To Infinity and Beyond

… or …

Bayesian Nonparametric Approaches for Reinforcement Learning in Partially Observable Domains

Finale Doshi-Velez
December 2, 2011

To Infinity and Beyond

… or …

Bayesian Nonparametric Approaches for Reinforcement Learning in Partially Observable Domains

Finale Doshi-Velez
December 2, 2011

Outline

- Introduction: The partially-observable reinforcement learning setting
- Framework: Bayesian reinforcement learning
- Applying nonparametrics:
  - Infinite Partially Observable Markov Decision Processes
  - Infinite State Controllers*
  - Infinite Dynamic Bayesian Networks*
- Conclusions and Continuing Work

* joint work with David Wingate

Outline

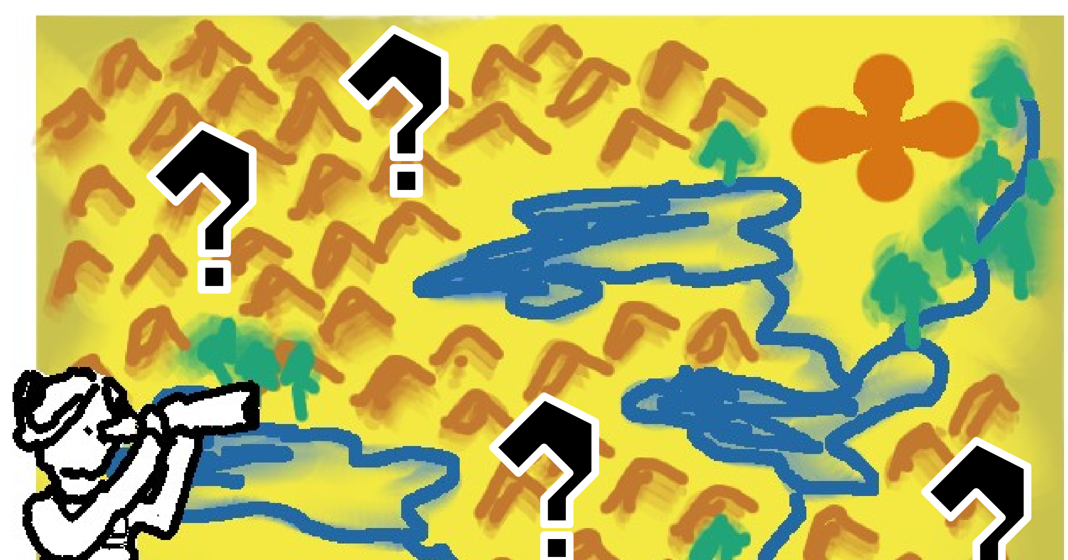
- Introduction: The partially-observable reinforcement learning setting
- Framework: Bayesian reinforcement learning
- Applying nonparametrics:
  - Infinite Partially Observable Markov Decision Processes
  - Infinite State Controllers
  - Infinite Dynamic Bayesian Networks
- Conclusions and Continuing Work

Example: Searching for Treasure

Example: Searching for Treasure

Example: Searching for Treasure

Partial Observability: can't tell where we are just by looking...

