Overestimation in Q-Learning

Deep Reinforcement Learning with Double Q-learning

Hado van Hasselt, Arthur Guez, David Silver. AAAI 2016

Non-delusional Q-learning and value-iteration

Tyler Lu, Dale Schuurmans, Craig Boutilier. NeurIPS 2018

Yang Liu

19.03.2019

TD Method

“Learn a guess from a guess”

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[Y_t - Q(s_t, a_t)]$$

Y_t: TD target combing Sample Reward and Current Estimate

Q-Learning: TD Control

"Learn a guess from a guess"

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[Y_t - Q(s_t, a_t)]$$

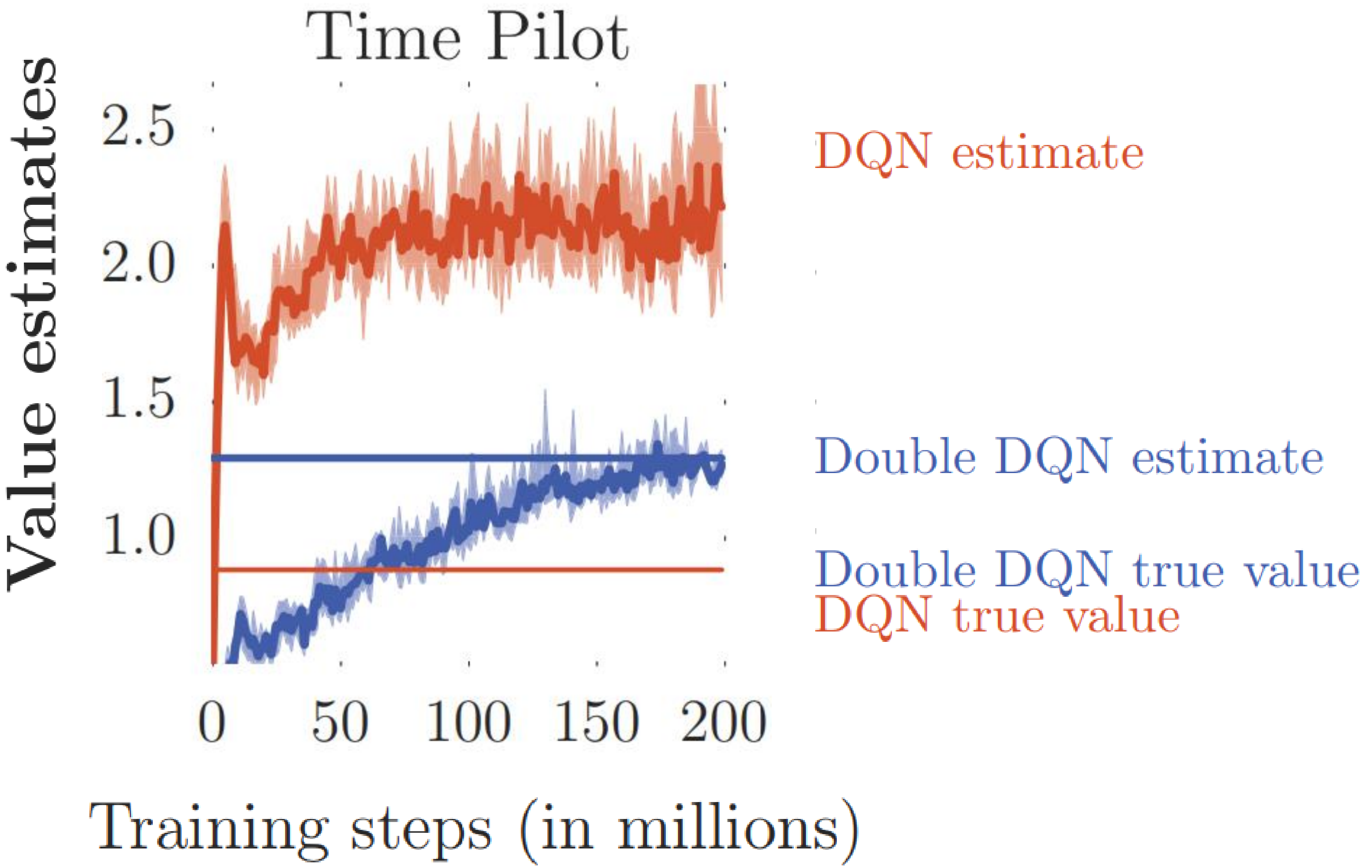
$$Y_t = r_{t+1} + \gamma \max_a Q(s_{t+1}, a)$$

Sample Reward Current Estimate

Converge to(Watkins 1989)

$$q_*(s, a) \coloneqq \max_\pi q_\pi(s, a) = \mathbb{E}[R_t + \gamma v_*(S_{t+1}) \mid S_t = s,\ A_t = a]$$

Overestimation in Q-Learning



Deep Reinforcement Learning with Double Q-learning Hado van Hasselt, Arthur Guez, David Silver. AAAI 2016

What's Wrong?

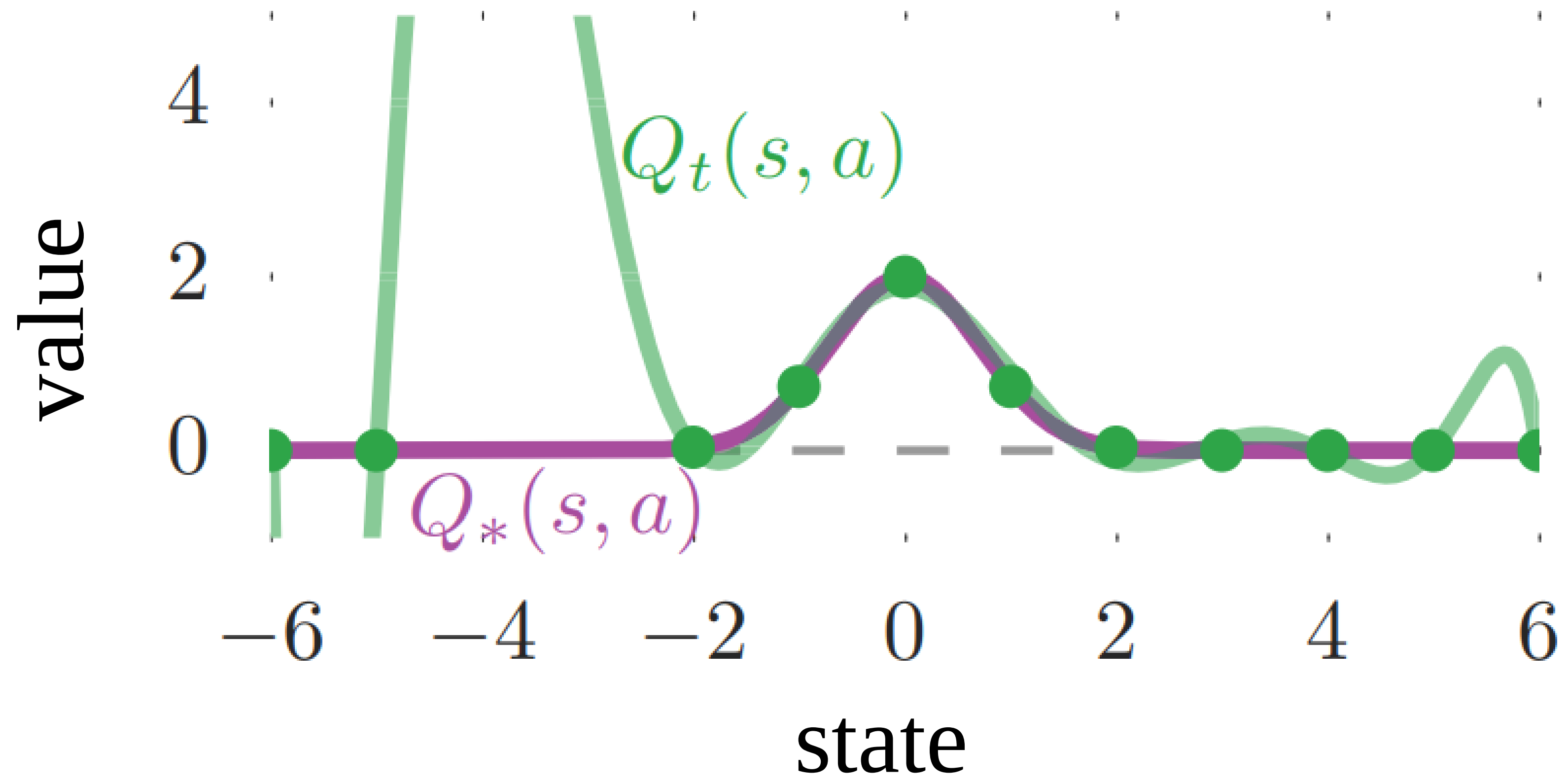
$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[Y_t - Q(s_t, a_t)]$$

