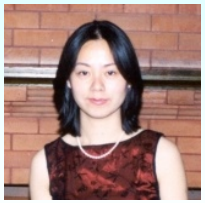
Bayesian Sparse Sampling for On-line Reward Optimization

Dale Schuurmans

UNIVERSITY OF ALBERTA

ALBERTA INGENUITY CENTRE FOR MACHINE LEARNING

With



Tao Wang

Dan Lizotte

Mike Bowling

Background Perspective

- Be *Bayesian* about reinforcement learning
- Ideal representation of uncertainty for action selection

Why are Bayesian approaches not prevalent in RL?

- Computational barriers

Our Recent Work

- Practical algorithms for approximating Bayes optimal decision making
- Analogy to *game-tree search*
  - on-line lookahead computation
  - \+ global value function approximation
- Use game-tree search ideas
  - but here expecti-max vs. mini-max
- Alternative approach to global value fun. approx.

Exploration vs. Exploitation

- Bayes decision theory
  - Value of information measured by ultimate return in reward
- Choose actions to max expected value
  - Exploration/exploitation tradeoff implicitly handled as side effect

Bayesian Approach

conceptually clean
but
computationally disastrous

versus

conceptually disastrous
but
computationally clean

Bayesian Approach

conceptually clean
but
computationally disastrous

versus

conceptually disastrous
but
computationally clean

Overview

- Efficient lookahead search for Bayesian RL
  - *Sparser* sparse sampling
  - Controllable computational cost
- Higher quality action selection than current methods

Greedy	
Epsilon - greedy	
Boltzmann	*(Luce 1959)*
Thompson Sampling	*(Thompson 1933)*
Bayes optimal	*(Hee 1978)*
Interval estimation	*(Lai 1987, Kaelbling 1994)*
Myopic value of perfect info.	*(Dearden, Friedman, Andre 1999)*
Standard sparse sampling	*(Kearns, Mansour, Ng 2001)*
Péret & Garcia	*(Péret & Garcia 2004)*

- General, can be combined with value fun. approx.

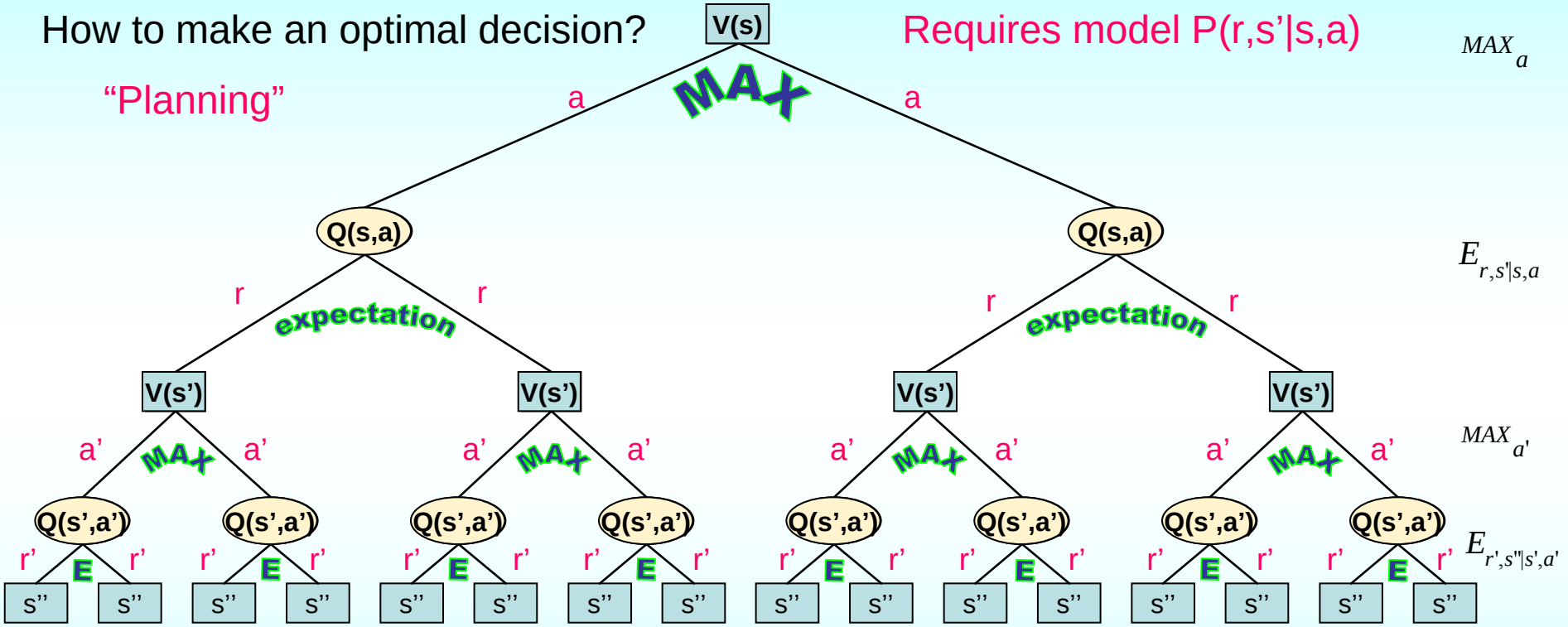
Goals

- Large (infinite) state and action spaces
- Exploit Bayesian modelling tools
  - E.g. Gaussian processes

Sequential Decision Making

How to make an optimal decision? Requires model P(r,s'|s,a)

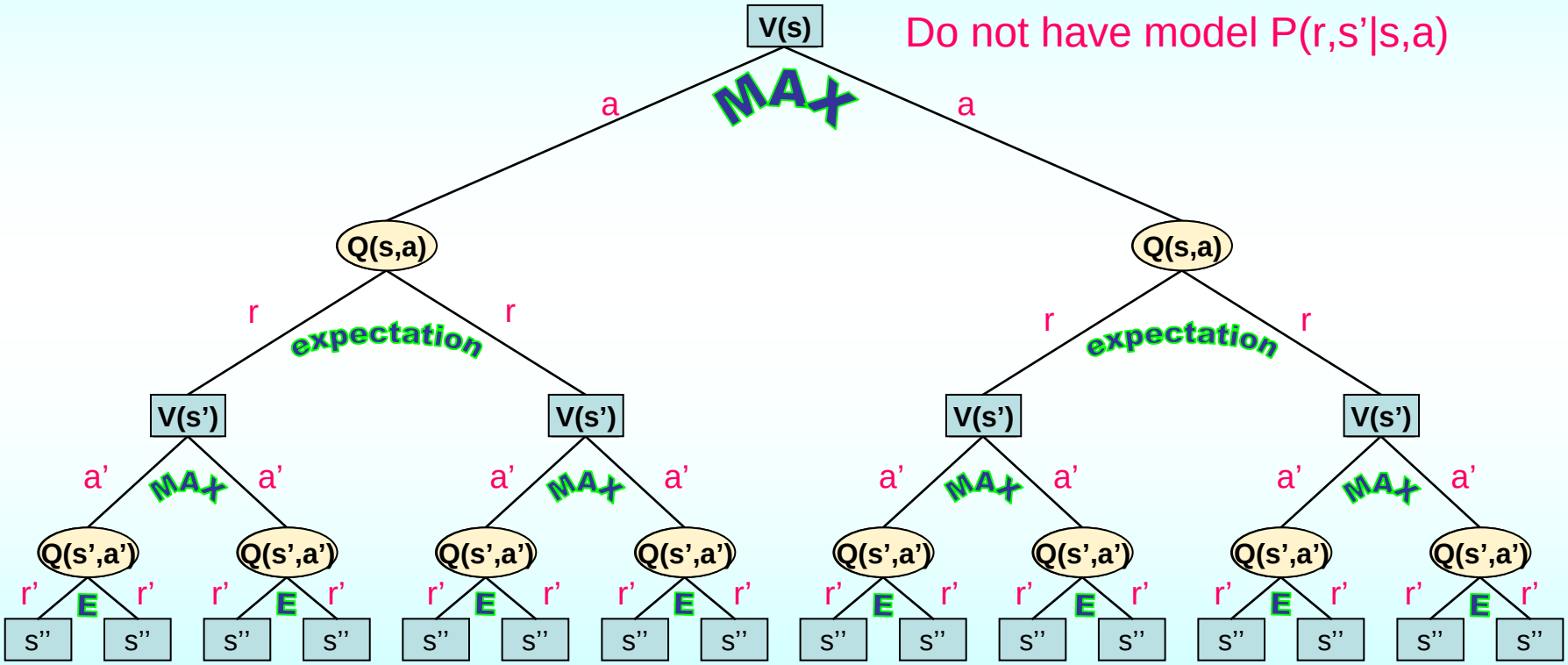
"Planning"



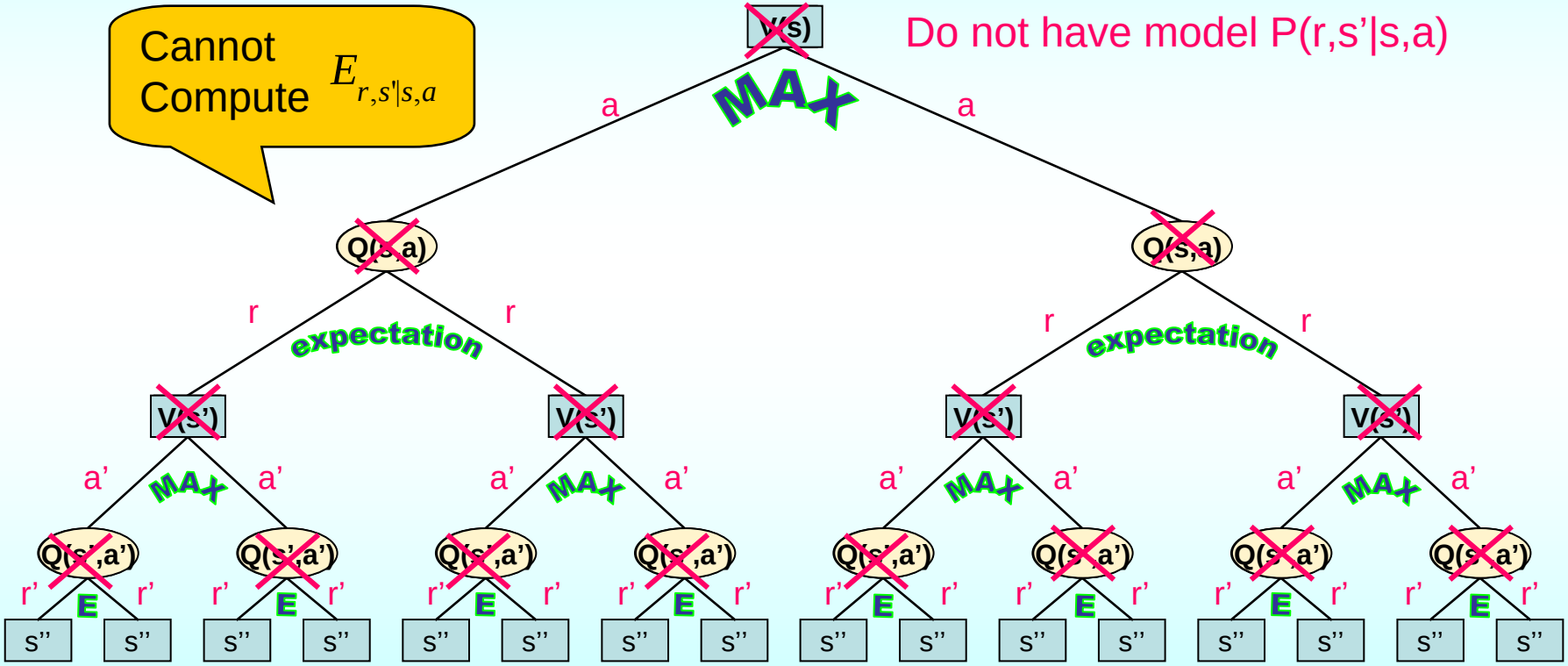
This is: *finite horizon, finite action, finite reward* case