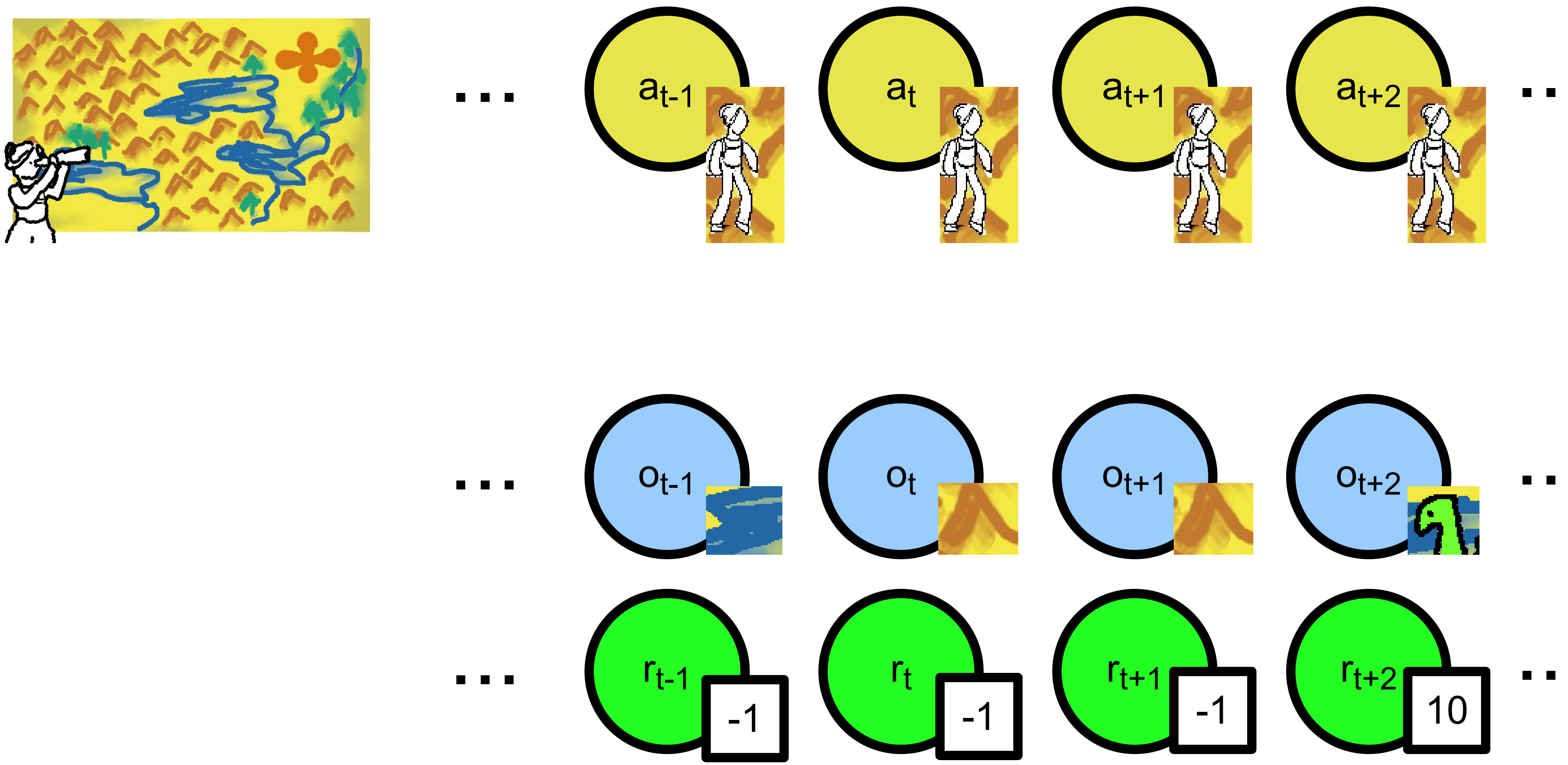


Example: Searching for Treasure

Example: Searching for Treasure

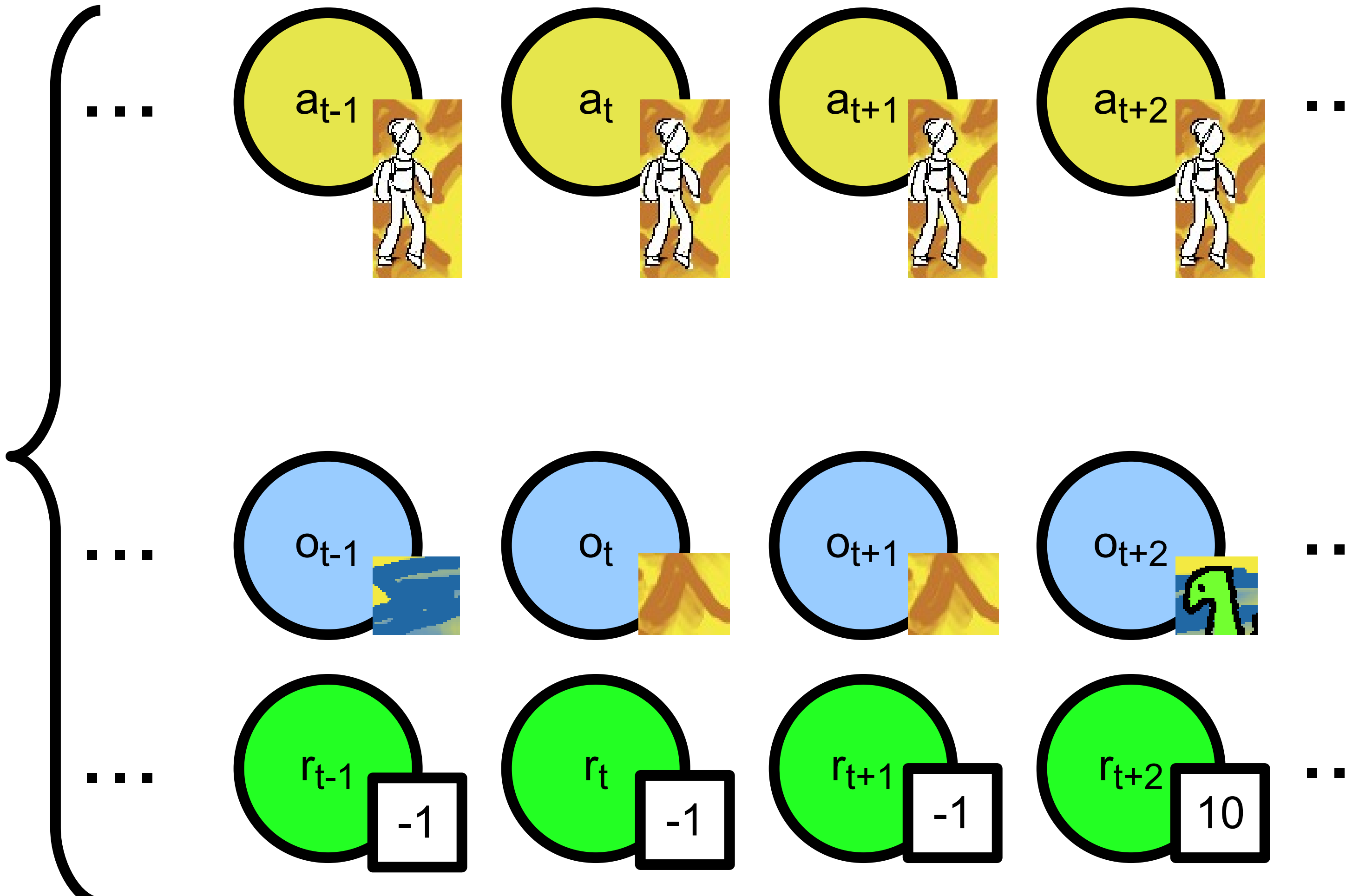
Reinforcement Learning:
don't even have a map!

The Partially Observable Reinforcement Learning Setting



The Partially Observable Reinforcement Learning Setting

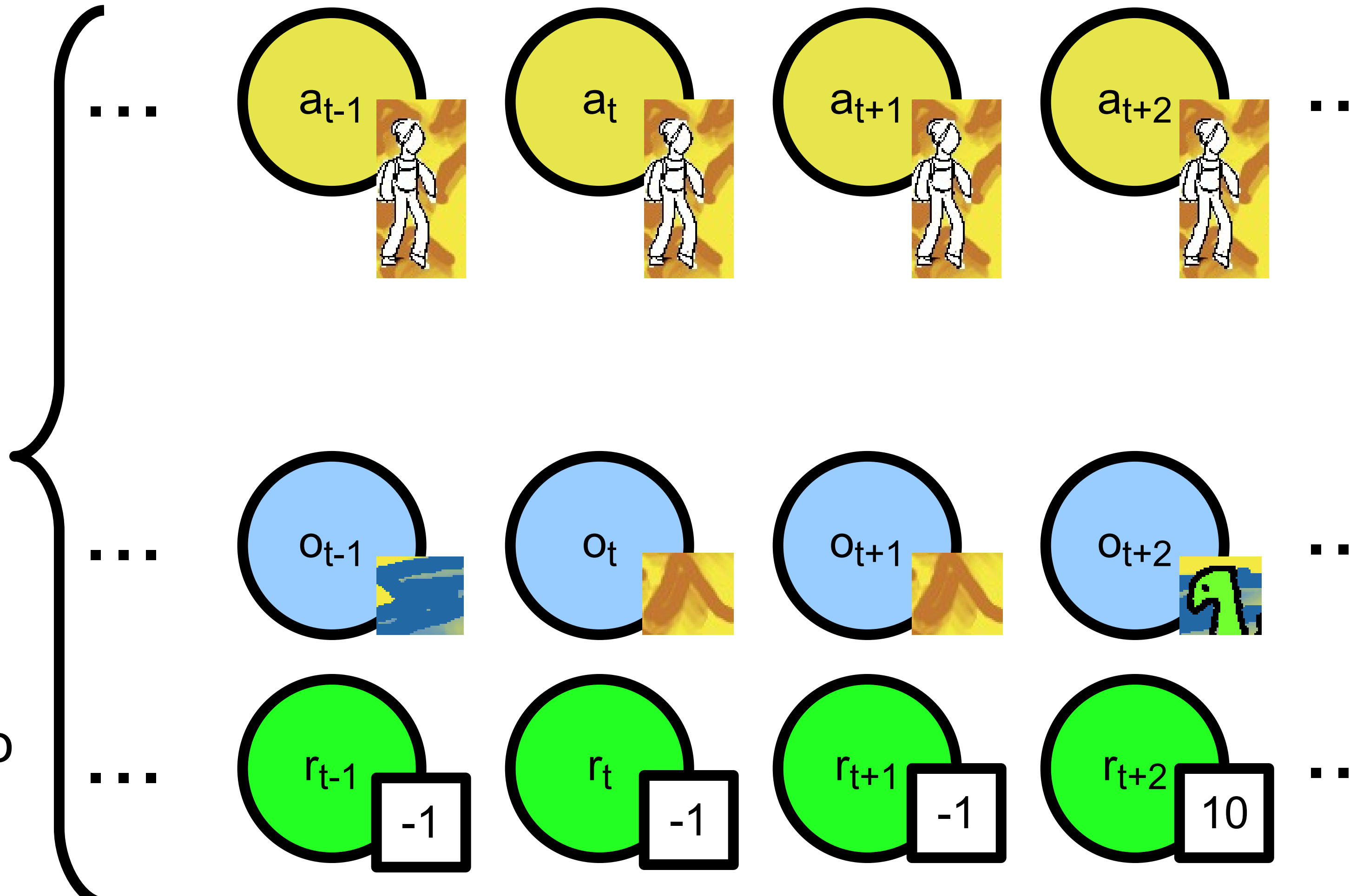
Given a **history** of actions, observations, and rewards