$$Y_t = R_{t+1} + \gamma \max_{a \in \mathcal{A}} Q(s_{t+1}, a)$$

$\max_{a \in \mathcal{A}}$ ⇒ two biases ⇒ Pathological Behavior

Maximization Bias	Inferior Policy
Delusional Bias	Divergence

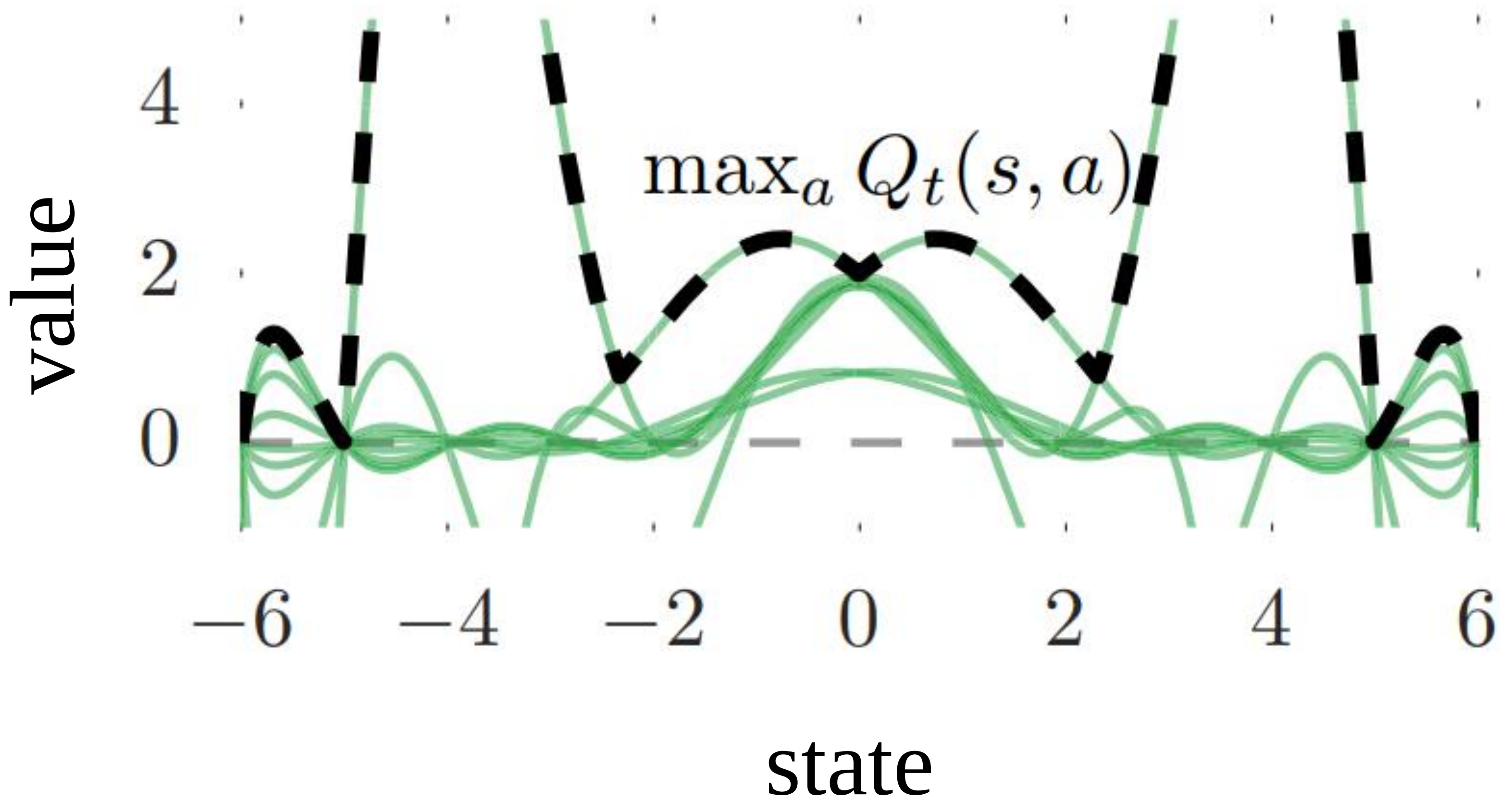
Maximization Bias



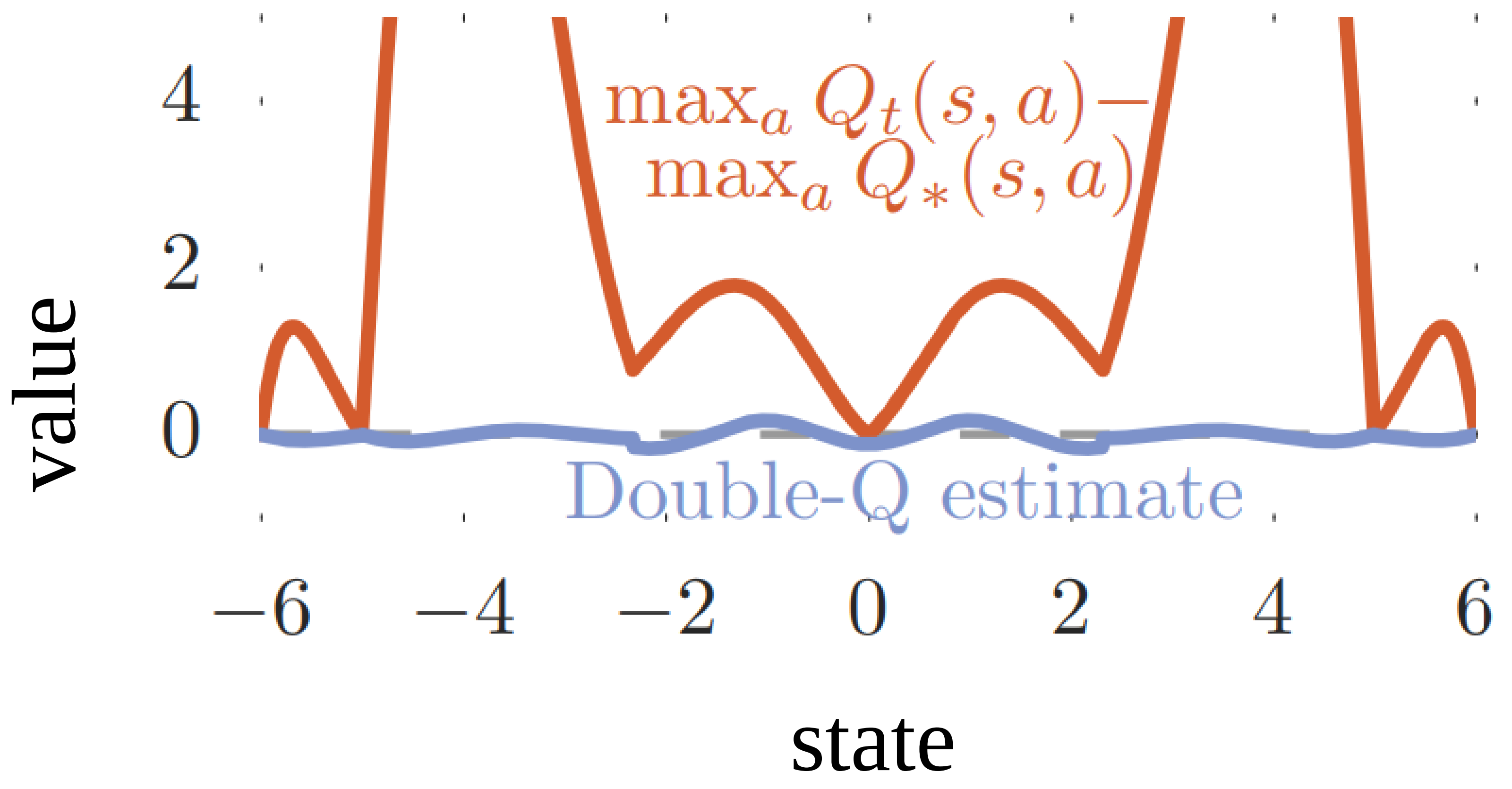
$Q_*(s,a) = V_*(s) = 2e^{-s^2}, \forall a$

$Q_t(s,a)$ 9-degree polynomial

Maximization Bias



Maximization Bias



Maximization Bias

By convexity of max function, for R.V. $X_1, \ldots, X_n$ and their sample mean $\mu_1, \ldots, \mu_n$ and their realization $\widehat{\mu_1}, \ldots, \widehat{\mu_n}$

$$\max_i \mathbb{E}[X_i] = \max_i \mathbb{E}[\mu_i] \leq \mathbb{E}[\max_i \mu_i] \approx \max_i \widehat{\mu_i}$$

$$\begin{aligned} q_*(s,a) &= \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a] + \gamma \mathbb{E}[\max_{a'} q_*(S_{t+1}, a') | S_t = s, A_t = a] \\ &\approx r_{t+1} + \gamma \mathbb{E}\left[\max_{a'} q_*(s_{t+1}, a')\right] \\ &\leq r_{t+1} + \gamma \mathbb{E}\left[\max_{a'} Q(s_{t+1}, a')\right] \\ &= r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') = Y_t \end{aligned}$$

Double Q-Learning

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[Y_t - Q(s_t, a_t)]$$

Single Q-Learning

$$Y_t = r_{t+1} + \gamma \max_a Q(s_{t+1}, a)$$

Double Q-Learning

$$Y_t = r_{t+1} + \gamma Q_2(s_{t+1}, \underset{a}{\operatorname{argmax}} Q_1)$$

Theorem: Double Estimator is bounded by $\max_i \mathbb{E}[X_i]$

Double Q-learning Hado van Hasselt. NIPS 2010

Double DQN

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha[Y_t - Q(\mathrm{s}_t, a_t; \boldsymbol{\theta}_t)]\nabla_\theta Q(\mathrm{s}_t, a_t; \boldsymbol{\theta}_t)$$

DQN

$$Y_t = r_{t+1} + \gamma \max_a Q(\mathrm{s}_{t+1}, a; \boldsymbol{\theta}_t^-)$$

