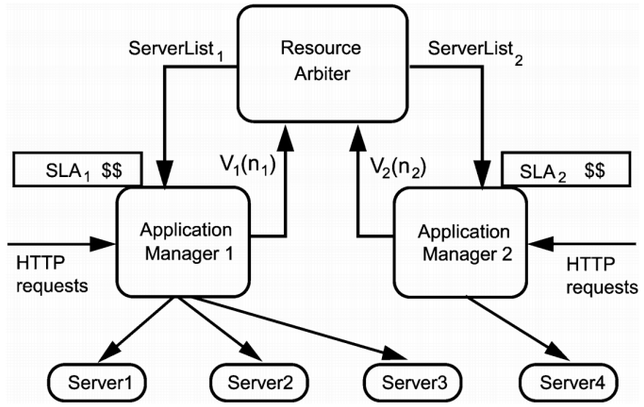




[Levine et al'16] [Ng et al'03] [Singh et al'02] [Lei et al'12]
[Mandel et al'16] [Tesauro et al'07]

Reinforcement Learning (RL) Applications

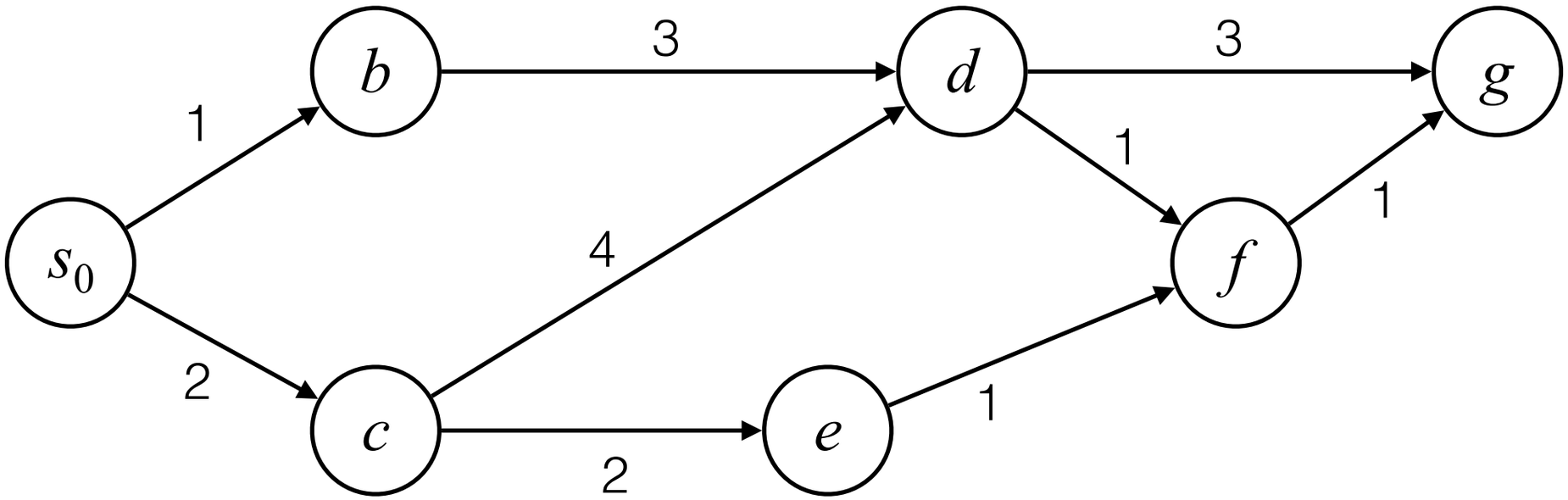




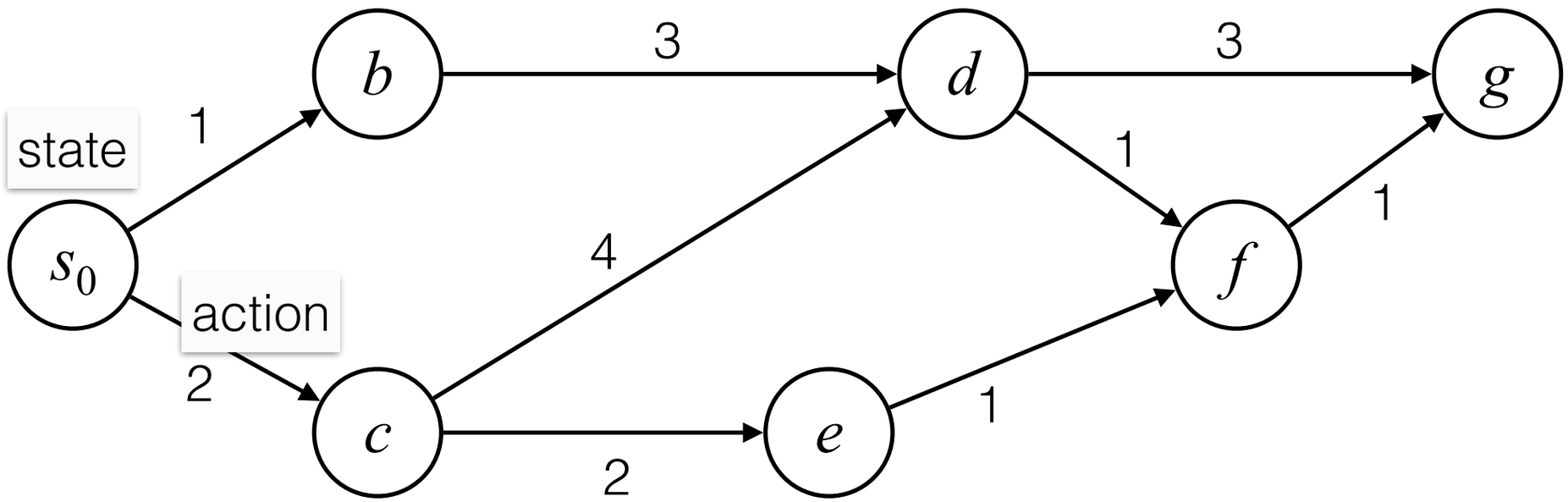
[Levine et al'16] [Ng et al'03] [Singh et al'02] [Lei et al'12]

[Mandel et al'16] [Tesauro et al'07] [Mnih et al'15] [Silver et al'16]