The Partially Observable Reinforcement Learning Setting

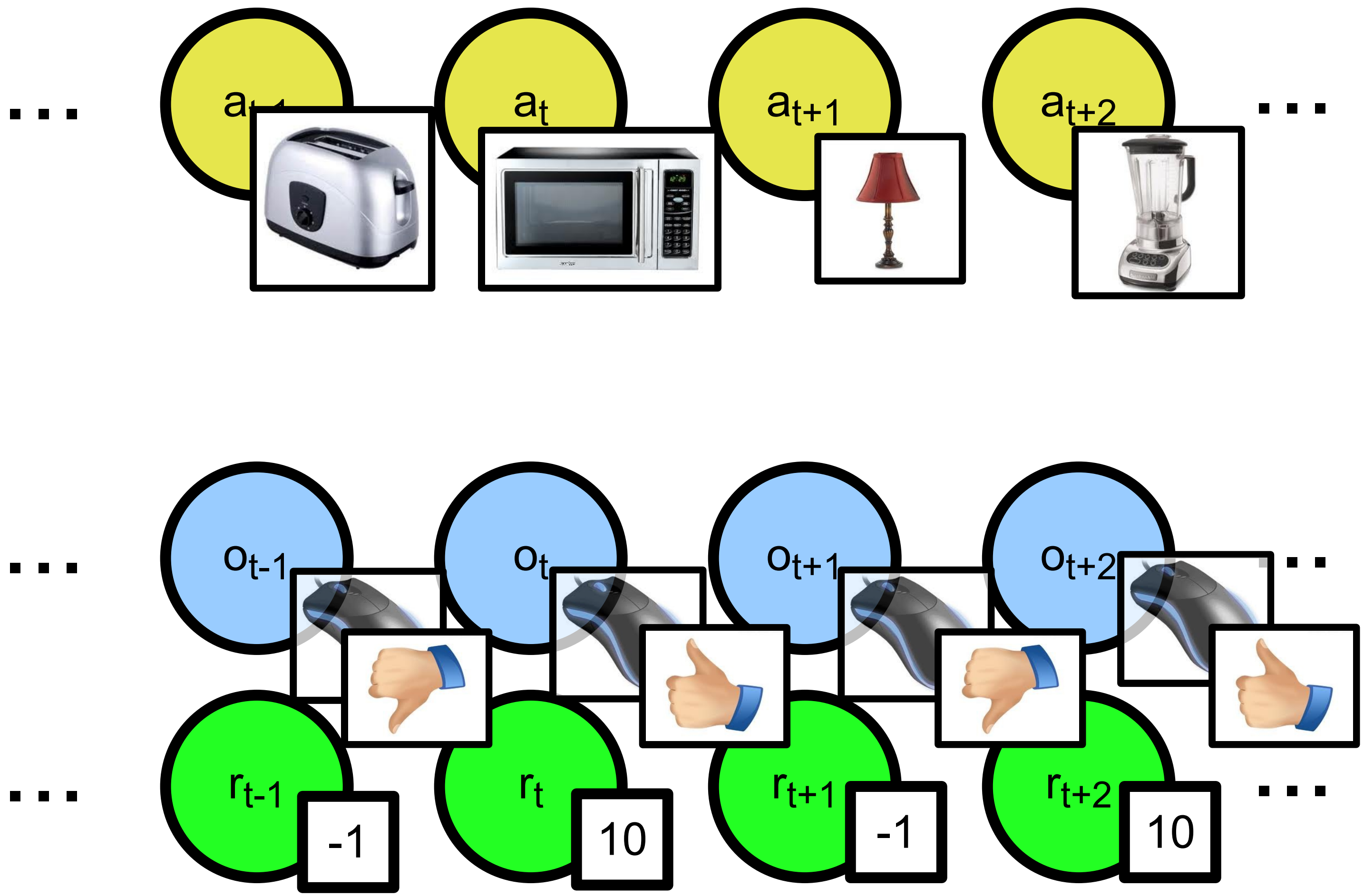
Given a **history** of actions, observations, and rewards

How can we act in order to **maximize long-term future rewards**?