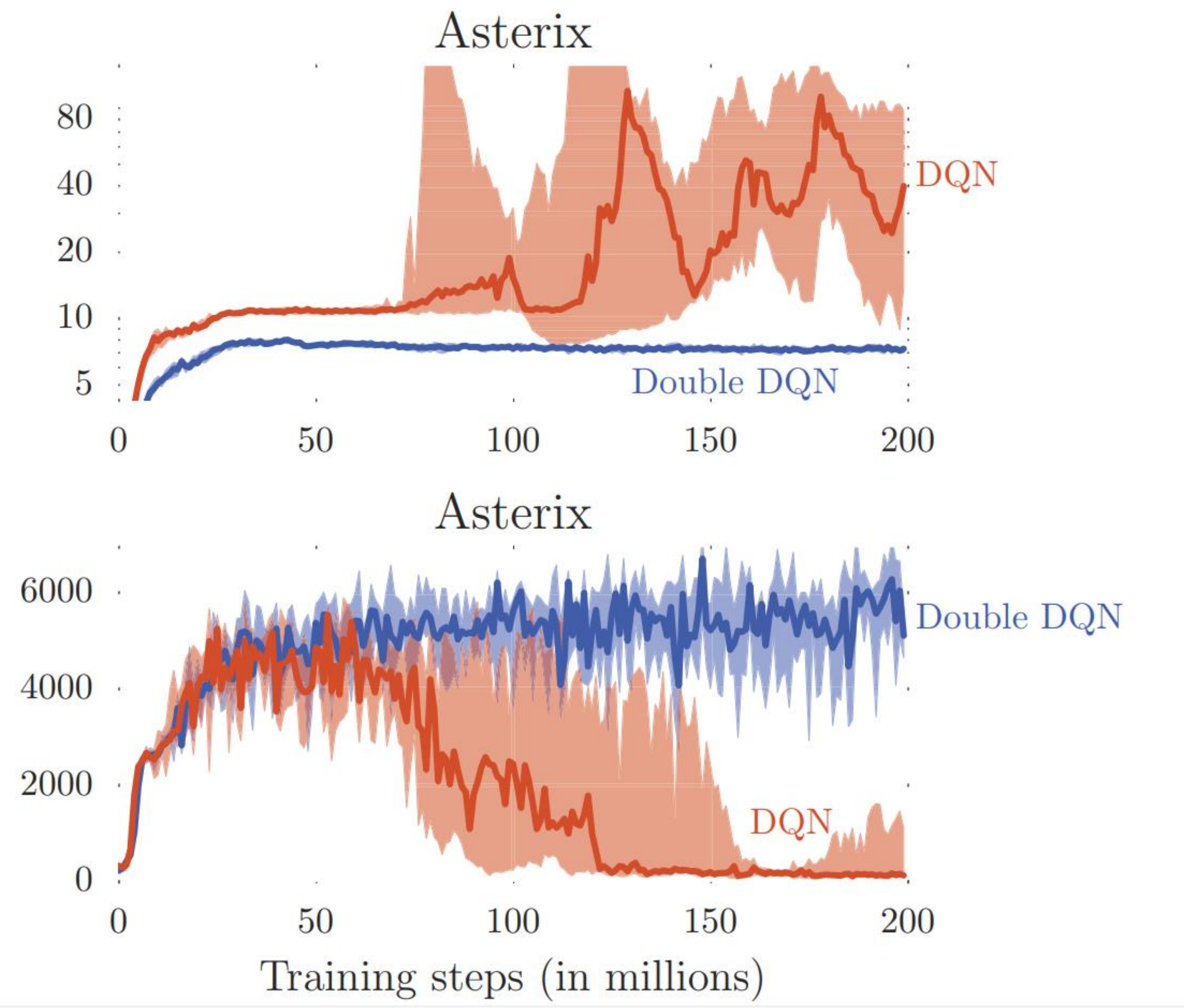
Double DQN

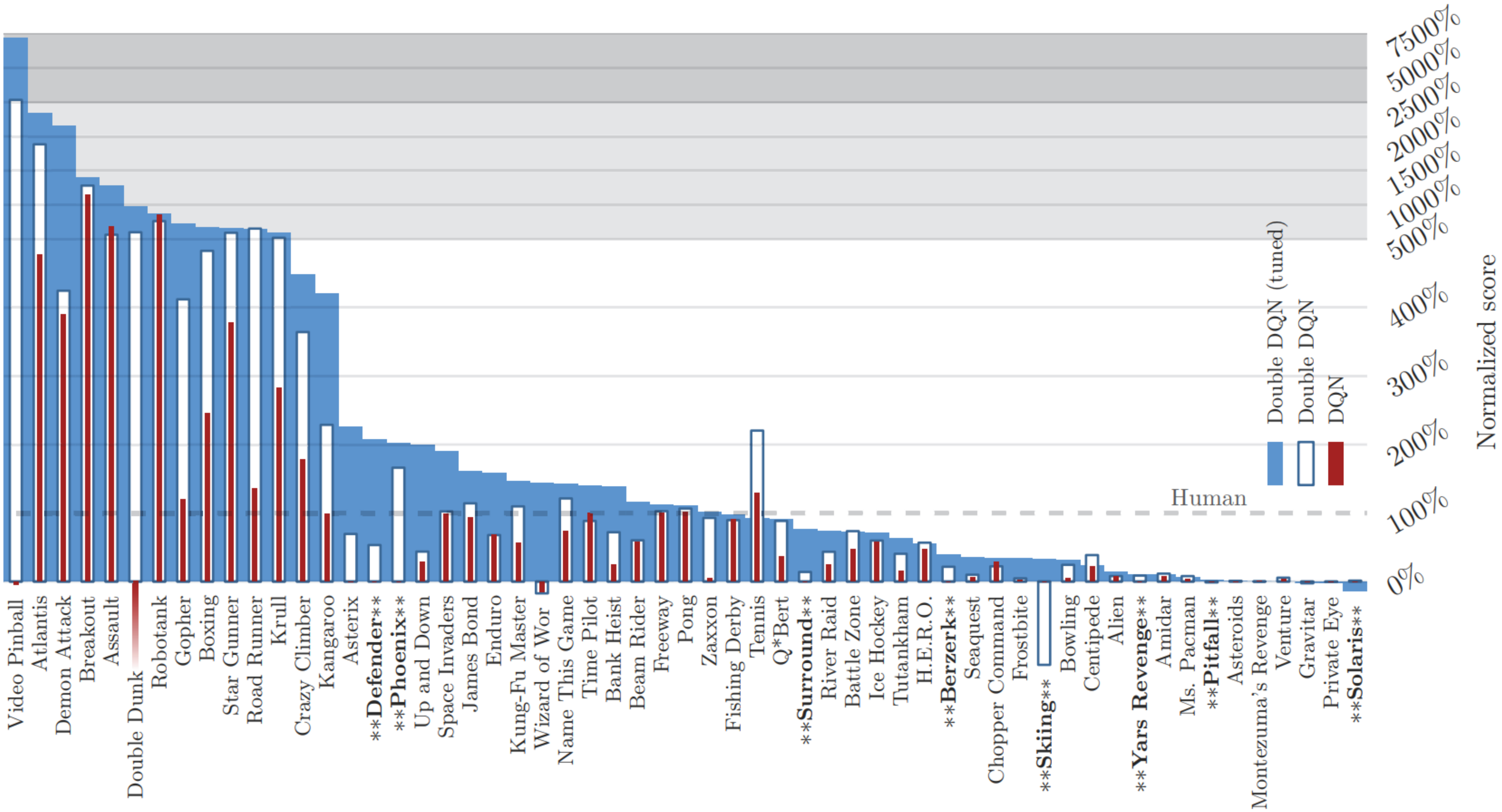
$$Y_t = r_{t+1} + \gamma Q(\mathrm{s}_{t+1}, \operatorname*{argmax}_a Q(\mathrm{s}_{t+1}, a; \boldsymbol{\theta}_t); \boldsymbol{\theta}_t^-)$$

Result

Value Estimate

Score




Video Pinball
Atlantis
Demon Attack
Breakout
Assault
Double Dunk
Robotank
Gopher
Boxing
Star Gunner
Road Runner
Krull
Crazy Climber
Kangaroo
Asterix
Defender
Phoenix
Up and Down
Space Invaders
James Bond
Enduro
Kung-Fu Master
Wizard of Wor
Name This Game
Time Pilot
Bank Heist
Beam Rider
Freeway
Pong
Zaxxon
Fishing Derby
Tennis
Q*Bert
Surround
River Raid
Battle Zone
Ice Hockey
Tutankham
H.E.R.O.
Berzerk
Seaquest
Chopper Command
Frostbite
Skiing
Bowling
Centipede
Alien
Yars Revenge
Amidar
Ms. Pacman
Pitfall
Asteroids
Montezuma's Revenge
Venture
Gravitar
Private Eye
Solaris
Human
Double DQN (tuned)
Double DQN
DQN
7500%
5000%
2500%
2000%
1500%
1000%
500%
400%
300%
200%
100%
0%
Normalized score

Comments

Pros

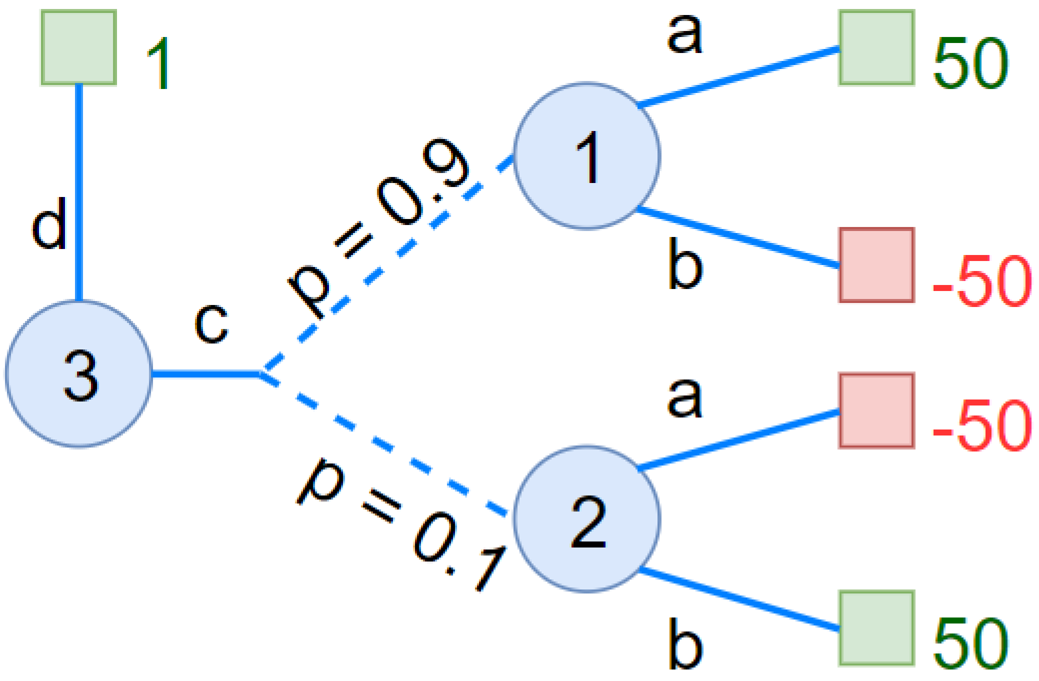
- Cheap modification
- Improvement on stability and score
- Widely adopted by following Deepmind's work

Cons

- Still biased estimate(proven underestimate)
- Experience Replay Buffer, the two estimates are not independent