General case: *Fixed point equations:* $V(s) = \sup_a Q(s,a)$ $\quad Q(s,a) = E_{r,s'|s,a}[r + \gamma V(s')]$

Reinforcement Learning



Reinforcement Learning



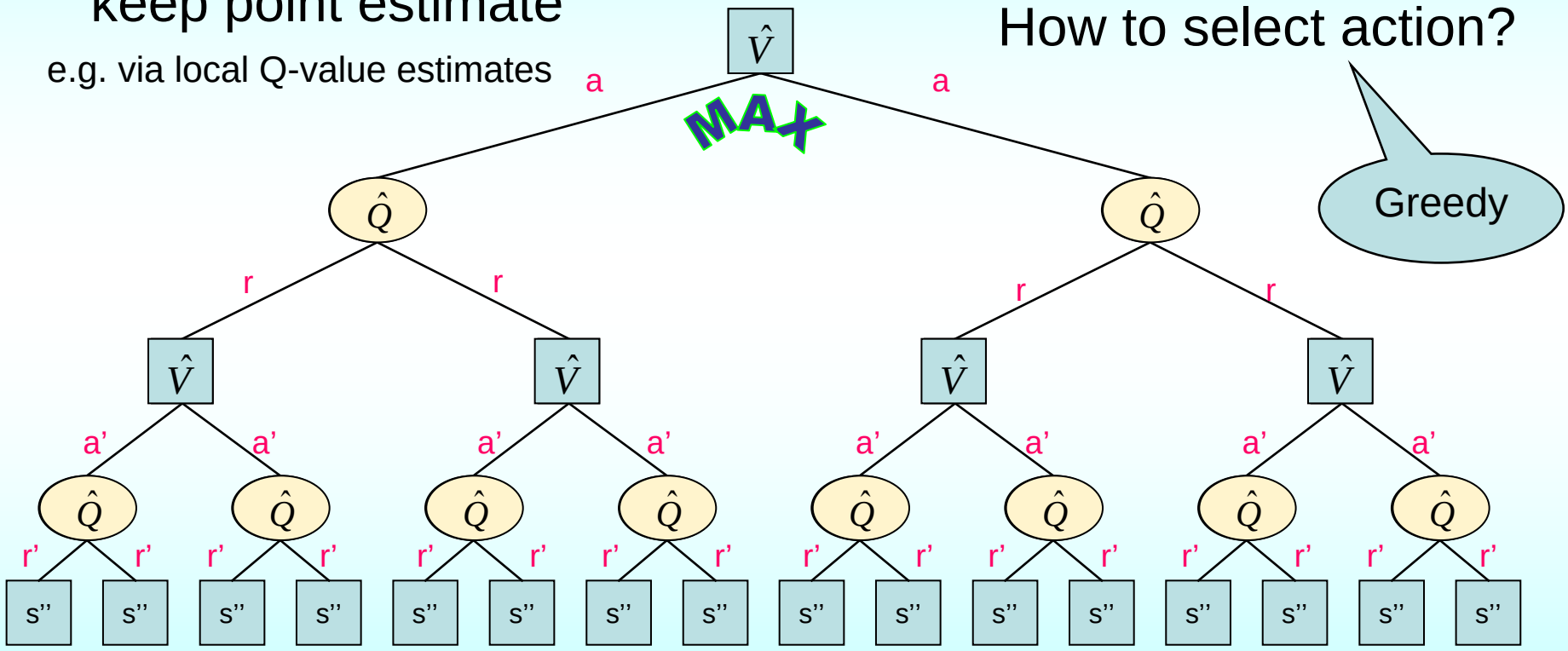
Reinforcement Learning

Standard approach:
keep point estimate

e.g. via local Q-value estimates

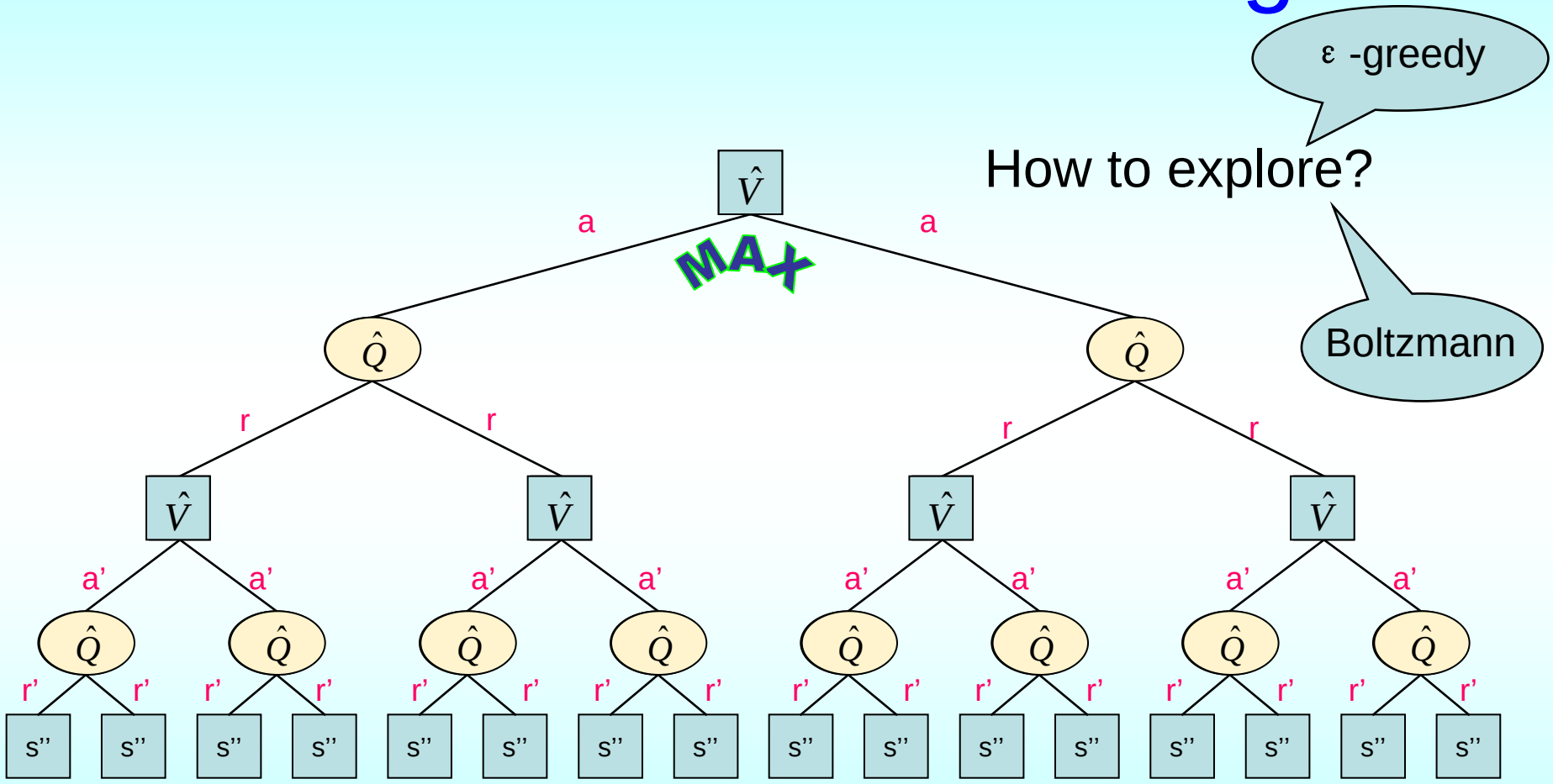
Do not have model P(r,s'|s,a)

How to select action?