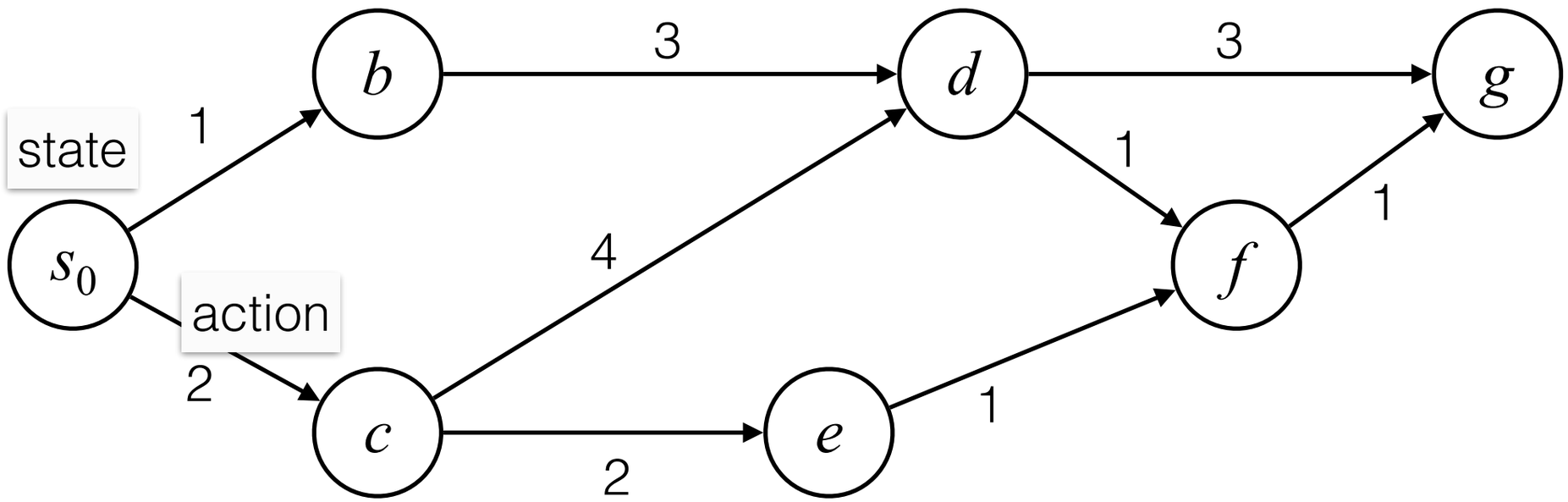
Shortest Path



Shortest Path



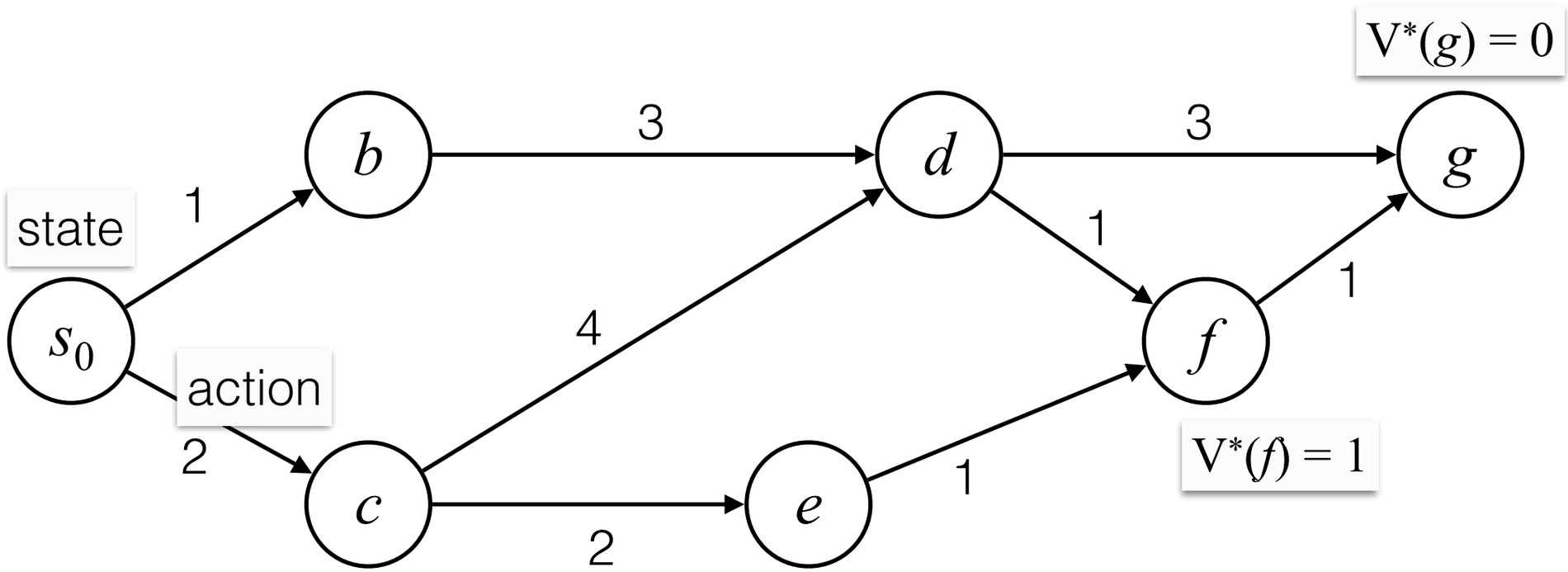
Shortest Path



Greedy is suboptimal due to delayed effects

Need **long-term planning**

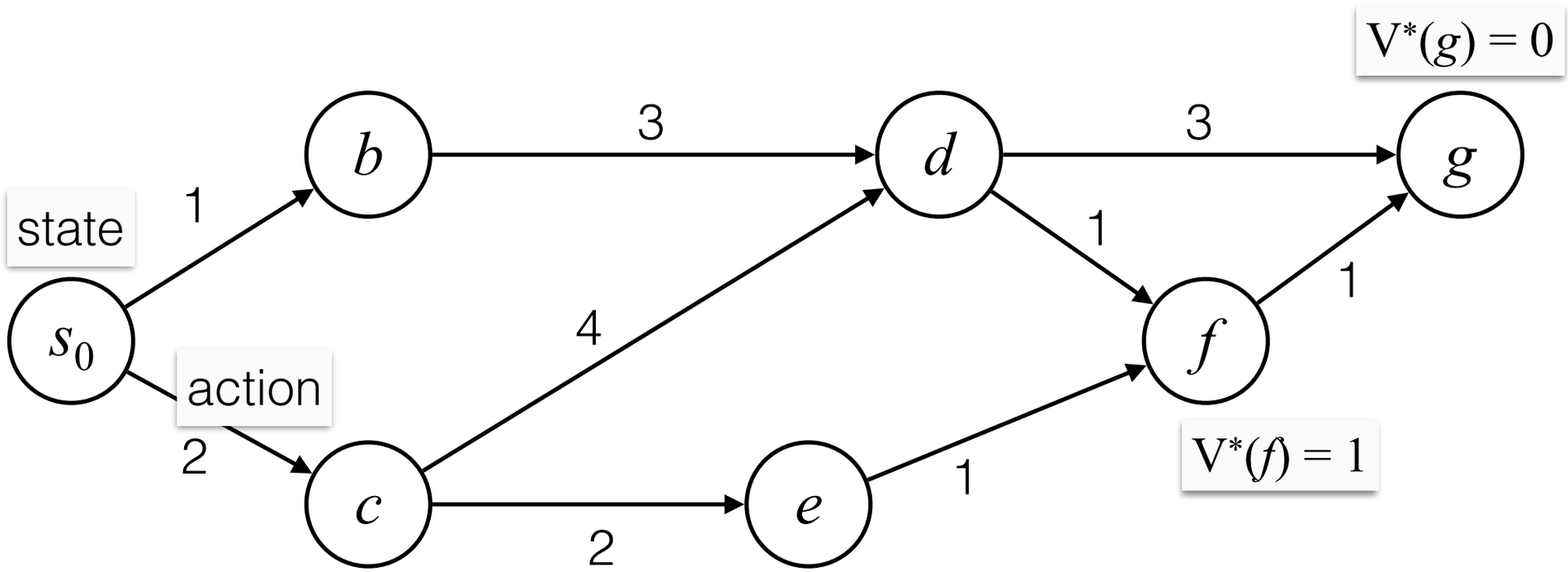
Shortest Path



Greedy is suboptimal due to delayed effects

Need **long-term planning**

Shortest Path

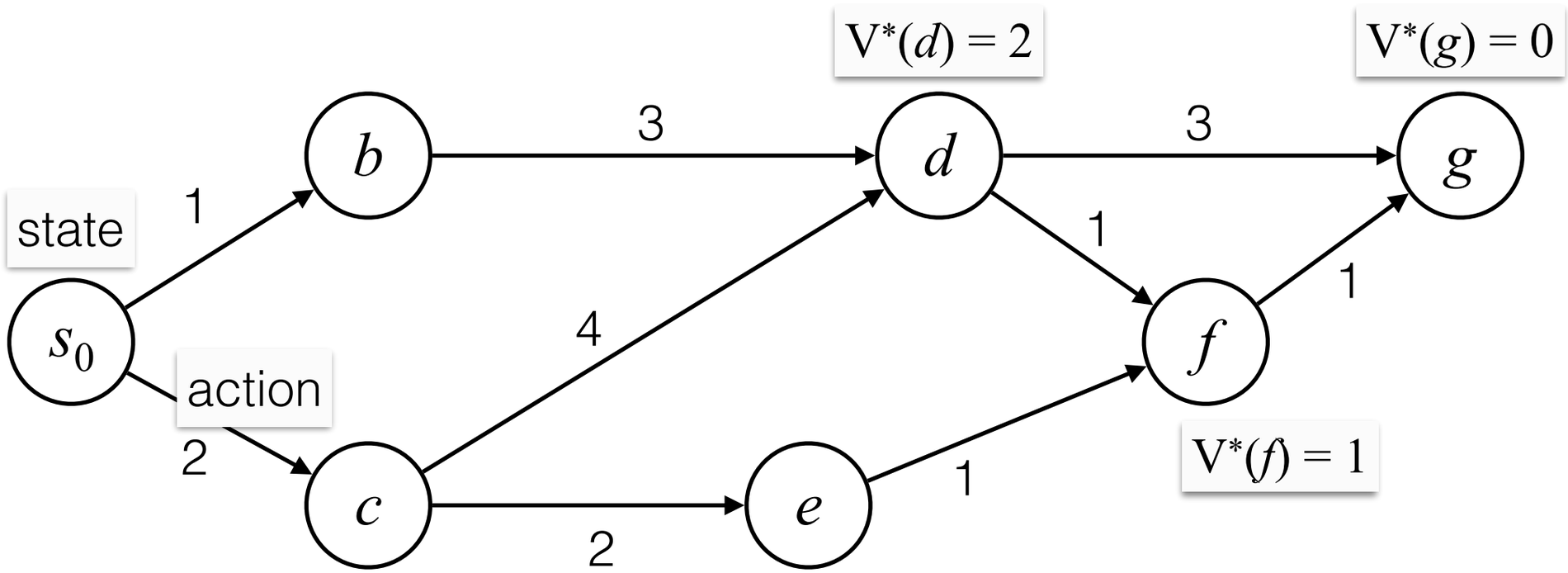


Bellman Equation $V^*(d) = \min\{3 + V^*(g)\ ,\ 2 + V^*(f)\}$

Greedy is suboptimal due to delayed effects

Need **long-term planning**

Shortest Path

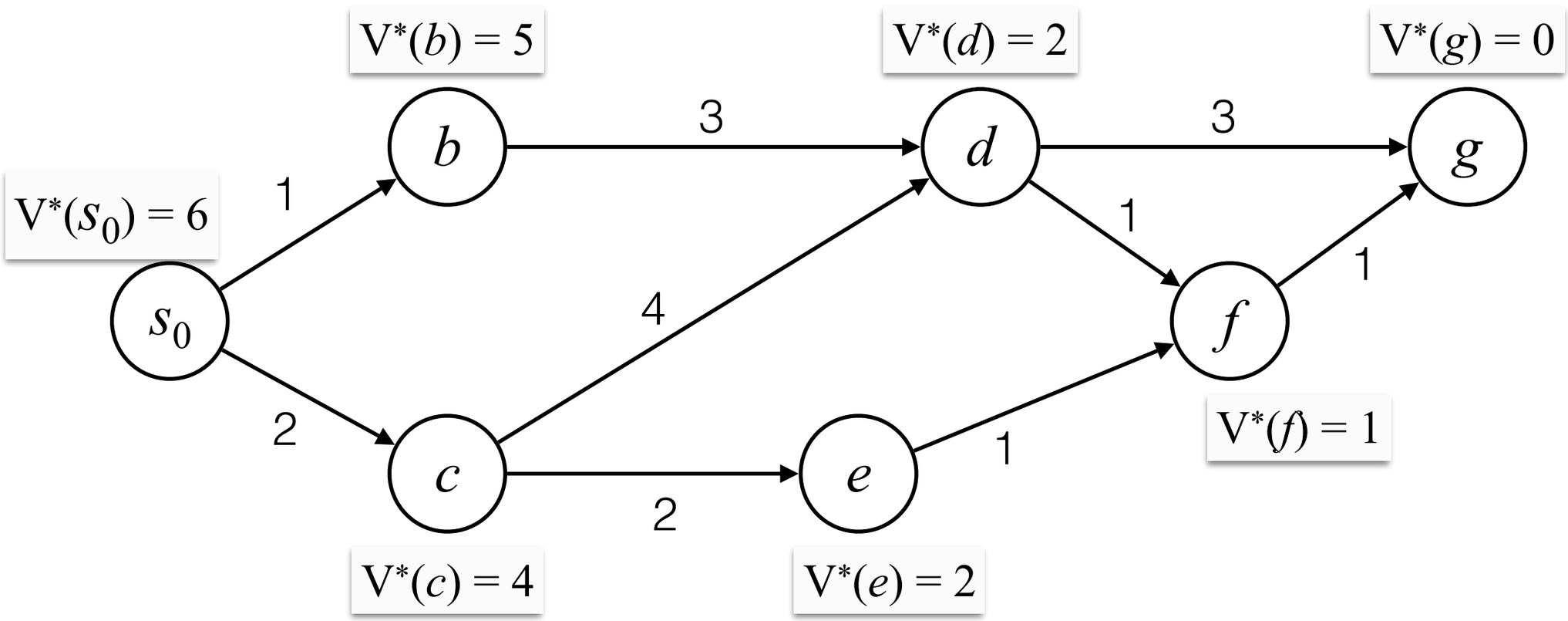


Bellman Equation $V^*(d) = \min\{3 + V^*(g)\ ,\ 2 + V^*(f)\}$

Greedy is suboptimal due to delayed effects

Need **long-term planning**

Shortest Path

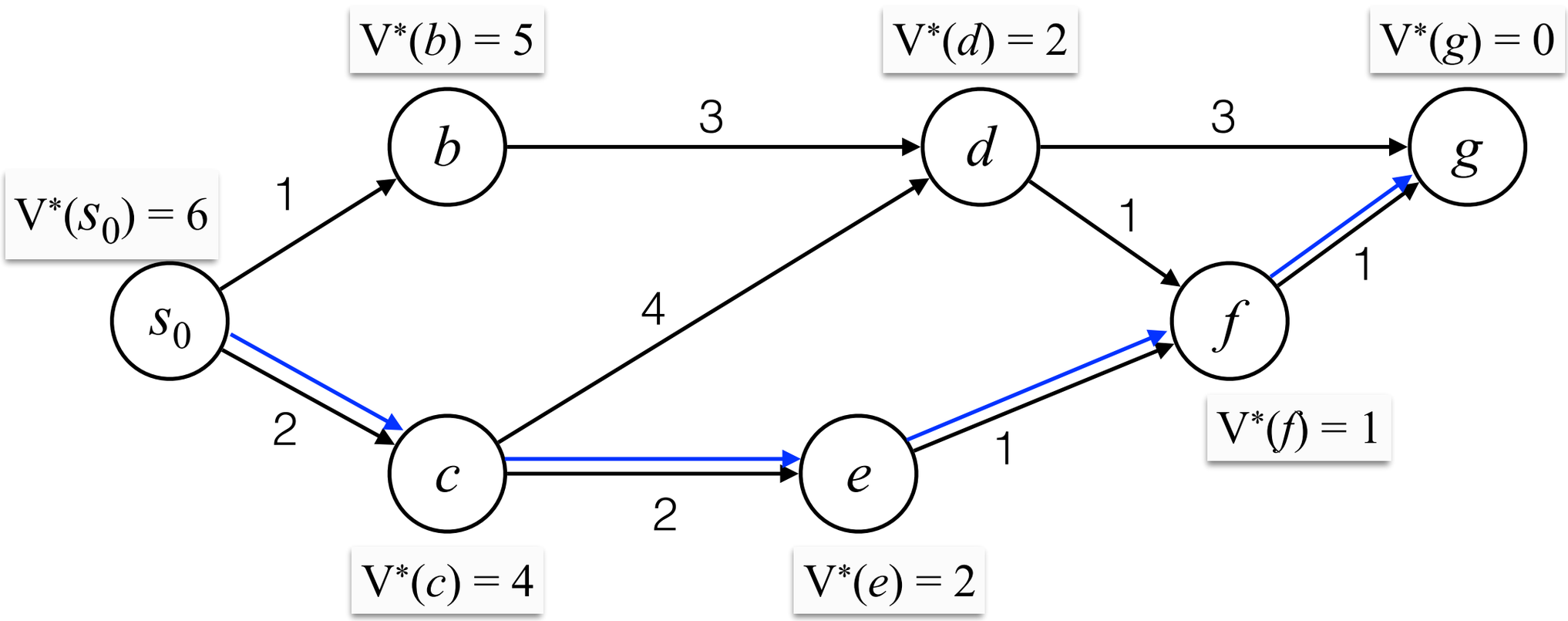


Bellman Equation $V^*(d) = \min\{3 + V^*(g)\ ,\ 2 + V^*(f)\}$

Greedy is suboptimal due to delayed effects

Need **long-term planning**

Shortest Path

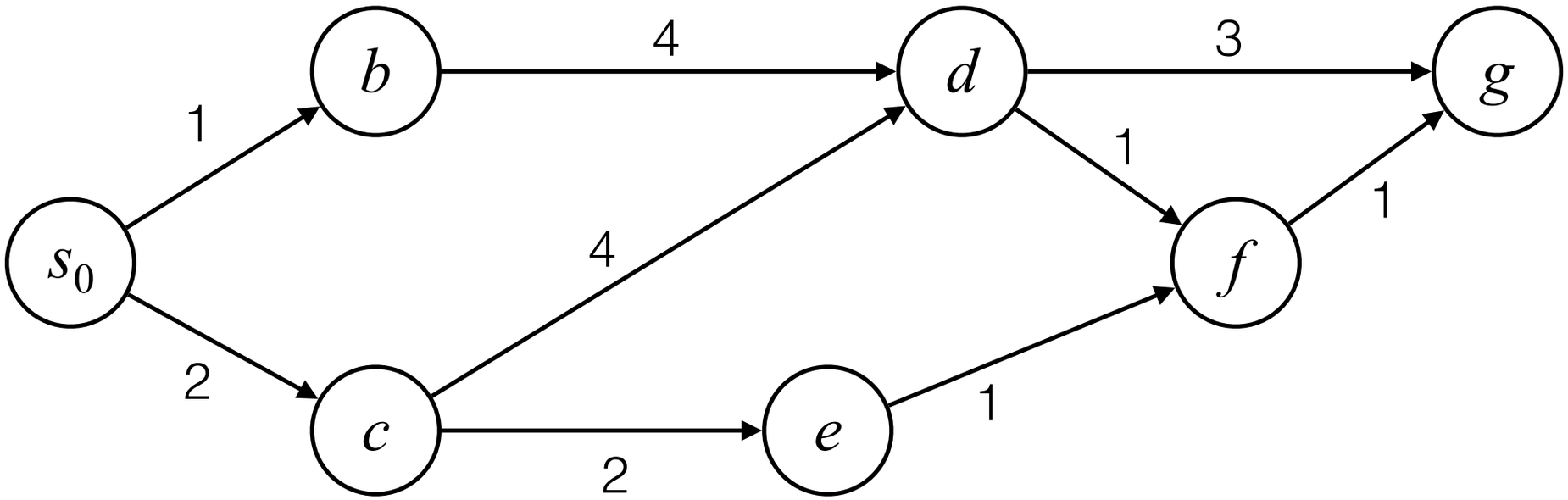


Bellman Equation $V^*(d) = \min\{3 + V^*(g)\ ,\ 2 + V^*(f)\}$

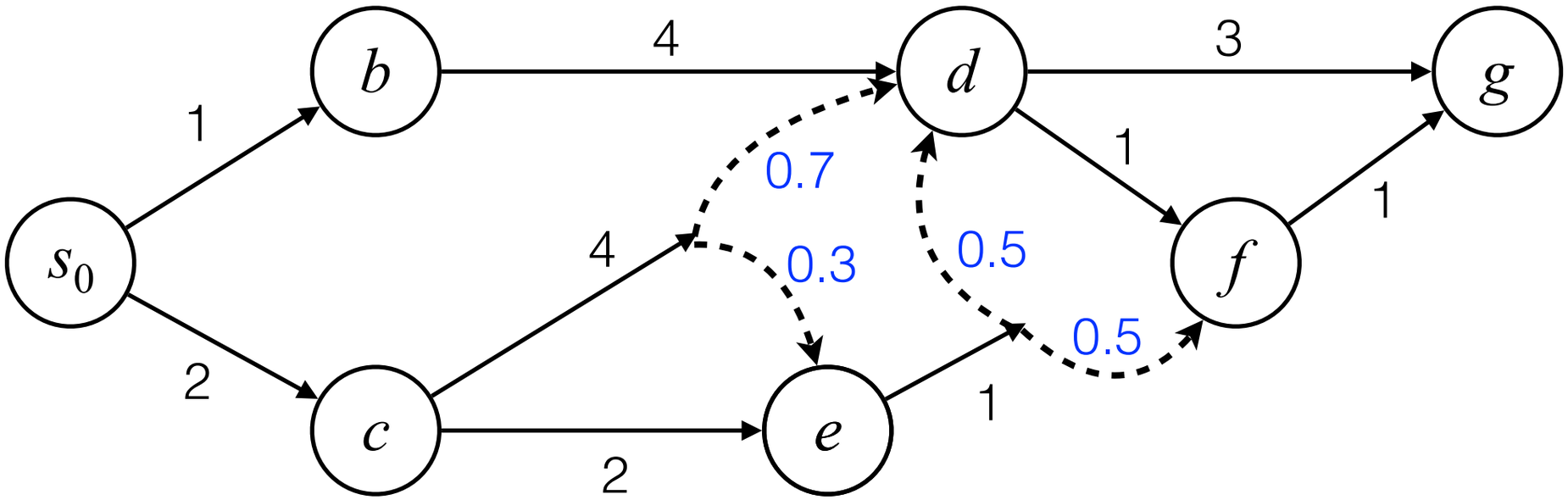
Greedy is suboptimal due to delayed effects

Need **long-term planning**

Shortest Path

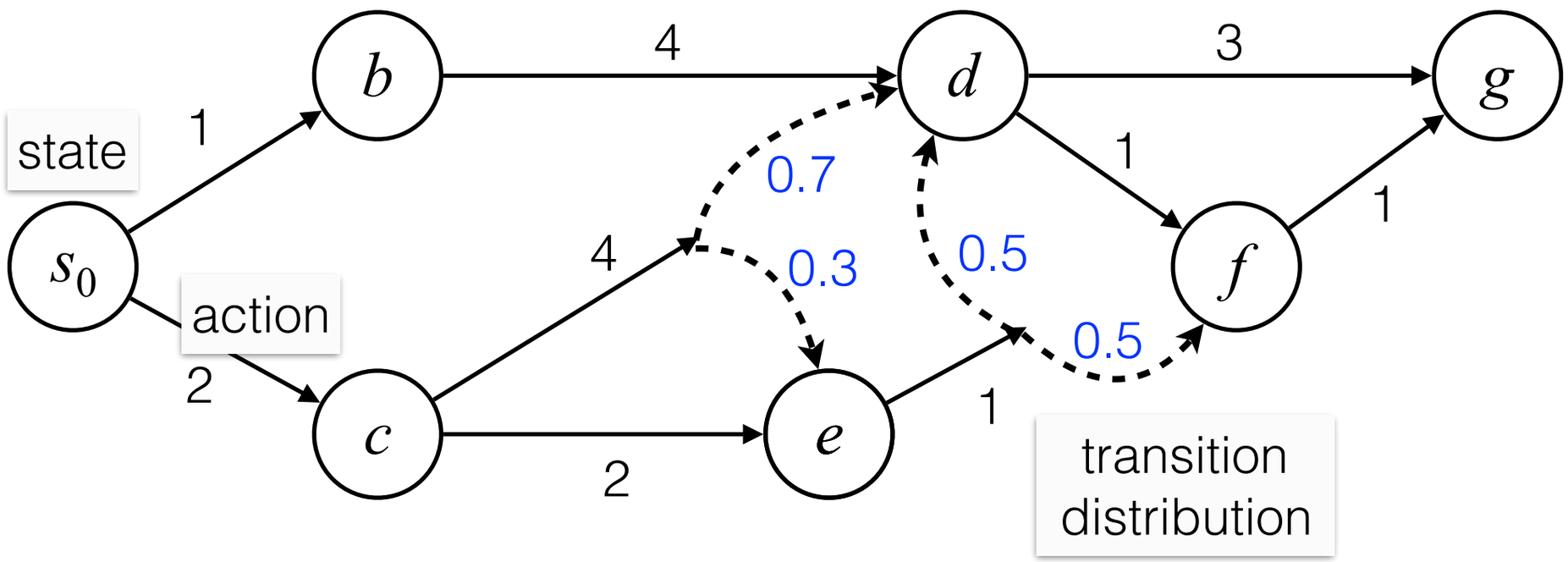


Stochastic Shortest Path

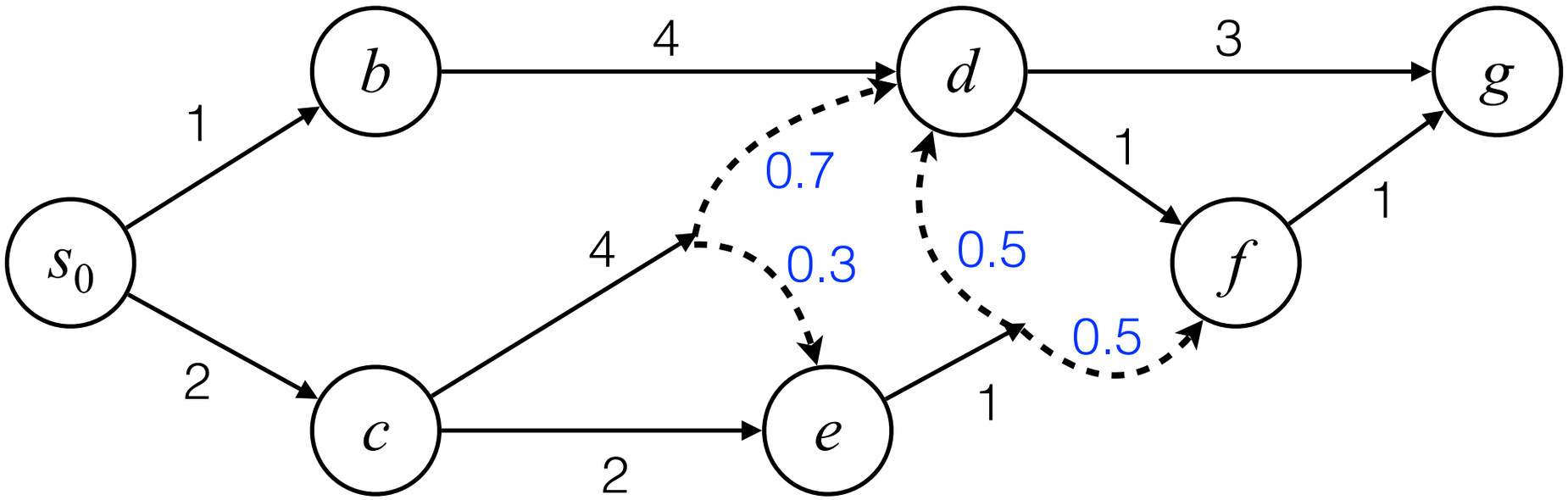


Stochastic Shortest Path

Markov Decision Process (MDP)



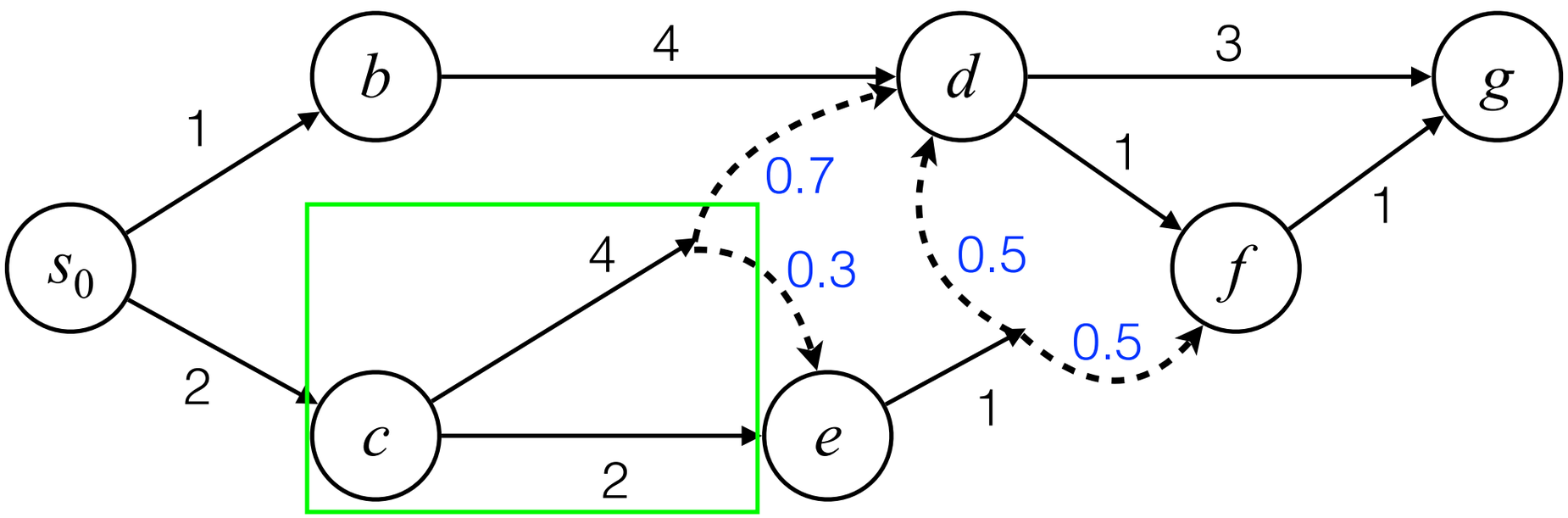
Stochastic Shortest Path



Greedy is suboptimal due to delayed effects

Need **long-term planning**
(also known as *temporal credit assignment* in RL)

Stochastic Shortest Path



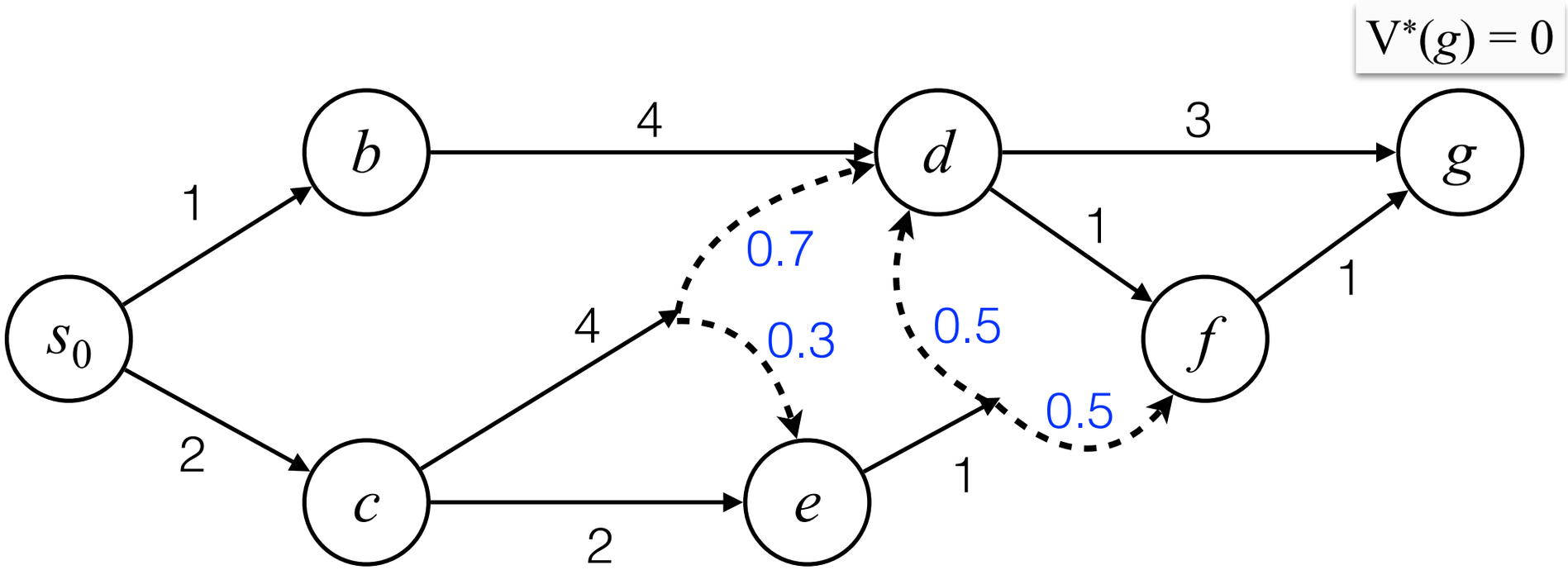
Bellman Equation

$$V^*(c) = \min\{4 + 0.7 \times V^*(d) + 0.3 \times V^*(e)\ ,\ 2 + V^*(e)\}$$

Greedy is suboptimal due to delayed effects

Need **long-term planning**
(also known as *temporal credit assignment* in RL)

Stochastic Shortest Path



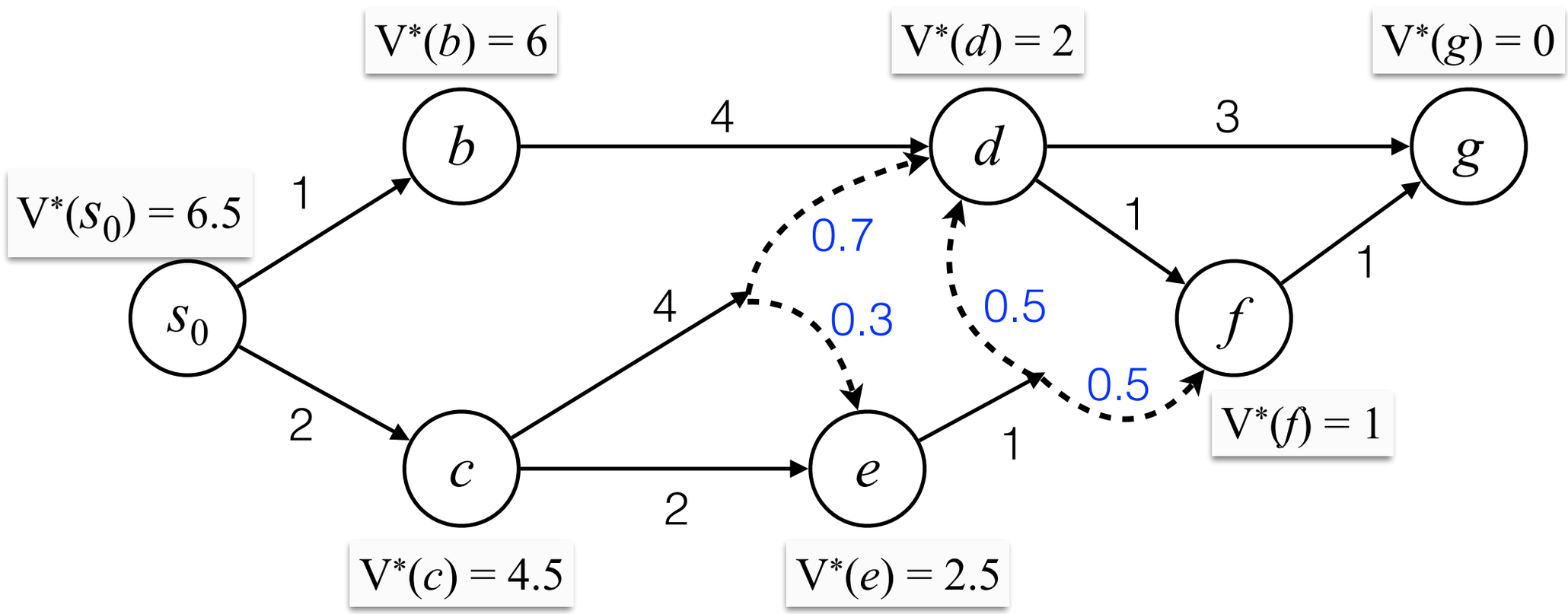
Bellman Equation

$$V^*(c) = \min\{4 + 0.7 \times V^*(d) + 0.3 \times V^*(e)\ ,\ 2 + V^*(e)\}$$

Greedy is suboptimal due to delayed effects

Need **long-term planning**
(also known as *temporal credit assignment* in RL)

Stochastic Shortest Path



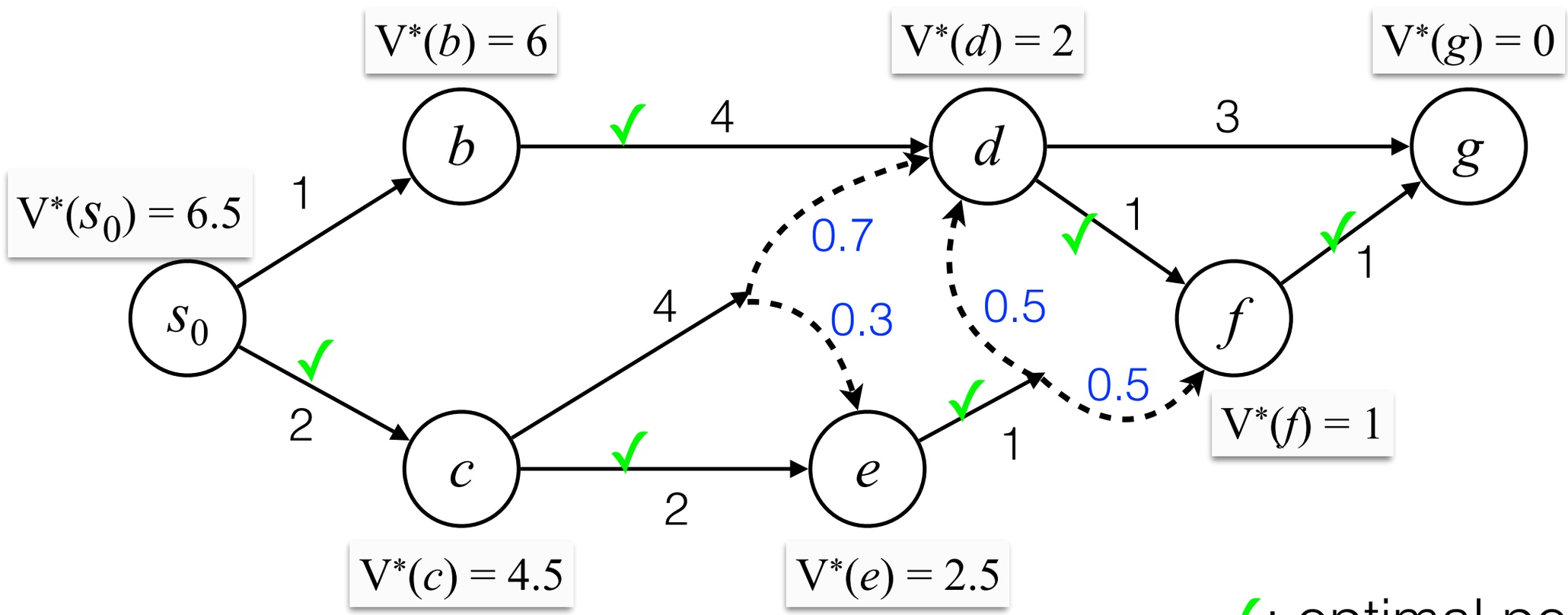
Bellman Equation

$$V^*(c) = \min\{4 + 0.7 \times V^*(d) + 0.3 \times V^*(e)\ ,\ 2 + V^*(e)\}$$

Greedy is suboptimal due to delayed effects

Need **long-term planning**
(also known as *temporal credit assignment* in RL)

Stochastic Shortest Path



✓: optimal policy π^*

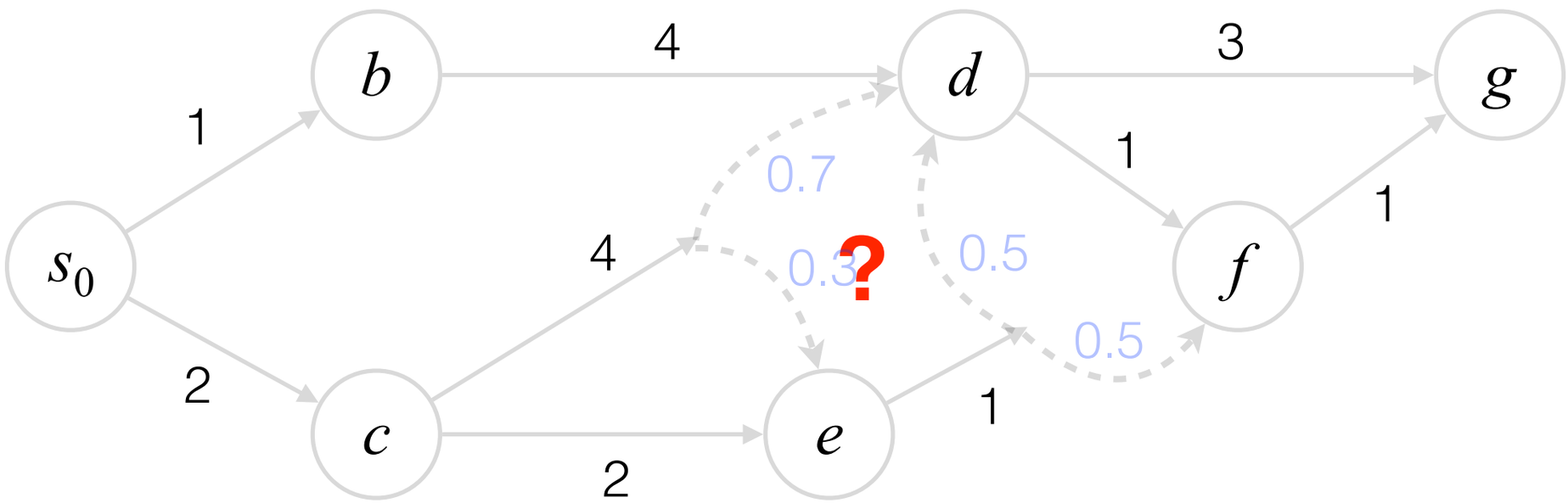
Bellman Equation

$$V^*(c) = \min\{4 + 0.7 \times V^*(d) + 0.3 \times V^*(e)\ ,\ 2 + V^*(e)\}$$

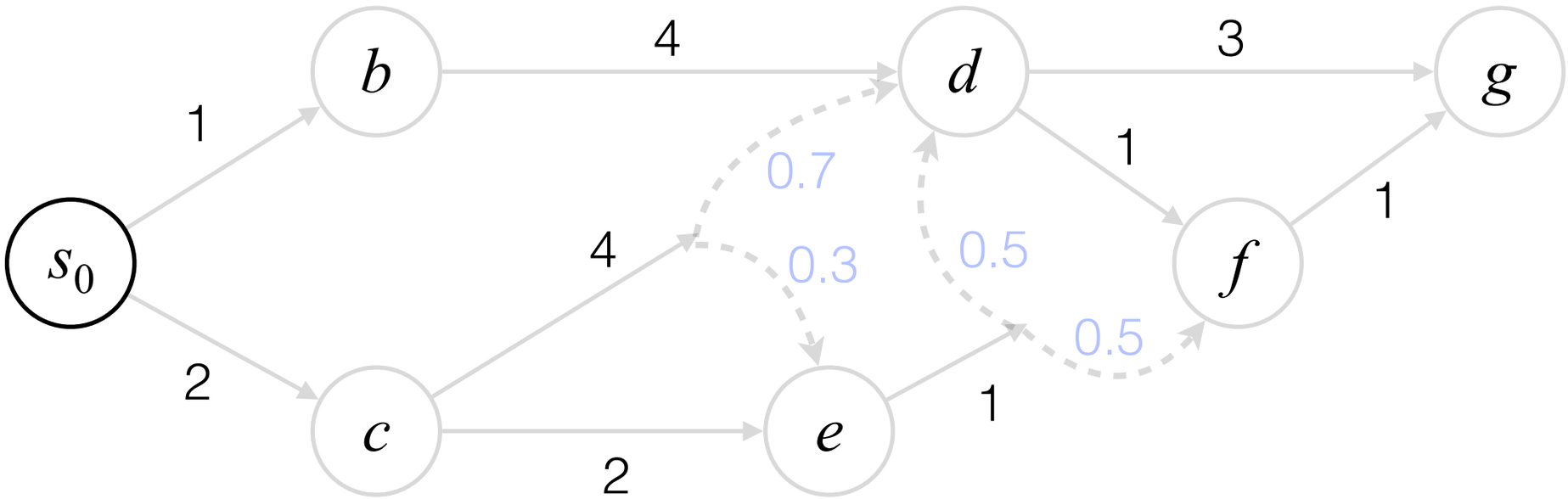
Greedy is suboptimal due to delayed effects