Delusional Bias

A Simple MDP

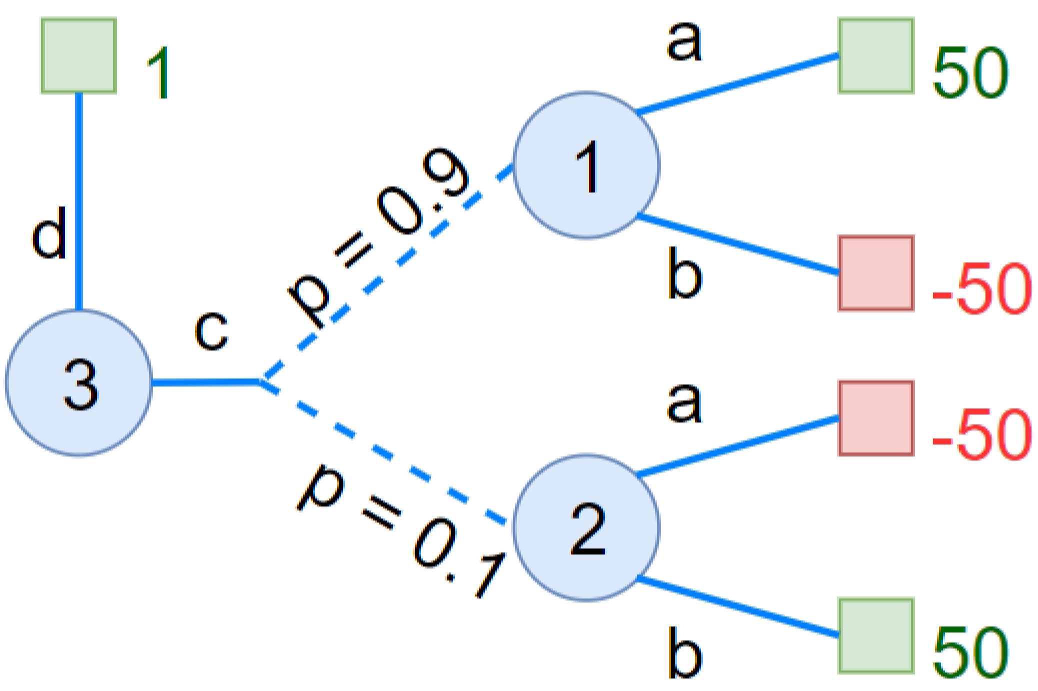


Optimal Policy: 50

3-c 1-a 2-b

Delusional Bias

Consider Linear Approximation $Q(s,a) = \theta\phi(s,a), \theta_0 > 0$



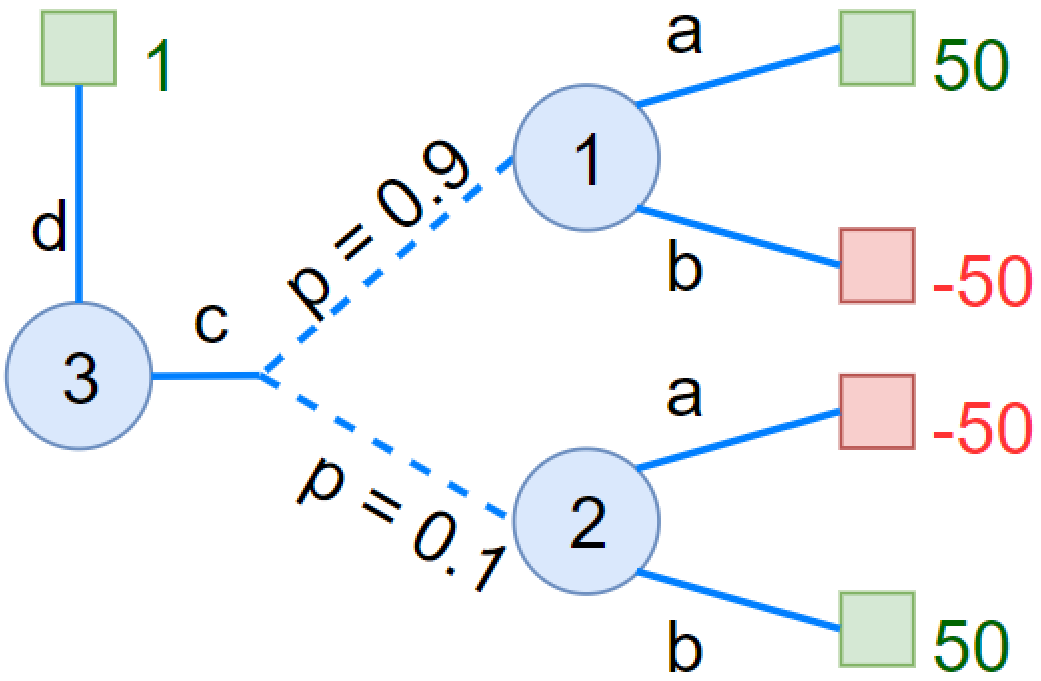
ϕ(1, a) = 1	ϕ(1, a) = -1
ϕ(2, a) = 1	ϕ(2, b) = -1

Optimal Policy: 40

3-c 1-a 2-a

Consider Linear Approximation $Q(s,a) = \theta\phi(s,a)$

ϕ(1, a) = 1	ϕ(1, a) = -1
ϕ(2, a) = 1	ϕ(2, b) = -1



Inconsistent Back up

Back up at (1, a): $\theta \uparrow$

Back up at (2, b) : $\theta \downarrow\downarrow$

$\Rightarrow \theta_* \approx 0 \Rightarrow Q(3,c) \approx 0$

$\Rightarrow \pi_*(3) = d$ with value 1

Delusional Bias

Delusional Bias *occurs whenever a backed-up value estimate is derived from action choices that are* ***not realizable*** *in the underlying class(Greedy Policy Class).*

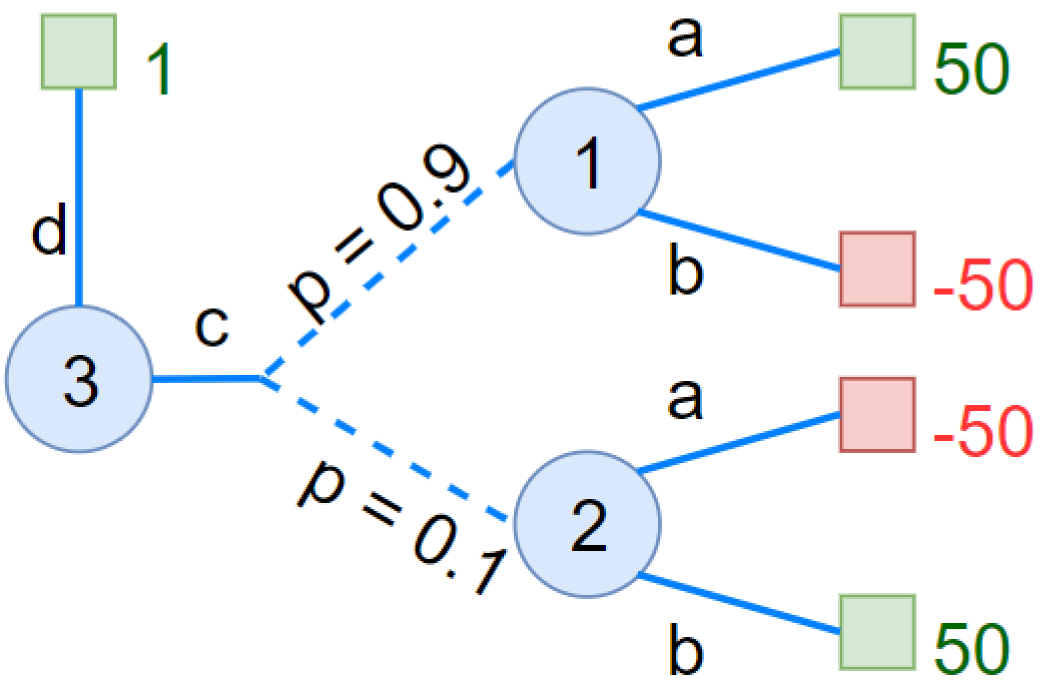
$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha[Y_t - Q(s_t, a_t; \boldsymbol{\theta}_t)]\nabla_\theta Q(s_t, a_t; \boldsymbol{\theta}_t)$$

$$Y_t = r_{t+1} + \gamma \max_{a \in \mathcal{A}} Q(s_{t+1}, a; \theta_t)$$

Unconstrained maximization ⇒ Overestimation in target Y_t

Delusional Bias

Delusional Bias *occurs whenever a backed-up value estimate is derived from action choices that are* ***not realizable*** *in the underlying class.*



Inconsistent Backup at (1,a) and (2,b)
⇒ Delusional Bias ⇒Poor policy

Delusional Bias

Explains following pathological behavior:

- Poor policy
- Divergence
- Cyclic Behavior
- Discounting Paradox

γ-score of model trained with γ' higher than model trained with γ

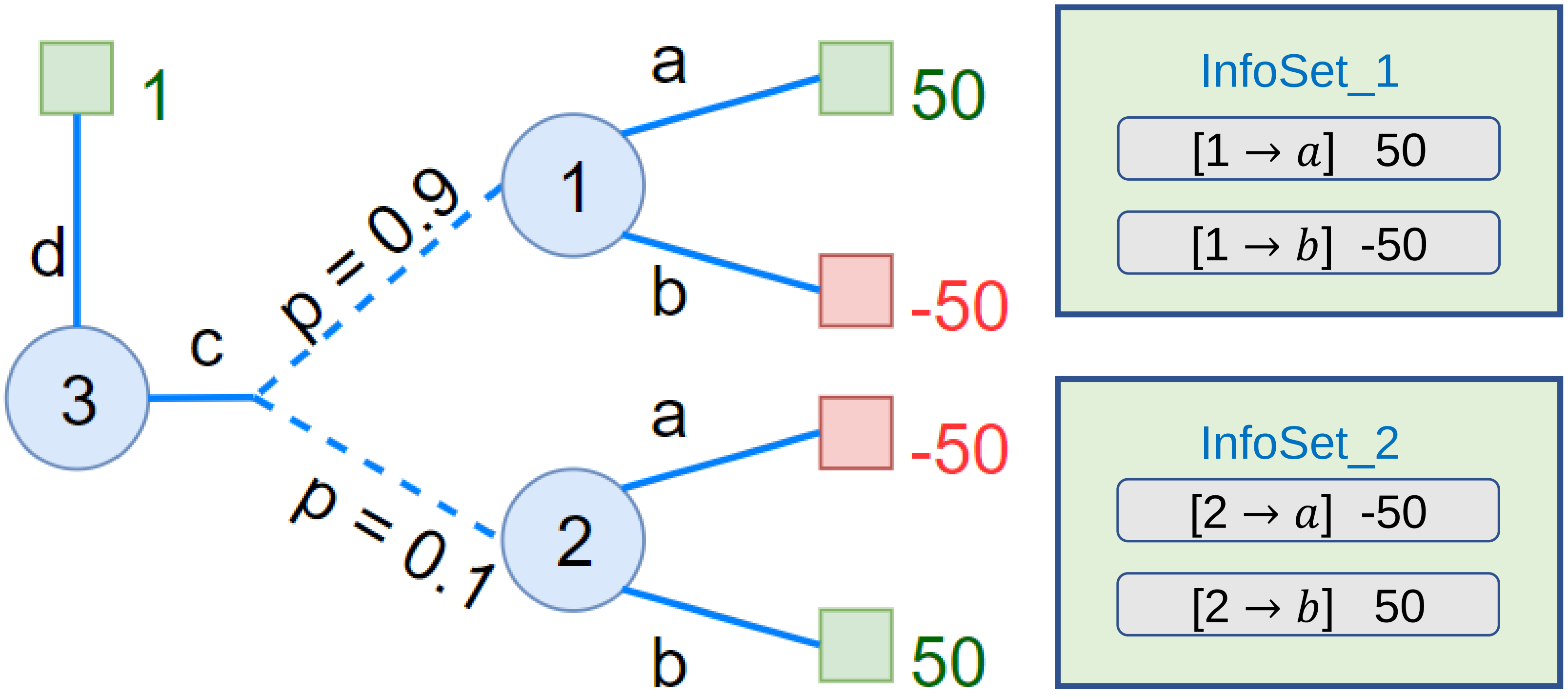
Also Arises in Value Iteration

$$V(s) \leftarrow \max_{a \in \mathcal{A}} \sum_{s'} p(s'|s,a)[r_{s,a} + \gamma V(s')]$$