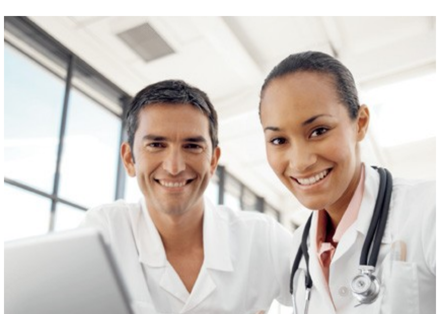


The Partially Observable Reinforcement Learning Setting

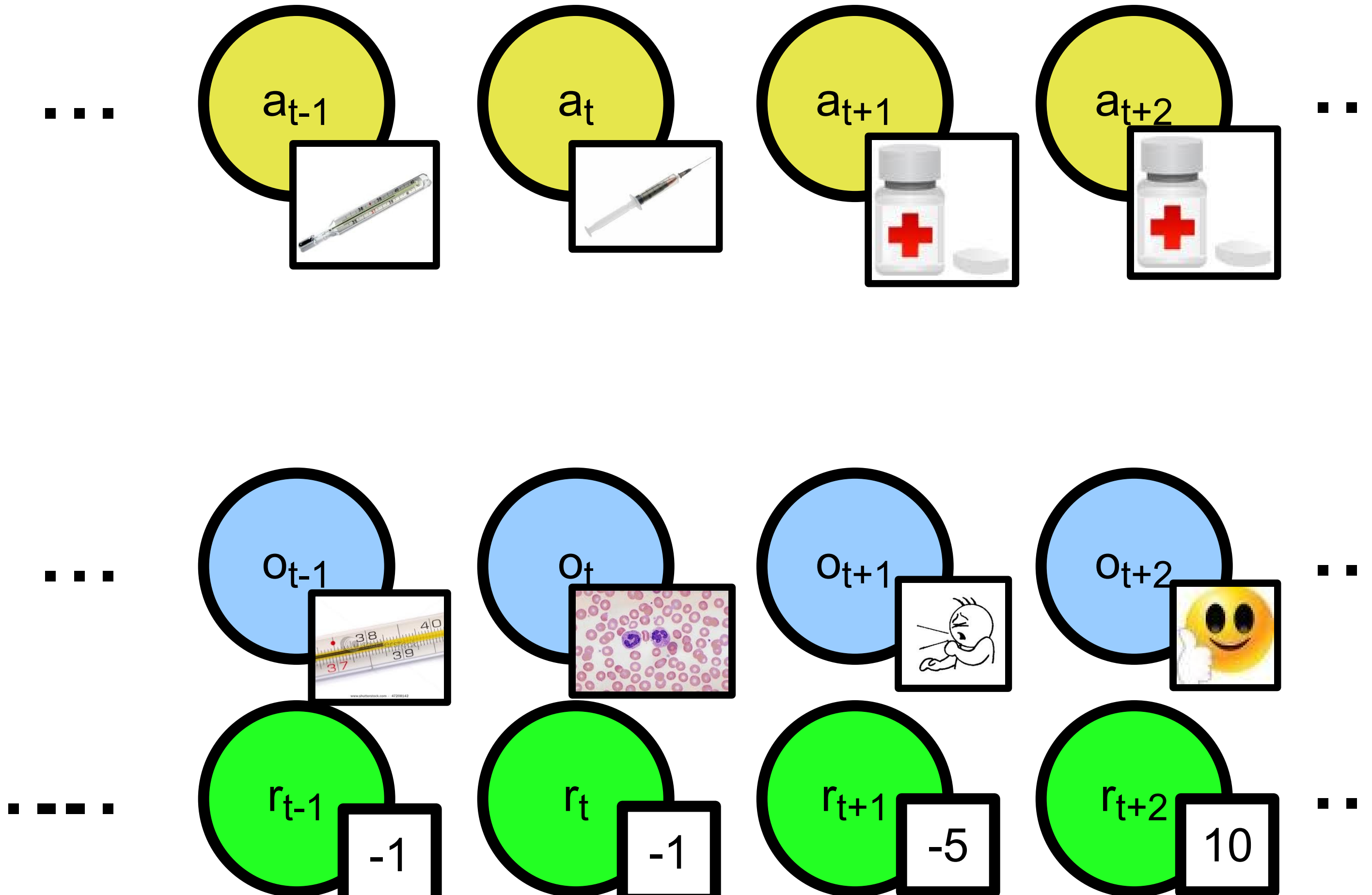
recommender systems



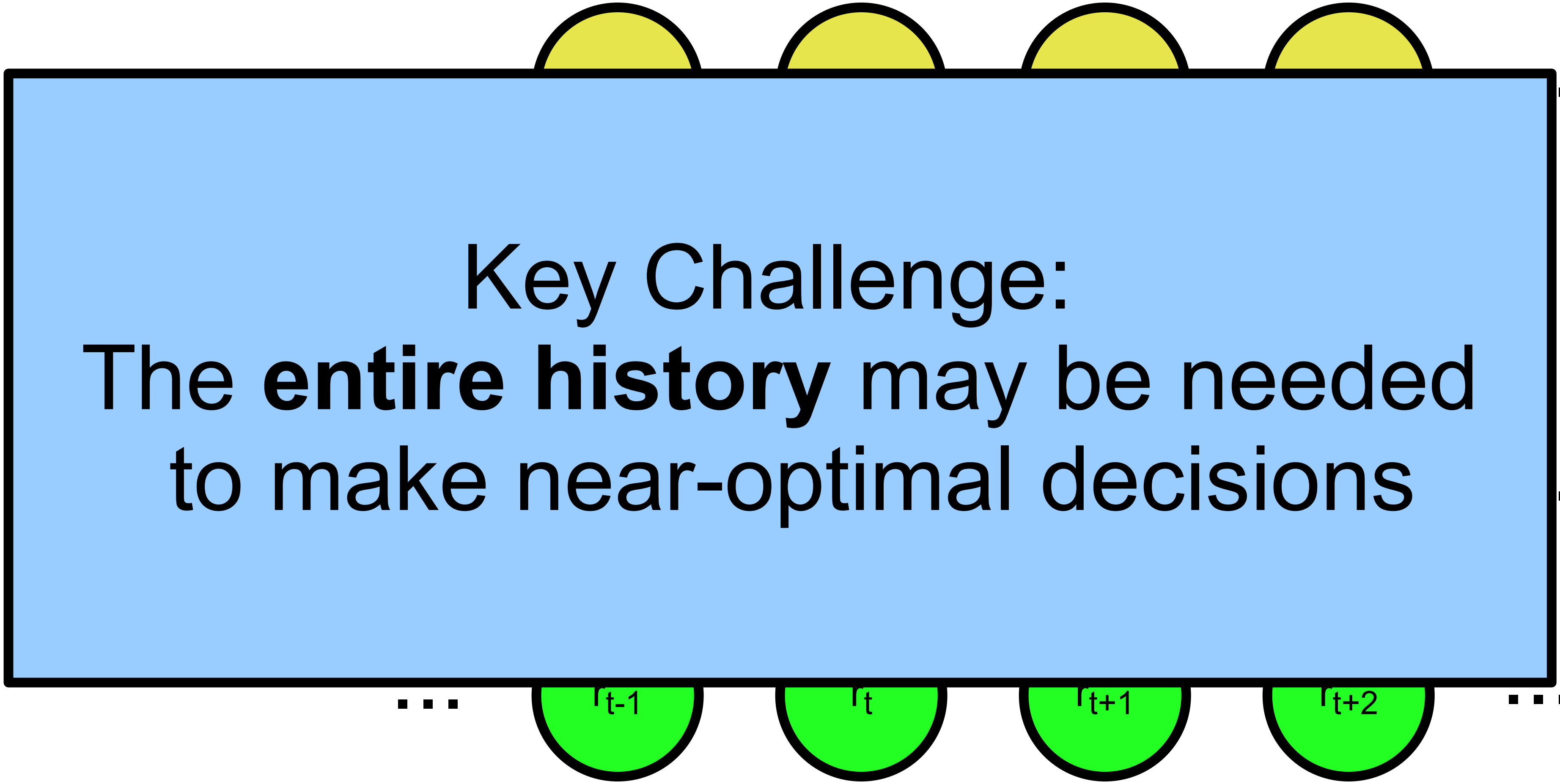
The Partially Observable Reinforcement Learning Setting



clinical diagnostic tools



Motivation:
the Reinforcement Learning Setting

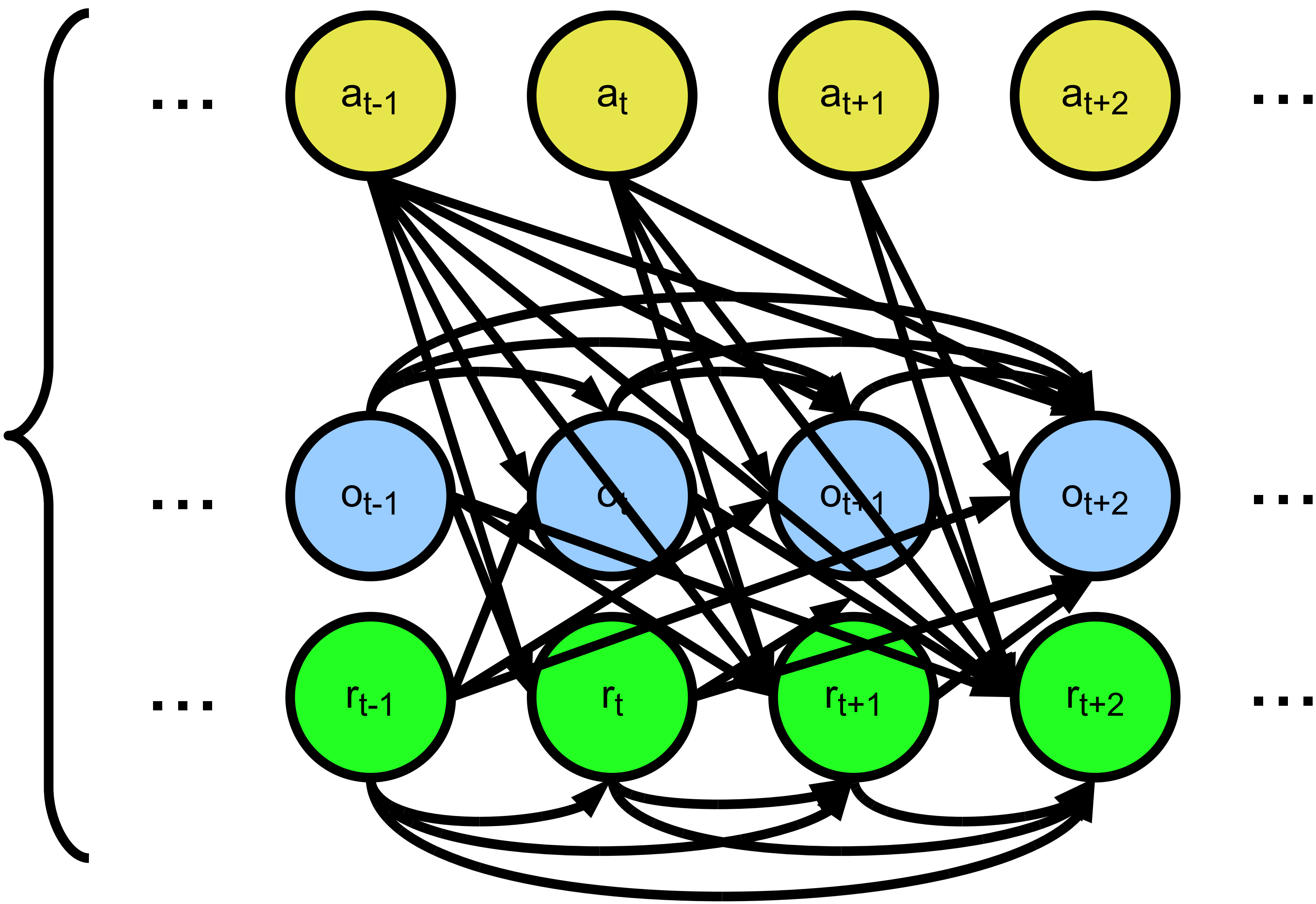


Key Challenge:

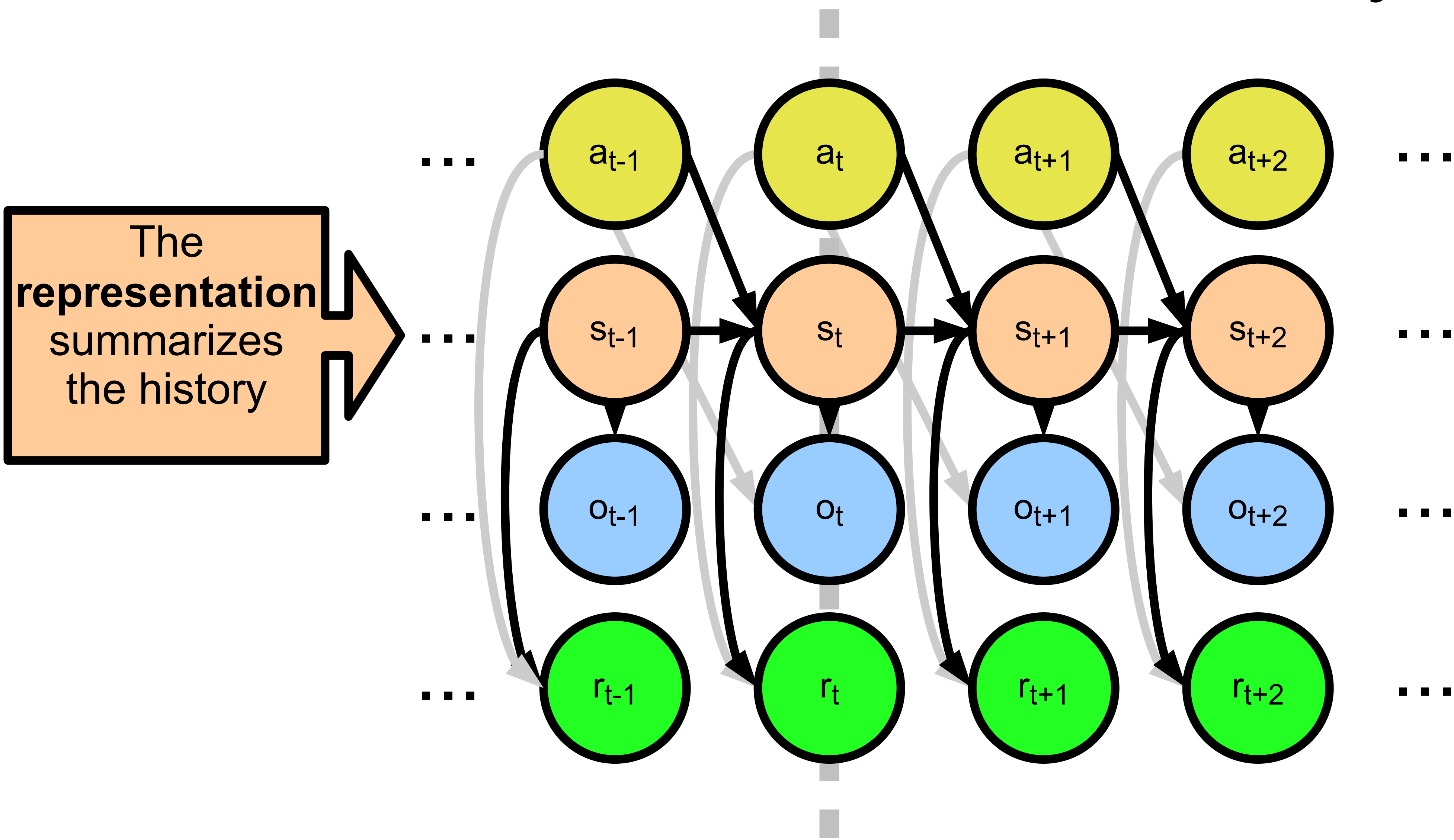
The **entire history** may be needed to make near-optimal decisions

The Partially Observable Reinforcement Learning Setting

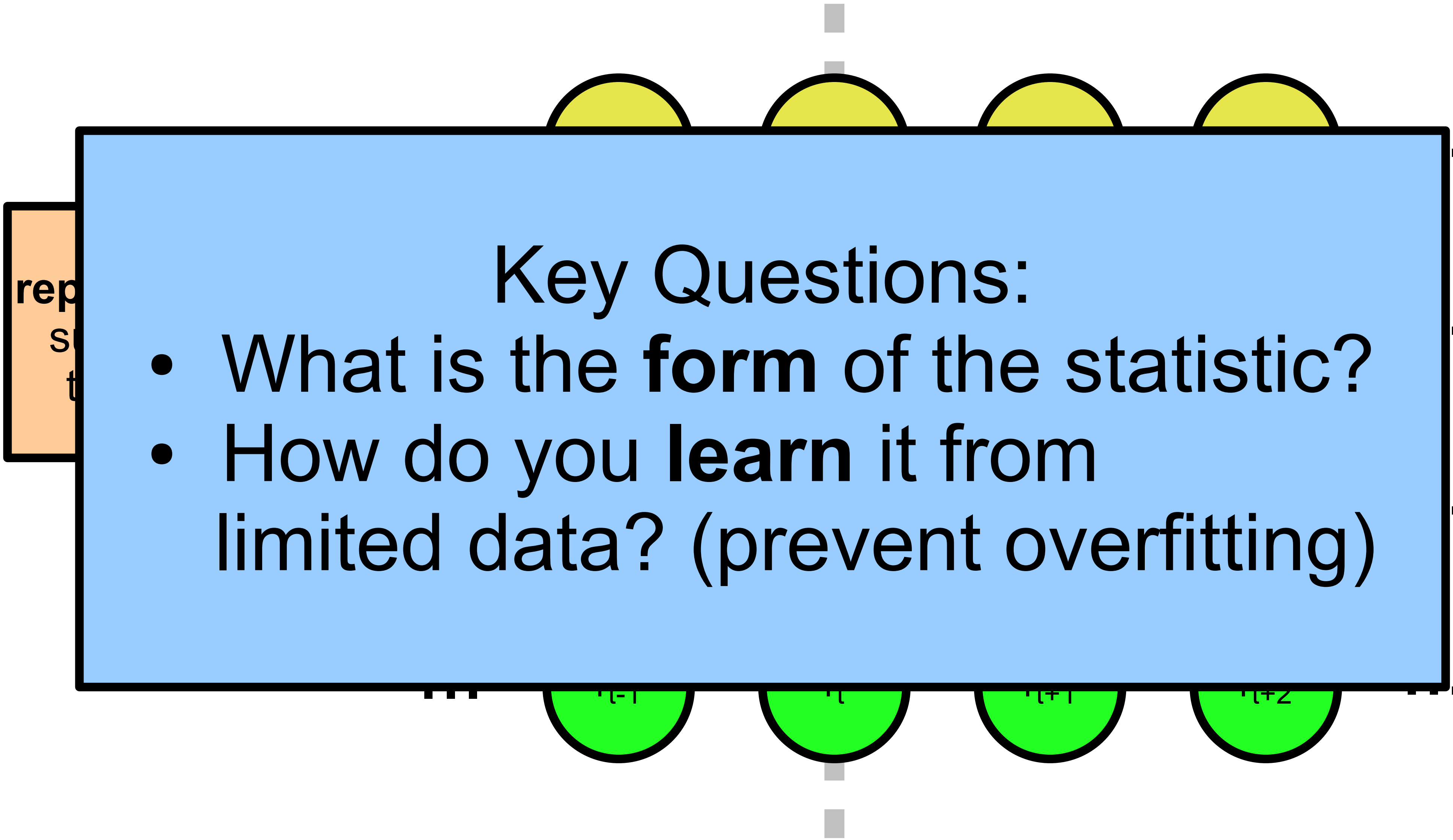
All past events are needed to predict future events