Need **long-term planning**
(also known as *temporal credit assignment* in RL)

Stochastic Shortest Path
via trial-and-error

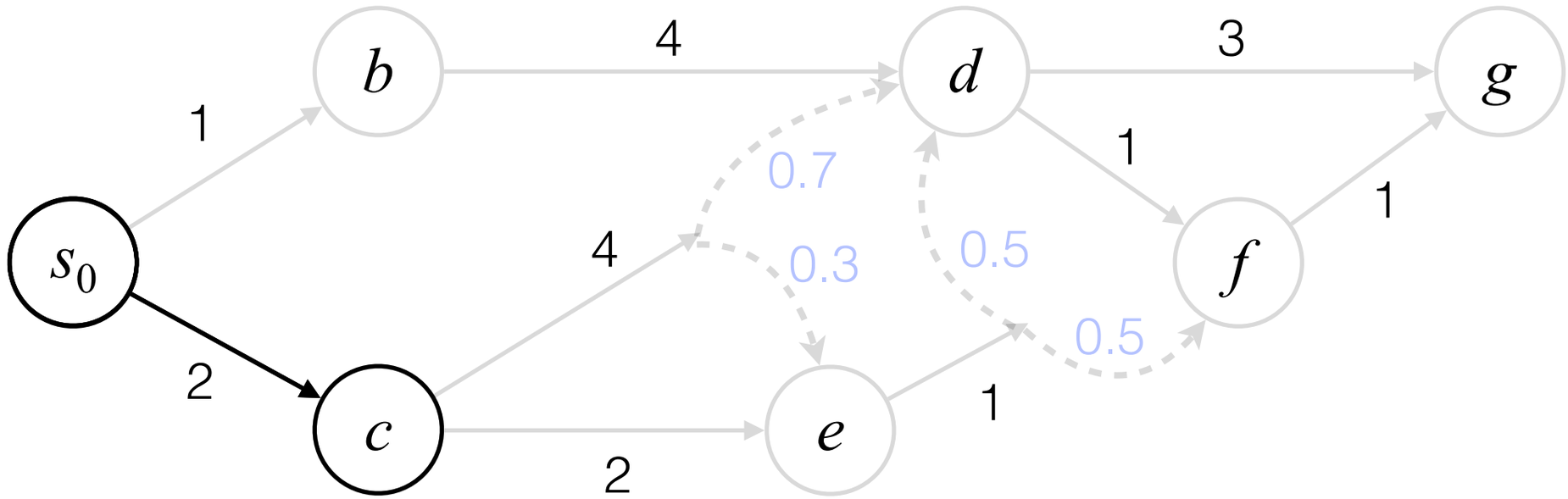


Stochastic Shortest Path
via trial-and-error



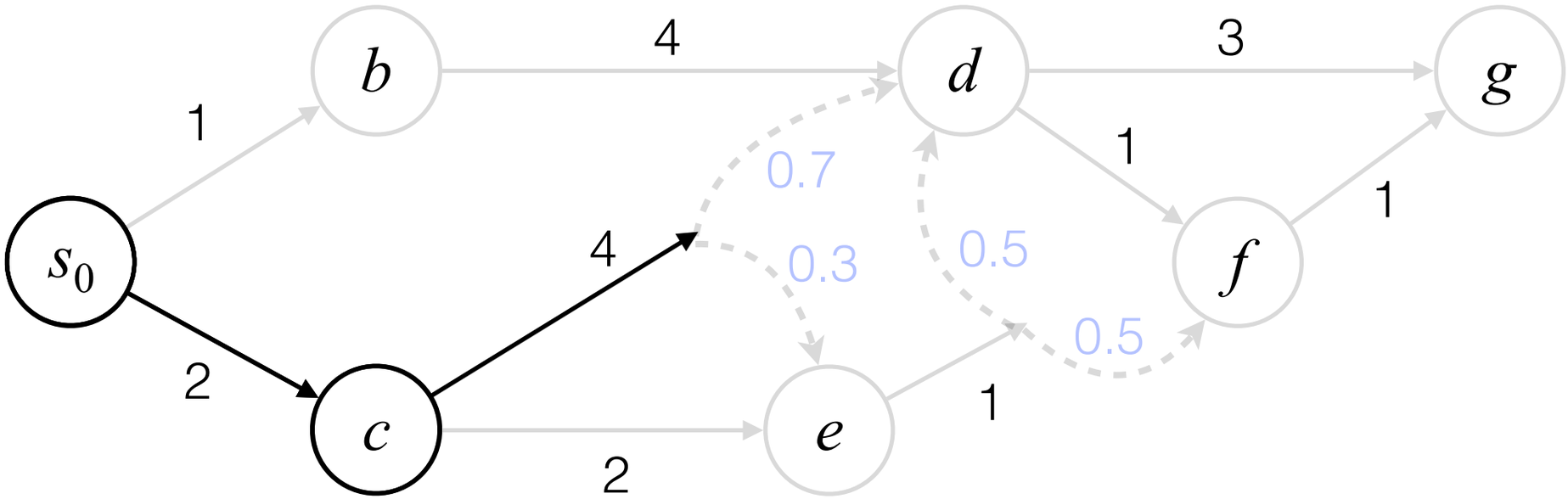
s_0

Stochastic Shortest Path
via trial-and-error



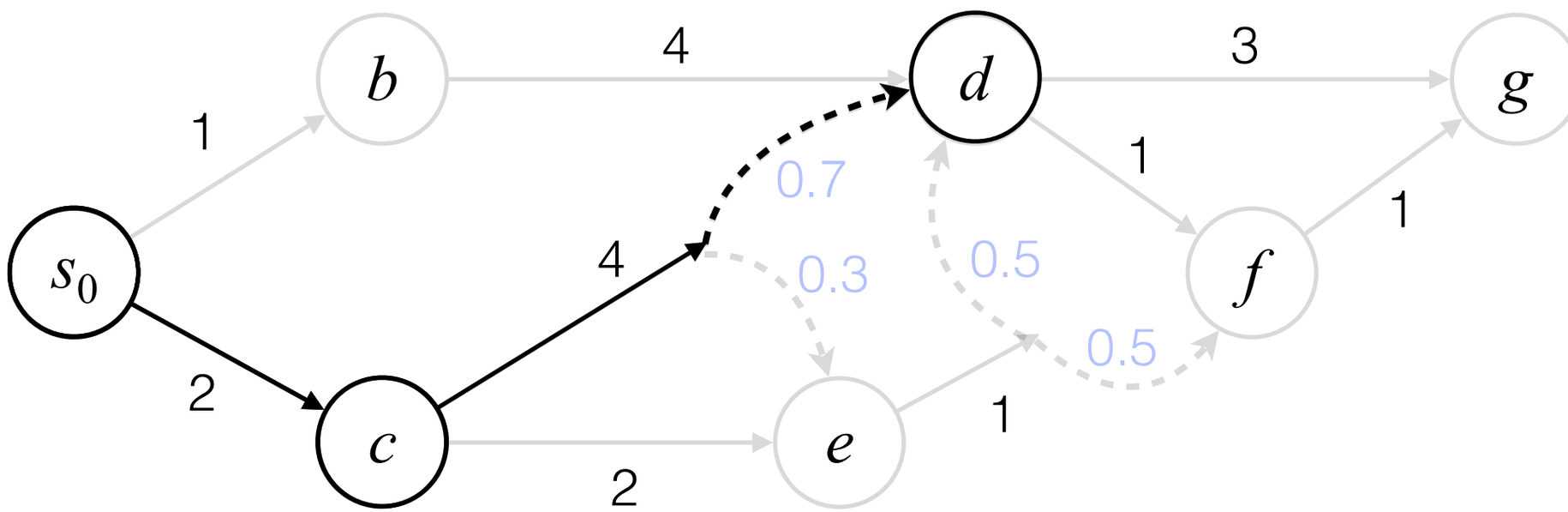
$s_0 \searrow c$

Stochastic Shortest Path
via trial-and-error



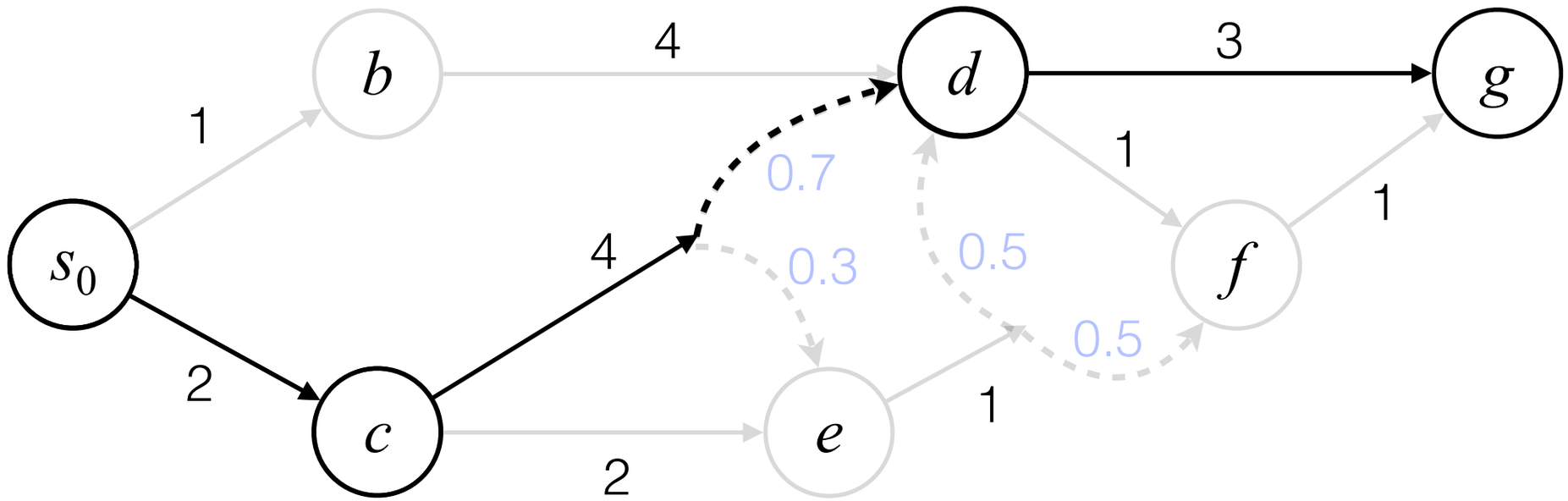
$s_0 \searrow c \nearrow$

Stochastic Shortest Path
via trial-and-error



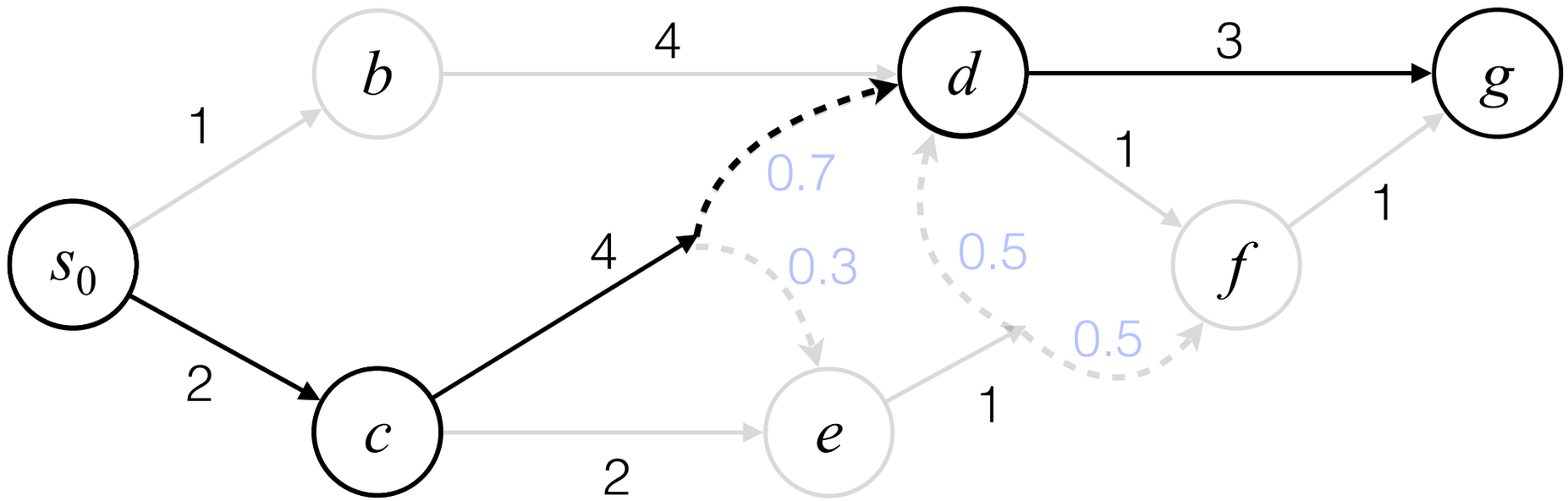
$s_0 \searrow c \nearrow d$

Stochastic Shortest Path
via trial-and-error



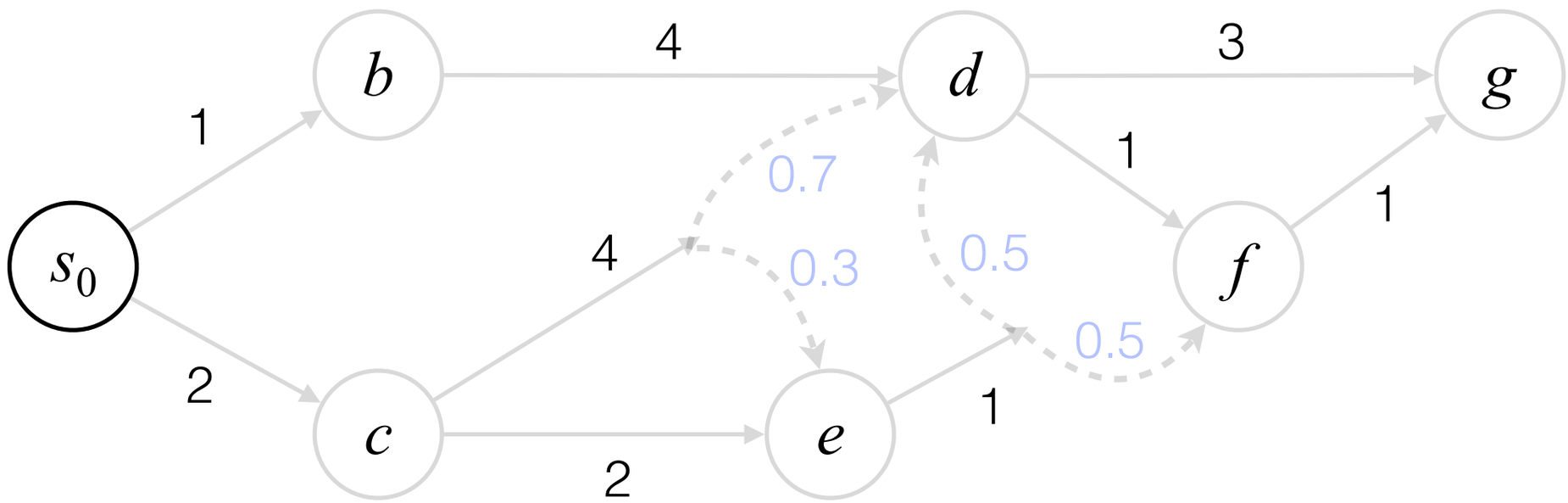
$s_0 \searrow c \nearrow d \rightarrow g$

Stochastic Shortest Path
via trial-and-error



Trajectory 1: $s_0 \searrow c \nearrow d \rightarrow g$

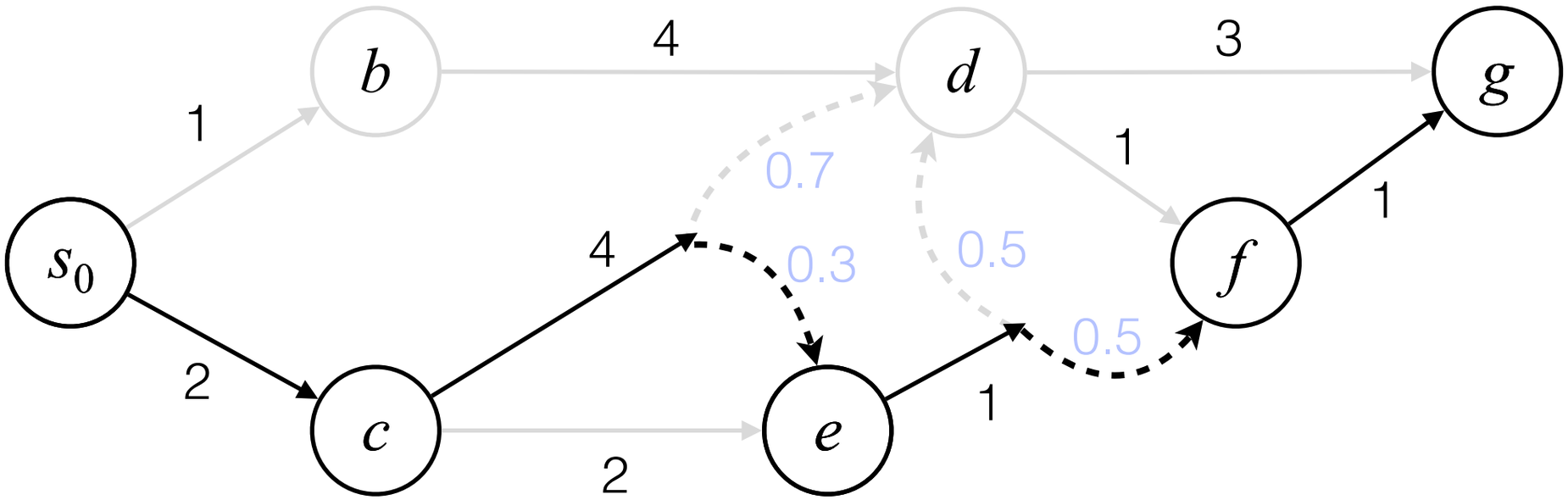
Stochastic Shortest Path
via trial-and-error



Trajectory 1: $s_0 \searrow c \nearrow d \rightarrow g$

Trajectory 2:

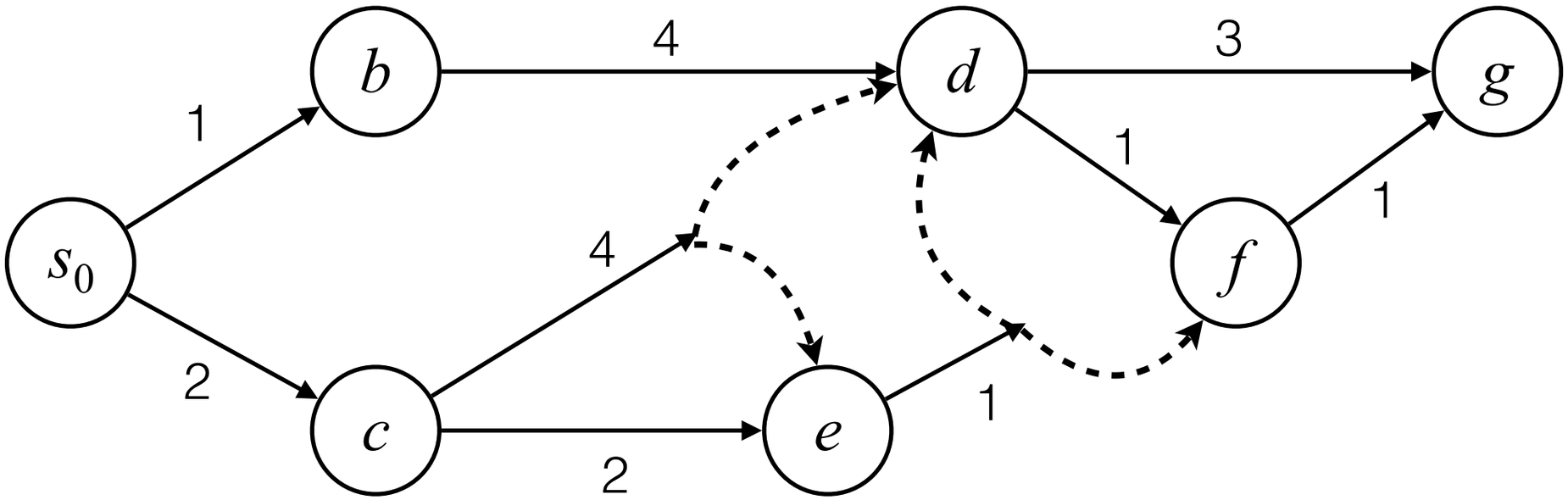
Stochastic Shortest Path
via trial-and-error



Trajectory 1: $s_0 \searrow c \nearrow d \rightarrow g$

Trajectory 2: $s_0 \searrow c \nearrow e \rightarrow f \nearrow g$

Stochastic Shortest Path
via trial-and-error

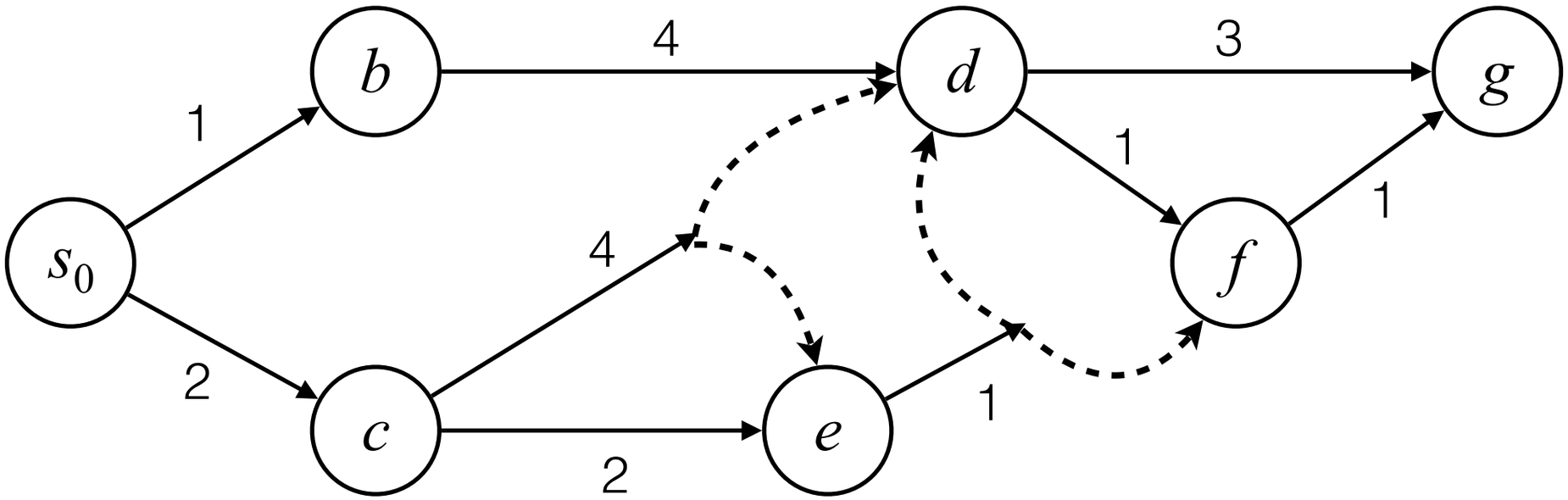


Trajectory 1: $s_0 \searrow c \nearrow d \rightarrow g$

Trajectory 2: $s_0 \searrow c \nearrow e \rightarrow f \nearrow g$

...