Full state Bellman Backup

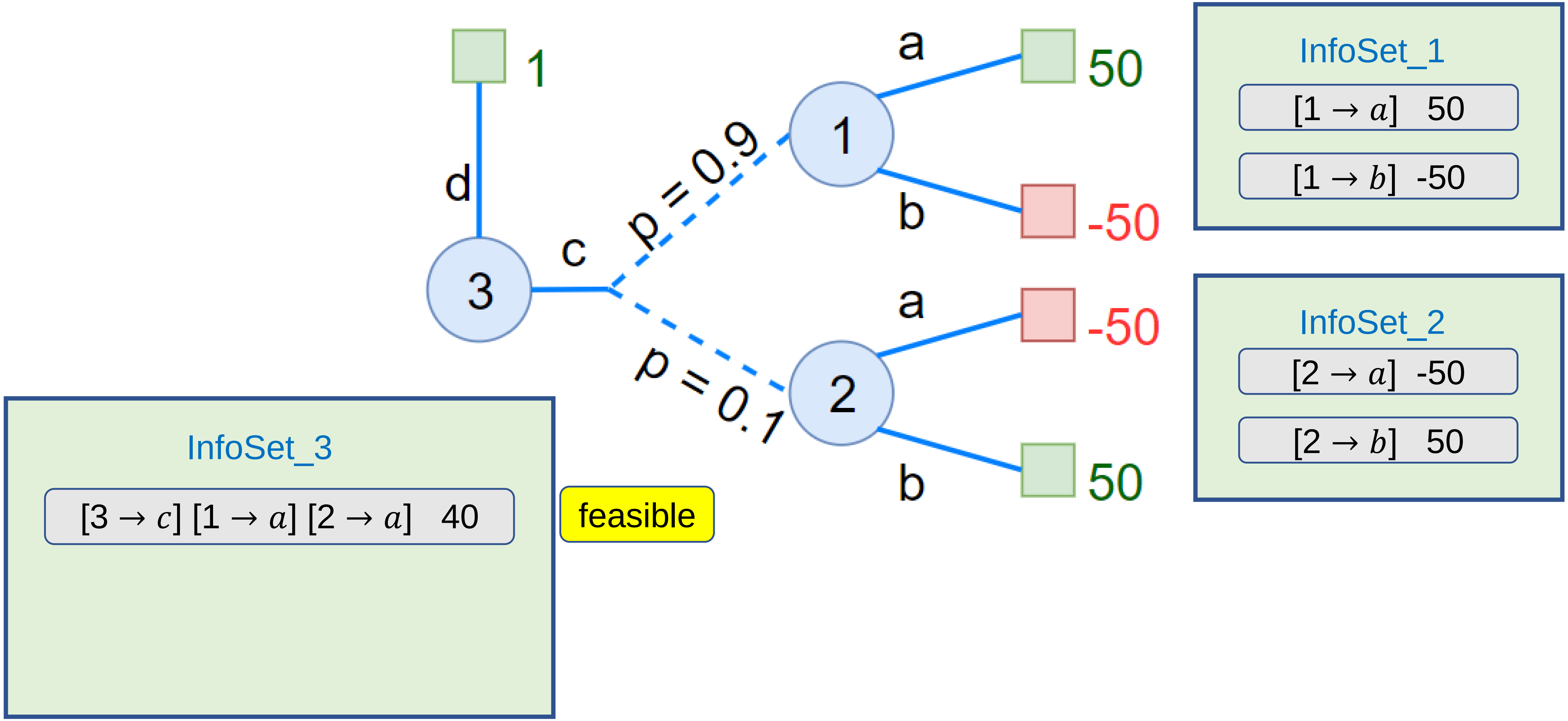
Policy-Class Value Iteration

Idea: just track every feasible paths using *information set*



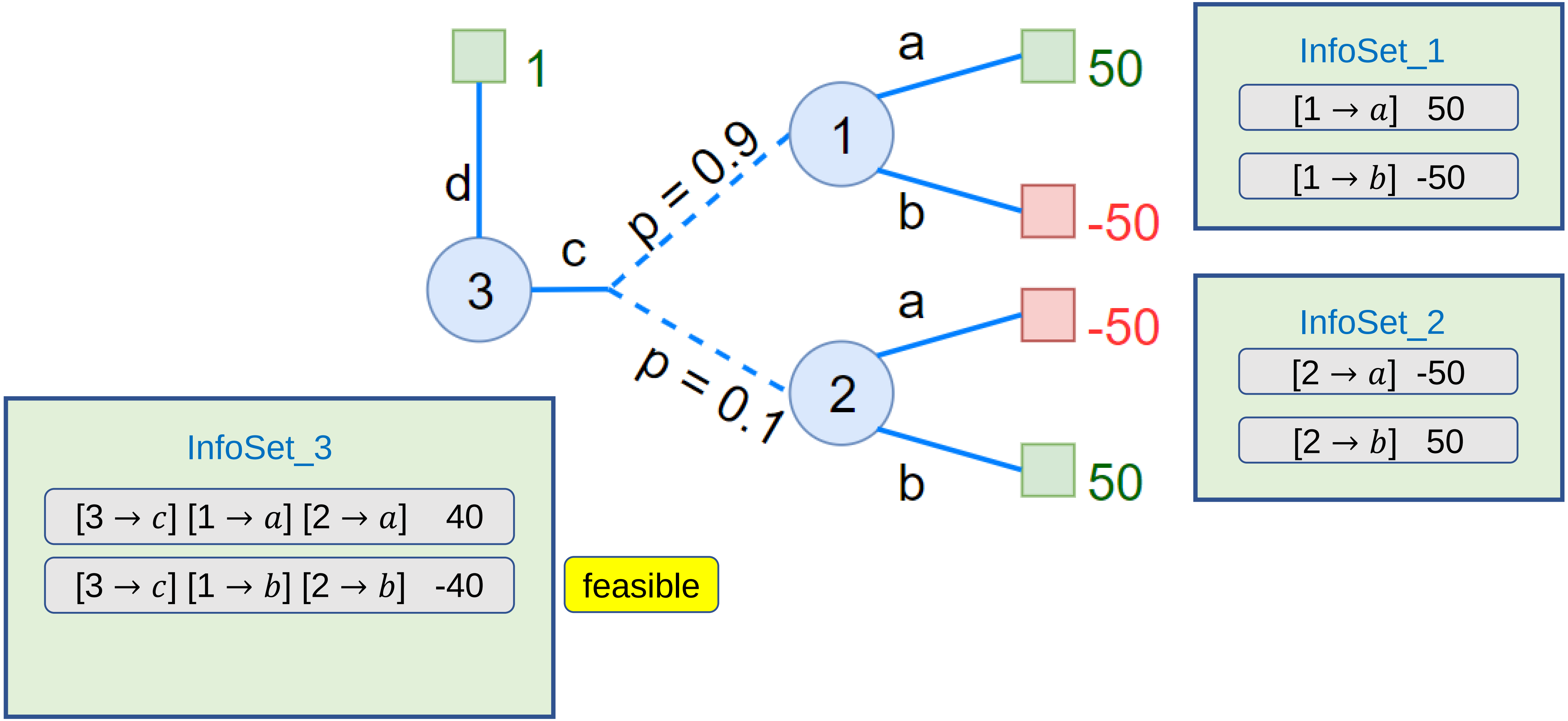
Policy-Class Value Iteration

Idea: just track every feasible paths using *information set*



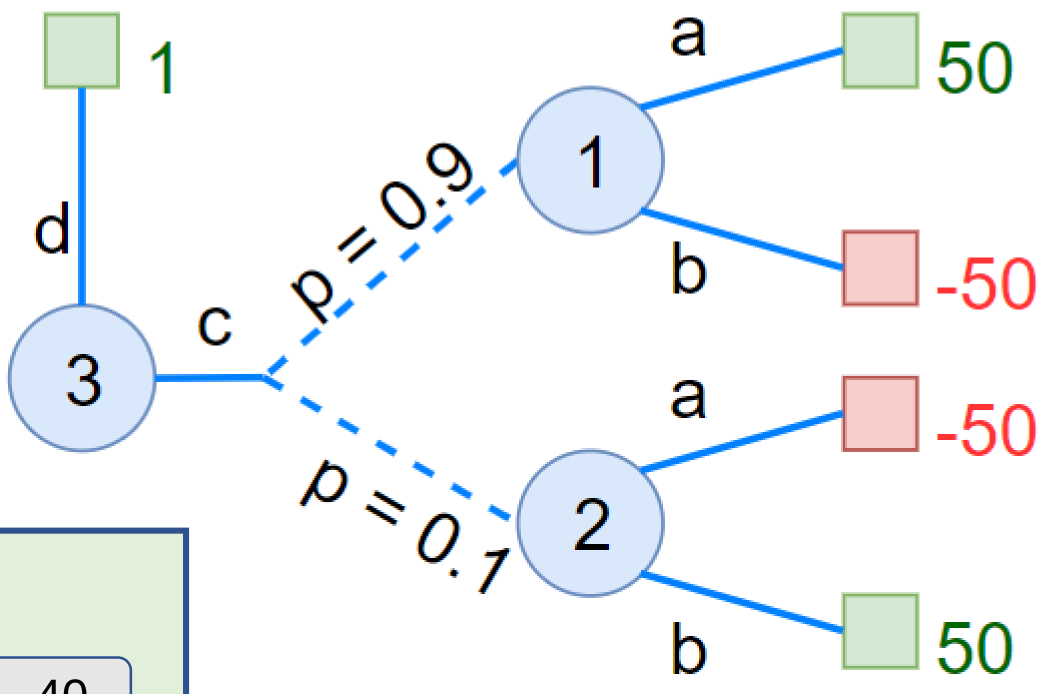
Policy-Class Value Iteration

Idea: just track every feasible paths using *information set*



Policy-Class Value Iteration

Idea: just track every feasible paths using *information set*



InfoSet_1

- $[1 \rightarrow a]$ 50
- $[1 \rightarrow b]$ -50

InfoSet_2

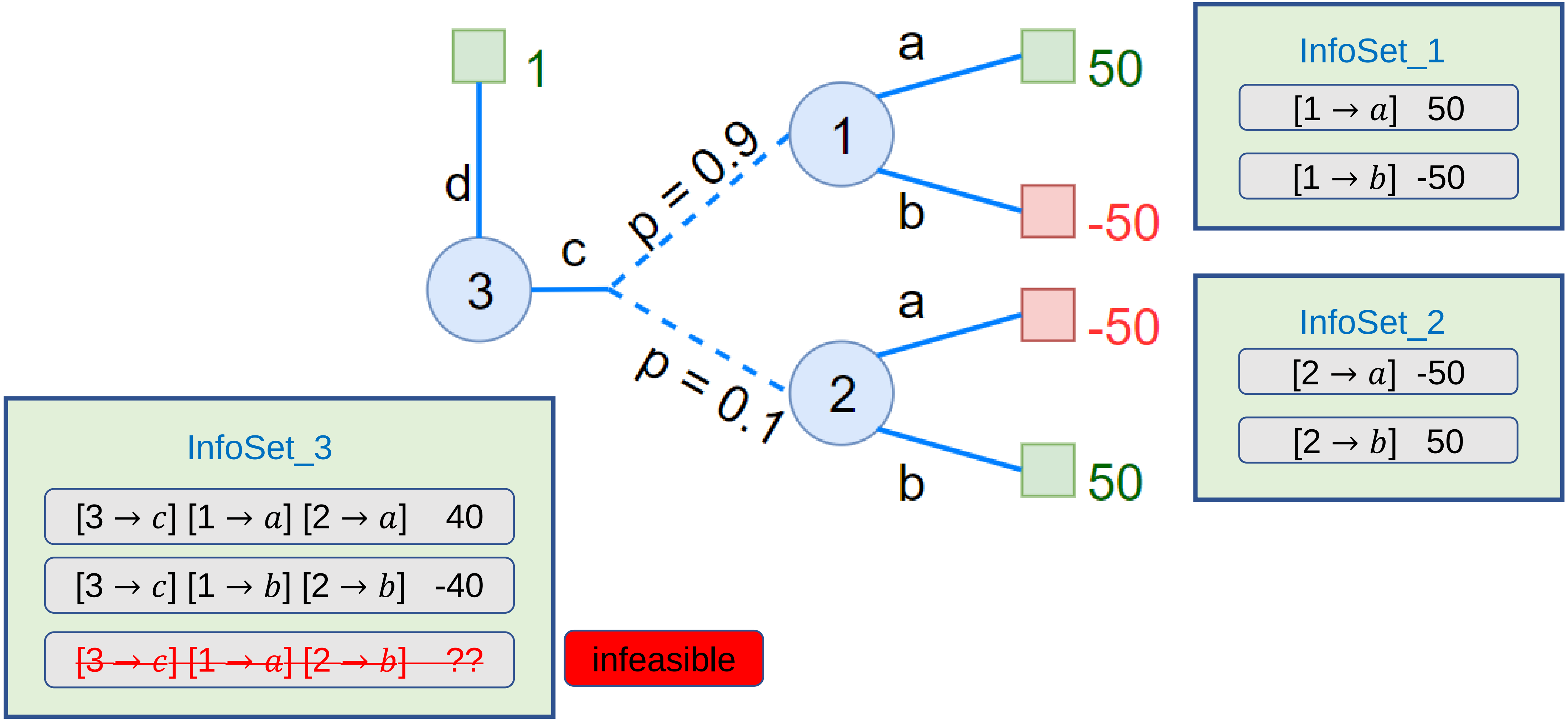
- $[2 \rightarrow a]$ -50
- $[2 \rightarrow b]$ 50

InfoSet_3

- $[3 \rightarrow c]\ [1 \rightarrow a]\ [2 \rightarrow a]$ 40
- $[3 \rightarrow c]\ [1 \rightarrow b]\ [2 \rightarrow b]$ -40
- $[3 \rightarrow c]\ [1 \rightarrow a]\ [2 \rightarrow b]$??