General Approach: Introduce a statistic that induces Markovianity



General Approach: Introduce a statistic that induces Markovianity



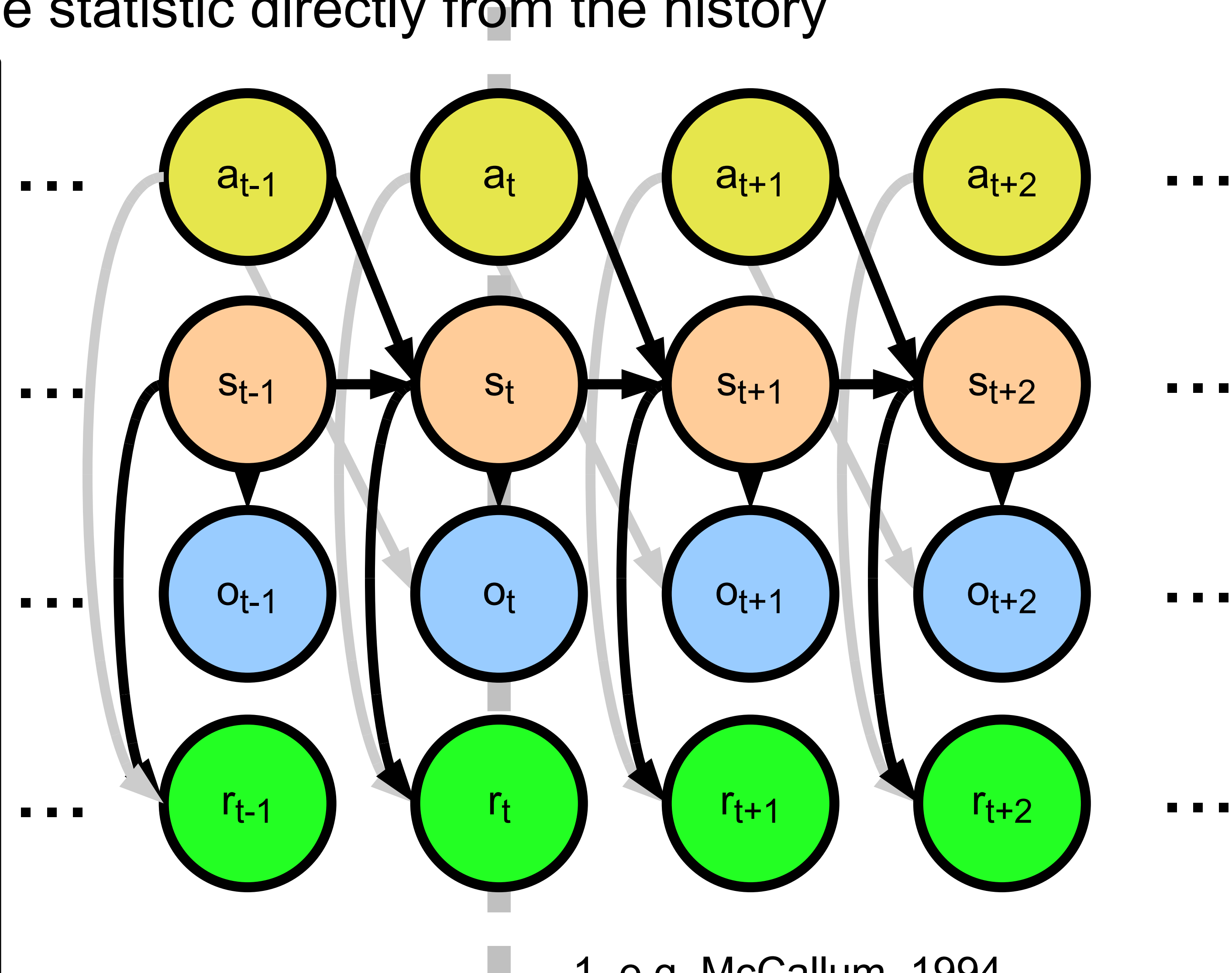
Key Questions:

- What is the **form** of the statistic?
- How do you **learn** it from limited data? (prevent overfitting)

History-Based Approaches

Idea: build the statistic directly from the history

Examples:
- U-Tree[1] (learn with statistical tests)
- Probabalistic Deterministic Finite Automata[2] (learned via validation sets)
- Predictive State Representations[3] (learned via eigenvalue decompositions)



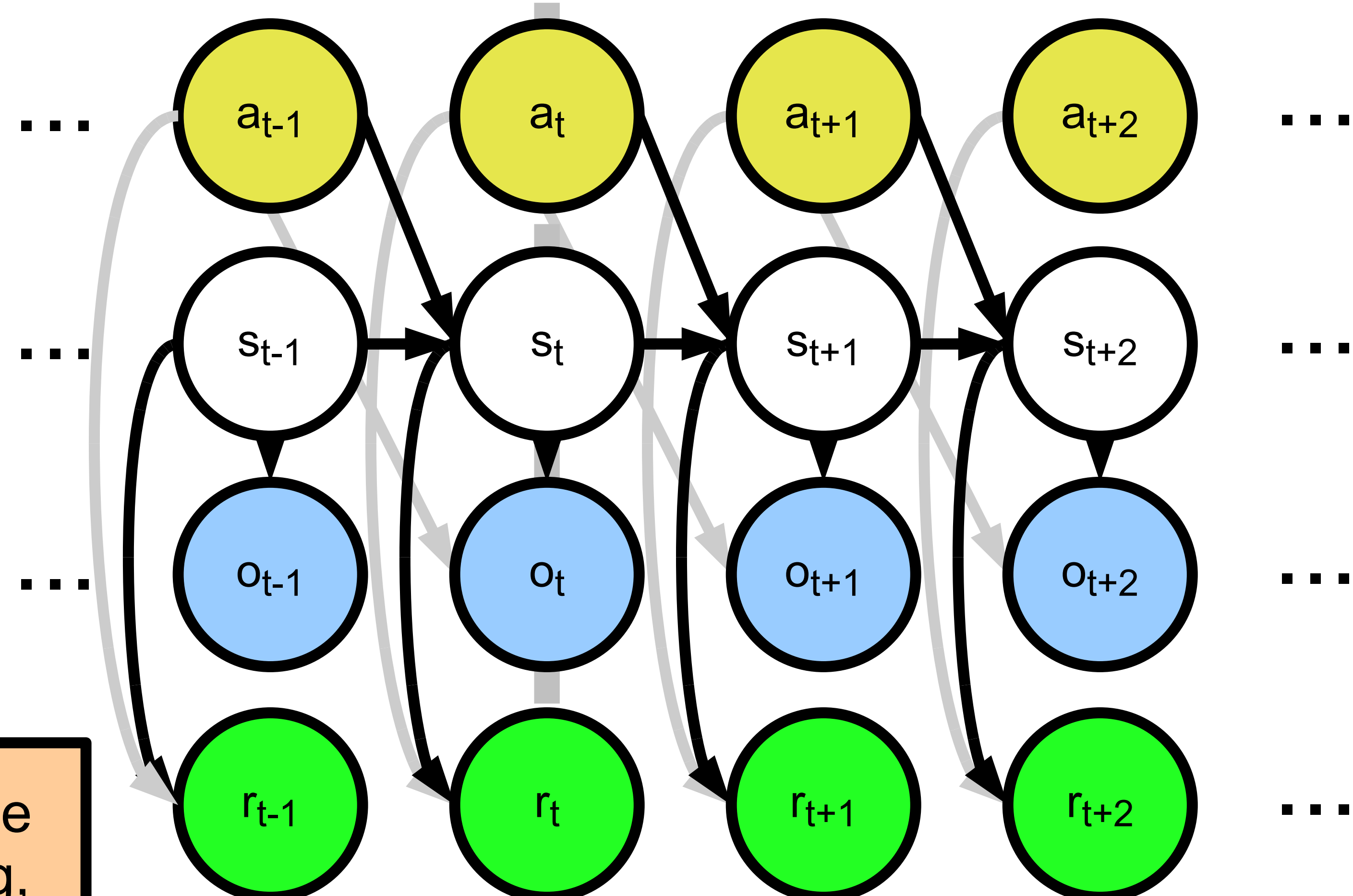
1. e.g. McCallum, 1994
2. e.g. Mahmud, 2010
3. e.g. Littman, Sutton, and Singh, 2002

Hidden-Variable Approaches

Idea: system is Markov if certain hidden variables are known

Examples:
POMDPs (and derivatives)[1] learned via
- Expectation-Maximization (validation sets)
- Bayesian methods (using Bayes rule)[2]

Our Focus: in the Bayesian setting, "belief" **p(s_t)** is a sufficient statistic