Stochastic Shortest Path
via trial-and-error

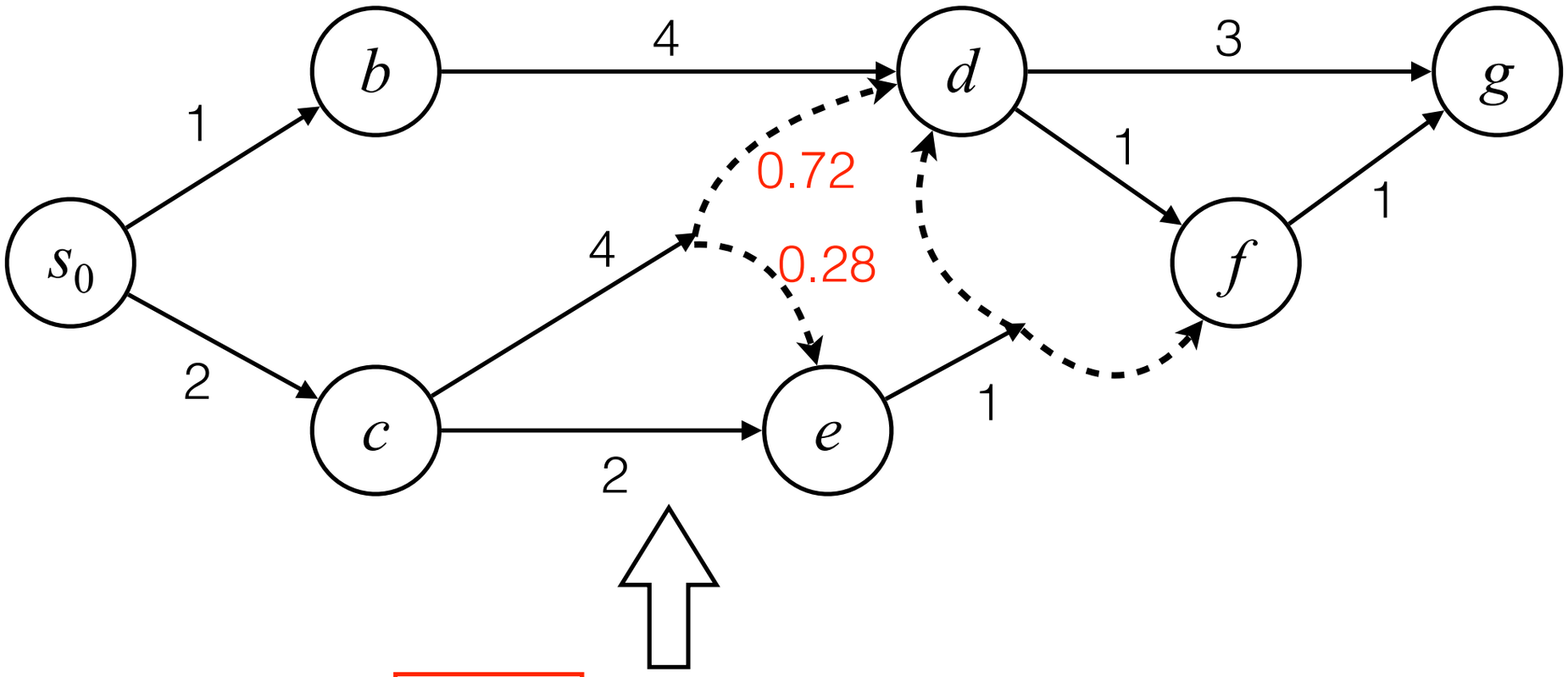


Trajectory 1: $s_0 \searrow \boxed{c \nearrow d} \rightarrow g$

Trajectory 2: $s_0 \searrow \boxed{c \nearrow e} \rightarrow f \nearrow g$

...

Stochastic Shortest Path
via trial-and-error

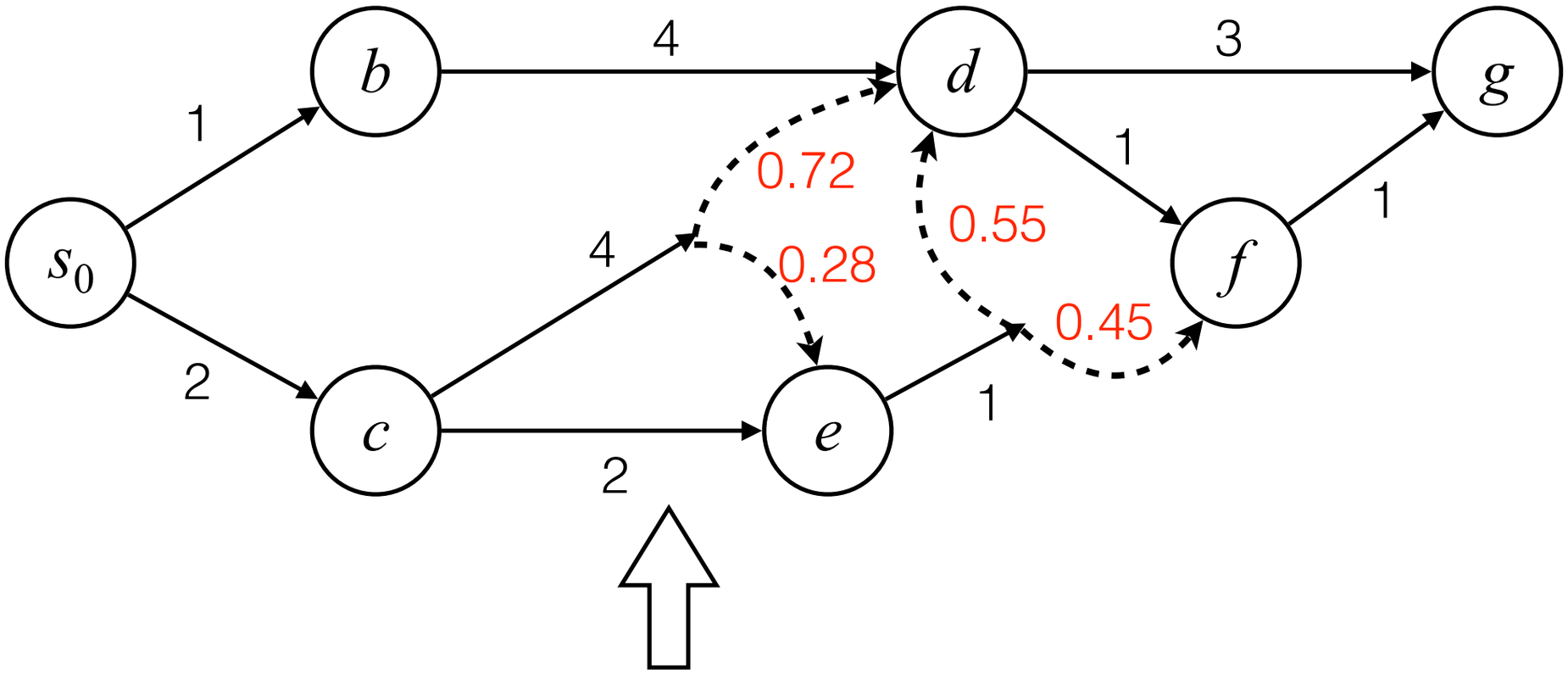


Trajectory 1: $s_0 \searrow c \nearrow d \rightarrow g$

Trajectory 2: $s_0 \searrow c \nearrow e \rightarrow f \nearrow g$

...

Stochastic Shortest Path
via trial-and-error

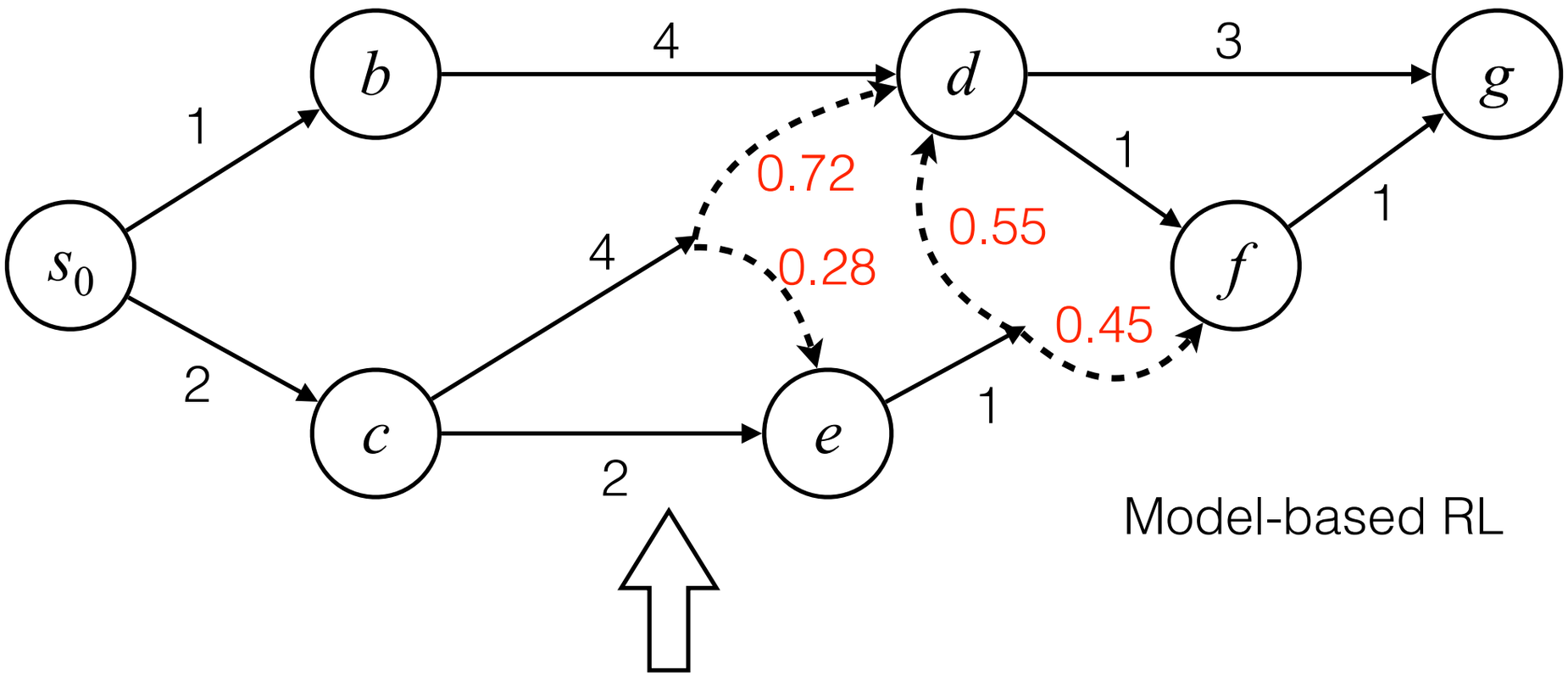


Trajectory 1: $s_0 \searrow c \nearrow d \rightarrow g$

Trajectory 2: $s_0 \searrow c \nearrow e \rightarrow f \nearrow g$

...

Stochastic Shortest Path
via trial-and-error



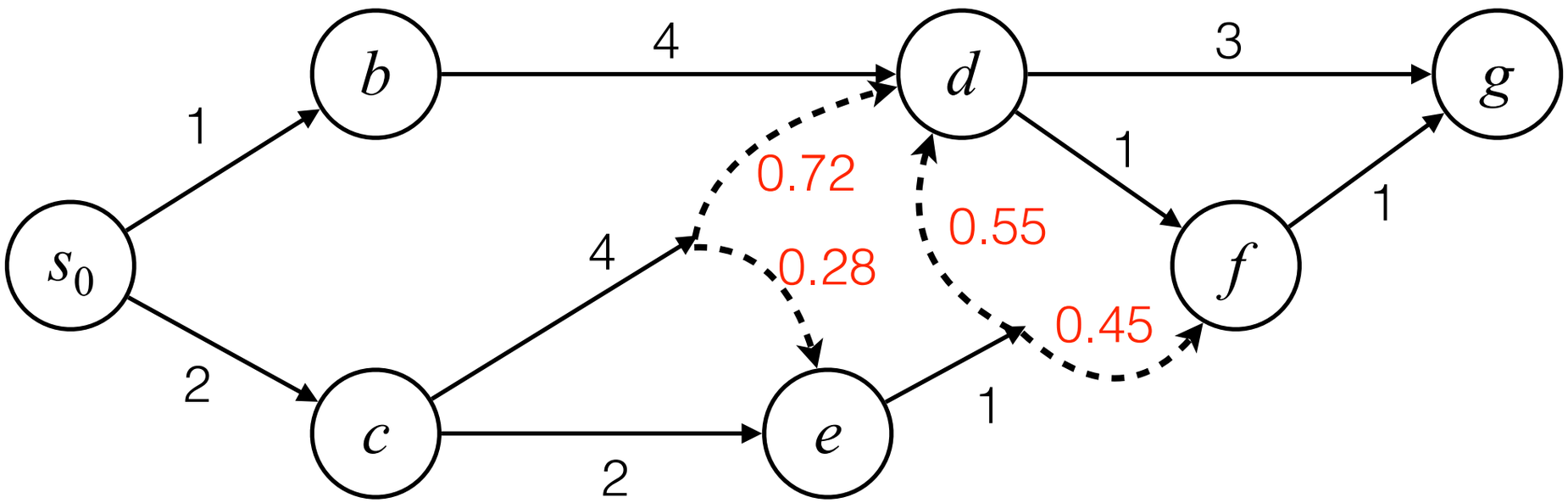
Model-based RL

Trajectory 1: $s_0 \searrow c \nearrow d \rightarrow g$

Trajectory 2: $s_0 \searrow c \nearrow e \rightarrow f \nearrow g$

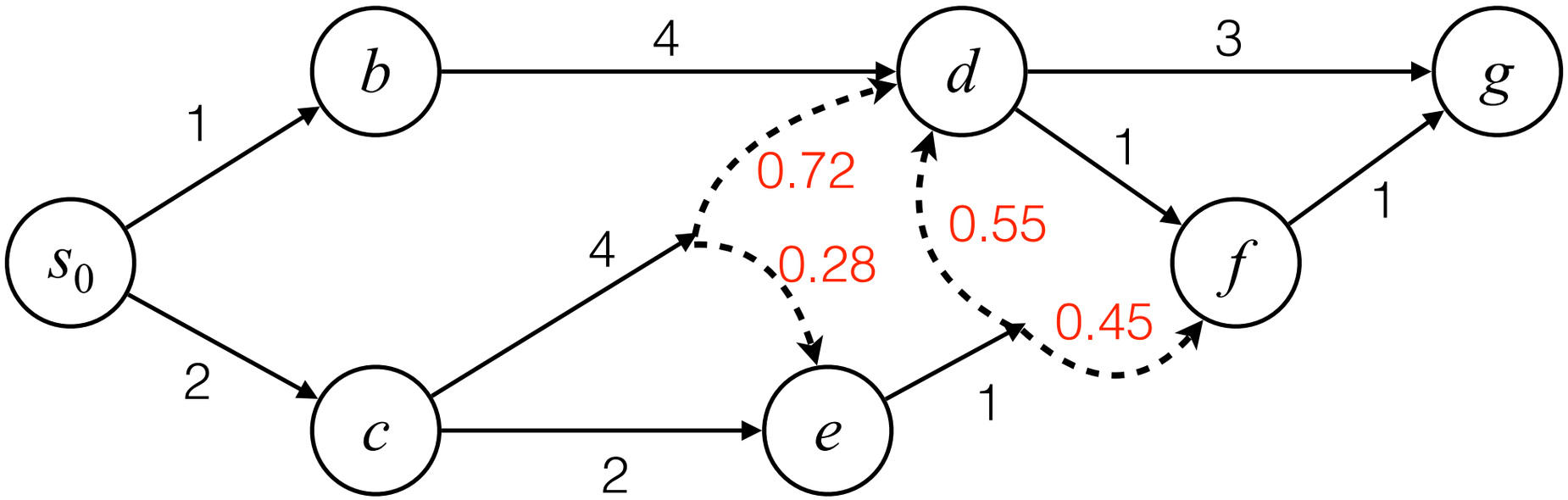
...

Stochastic Shortest Path
via trial-and-error



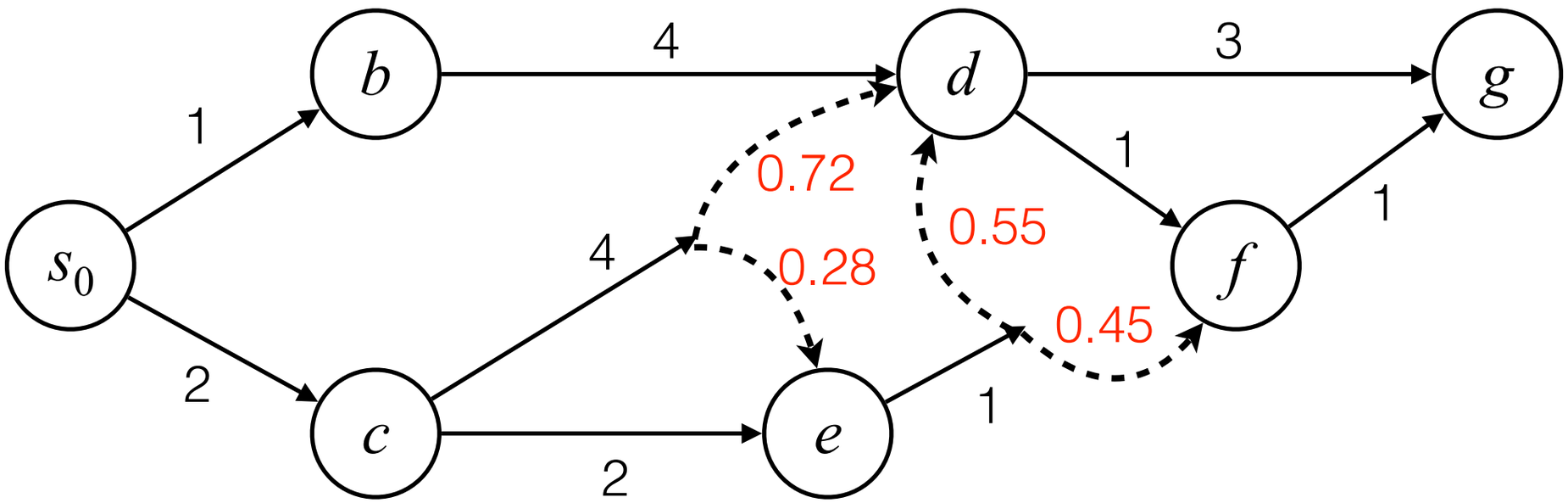
- Assume states & actions are visited uniformly

Stochastic Shortest Path
via trial-and-error



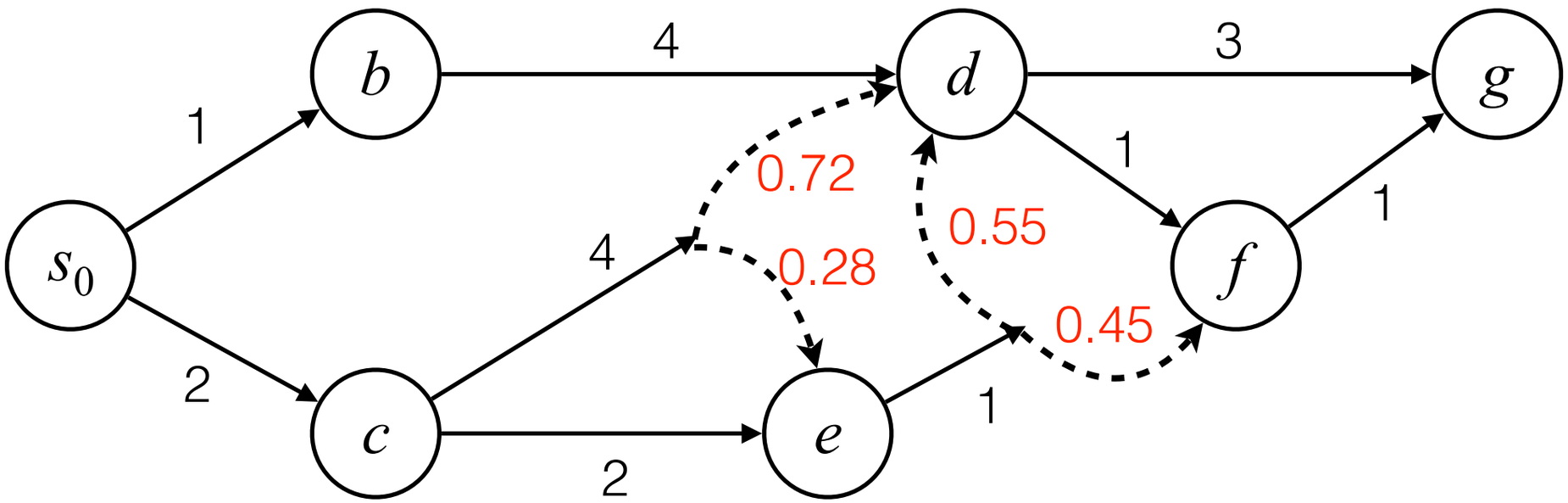
- Assume states & actions are visited uniformly
- As we have more data, the estimated distributions will be closer to true distributions, and the learned policy will be closer to optimal

Stochastic Shortest Path
via trial-and-error



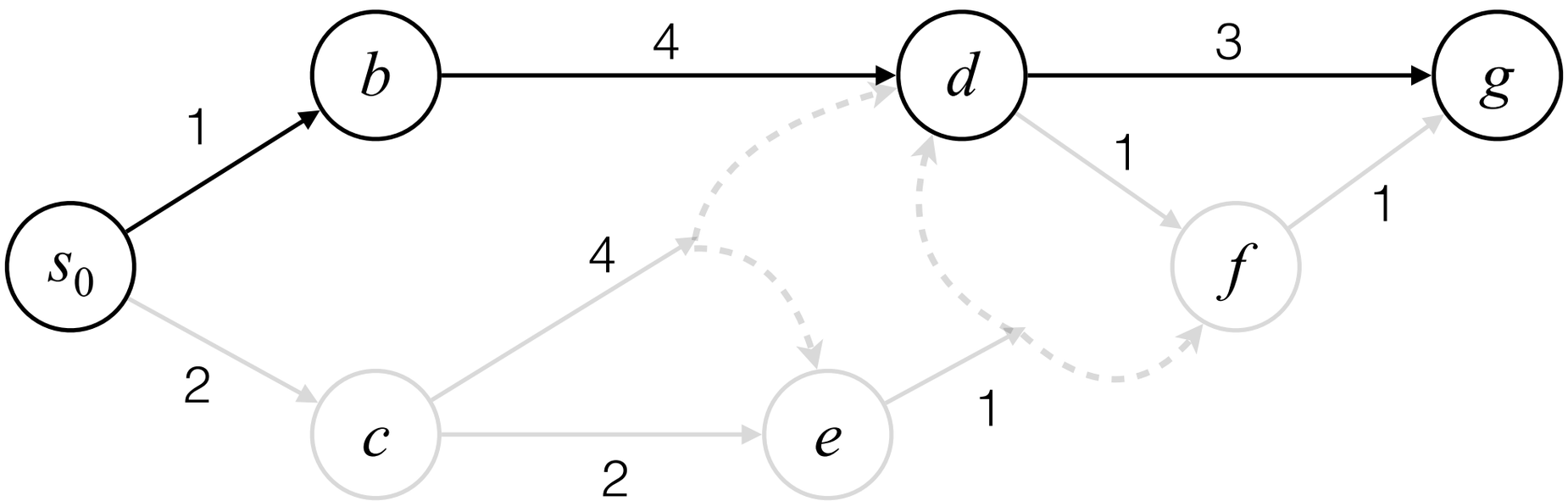
- Assume states & actions are visited uniformly
- As we have more data, the estimated distributions will be closer to true distributions, and the learned policy will be closer to optimal
- Sample complexity: the amount of data needed to guarantee that a certain level of near-optimality is achieved

Stochastic Shortest Path
via trial-and-error



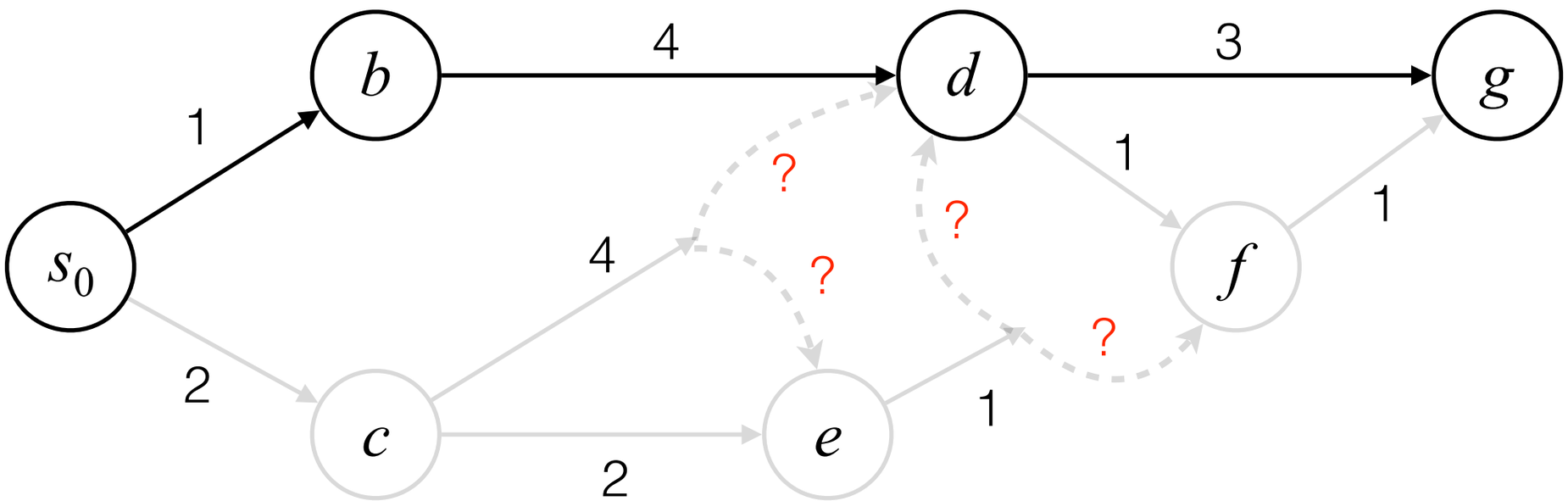
- Assume states & actions are visited uniformly
- As we have more data, the estimated distributions will be closer to true distributions, and the learned policy will be closer to optimal
- Sample complexity: the amount of data needed to guarantee that a certain level of near-optimality is achieved