Problem: greedy does not explore

Reinforcement Learning



Problem: do not account for uncertainty in estimates

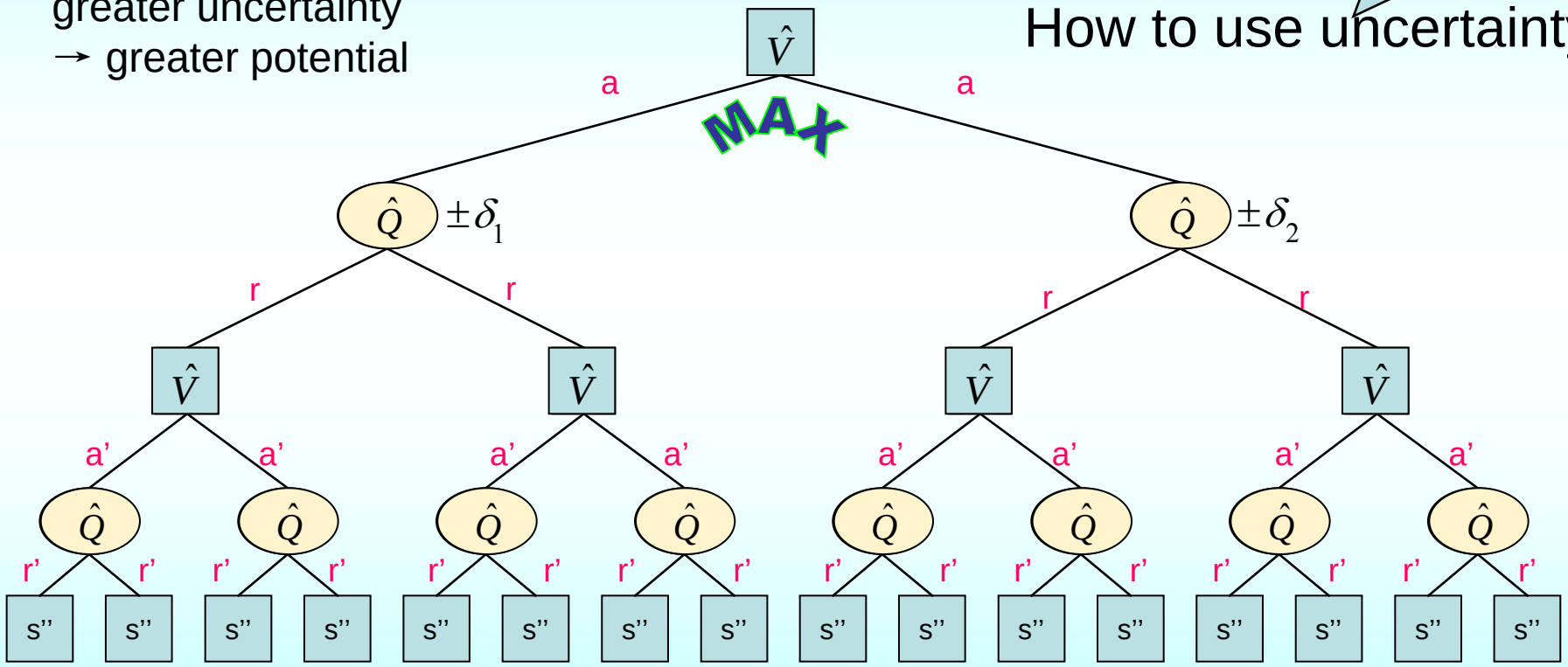
Reinforcement Learning

Interval estimation

How to use uncertainty?

Intuition:

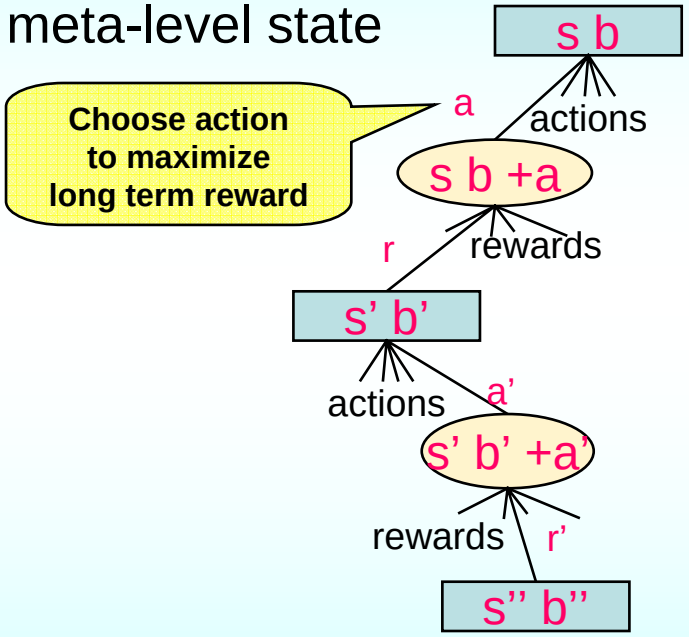
greater uncertainty
→ greater potential



Problem: δ 's computed myopically: doesn't consider horizon

Bayesian Reinforcement Learning

Prior $P(\theta)$ on model $P(rs' \mid sa, \theta)$ Belief state $b=P(\theta)$



Meta-level MDP

decision
a

outcome
r, s', b'

Meta-level Model
P(r,s'b'|s b,a)

decision
a'

outcome
r', s'', b''

Meta-level Model
P(r',s''b''|s' b',a')

Have a model for meta-level transitions!

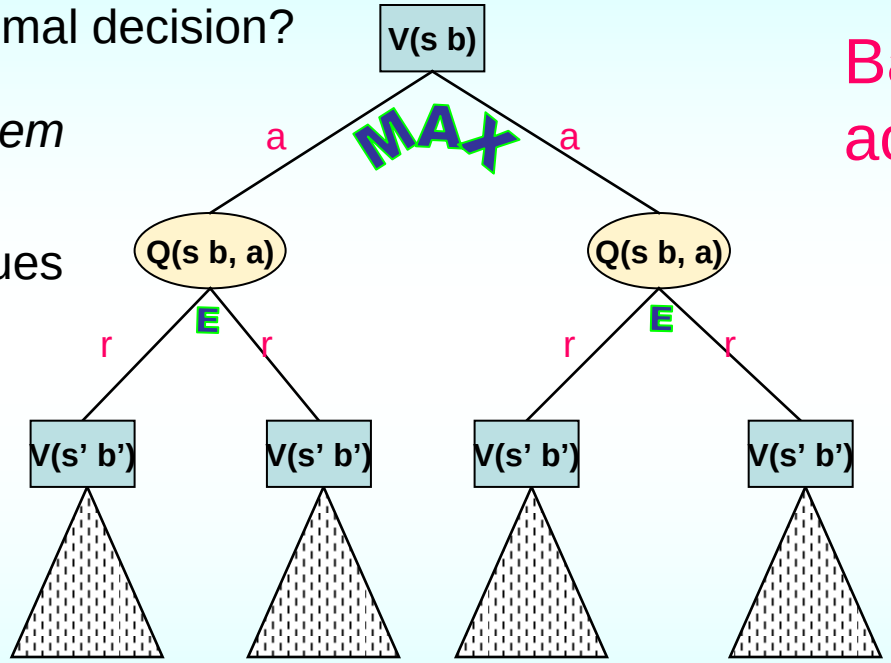
- based on posterior update and expectations over base-level MDPs

Bayesian RL Decision Making

How to make an optimal decision?

Solve planning problem in meta-level MDP:

- Optimal Q,V values



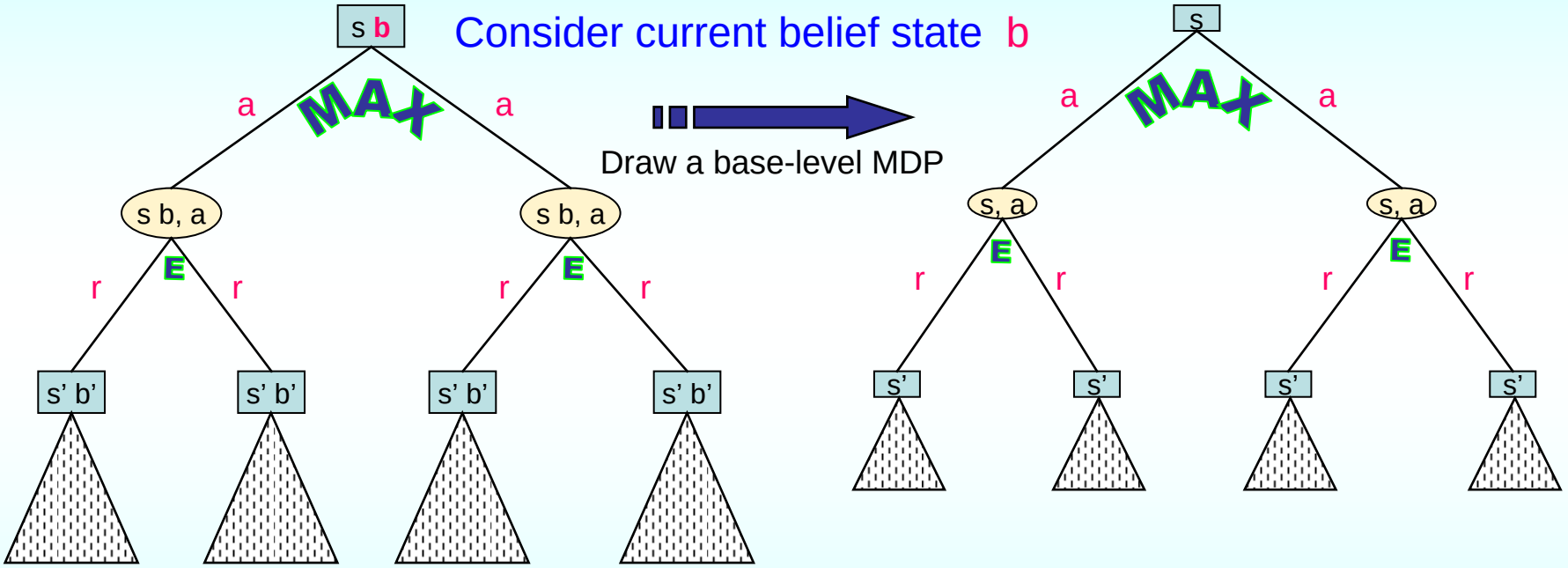
Bayes optimal action selection

Problem: meta-level MDP *much larger* than base-level MDP

Impractical

Bayesian RL Decision Making

Current approximation strategies:



Greedy approach:

current b → mean base-level MDP model

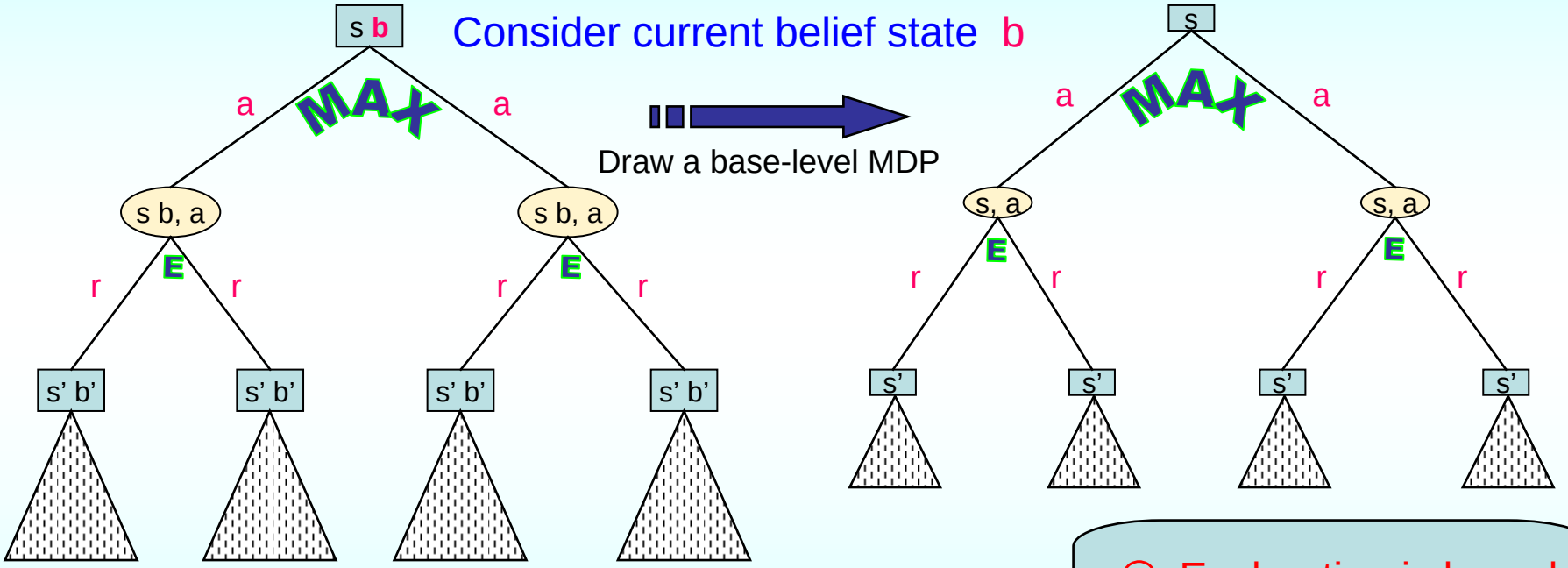
→ point estimate for Q, V

→ choose greedy action

But doesn't consider uncertainty

Bayesian RL Decision Making

Current approximation strategies:



Thompson approach:

current b → sample a base-level MDP model

→ point estimate for Q, V

(Choose action proportional to probability it is max Q)

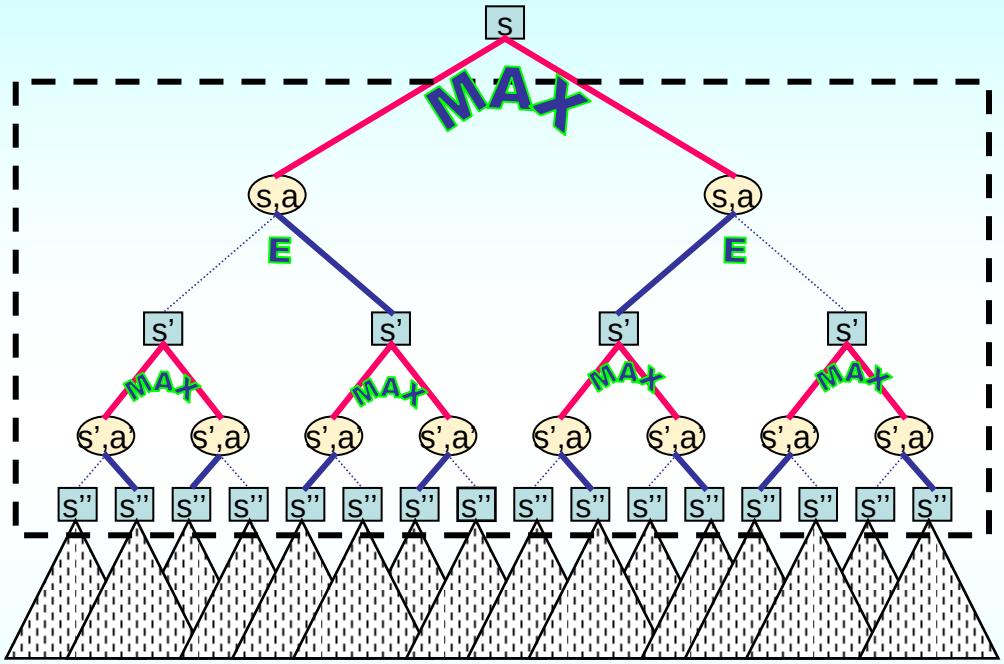
☺ Exploration is based on uncertainty

But still myopic

Our Approach

- Try to better approximate Bayes optimal action selection by performing lookahead
- Adapt “sparse sampling” (Kearns, Mansour ,Ng)
  - *Make some practical improvements*

Sparse Sampling



(Kearns, Mansour, Ng 2001)

Approximate values

Enumerate action choices

Subsample action outcomes

Bound depth

Back up approx values

+ Chooses approximately optimal action with high probability
  (if depth, sampling large enough)

− Achieving guarantees too expensive

+ But can control depth, sampling

Bayesian Sparse Sampling

Bayesian Sparse Sampling

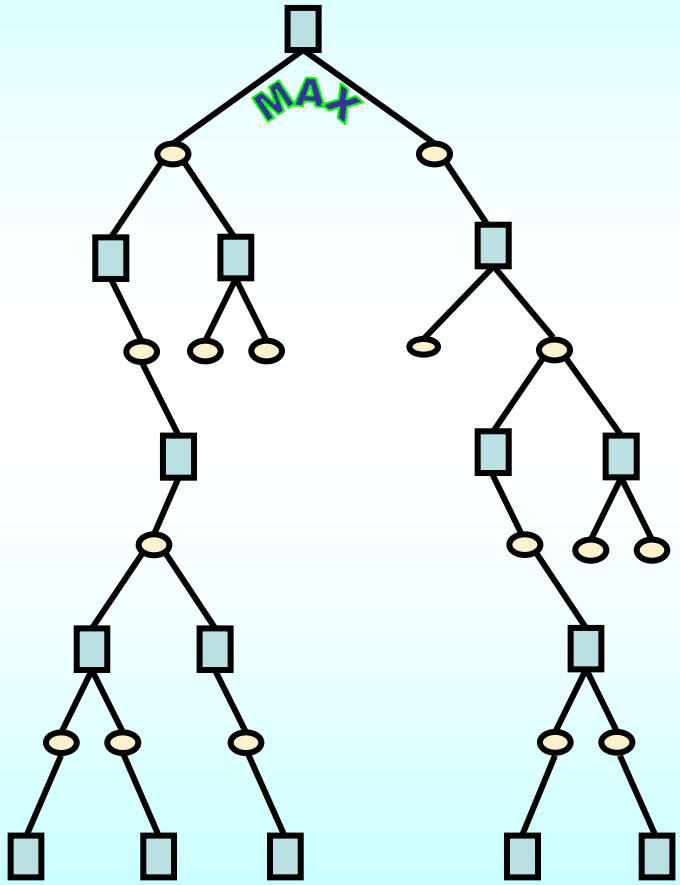
Observation 1

- Do not need to enumerate actions in a Bayesian setting
  - Given random variables $Q_1,\ldots,Q_K$
  - and a prior $P(Q_1,\ldots,Q_K)$
  - Can approximate $\max(Q_1,\ldots,Q_K)$
  - Without observing every variable

(Stop when posterior probability of a significantly better Q-value is small)

Bayesian Sparse Sampling

Observation 2



- Action value estimates are not equally important
  - Need better Q value estimates for some actions but not all
  - Preferentially expand tree under actions that might be optimal