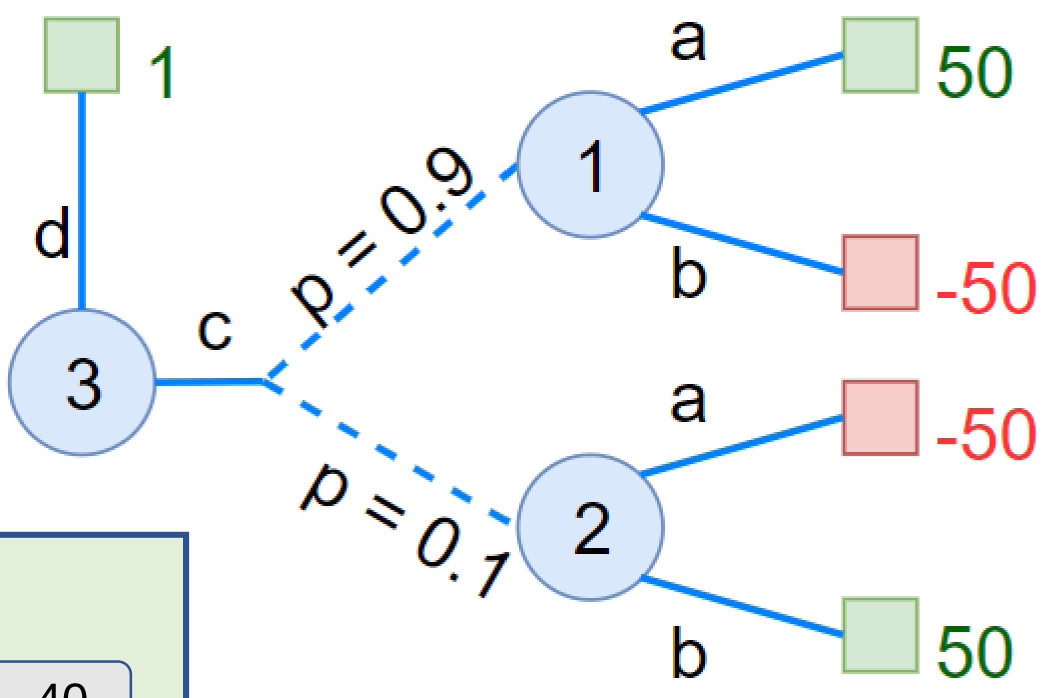
Policy-Class Value Iteration

Idea: just track every feasible paths using *information set*



Policy-Class Value Iteration

Idea: just track every feasible paths using *information set*



InfoSet_1

- $[1 \to a]$ 50
- $[1 \to b]$ -50

InfoSet_2

- $[2 \to a]$ -50
- $[2 \to b]$ 50

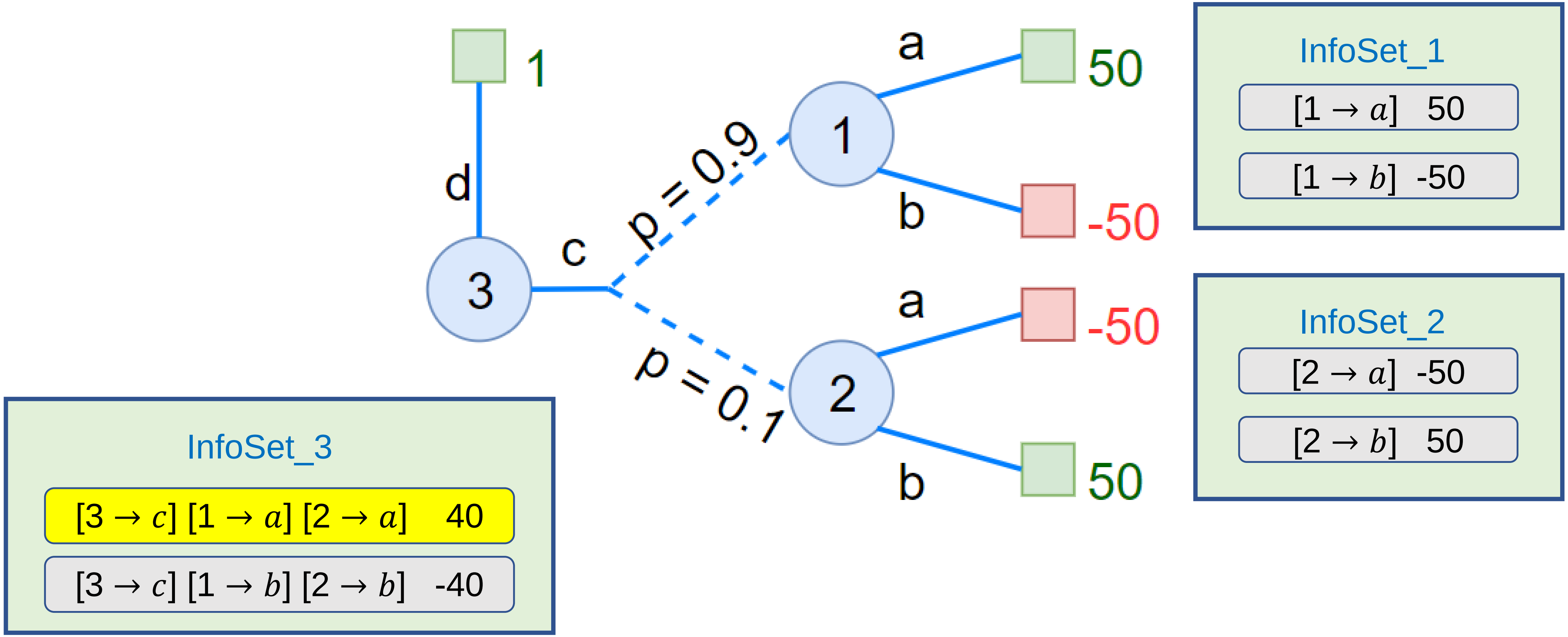
InfoSet_3

- $[3 \to c]$ $[1 \to a]$ $[2 \to a]$ 40
- $[3 \to c]$ $[1 \to b]$ $[2 \to b]$ -40
- ~~$[3 \to c]$ $[1 \to a]$ $[2 \to b]$??~~
- ~~$[3 \to c]$ $[1 \to b]$ $[2 \to a]$??~~

infeasible

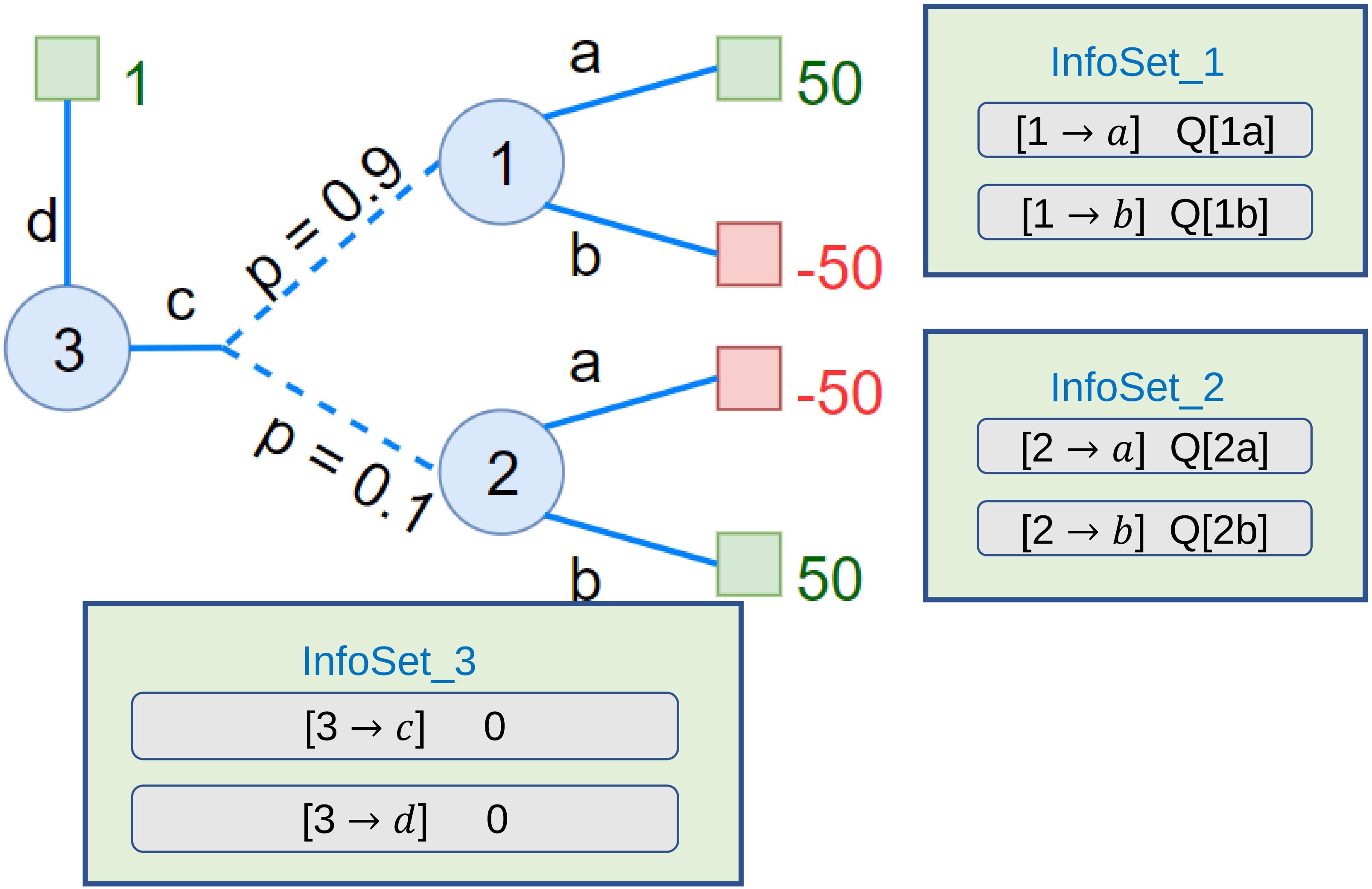
Policy-Class Value Iteration

Idea: just track every feasible paths using *information set*



Policy-Class Q-Learning

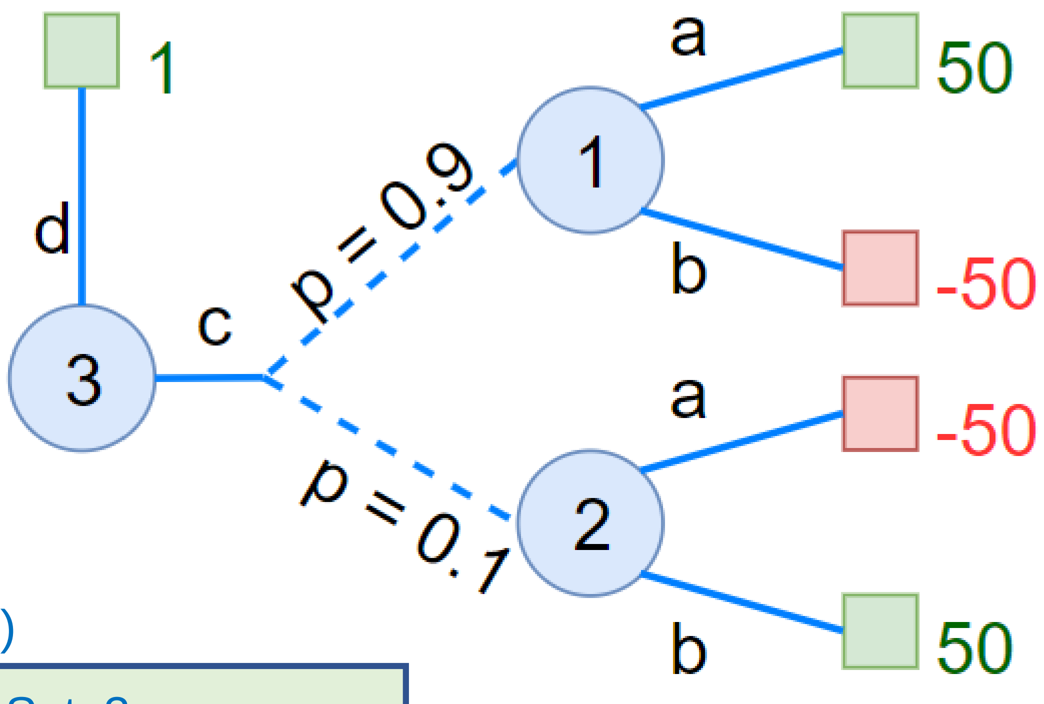
Sample Backup instead of full state Backup



Initialization

Policy-Class Q-Learning

Sample Backup instead of full state Backup



InfoSet_1

$[1 \rightarrow a]$	Q[1a]
$[1 \rightarrow b]$	Q[1b]

InfoSet_2

$[2 \rightarrow a]$	Q[2a]
$[2 \rightarrow b]$	Q[2b]

Sample experience (3, c, 0, 1)

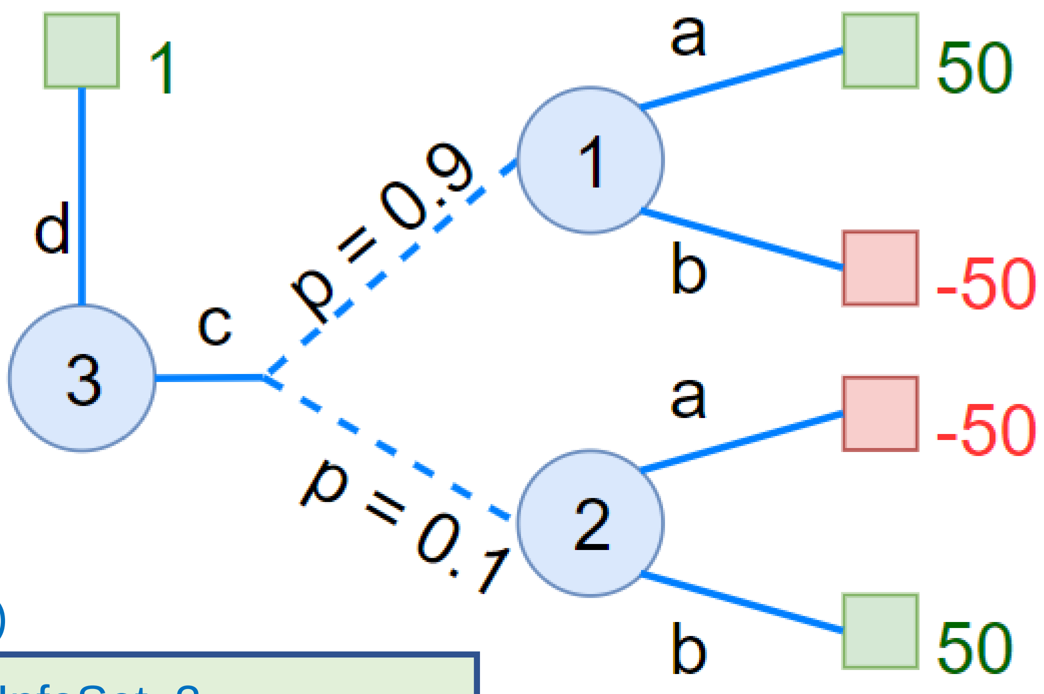
InfoSet_3

$[3 \rightarrow c]\ [1 \rightarrow a]$	$\longleftarrow$
$[3 \rightarrow c]$	0
$[3 \rightarrow d]$	0

feasible $Q[3c][1a] \longleftarrow Q[3c] + \alpha_t^{sa}(r + \gamma Q[1a] - Q[3c])$