Stochastic Shortest Path
via trial-and-error



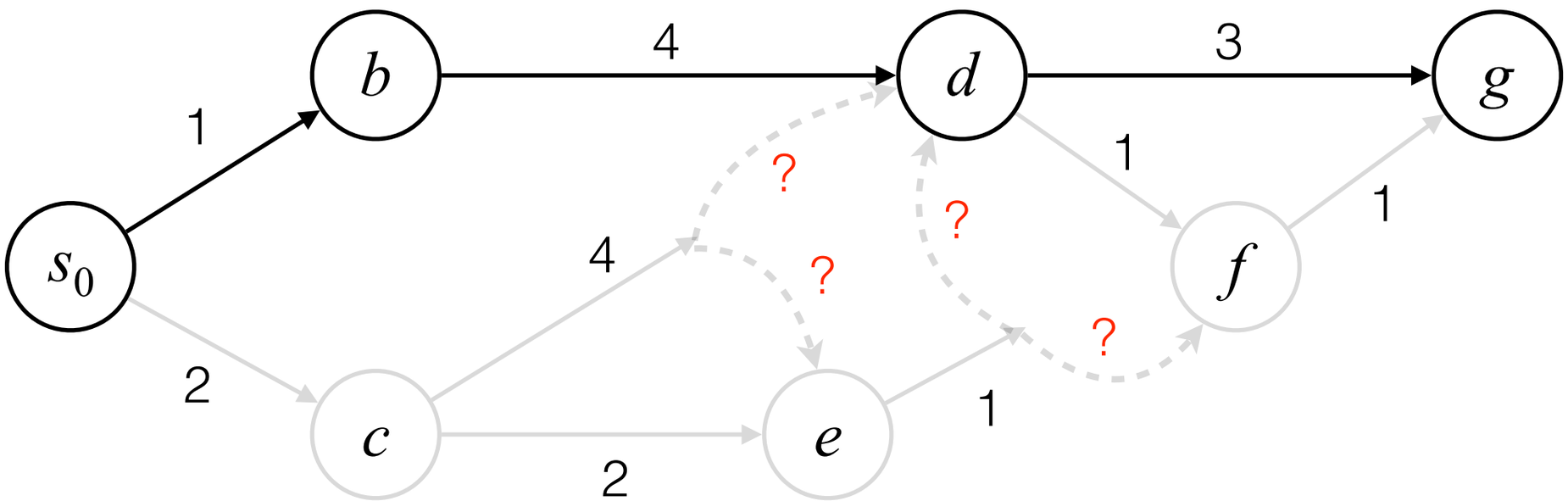
- Assume states & actions are visited uniformly
- As we have more data, the estimated distributions will be closer to true distributions, and the learned policy will be closer to optimal
- Sample complexity: the amount of data needed to guarantee that a certain level of near-optimality is achieved

Stochastic Shortest Path
via trial-and-error



- Assume states & actions are visited uniformly
- As we have more data, the estimated distributions will be closer to true distributions, and the learned policy will be closer to optimal
- Sample complexity: the amount of data needed to guarantee that a certain level of near-optimality is achieved

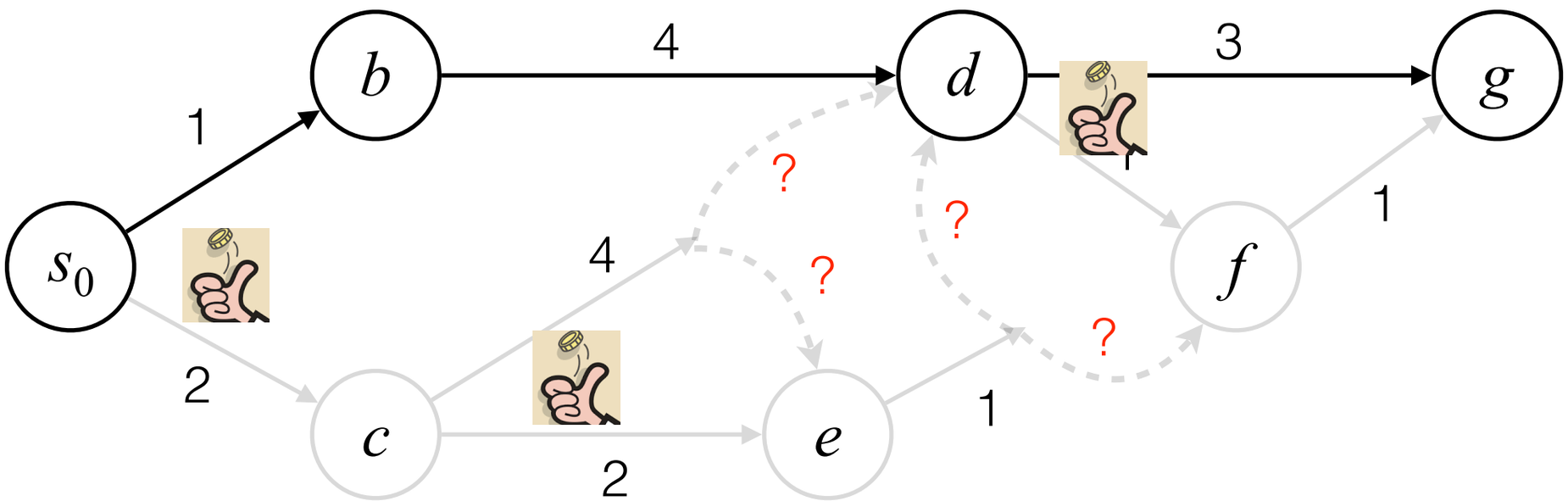
Stochastic Shortest Path
via trial-and-error



Nontrivial! Need **exploration**

- Assume states & actions are visited uniformly
- As we have more data, the estimated distributions will be closer to true distributions, and the learned policy will be closer to optimal
- Sample complexity: the amount of data needed to guarantee that a certain level of near-optimality is achieved

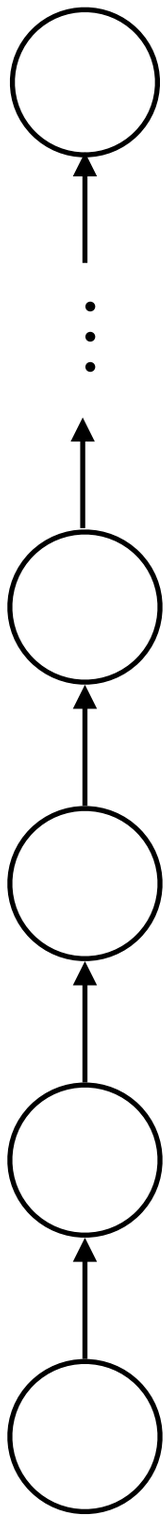
Stochastic Shortest Path
via trial-and-error



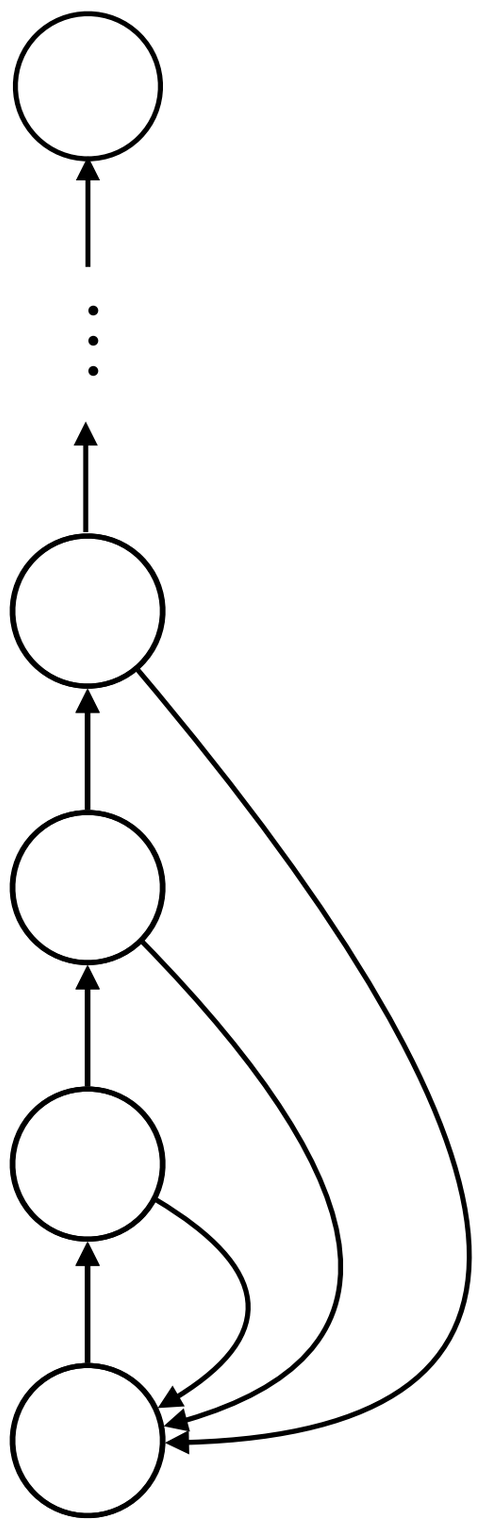
Nontrivial! Need **exploration**

- Assume states & actions are visited uniformly
- As we have more data, the estimated distributions will be closer to true distributions, and the learned policy will be closer to optimal
- Sample complexity: the amount of data needed to guarantee that a certain level of near-optimality is achieved

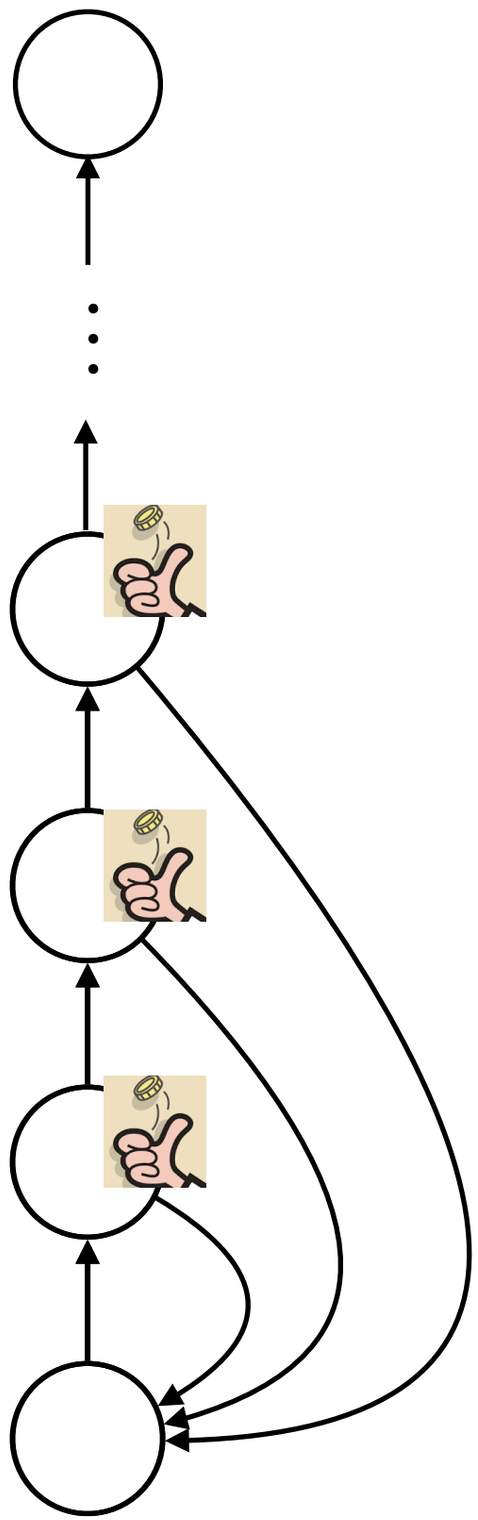
Random exploration can be inefficient



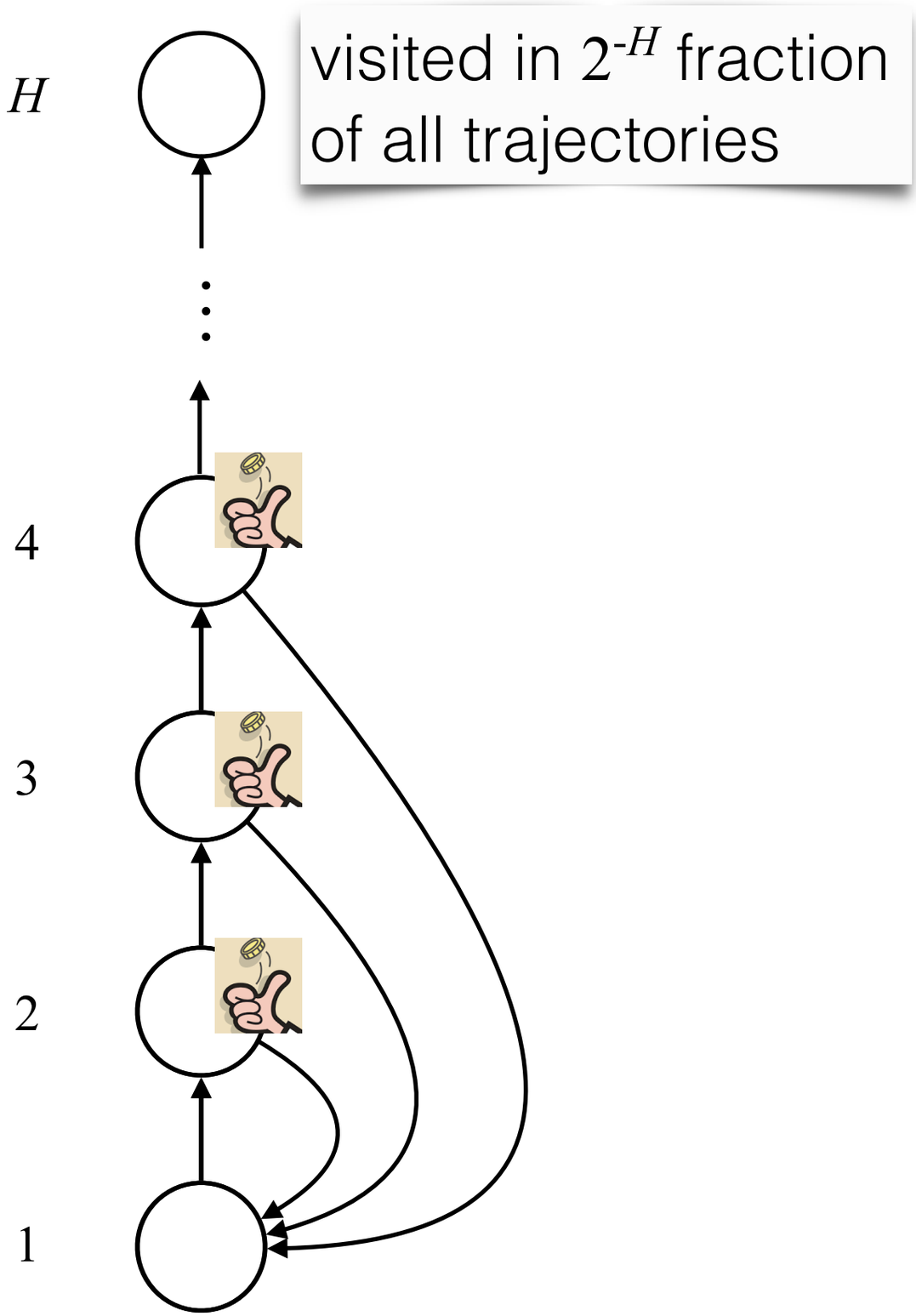
Random exploration can be inefficient



Random exploration can be inefficient

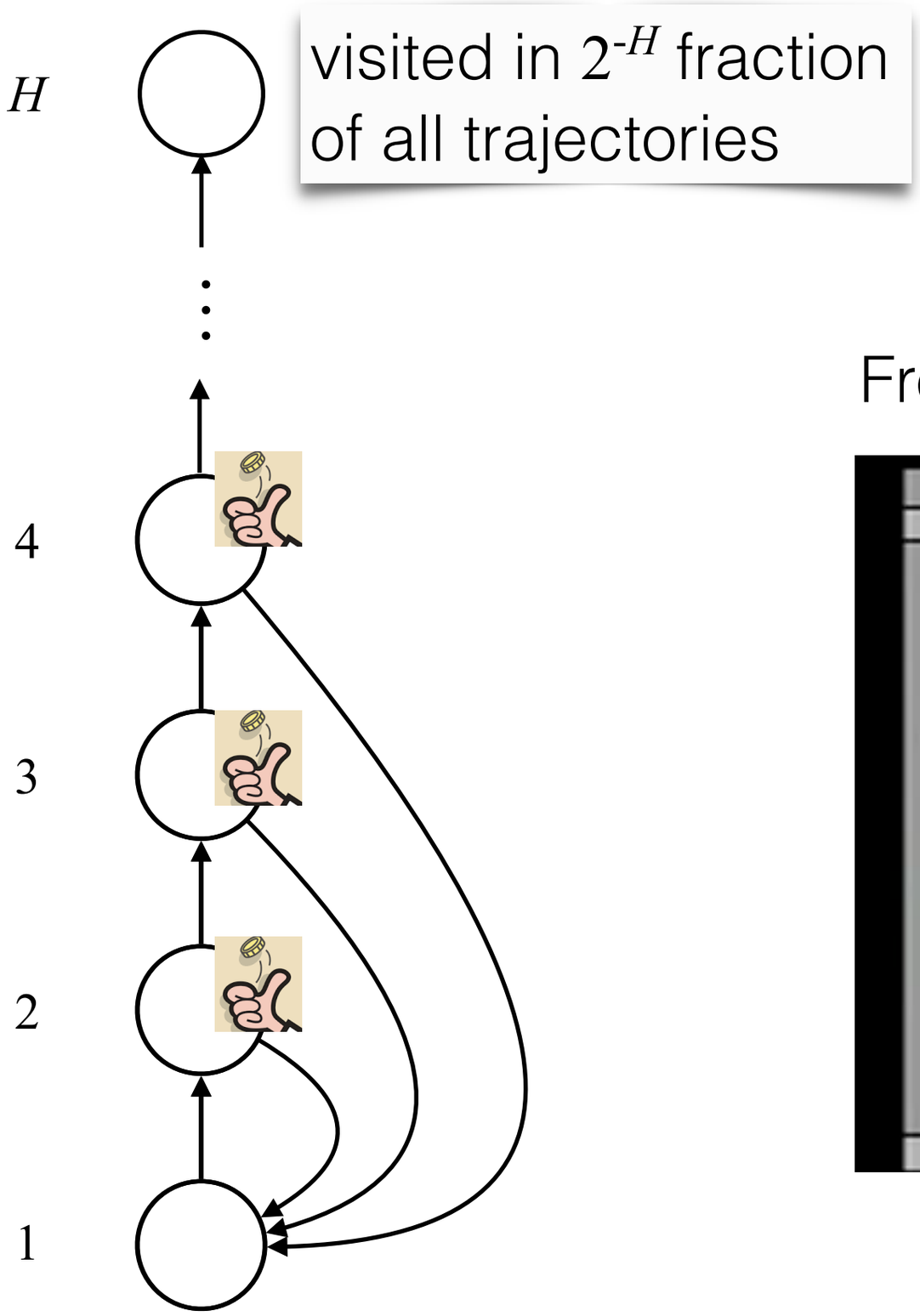


Random exploration can be inefficient

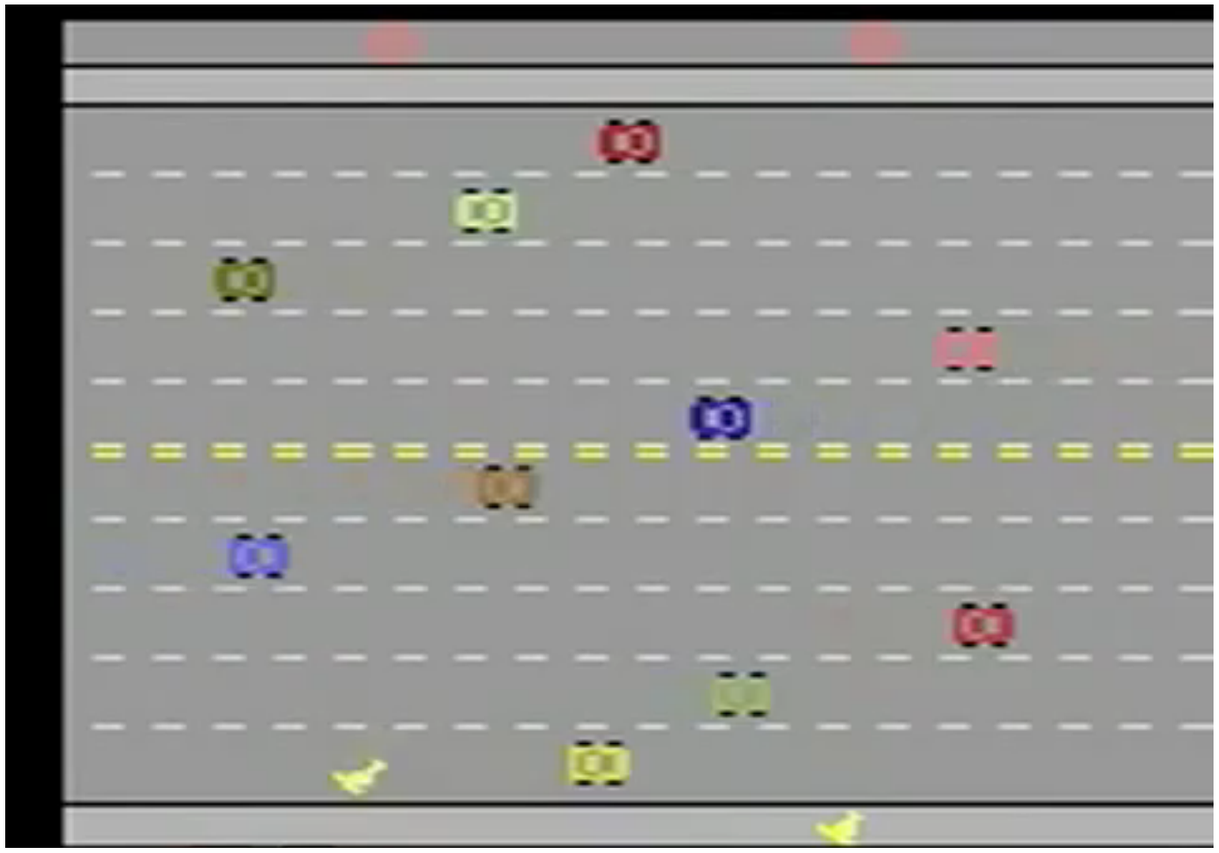


visited in 2^{-H} fraction of all trajectories

Random exploration can be inefficient

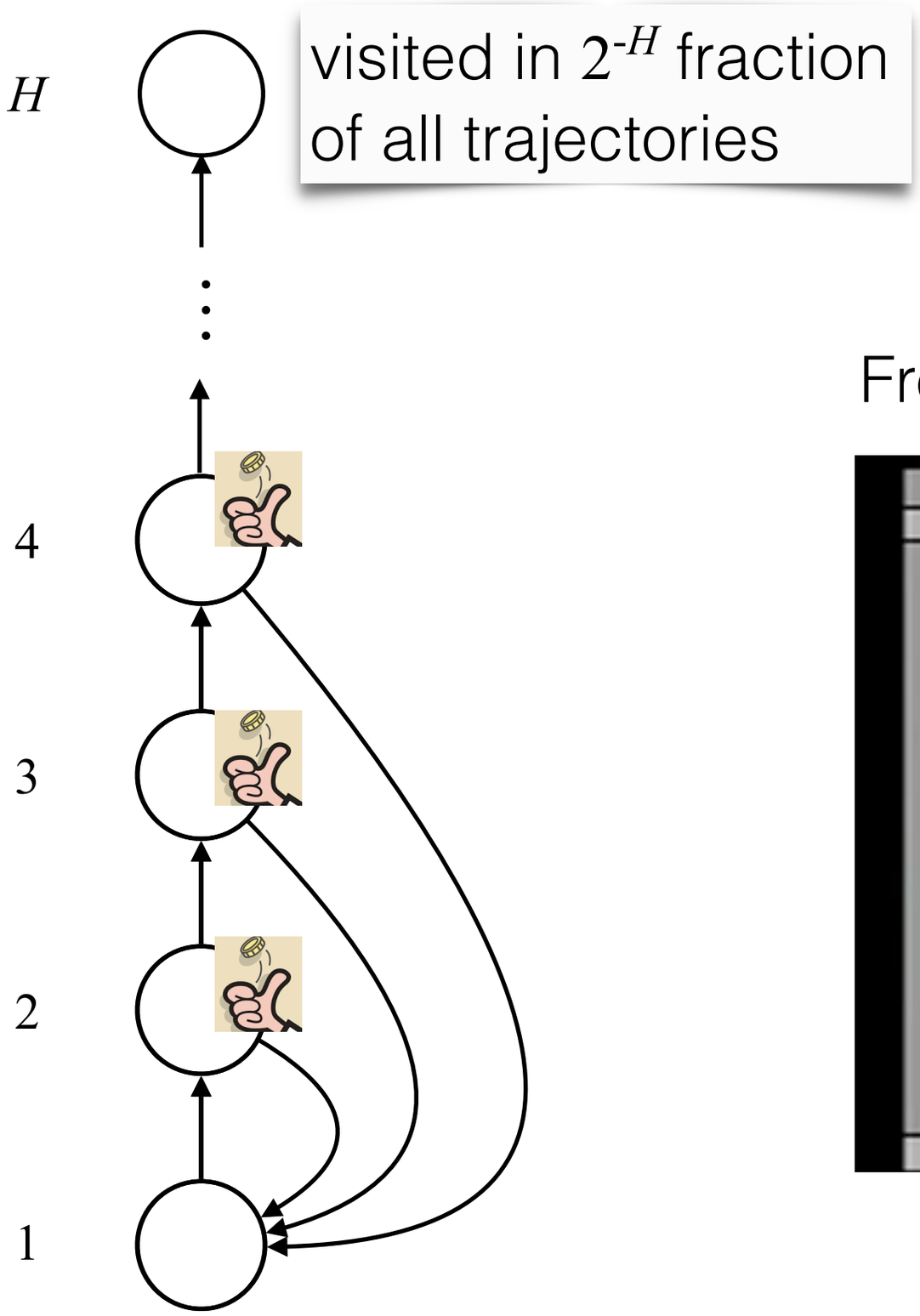


Freeway (one of the Atari games)