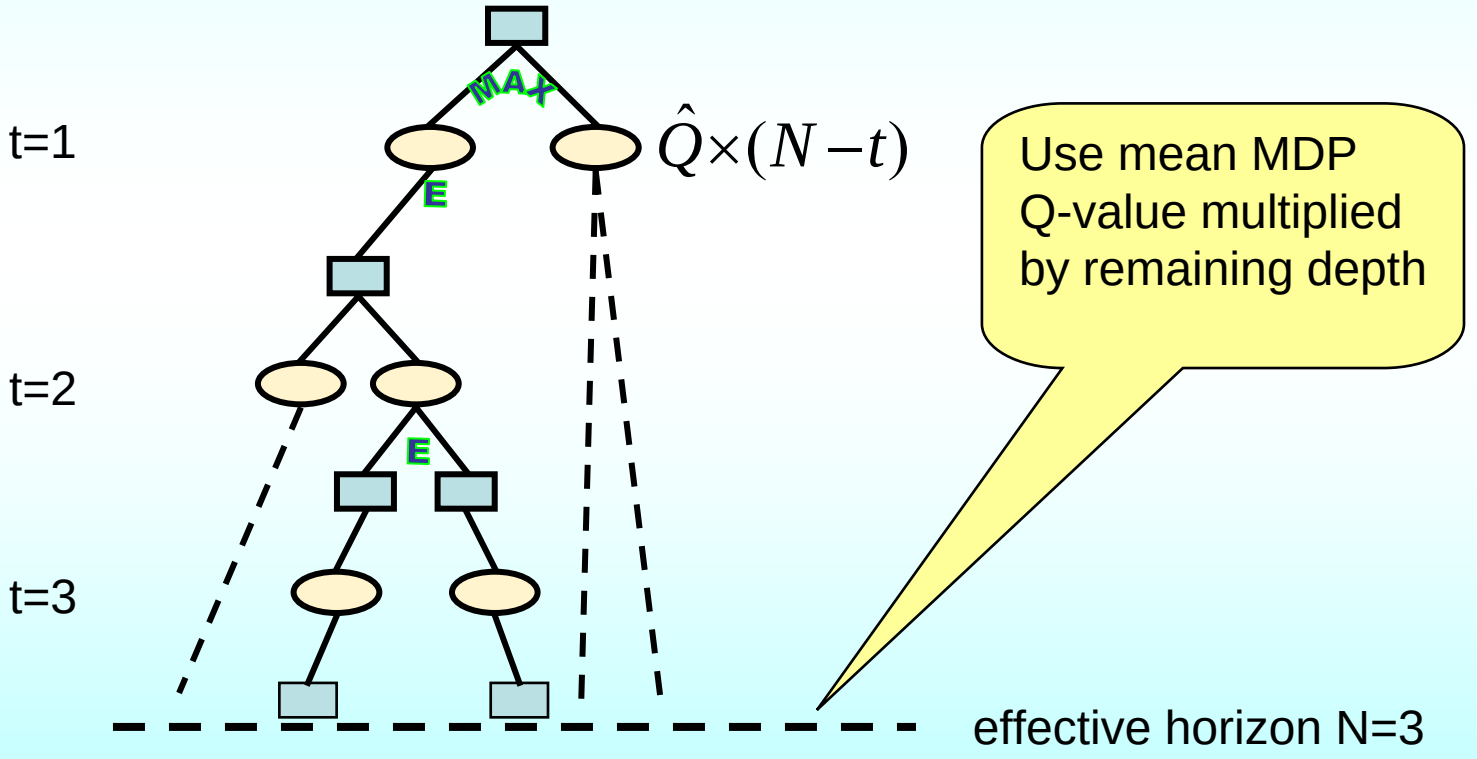
Biased tree growth

Use Thompson sampling to select actions to expand

Bayesian Sparse Sampling

Observation 3

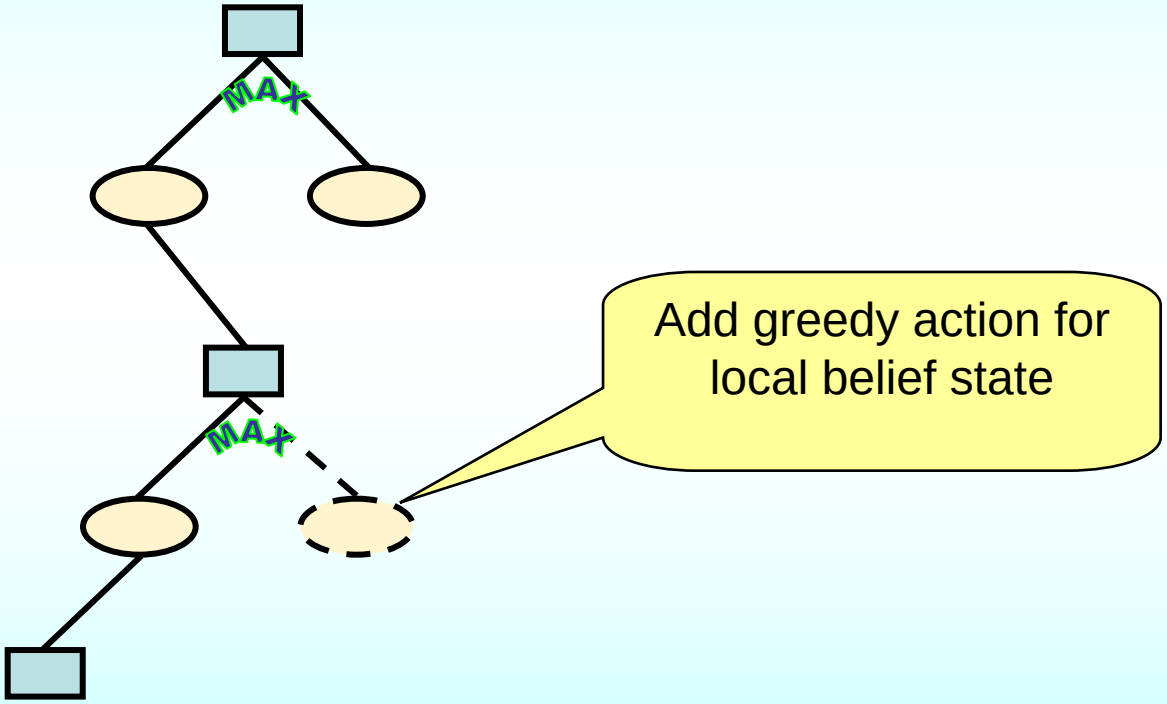
Correct leaf value estimates to same depth



Bayesian Sparse Sampling

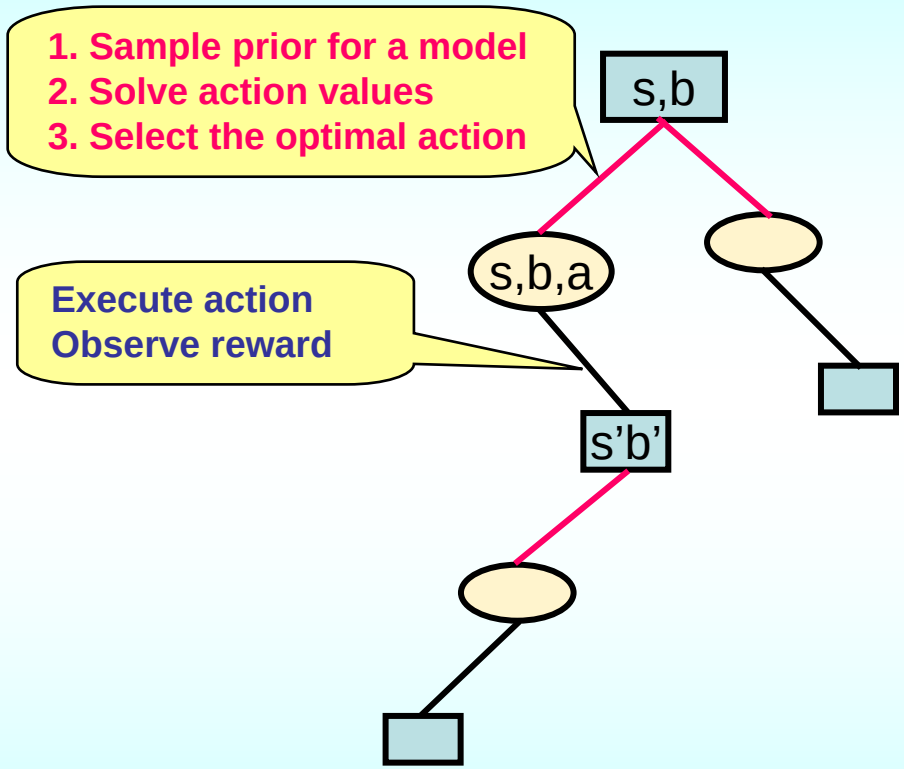
Observation 4

Include greedy action at decision nodes (if not sampled)