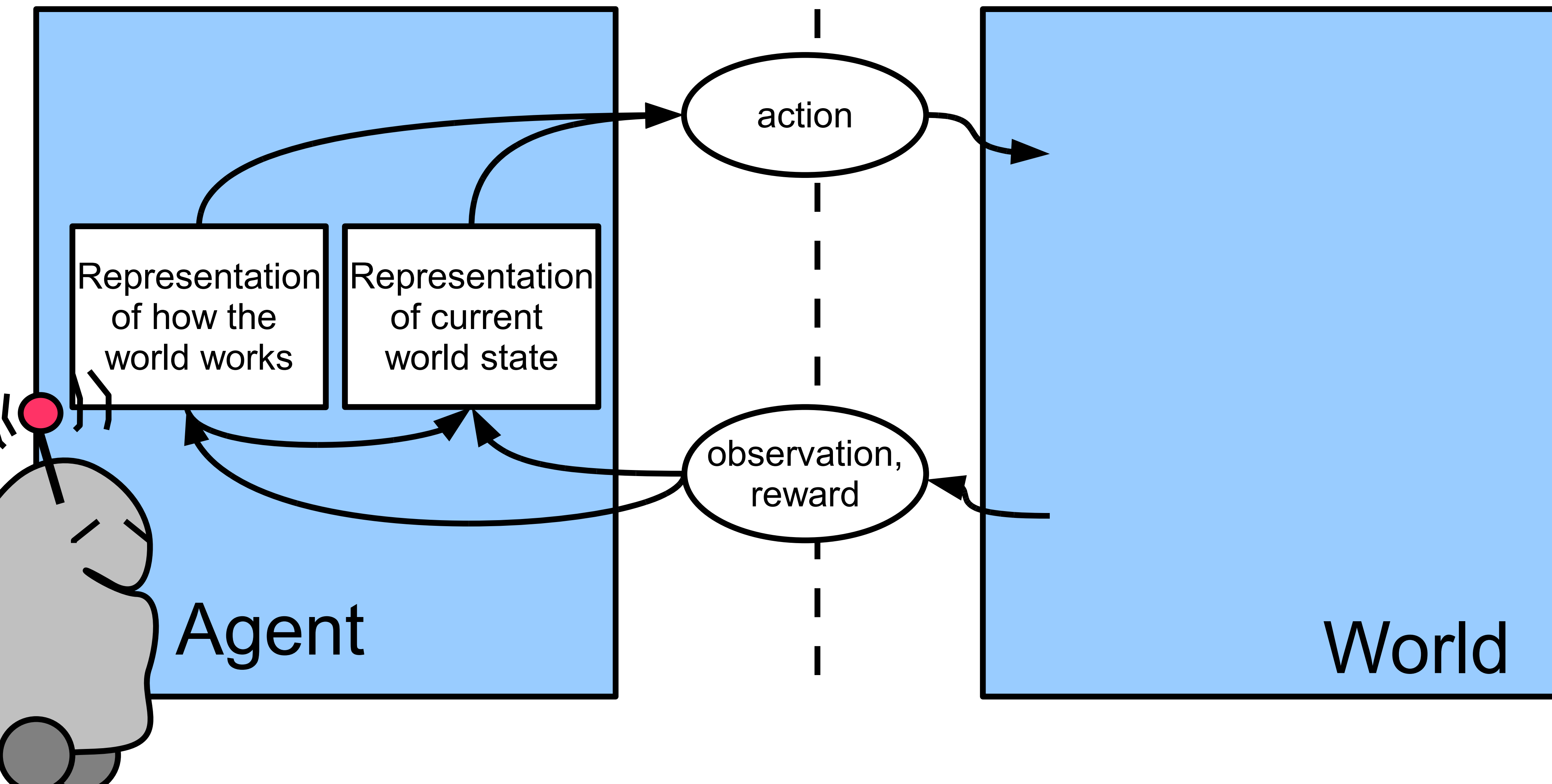
1. e.g. Sondik 1971, Kaelbling, Littman, and Cassandra 1995, McAllester and Singh 1999
2. e.g. Ross, Chaib-draa, Pineau 2007, Poupart and Vlassis, 2008

Outline

- Introduction: The partially-observable reinforcement learning setting
- Framework: Bayesian reinforcement learning
- Applying nonparametrics:
  - Infinite Partially Observable Markov Decision Processes
  - Infinite State Controllers
  - Infinite Dynamic Bayesian Networks
- Conclusions and Continuing Work

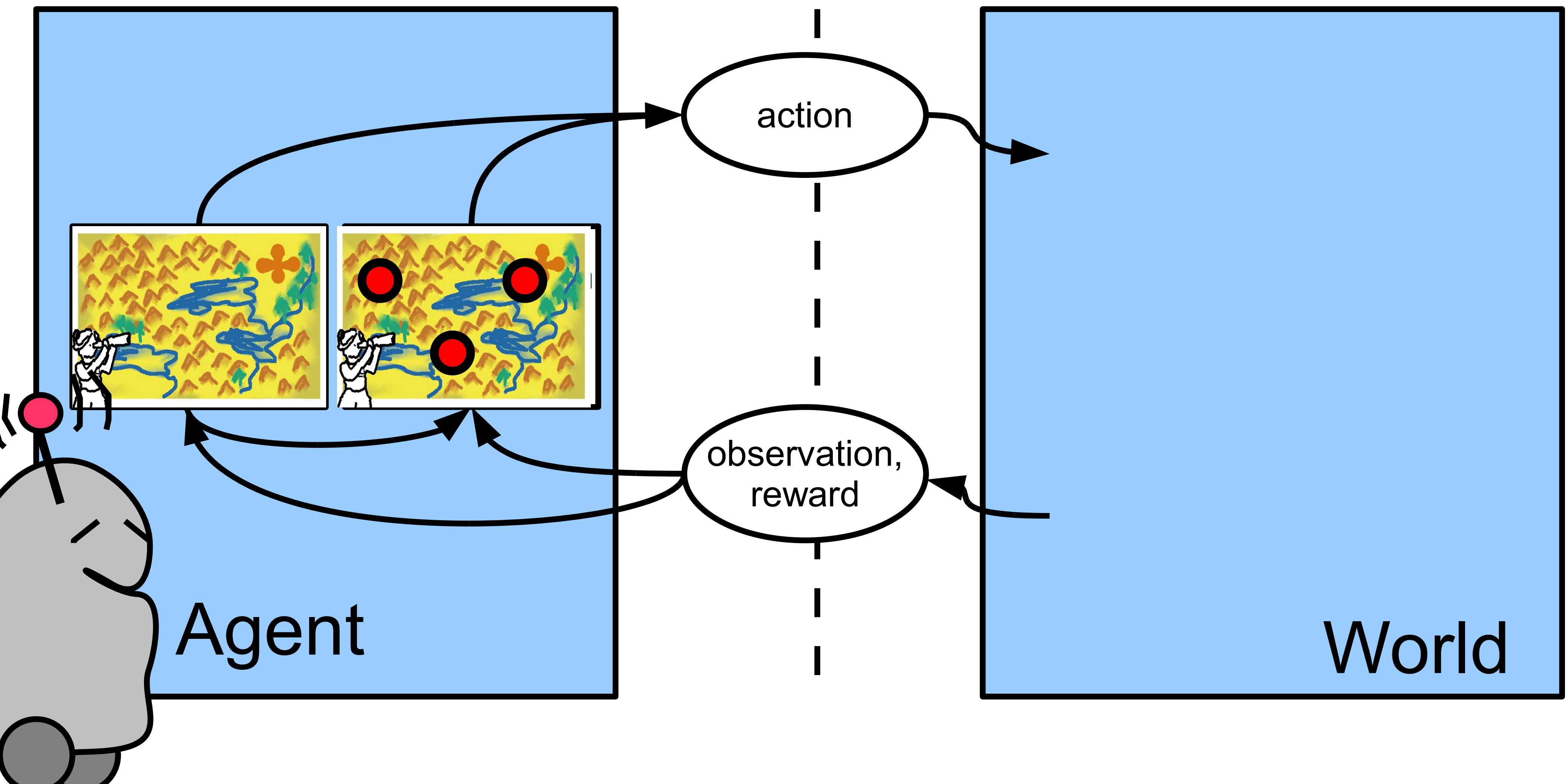
Formalizing the Problem

The agent maintains a representation of how the world works as well as the world's current state