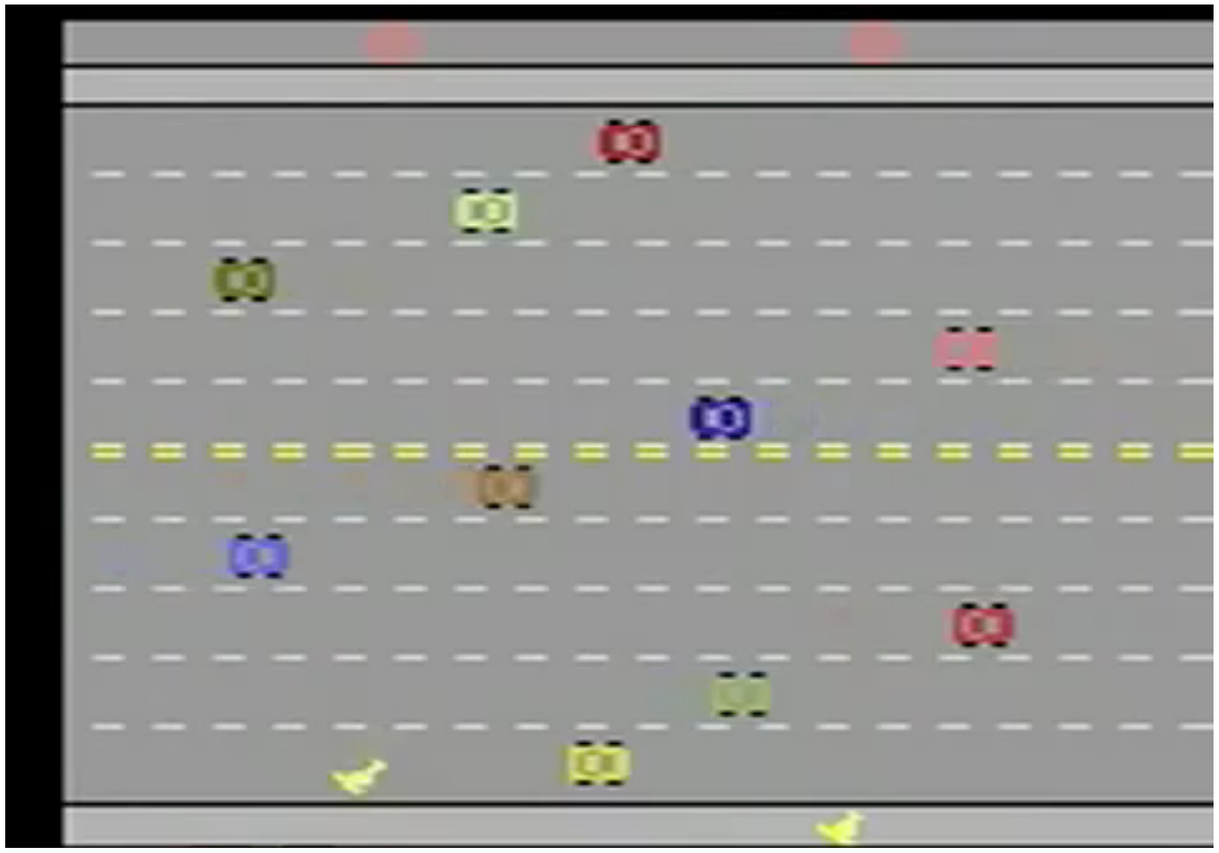


"Freeway + RL": https://youtu.be/44CilPmlimQ

Random exploration can be inefficient



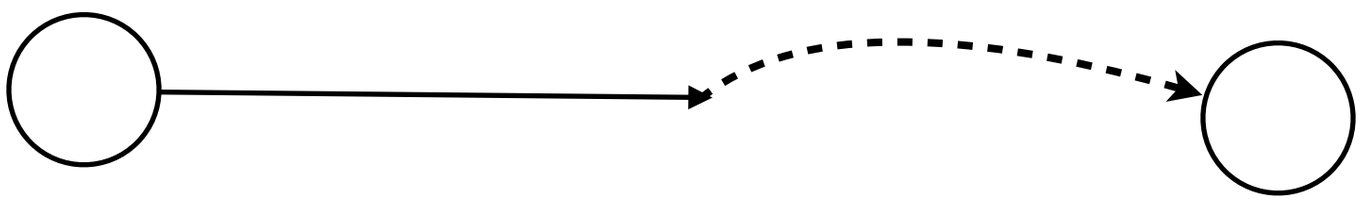
Freeway (one of the Atari games)



"Freeway + RL": https://youtu.be/44CilPmlimQ

Video game playing

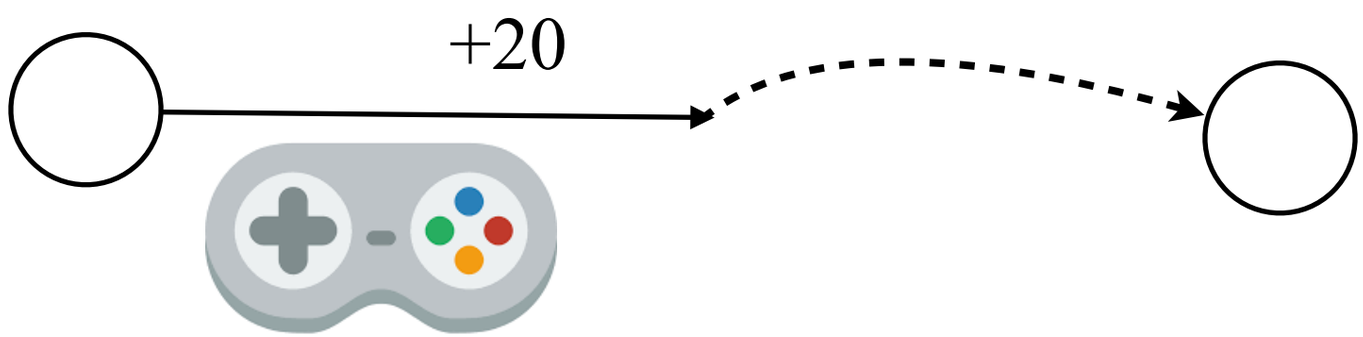
state
$s_t \in S$



Video game playing

reward $r_t = R\,(s_t\ ,\ a_t)$

state $s_t \in S$

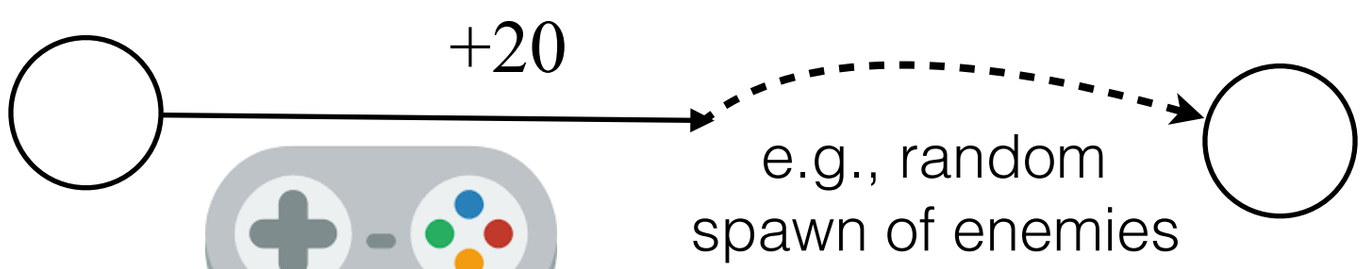


action $a_t \in A$

Video game playing

reward $r_t = R(s_t, a_t)$

state $s_t \in S$



action $a_t \in A$

transition dynamics
$P(\cdot \mid s_t, a_t)$

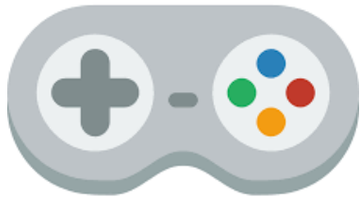
(unknown)

Video game playing

reward $r_t = R\,(s_t\,,\,a_t)$

state
$s_t \in S$

$+20$



e.g., random
spawn of enemies

policy
$\pi\colon S \to A$

action $a_t \in A$

transition dynamics
$P\,(\,\cdot \mid s_t\,,\,a_t)$

(unknown)

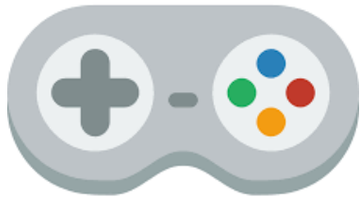
objective: maximize $\mathbb{E}\left[\sum_{t=1}^{H} r_t \mid \pi\right]$

Video game playing

reward $r_t = R(s_t, a_t)$

state
$s_t \in S$

+20



e.g., random
spawn of enemies

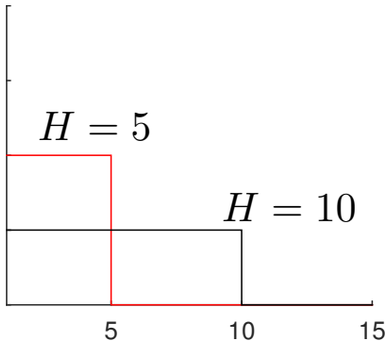
policy
$\pi: S \rightarrow A$

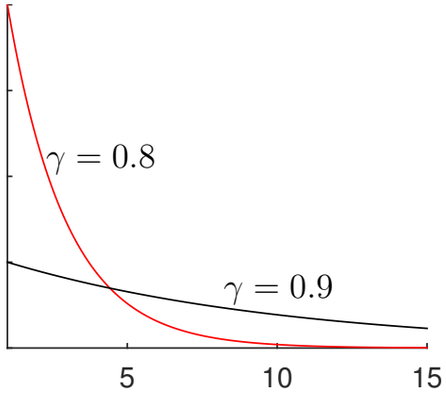
action $a_t \in A$

transition dynamics
$P(\cdot \mid s_t, a_t)$

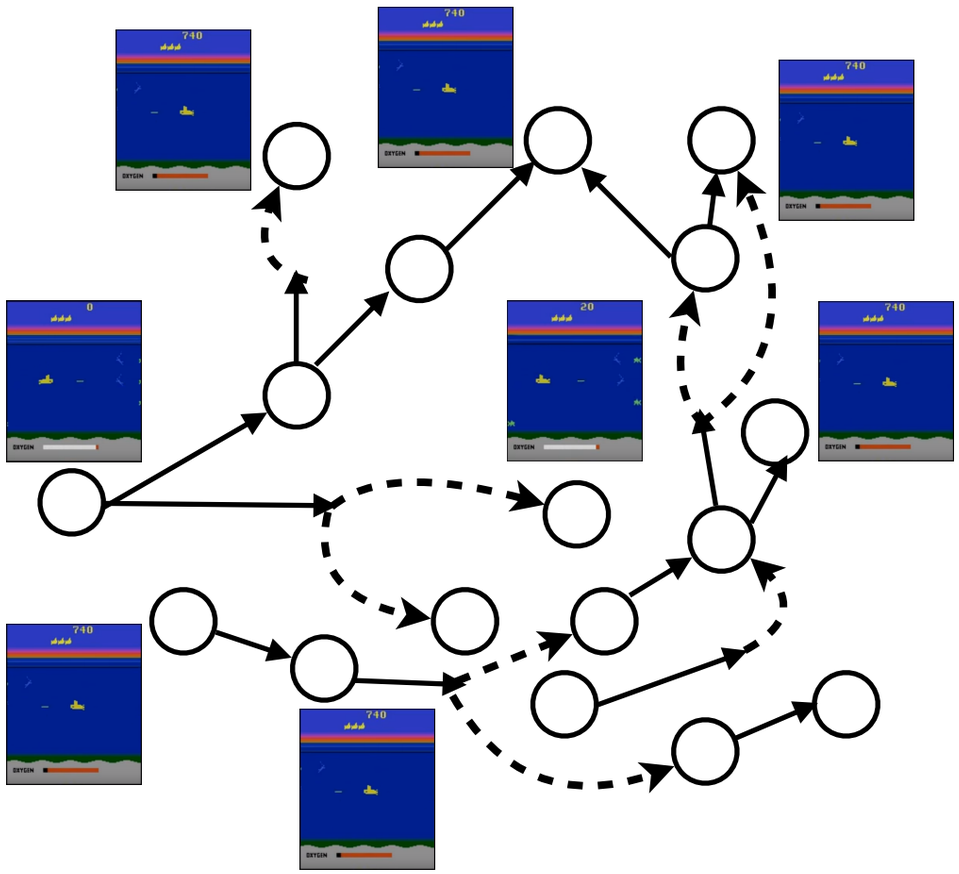
(unknown)

objective: maximize $\mathbb{E}\left[\sum_{t=1}^{H} r_t \mid \pi\right]$ (or $\mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r_t \mid \pi\right]$)