Bayesian Sparse Sampling

Tree growing procedure

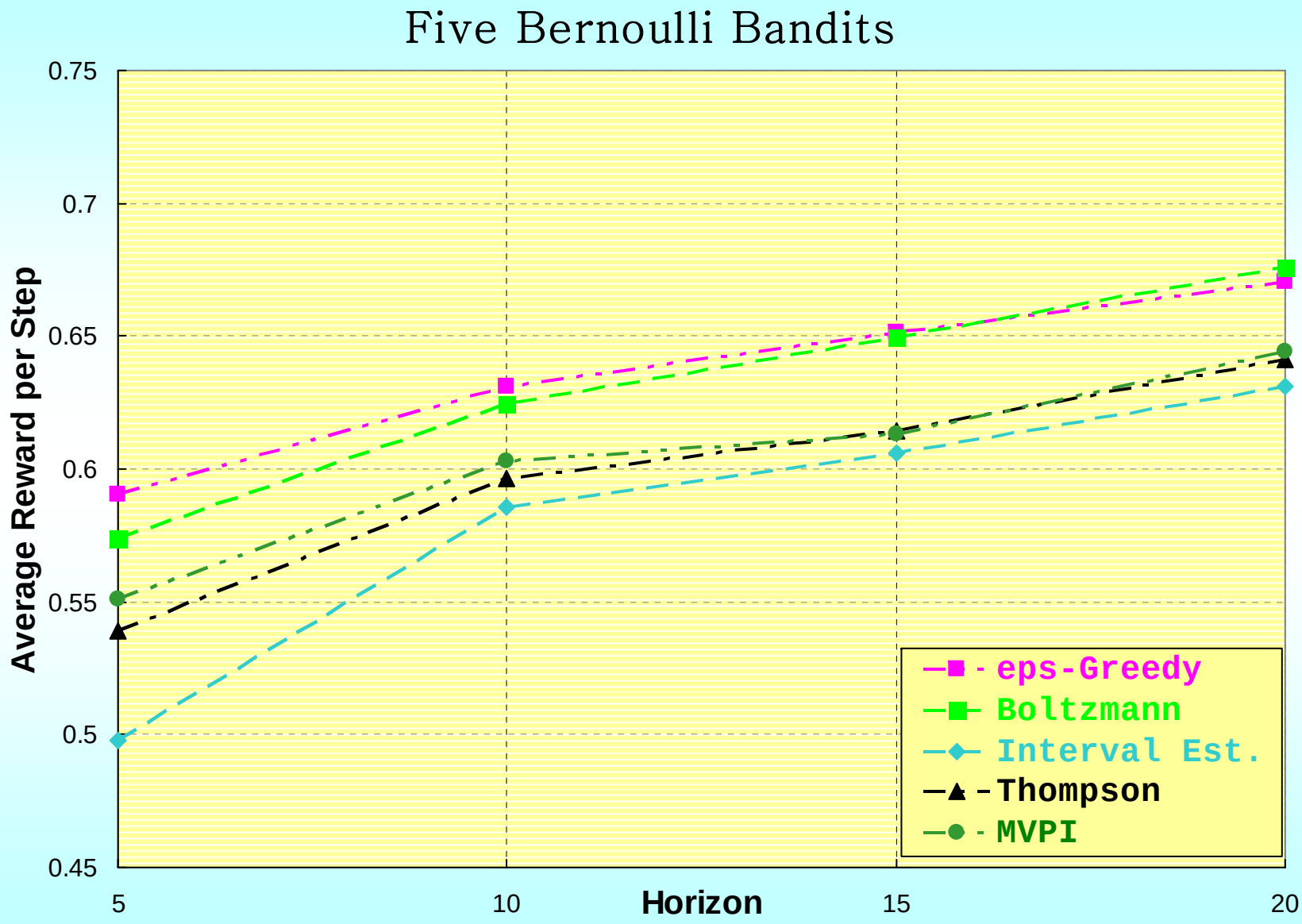


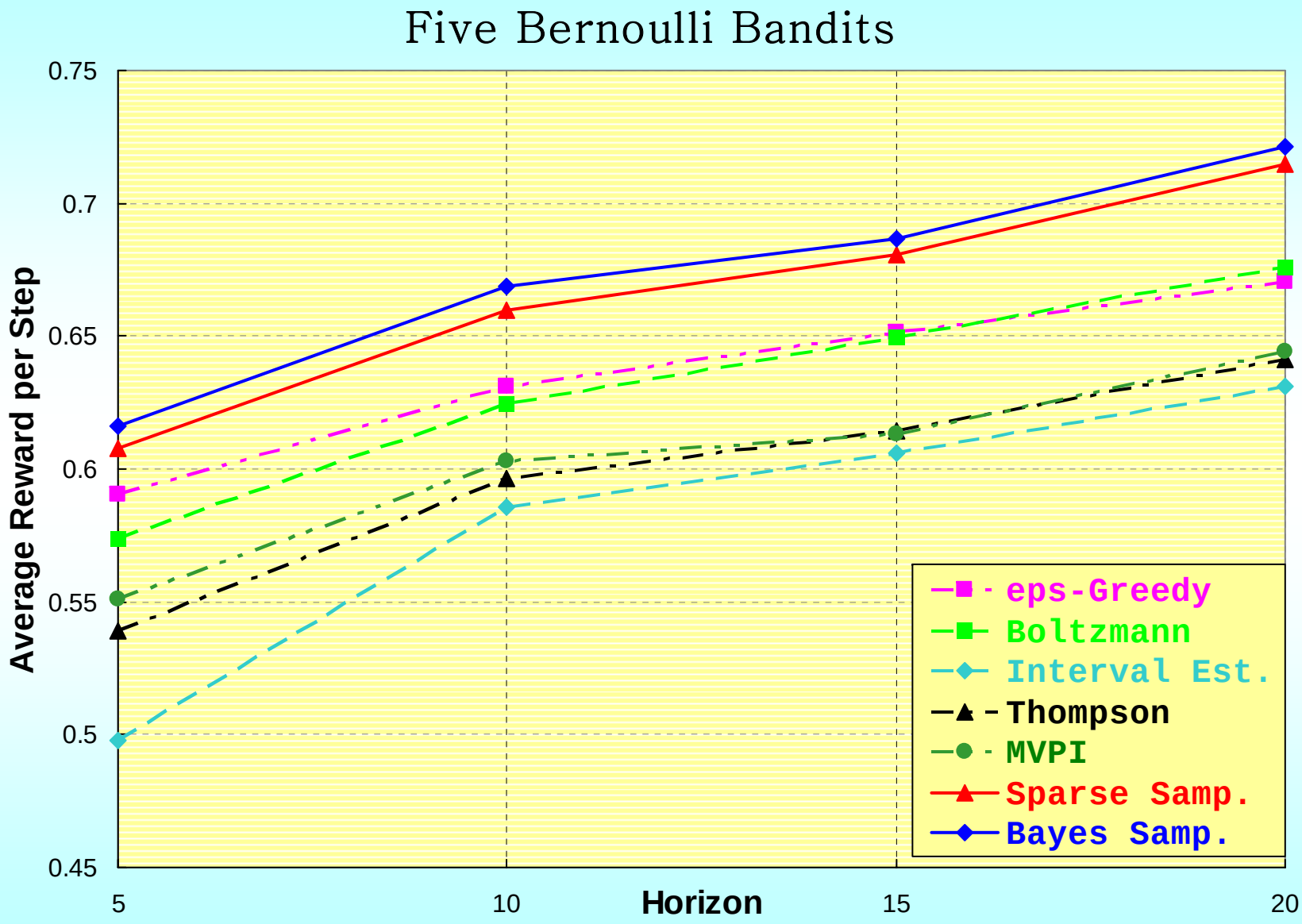
- Descend sparse tree from root
  - **Thompson sample actions**
  - Sample outcome
- Until new node added
- Repeat until tree size limit reached

Control computation by controlling tree size

Simple experiments

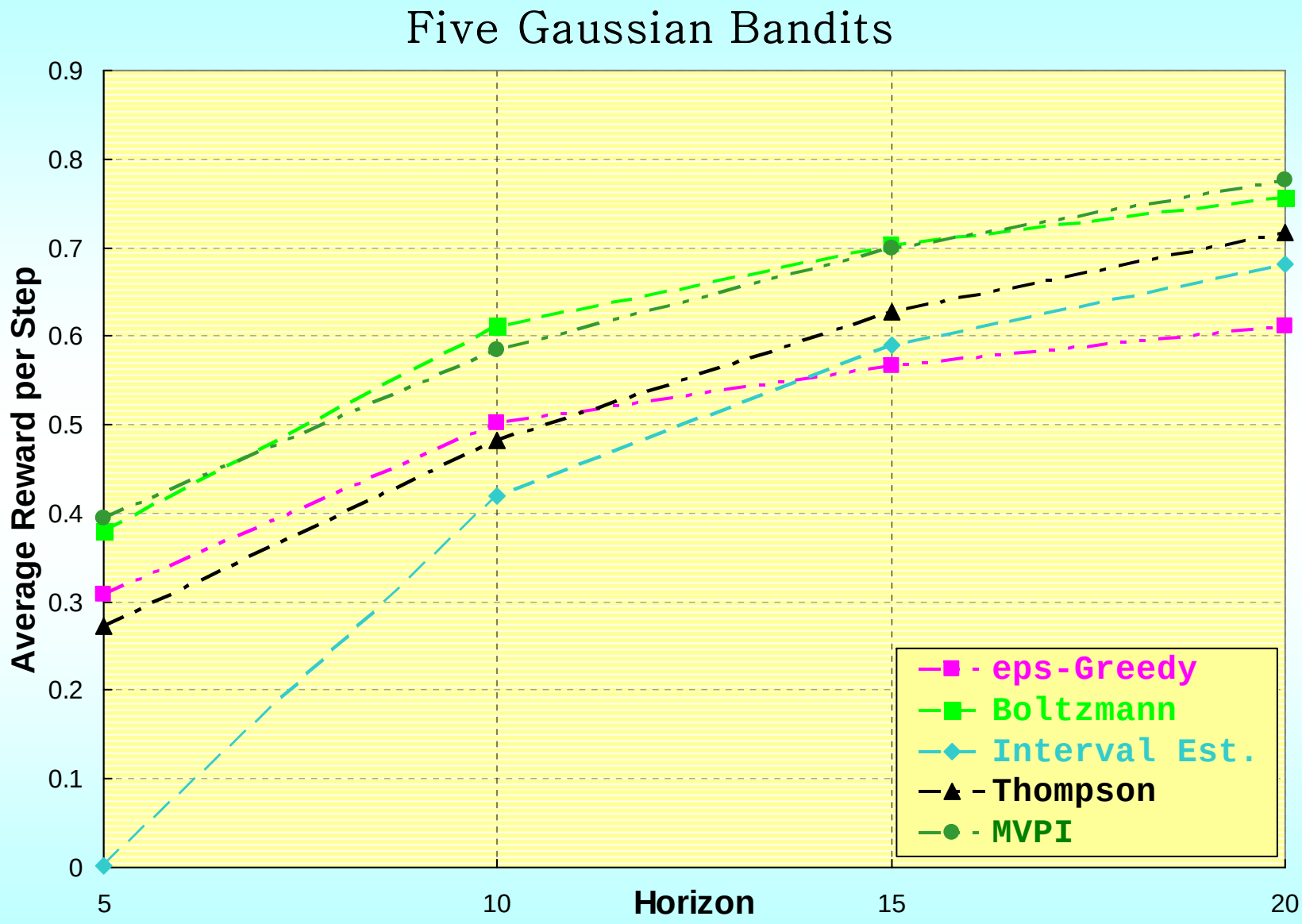
- 5 Bernoulli bandits $a_1,\ldots,a_5$
- *Beta* priors
- Sampled model from prior
- Run action selection strategies
- Repeat 3000 times
- Average accumulated reward per step

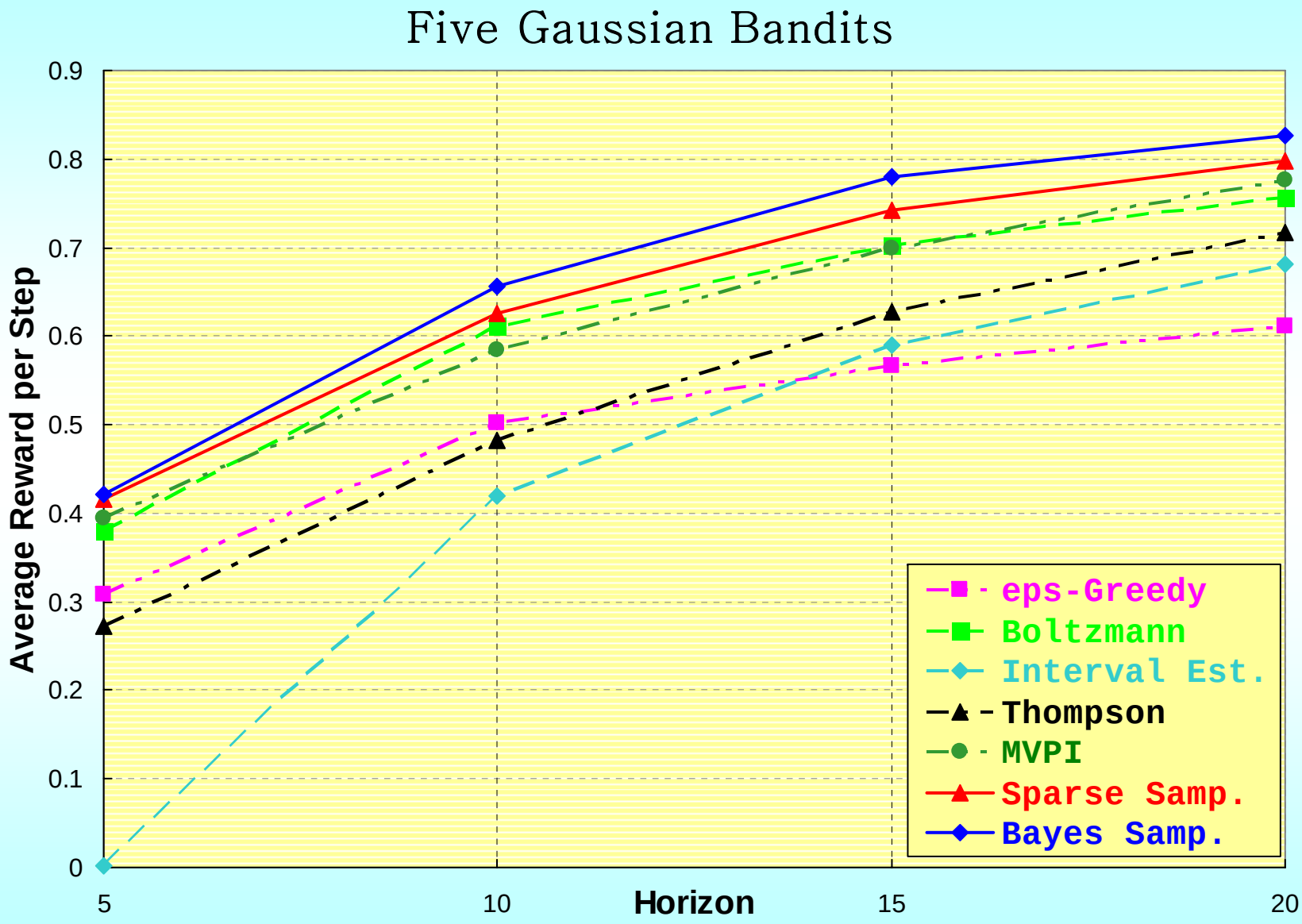

Five Bernoulli Bandits
Average Reward per Step
0.75
0.7
0.65
0.6
0.55
0.5
0.45
5
10
15
20
Horizon
eps-Greedy
Boltzmann
Interval Est.
Thompson
MVPI


Five Bernoulli Bandits
Average Reward per Step
0.75
0.7
0.65
0.6
0.55
0.5
0.45
5
10
15
20
Horizon
eps-Greedy
Boltzmann
Interval Est.
Thompson
MVPI
Sparse Samp.
Bayes Samp.