Policy-Class Q-Learning

Sample Backup instead of full state Backup



InfoSet_1

$[1 \to a]$ Q[1a]

$[1 \to b]$ Q[1b]

InfoSet_2

$[2 \to a]$ Q[2a]

$[2 \to b]$ Q[2b]

Sample experience (3, c, 0, 1)

InfoSet_3

$[3 \to c]\ [1 \to a]$ $\longleftarrow$

$[3 \to c]\ [1 \to b]$ $\longleftarrow$

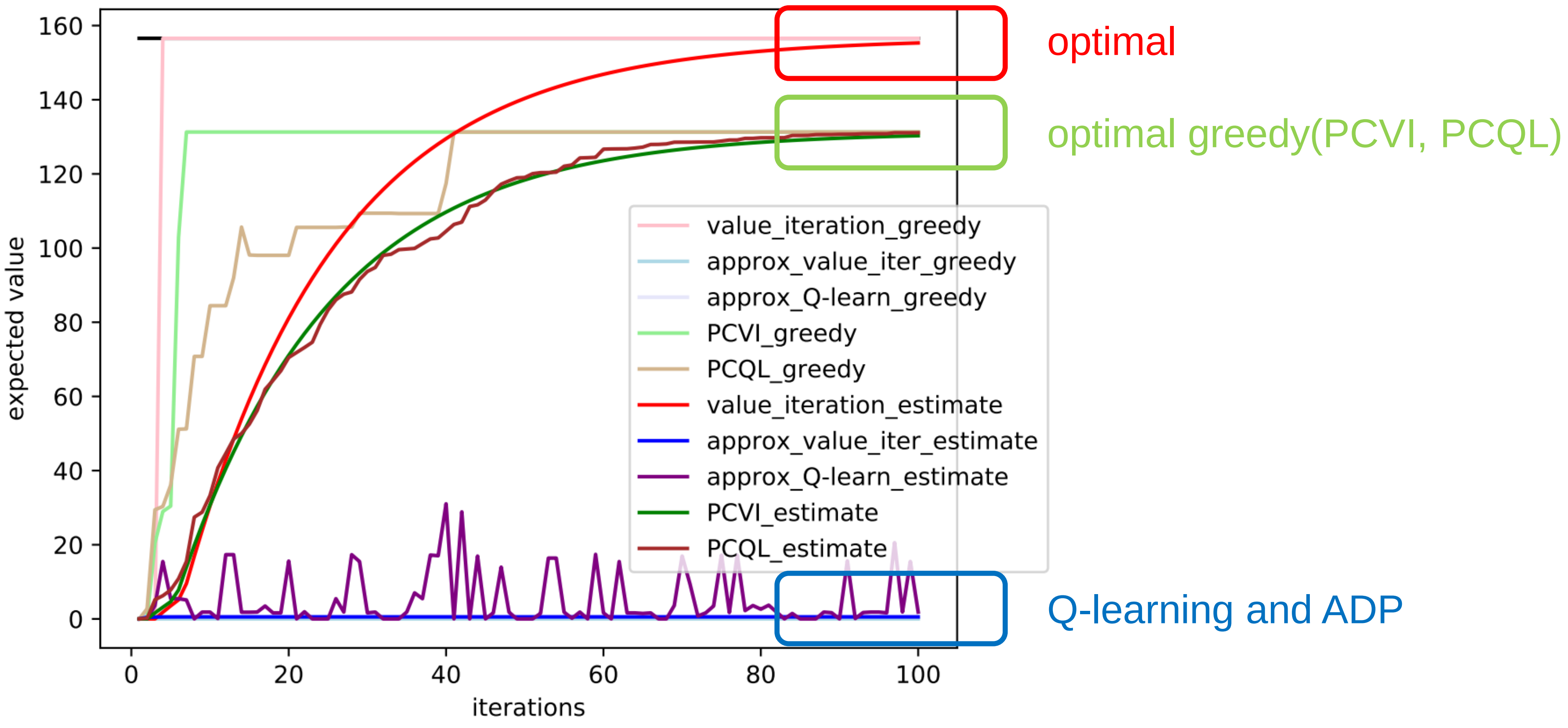
$[3 \to d]$ 0

feasible $Q[3c][1b] \longleftarrow Q[3c] + \alpha_t^{sa}(r + \gamma Q[1b] - Q[3c])$

Theorems

- PCVI and PCQL converge
- Converge to optimal policy in given class/No delusion
- Bounded runtime

Result



4x4 Grid world using 4 features

Comment:

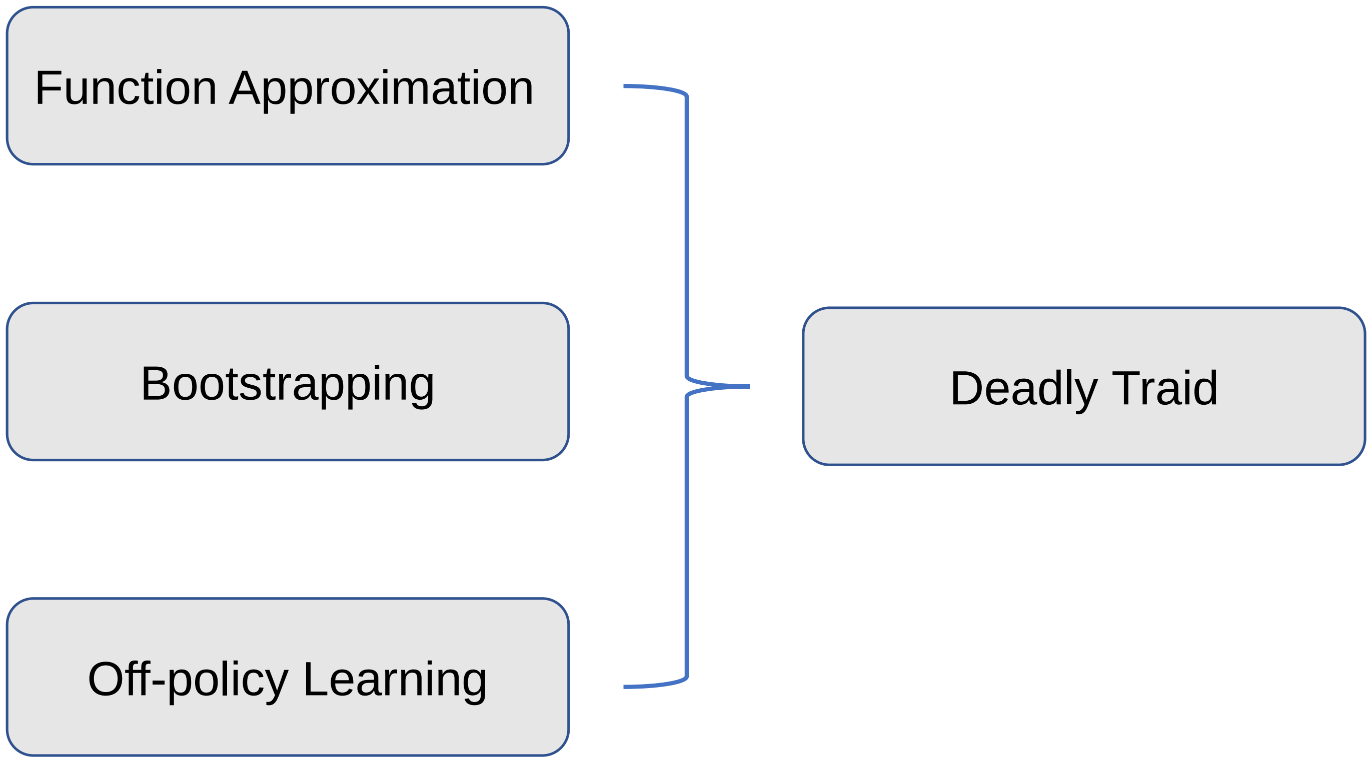
Pros:

- Identified Delusional Bias and its consequence
- Come up with proven algorithm PCVI and PCQL
- Heuristic methods for large models

Cons:

- No scalable solution
- Provide no result on DNN

More on Deadly Traid



Reference

The Value Function Polytope in Reinforcement Learning Robert Dadashi, Adrien Ali Taïga, Nicolas Le Roux, Dale Schuurmans, Marc G. Bellemare. Arxiv 2019

Deep Reinforcement Learning and the Deadly Triad Hado Van Hasselt, Yotam Doron, Florian Strub, Matteo Hessel, Nicolas Sonnerat, Joseph Modayil. Arxiv 2018

Non-delusional Q-learning and value-iteration Tyler Lu, Dale Schuurmans, Craig Boutilier. NeurIPS 2018

Deep Reinforcement Learning with Double Q-learning Hado van Hasselt, Arthur Guez, David Silver. AAAI 2016

Reinforcement Learning: An Introduction Richard Sutton and Andrew Barto. 2016

Double Q-learning Hado van Hasselt. NeurIPS 2010

The Optimizer's Curse: Skepticism and Postdecision Surprise in Decision Analysis James Smith and Robert Winkler. Management Science 2006

Q-learning Christopher Watkins and Peter Dayan. Machine Learning 1992