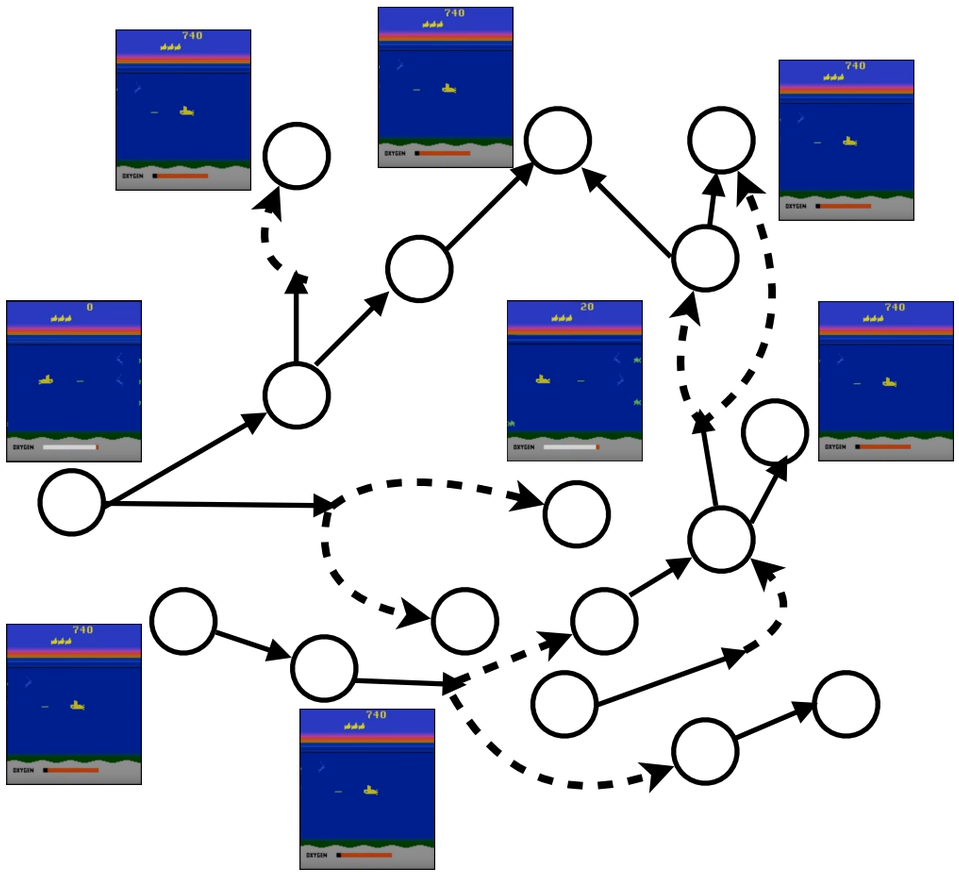




Video game playing

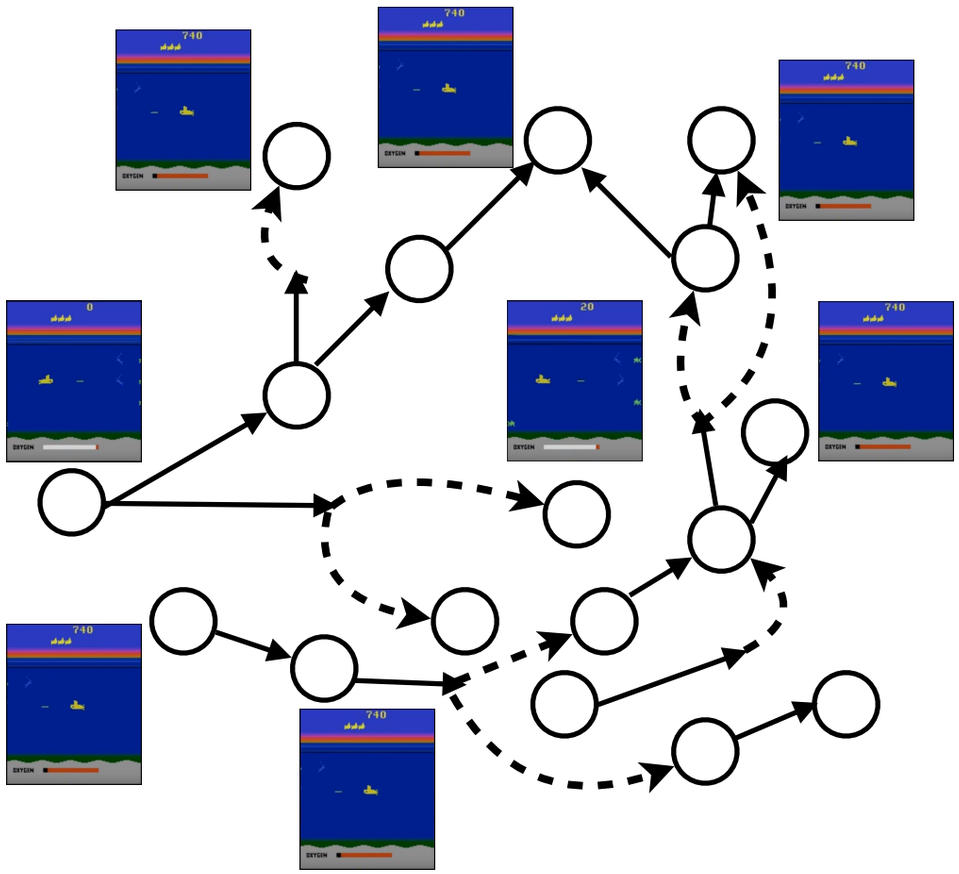


Video game playing



Need **generalization**

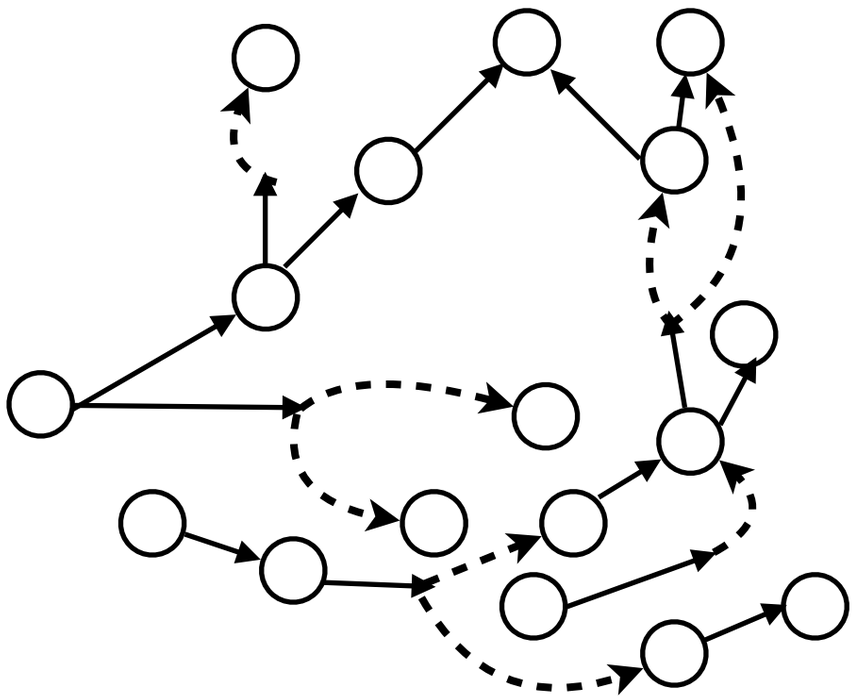
Video game playing

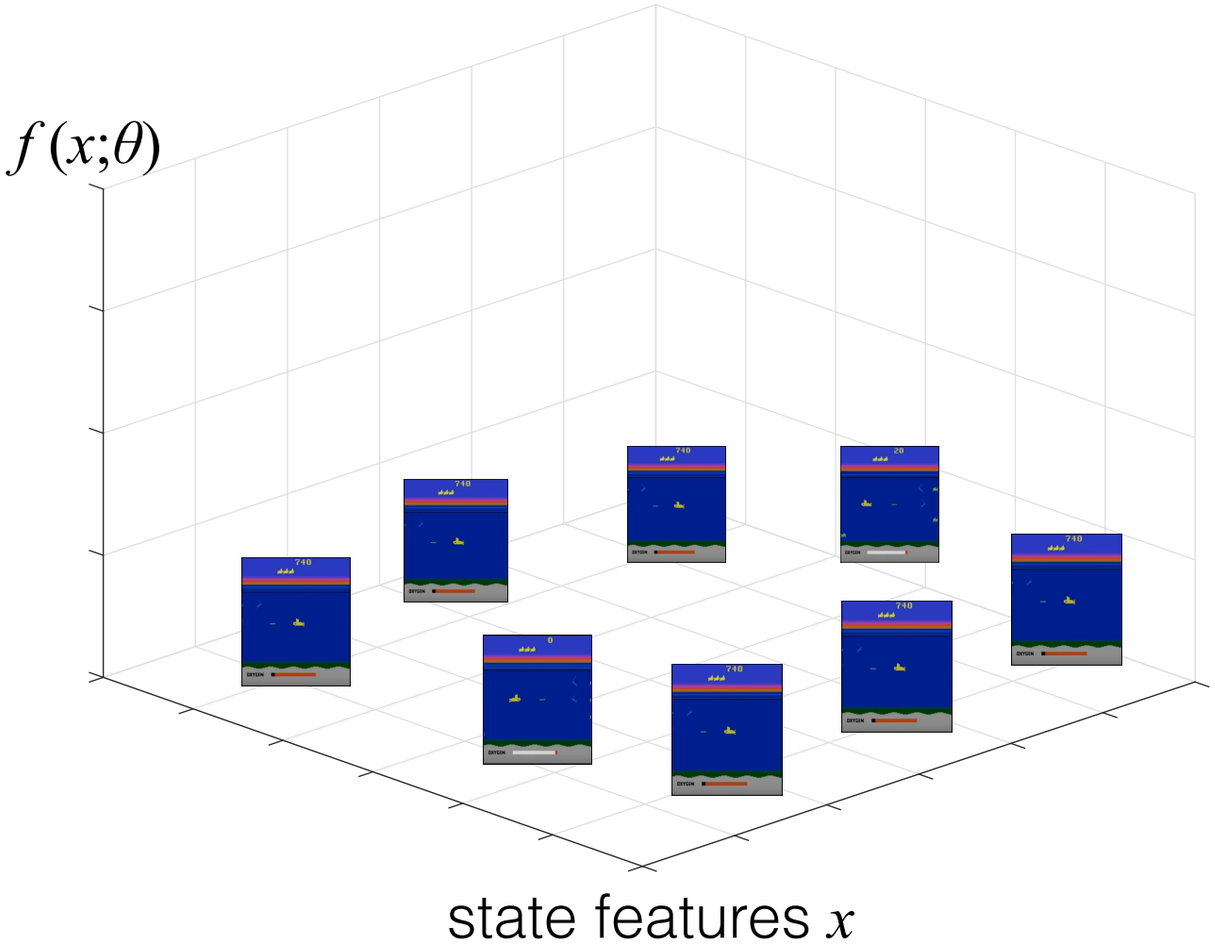


Need **generalization**

Value function approximation

Video game playing

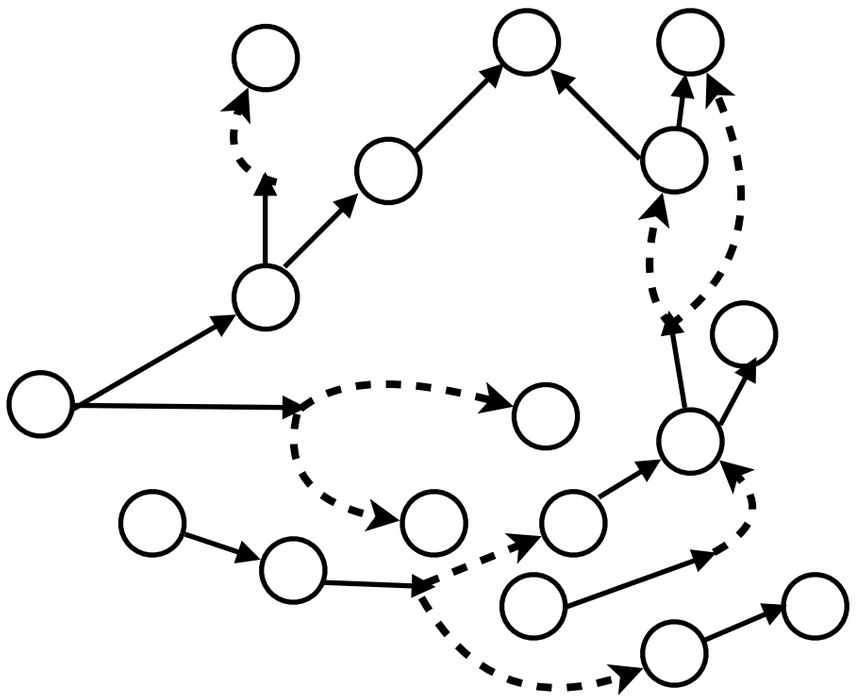


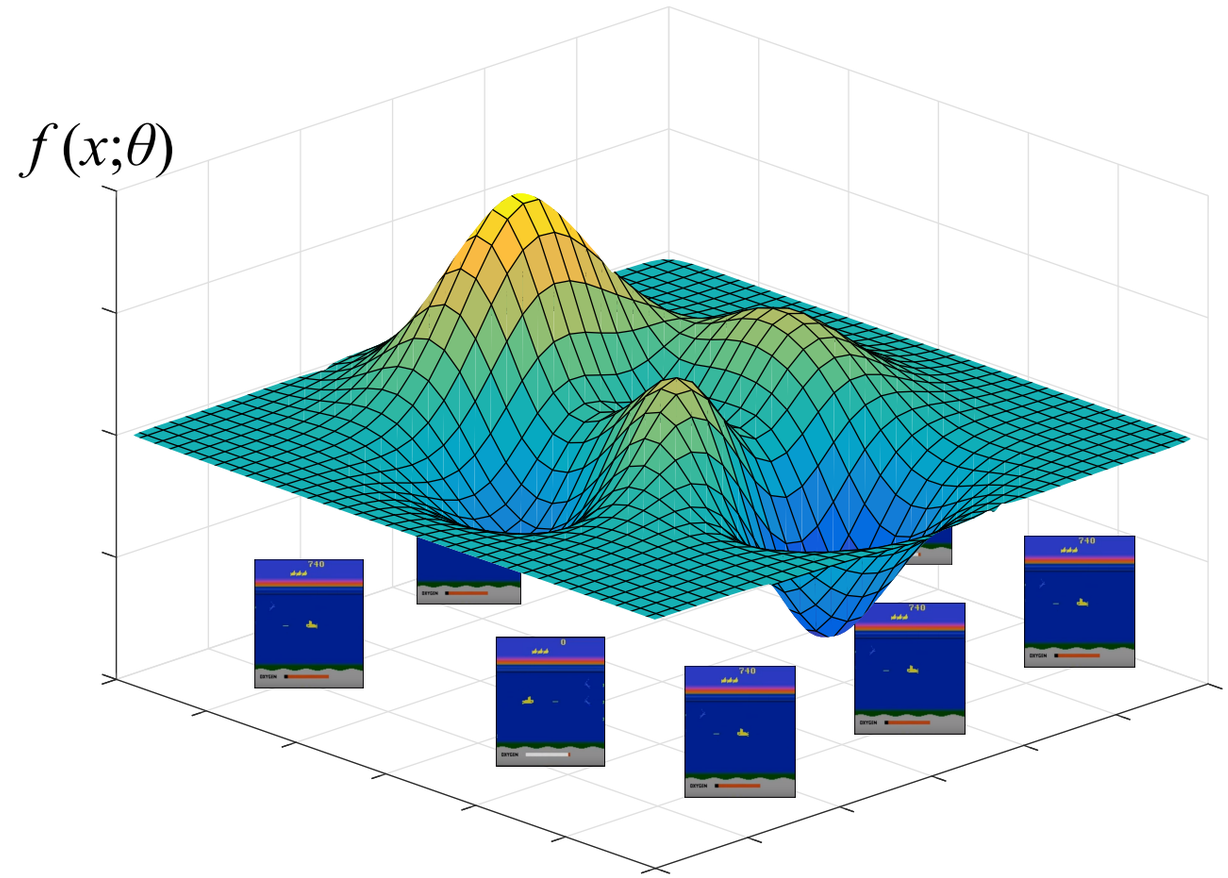


Need **generalization**

Value function approximation

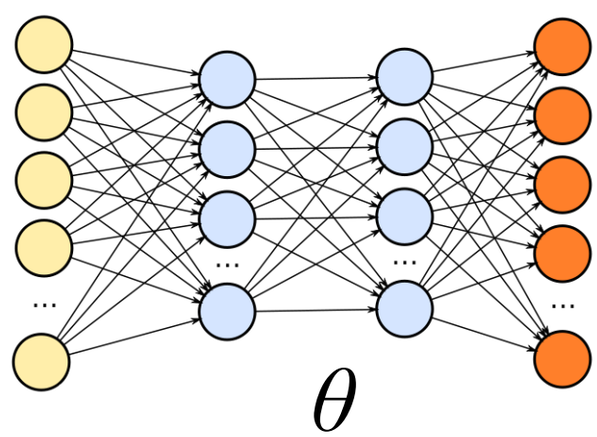
Video game playing





state features x

state features x

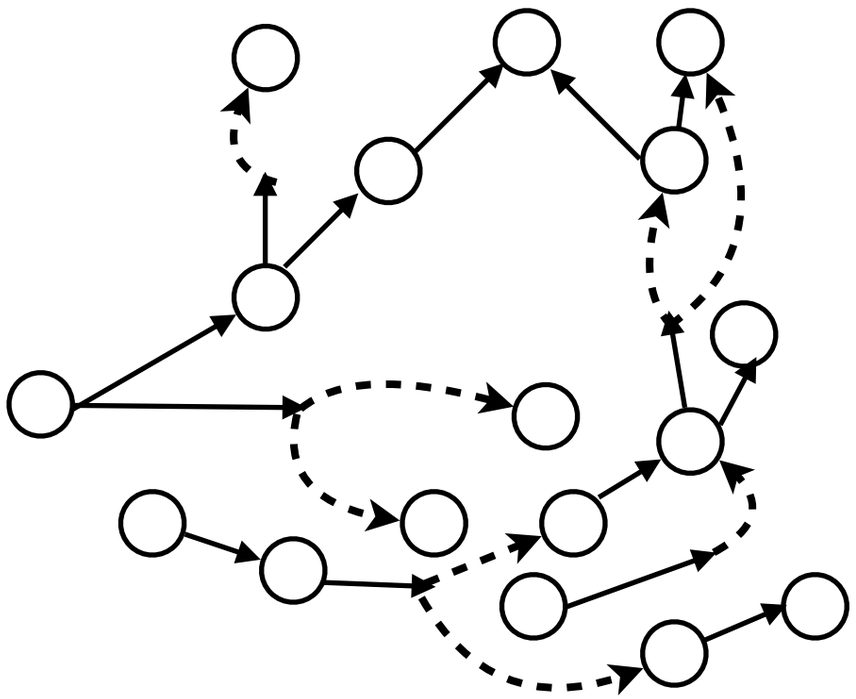


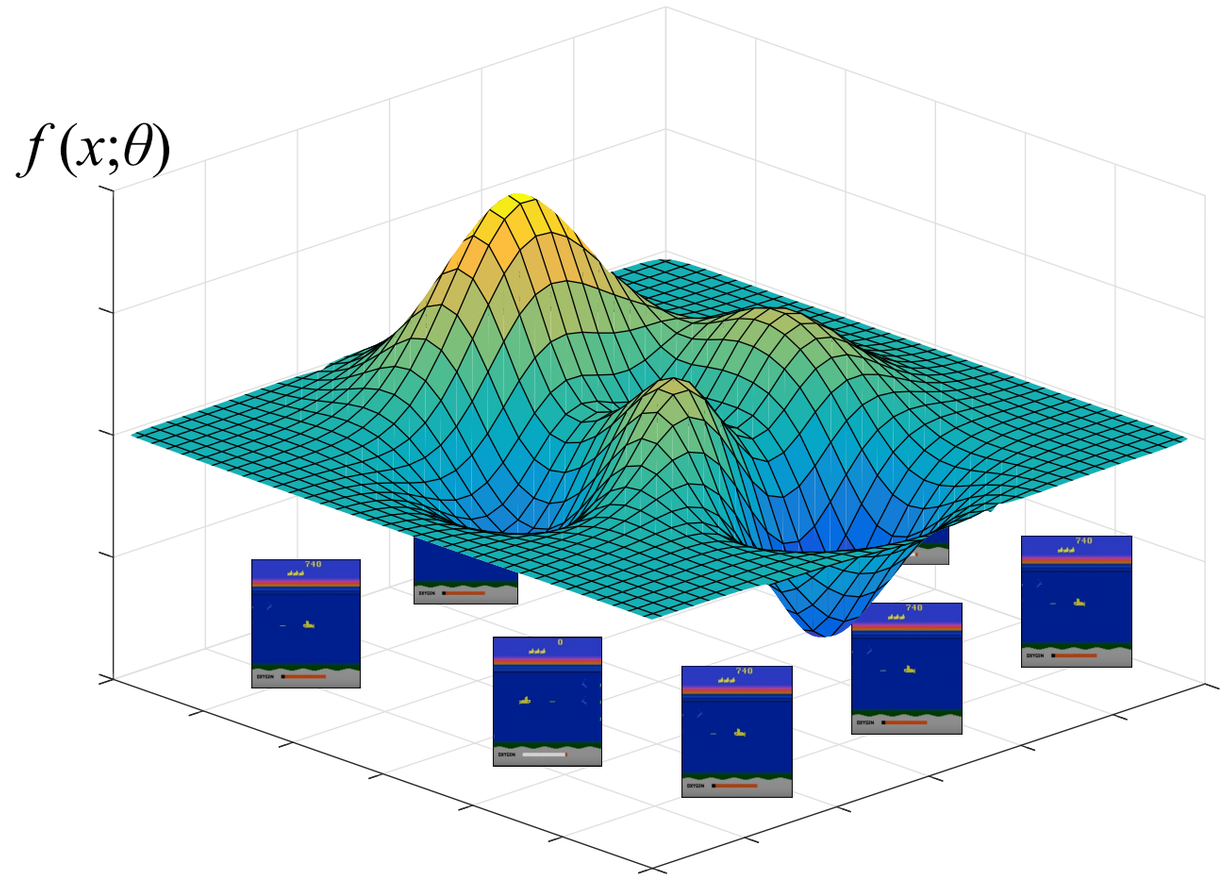
$f(x;\theta)$

Need **generalization**

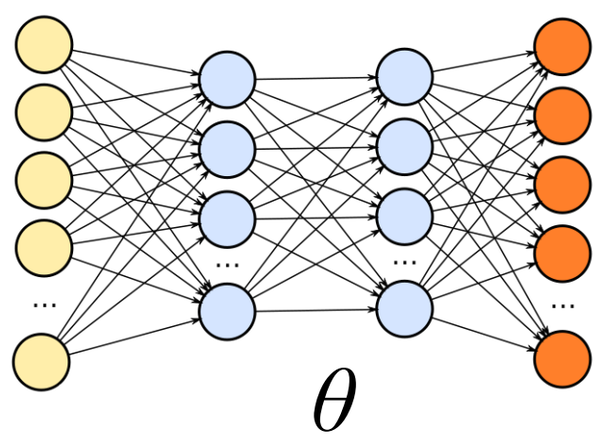
Value function approximation

Video game playing





state features x



$f(x;\theta)$

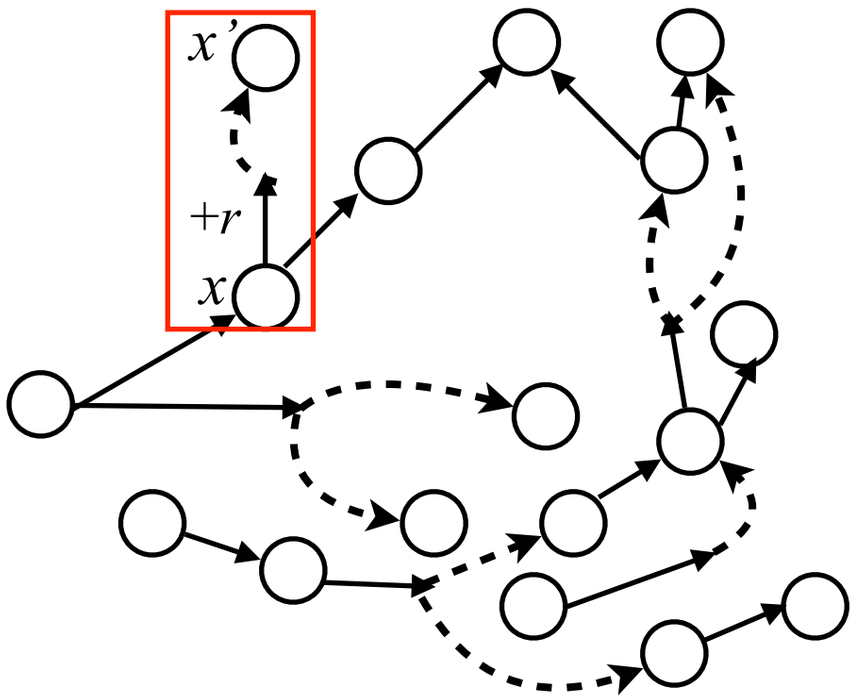
Find θ s.t.

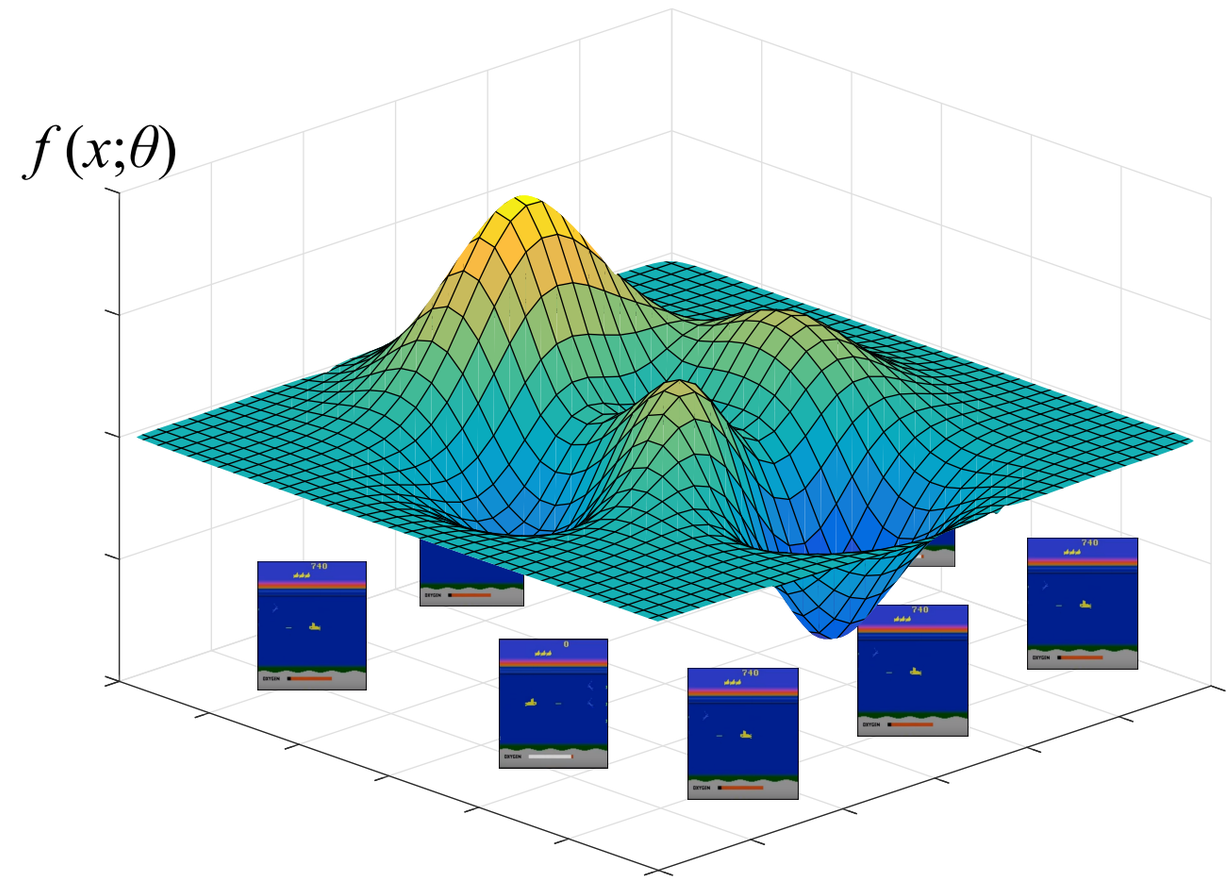
Need **generalization**

Value function approximation

$$f(\cdot\,;\theta) \approx \mathrm{V}^*$$

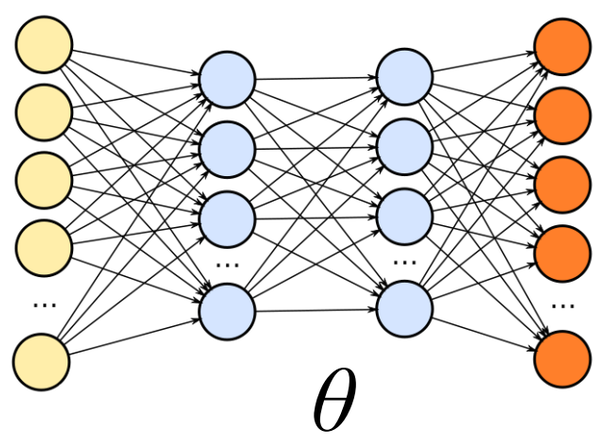
Video game playing





state features x

state features x ⇨



⇨ $f(x;\theta)$

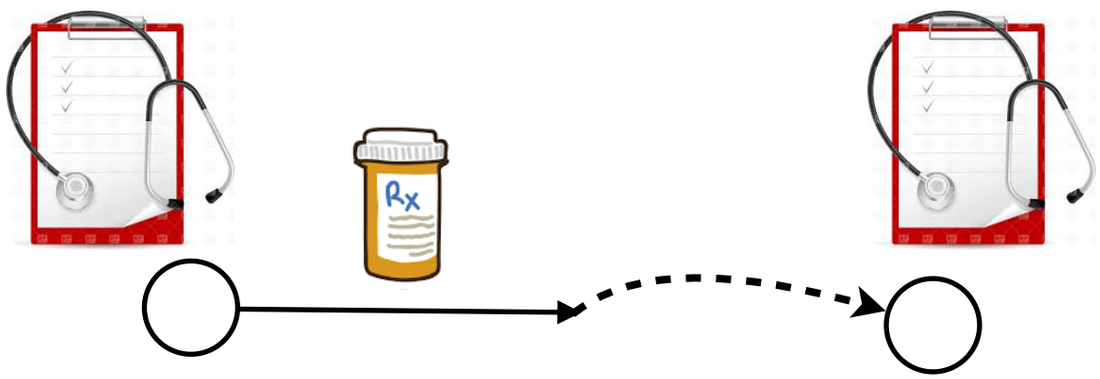
Need **generalization**

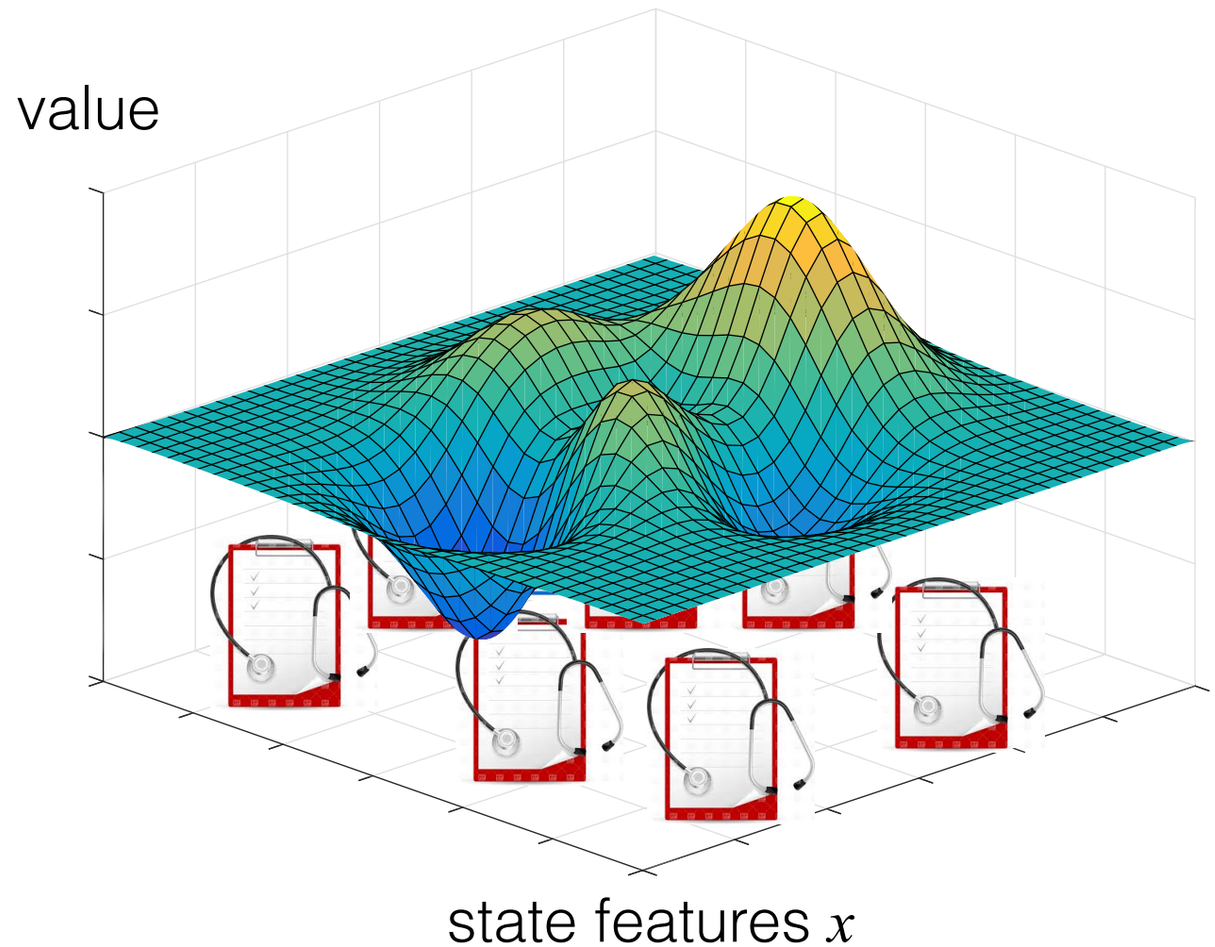
Value function approximation

Find θ s.t. $f(x;\theta) \approx r + \gamma \cdot \mathrm{E}_{x'|x}[f(x';\theta)] \;\Rightarrow\; f(\cdot\,;\theta) \approx \mathrm{V}^*$