Simple experiments

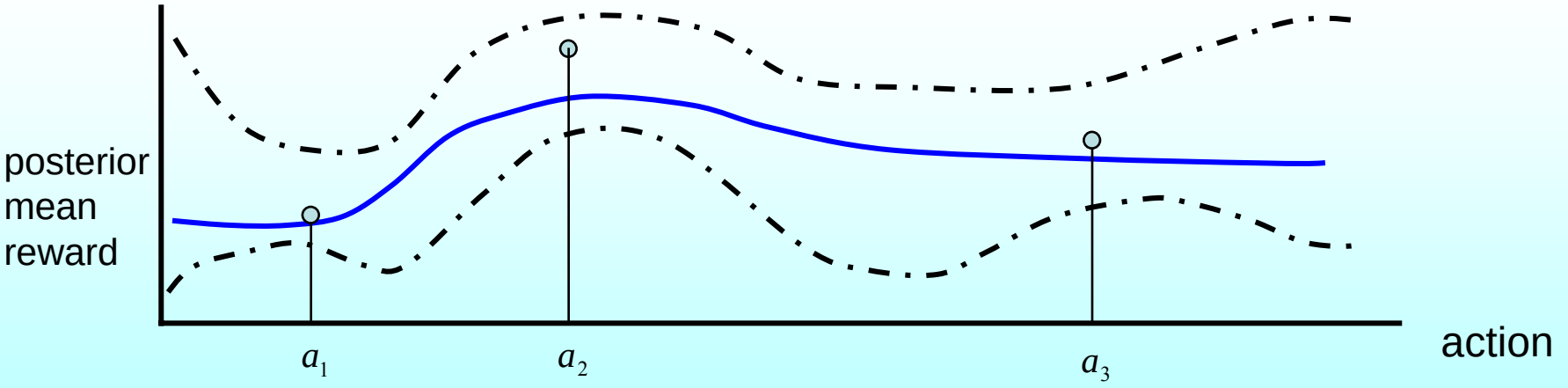
- 5 Gaussian bandits $a_1,\ldots,a_5$
- *Gaussian* priors
- Sampled model from prior
- Run action selection strategies
- Repeat 3000 times
- Average accumulated reward per step


Five Gaussian Bandits
Average Reward per Step
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
5
10
15
20
Horizon
eps-Greedy
Boltzmann
Interval Est.
Thompson
MVPI


Five Gaussian Bandits
Average Reward per Step
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
5
10
15
20
Horizon
eps-Greedy
Boltzmann
Interval Est.
Thompson
MVPI
Sparse Samp.
Bayes Samp.

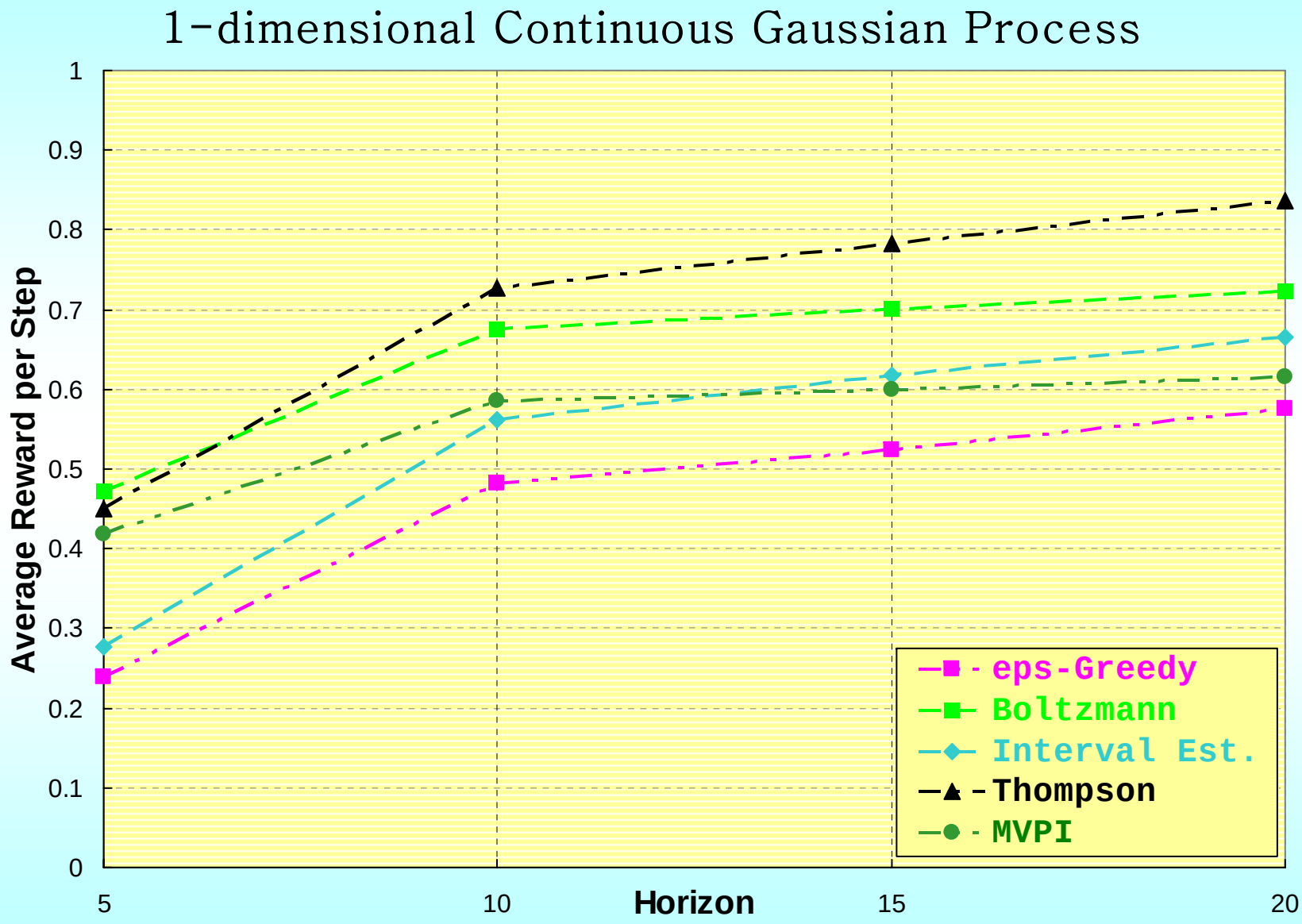
Gaussian process bandits

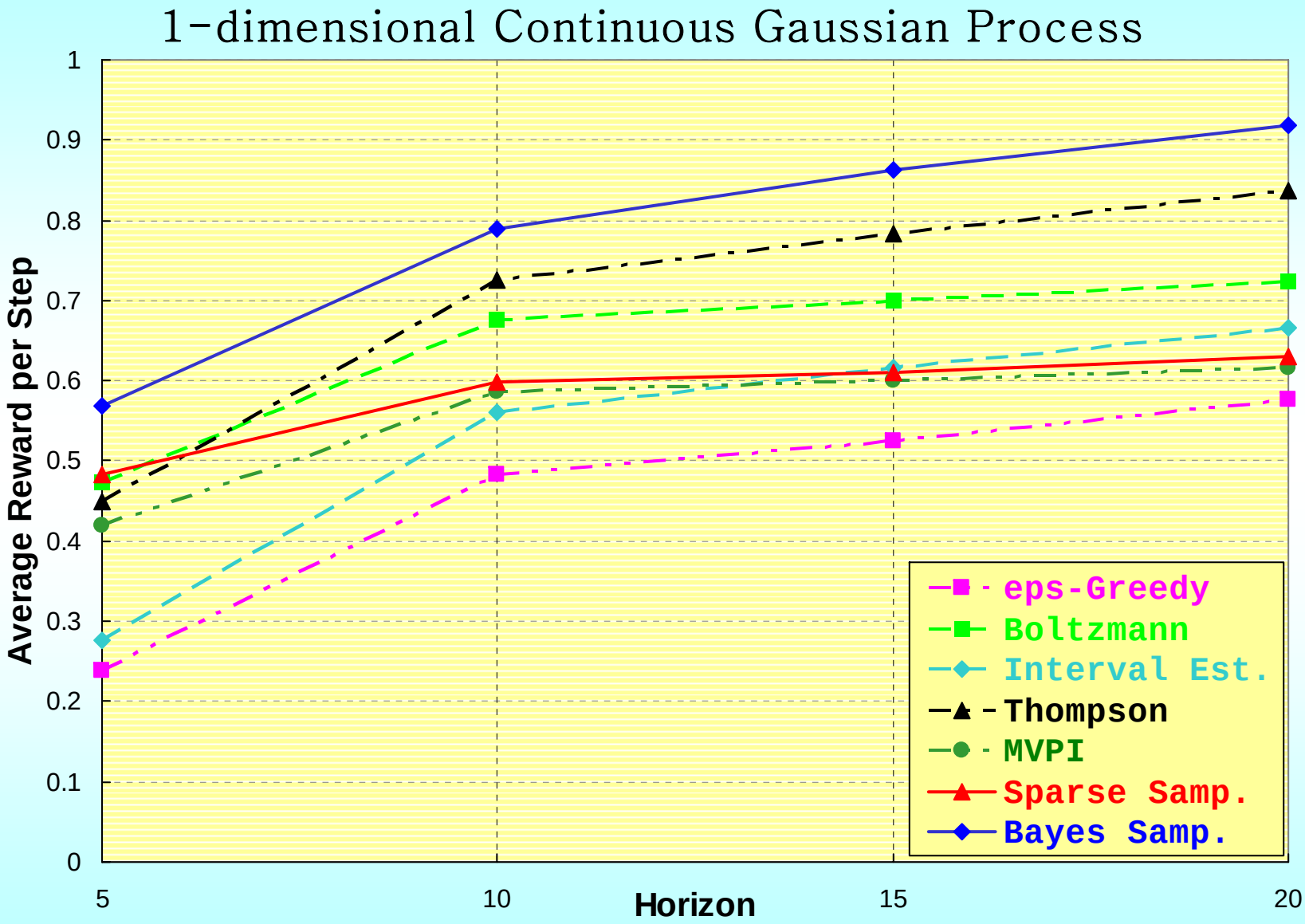
- General action spaces
  - Continuous actions, multidimensional actions
- Gaussian process prior over reward models
  - Covariance kernel between actions
- Action rewards correlated
- Posterior is a Gaussian process