$\ldots$

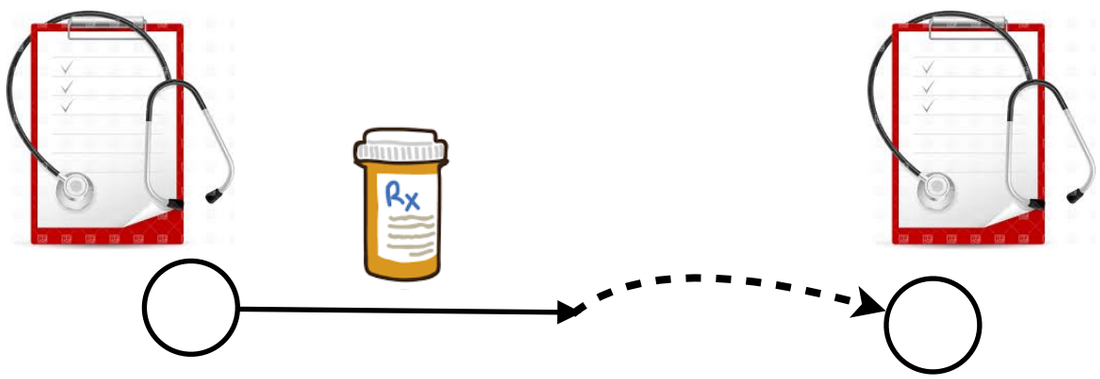
Adaptive medical treatment

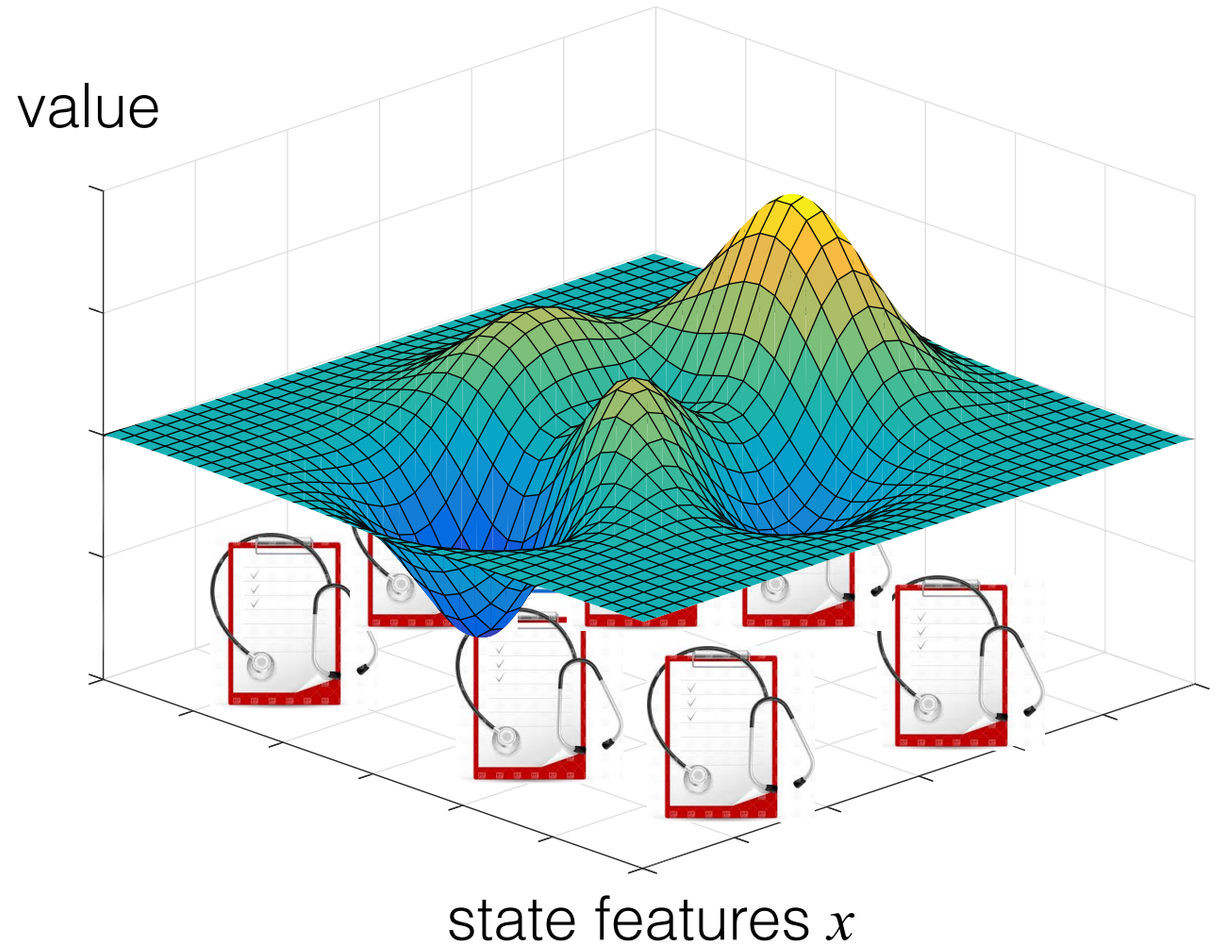




- State: diagnosis
- Action: treatment
- Reward: progress in recovery

Adaptive medical treatment





- State: diagnosis
- Action: treatment
- Reward: progress in recovery

Formulating a real problem in the RL framework is difficult, and defining your state, action, horizon, and reward can be tricky—will come back later in the course.

Summary of this Part: 3 core challenges of RL

Summary of this Part: 3 core challenges of RL

- Temporal credit assignment

Summary of this Part: 3 core challenges of RL

- Temporal credit assignment
  - A sequence of actions led to success/failure: which action(s) to attribute the consequence to?

Summary of this Part: 3 core challenges of RL

- Temporal credit assignment
  - A sequence of actions led to success/failure: which action(s) to attribute the consequence to?
- Exploration

Summary of this Part: 3 core challenges of RL

- Temporal credit assignment
  - A sequence of actions led to success/failure: which action(s) to attribute the consequence to?
- Exploration
  - How to take actions during data collection to guarantee that the dataset provide a comprehensive description of the environment?

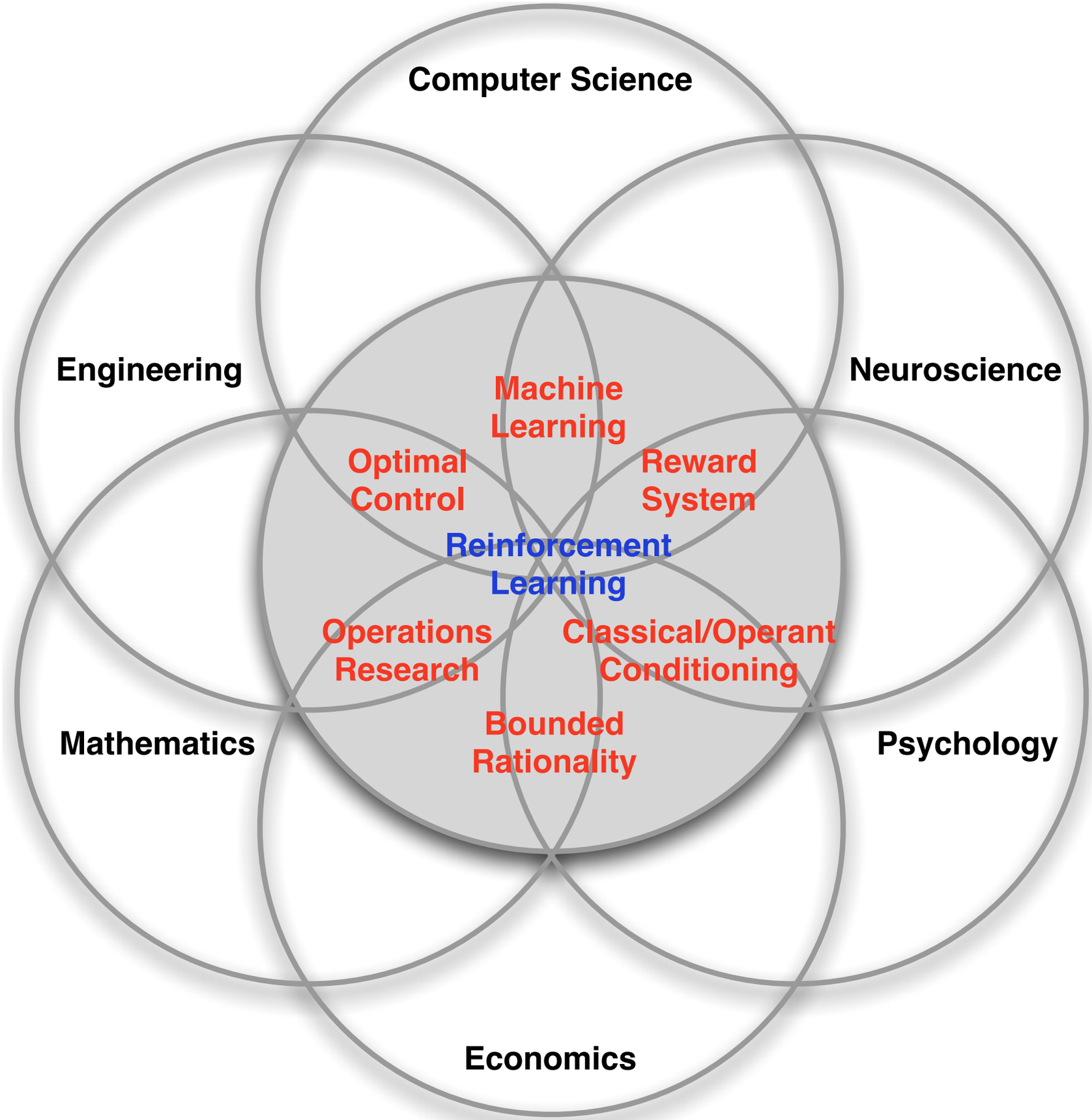
Summary of this Part: 3 core challenges of RL

- Temporal credit assignment
  - A sequence of actions led to success/failure: which action(s) to attribute the consequence to?
- Exploration
  - How to take actions during data collection to guarantee that the dataset provide a comprehensive description of the environment?
- Generalization

Summary of this Part: 3 core challenges of RL

- Temporal credit assignment
  - A sequence of actions led to success/failure: which action(s) to attribute the consequence to?
- Exploration
  - How to take actions during data collection to guarantee that the dataset provide a comprehensive description of the environment?
- Generalization
  - How do deal with (very) large state spaces?

A Machine Learning view of RL



slide credit: David Silver

Supervised Learning

Given $\{(x^{(i)}, y^{(i)})\}$, learn $f : x \mapsto y$

Supervised Learning

Given $\{(x^{(i)}, y^{(i)})\}$, learn $f : x \mapsto y$

- Online version: for round $t = 1, 2, \ldots,$ **the learner**

Supervised Learning

Given $\{(x^{(i)}, y^{(i)})\}$, learn $f : x \mapsto y$

- Online version: for round $t = 1, 2, \ldots,$ **the learner**
  - observes $x^{(t)}$

Supervised Learning

Given $\{(x^{(i)}, y^{(i)})\}$, learn $f : x \mapsto y$

- Online version: for round $t = 1, 2, \ldots,$ **the learner**
  - observes $x^{(t)}$
  - predicts $\hat{y}^{(t)}$

Supervised Learning

Given $\{(x^{(i)}, y^{(i)})\}$, learn $f : x \mapsto y$

- Online version: for round $t = 1, 2, \ldots,$ **the learner**
  - observes $x^{(t)}$
  - predicts $\hat{y}^{(t)}$
  - receives $y^{(t)}$

Supervised Learning

Given $\{(x^{(i)}, y^{(i)})\}$, learn $f : x \mapsto y$

- Online version: for round $t = 1, 2, \ldots,$ **the learner**
  - observes $x^{(t)}$
  - predicts $\hat{y}^{(t)}$
  - receives $y^{(t)}$
- Want to maximize # of correct predictions

Supervised Learning

Given $\{(x^{(i)}, y^{(i)})\}$, learn $f : x \mapsto y$

- Online version: for round $t = 1, 2, \ldots,$ **the learner**
  - observes $x^{(t)}$
  - predicts $\hat{y}^{(t)}$
  - receives $y^{(t)}$
- Want to maximize # of correct predictions
- e.g., classifies if an image is about a dog, a cat, a plane, etc. (multi-class classification)

Supervised Learning

Given $\{(x^{(i)}, y^{(i)})\}$, learn $f : x \mapsto y$

- Online version: for round $t = 1, 2, \ldots,$ **the learner**
  - observes $x^{(t)}$
  - predicts $\hat{y}^{(t)}$
  - receives $y^{(t)}$
- Want to maximize # of correct predictions
- e.g., classifies if an image is about a dog, a cat, a plane, etc. (multi-class classification)
- Dataset is fixed for everyone

Supervised Learning

Given $\{(x^{(i)}, y^{(i)})\}$, learn $f : x \mapsto y$

- Online version: for round $t = 1, 2, \ldots,$ **the learner**
  - observes $x^{(t)}$
  - predicts $\hat{y}^{(t)}$
  - receives $y^{(t)}$
- Want to maximize # of correct predictions
- e.g., classifies if an image is about a dog, a cat, a plane, etc. (multi-class classification)
- Dataset is fixed for everyone
- “Full information setting”

Supervised Learning

Given $\{(x^{(i)}, y^{(i)})\}$, learn $f : x \mapsto y$

- Online version: for round $t = 1, 2, \ldots,$ **the learner**
  - observes $x^{(t)}$
  - predicts $\hat{y}^{(t)}$
  - receives $y^{(t)}$
- Want to maximize # of correct predictions
- e.g., classifies if an image is about a dog, a cat, a plane, etc. (multi-class classification)
- Dataset is fixed for everyone
- “Full information setting”
- Core challenge: generalization

Contextual bandits

For round $t = 1, 2, \ldots,$ **the learner**

Contextual bandits

For round $t = 1, 2, \ldots,$ **the learner**

- Given $x^{(t)}$, chooses from a set of actions $a^{(t)} \in A$

Contextual bandits

For round $t = 1, 2, \ldots,$ **the learner**

- Given $x^{(t)}$, chooses from a set of actions $a^{(t)} \in A$
- Receives reward $r^{(t)} \sim R(x^{(t)}, a^{(t)})$ (i.e., can be random)

Contextual bandits

For round $t = 1, 2, \ldots,$ **the learner**

- Given $x^{(t)}$, chooses from a set of actions $a^{(t)} \in A$
- Receives reward $r^{(t)} \sim R(x^{(t)}, a^{(t)})$ (i.e., can be random)
- Want to maximize total reward

Contextual bandits

For round $t = 1, 2, \ldots,$ **the learner**

- Given $x^{(t)}$, chooses from a set of actions $a^{(t)} \in A$
- Receives reward $r^{(t)} \sim R(x^{(t)}, a^{(t)})$ (i.e., can be random)
- Want to maximize total reward
- You generate your own dataset $\{(x^{(t)}, a^{(t)}, r^{(t)})\}$!

Contextual bandits

For round $t = 1, 2, \ldots,$ **the learner**

- Given $x^{(t)}$, chooses from a set of actions $a^{(t)} \in A$
- Receives reward $r^{(t)} \sim R(x^{(t)}, a^{(t)})$ (i.e., can be random)
- Want to maximize total reward
- You generate your own dataset $\{(x^{(t)}, a^{(t)}, r^{(t)})\}$!
  - e.g., for an image, the learner guesses a label, and is told whether correct or not (reward = 1 if correct and 0 otherwise). **Do not know what's the true label.**

Contextual bandits

For round $t = 1, 2, \ldots,$ **the learner**

- Given $x^{(t)}$, chooses from a set of actions $a^{(t)} \in A$
- Receives reward $r^{(t)} \sim R(x^{(t)}, a^{(t)})$ (i.e., can be random)
- Want to maximize total reward
- You generate your own dataset $\{(x^{(t)}, a^{(t)}, r^{(t)})\}$!
  - e.g., for an image, the learner guesses a label, and is told whether correct or not (reward = 1 if correct and 0 otherwise). **Do not know what's the true label.**
  - e.g., for an user, the website recommends a movie, and observes whether the user likes it or not. **Do not know what movies the user really want to see.**

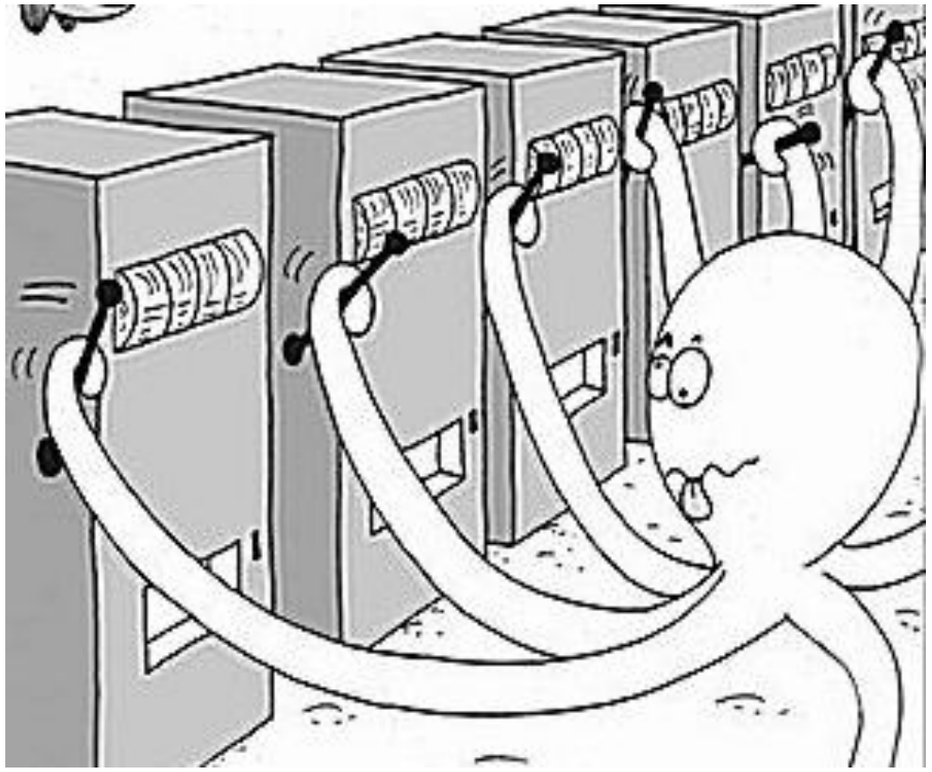
Contextual bandits

For round $t = 1, 2, \ldots,$ **the learner**

- Given $x^{(t)}$, chooses from a set of actions $a^{(t)} \in A$
- Receives reward $r^{(t)} \sim R(x^{(t)}, a^{(t)})$ (i.e., can be random)
- Want to maximize total reward
- You generate your own dataset $\{(x^{(t)}, a^{(t)}, r^{(t)})\}$!
  - e.g., for an image, the learner guesses a label, and is told whether correct or not (reward = 1 if correct and 0 otherwise). **Do not know what's the true label.**
  - e.g., for an user, the website recommends a movie, and observes whether the user likes it or not. **Do not know what movies the user really want to see.**
  - "Partial information setting"

Contextual bandits

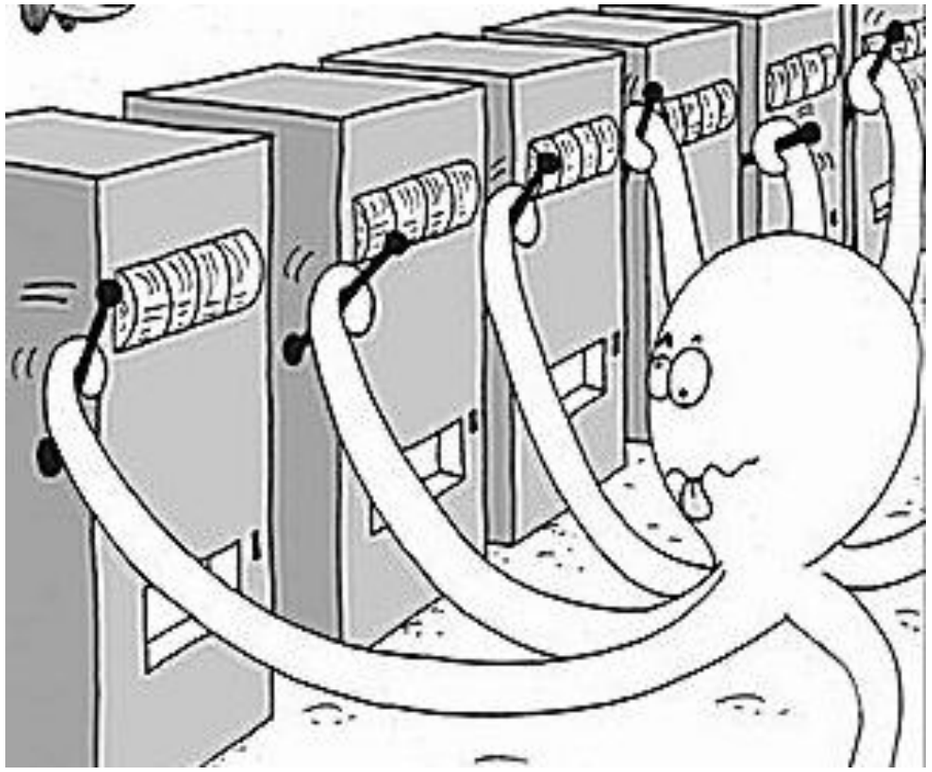
Contextual Bandits (cont.)



Contextual bandits

Contextual Bandits (cont.)

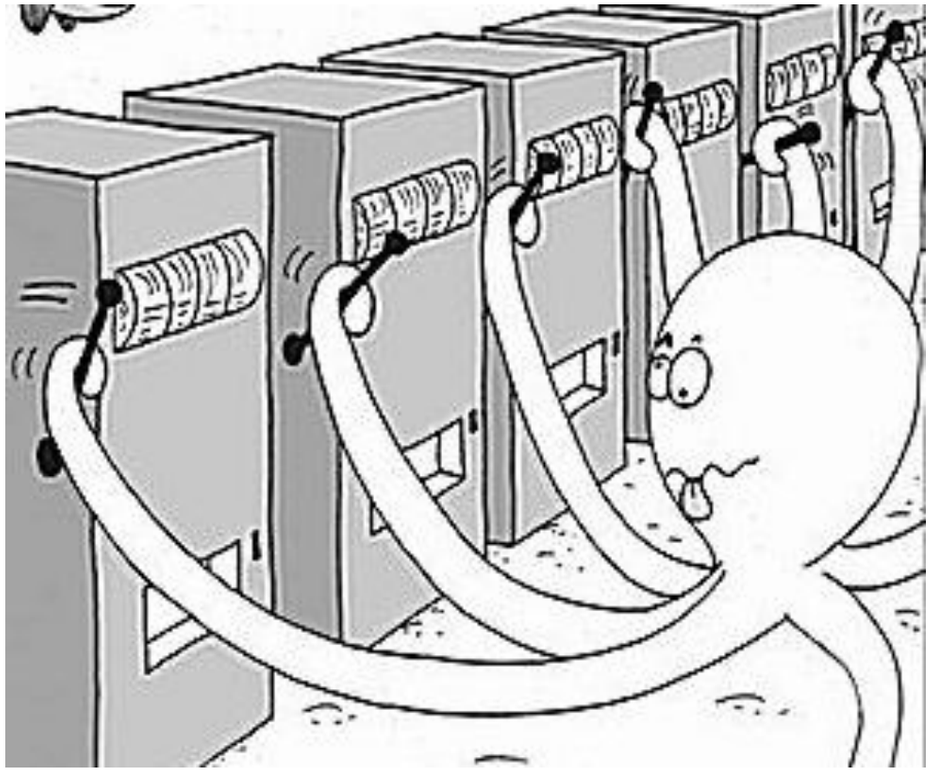
- Simplification: no x, Multi-Armed Bandits (MAB)



Contextual bandits

Contextual Bandits (cont.)

- Simplification: no x, Multi-Armed Bandits (MAB)
- Bandit is a research area by itself. we will not do a lot of bandits but may go through some material that have important implications on general RL (e.g., lower bounds)



RL

For round $t = 1, 2, \ldots,$

RL

For round $t = 1, 2, \ldots,$

- For time step h=1, 2, …, H, the learner

RL

For round $t = 1, 2, \ldots,$

- For time step h=1, 2, …, H, the learner
  - Observes $x_h^{(t)}$

RL

For round $t = 1, 2, \ldots,$

- For time step h=1, 2, …, H, the learner
  - Observes $x_h^{(t)}$
  - Chooses $a_h^{(t)}$

RL

For round $t = 1, 2, \ldots,$

- For time step h=1, 2, …, H, the learner
  - Observes $x_h^{(t)}$
  - Chooses $a_h^{(t)}$
  - Receives $r_h^{(t)} \sim R(x_h^{(t)}, a_h^{(t)})$

RL

For round $t = 1, 2, \ldots,$

- For time step h=1, 2, …, H, the learner
  - Observes $x_h^{(t)}$
  - Chooses $a_h^{(t)}$
  - Receives $r_h^{(t)} \sim R(x_h^{(t)}, a_h^{(t)})$
  - Next $x_{h+1}^{(t)}$ is generated as a function of $x_h^{(t)}$ and $a_h^{(t)}$ (or sometimes, all previous x's and a's within round t)

RL

For round $t = 1, 2, \ldots,$

- For time step h=1, 2, …, H, the learner
  - Observes $x_h^{(t)}$
  - Chooses $a_h^{(t)}$
  - Receives $r_h^{(t)} \sim R(x_h^{(t)}, a_h^{(t)})$
  - Next $x_{h+1}^{(t)}$ is generated as a function of $x_h^{(t)}$ and $a_h^{(t)}$ (or sometimes, all previous x's and a's within round t)
- Bandits + "Delayed rewards/consequences"

RL

For round $t = 1, 2, \ldots,$

- For time step h=1, 2, …, H, the learner
  - Observes $x_h^{(t)}$
  - Chooses $a_h^{(t)}$
  - Receives $r_h^{(t)} \sim R(x_h^{(t)}, a_h^{(t)})$
  - Next $x_{h+1}^{(t)}$ is generated as a function of $x_h^{(t)}$ and $a_h^{(t)}$ (or sometimes, all previous x's and a's within round t)
- Bandits + "Delayed rewards/consequences"
- The protocol here is for episodic RL (each t is an *episode*).