Gaussian process experiments

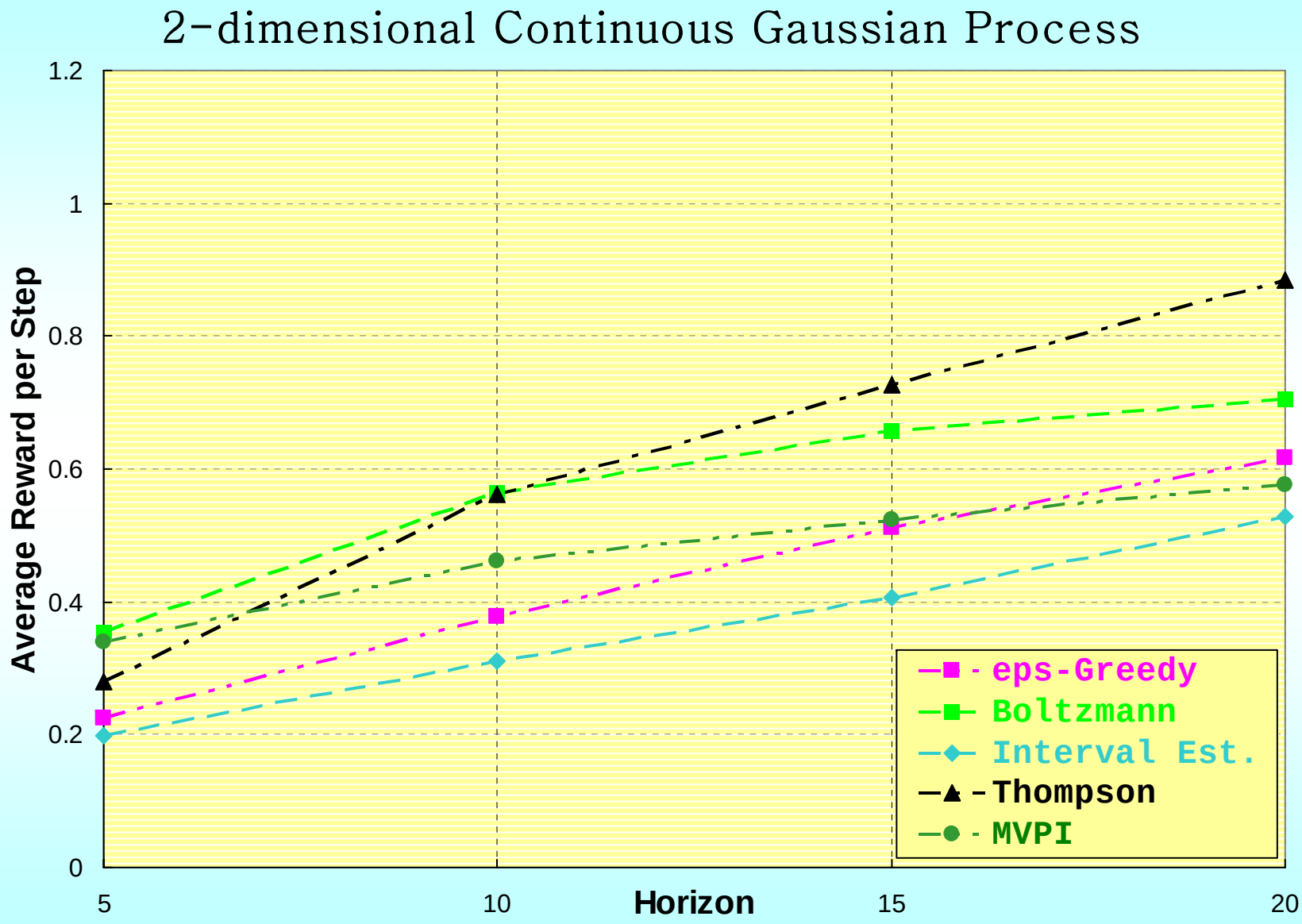
- 1 dimensional continuous action space
- GP priors RBF kernel
- Sampled model from prior
- Run action selection strategies
- Repeat 3000 times
- Average accumulated reward per step

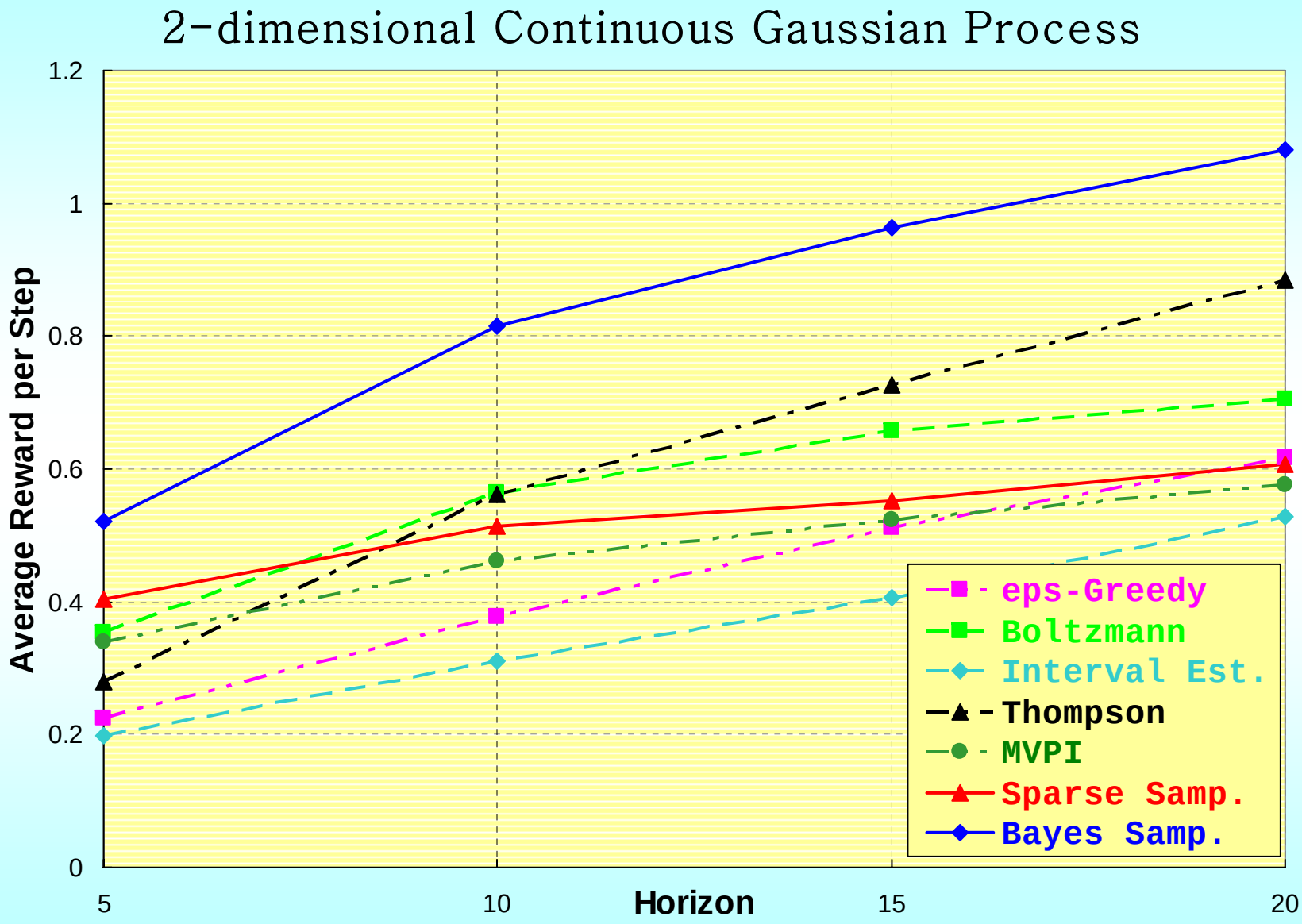

1-dimensional Continuous Gaussian Process
Average Reward per Step
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
5
10
15
20
Horizon
eps-Greedy
Boltzmann
Interval Est.
Thompson
MVPI


1-dimensional Continuous Gaussian Process
Average Reward per Step
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
5
10
15
20
Horizon
eps-Greedy
Boltzmann
Interval Est.
Thompson
MVPI
Sparse Samp.
Bayes Samp.

Gaussian process experiments

- 2 dimensional continuous action space
- GP priors RBF kernel
- Sampled model from prior
- Run action selection strategies
- Repeat 3000 times
- Average accumulated reward per step


2-dimensional Continuous Gaussian Process
Average Reward per Step
1.2
1
0.8
0.6
0.4
0.2
0
5
10
15
20
Horizon
eps-Greedy
Boltzmann
Interval Est.
Thompson
MVPI


2-dimensional Continuous Gaussian Process
Average Reward per Step
1.2
1
0.8
0.6
0.4
0.2
0
5
10
15
20
Horizon
eps-Greedy
Boltzmann
Interval Est.
Thompson
MVPI
Sparse Samp.
Bayes Samp.

Gaussian Process Bandits

- Very flexible model
- Actions can be complicated
  - e.g. a parameterized policy
  - Just need a kernel between policies
- Applications in robotics & game playing
- Reward = total reward accumulated by a policy in an episode

Summary

Bayesian sparse sampling

- Flexible and practical technique for improving action selection
- Reasonably straightforward

- Bandit problems
  - Planning is “easy”
    (at least approximate planning is “easy”)

Other Work

AIBO dog walking

Opponent modeling (Kuhn poker)

Vendor-bot (Pioneer)

Improve tree search?

Theoretical guarantees?

Cheaper re-planning?

Incorporate value fun. approx.

That's it