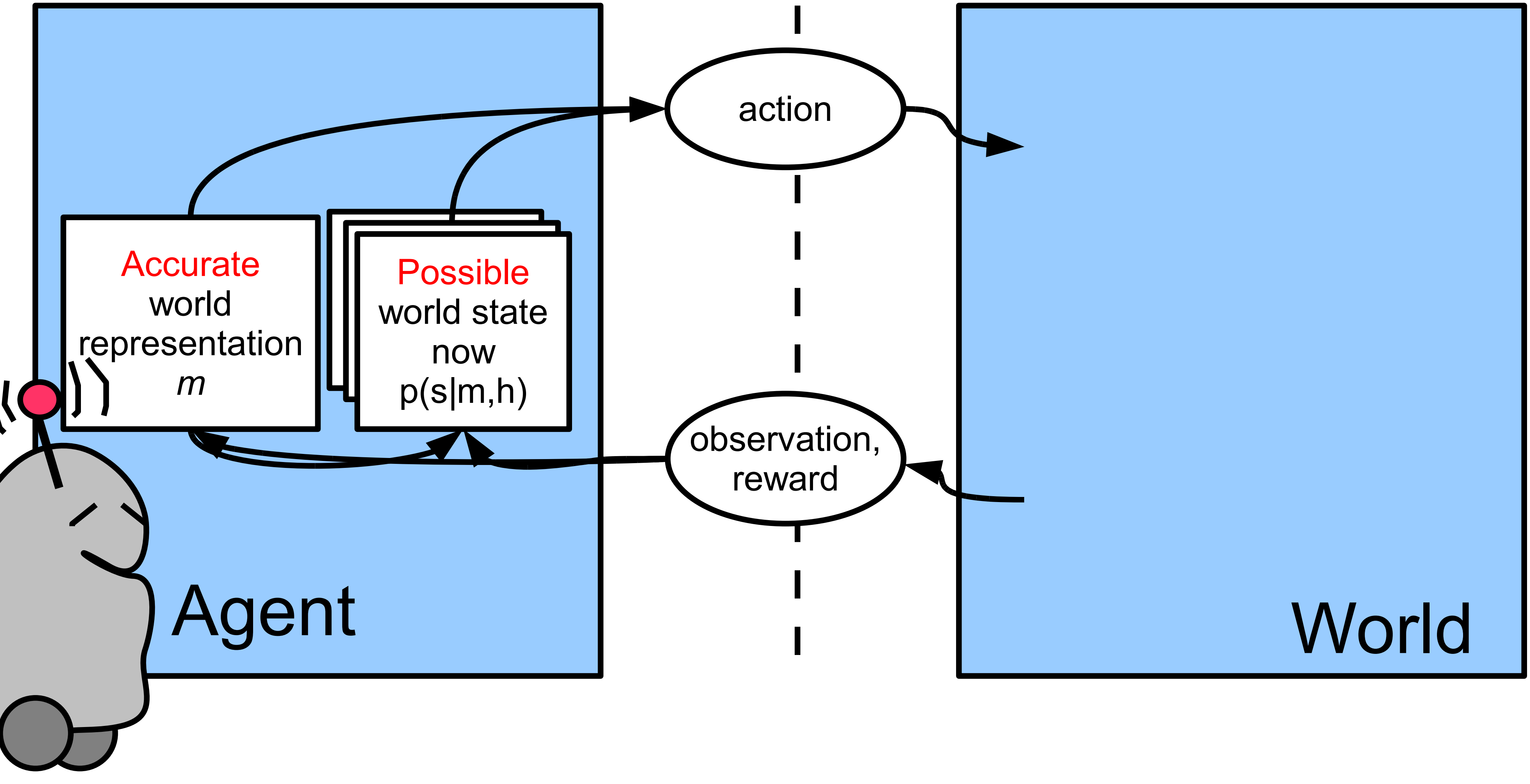
Model-Based Approach

The agent maintains a representation of how the world works as well as the world's current state



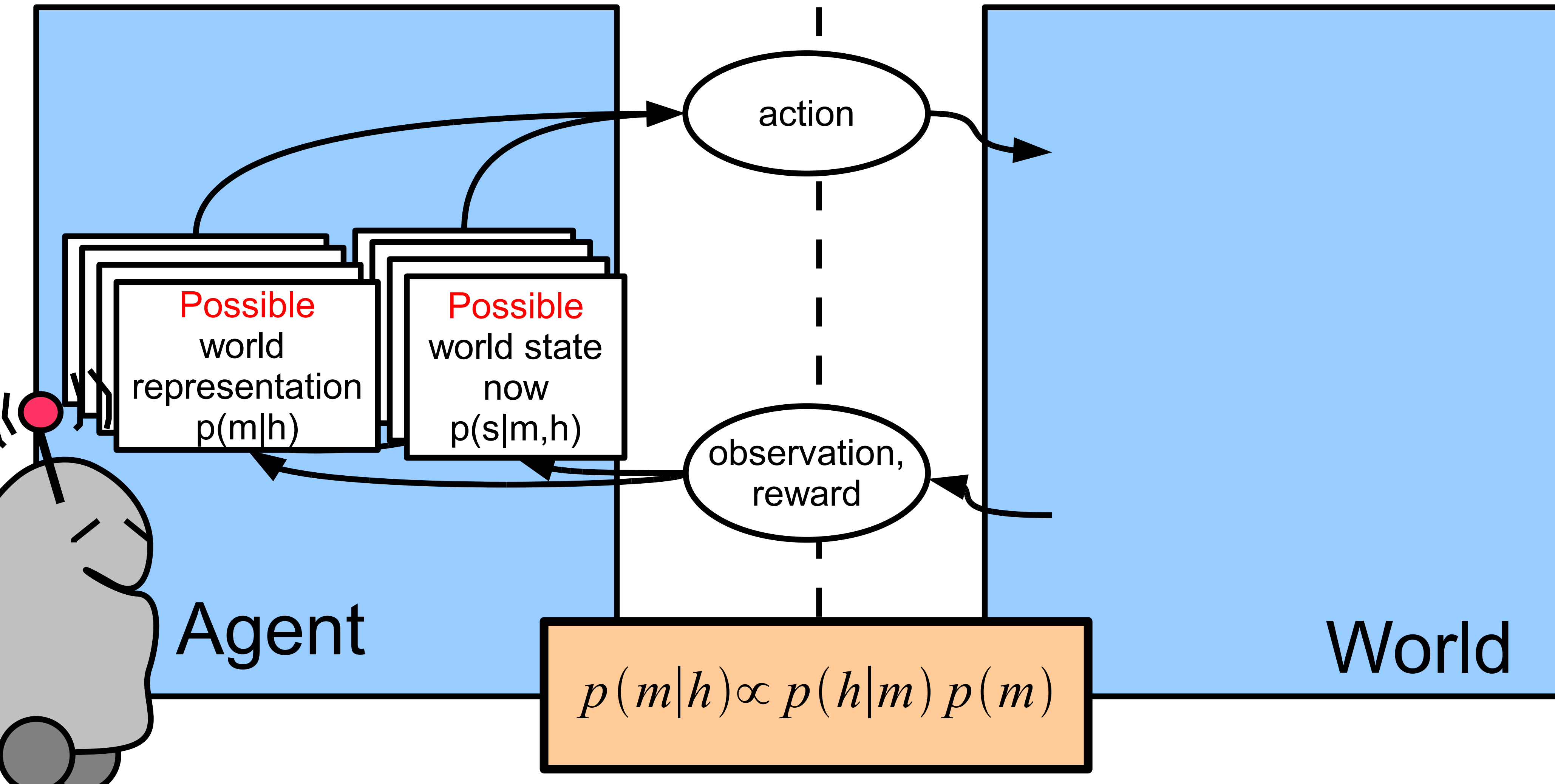
Being Bayesian

If the agent has an accurate world representation, we can keep a distribution over current states..



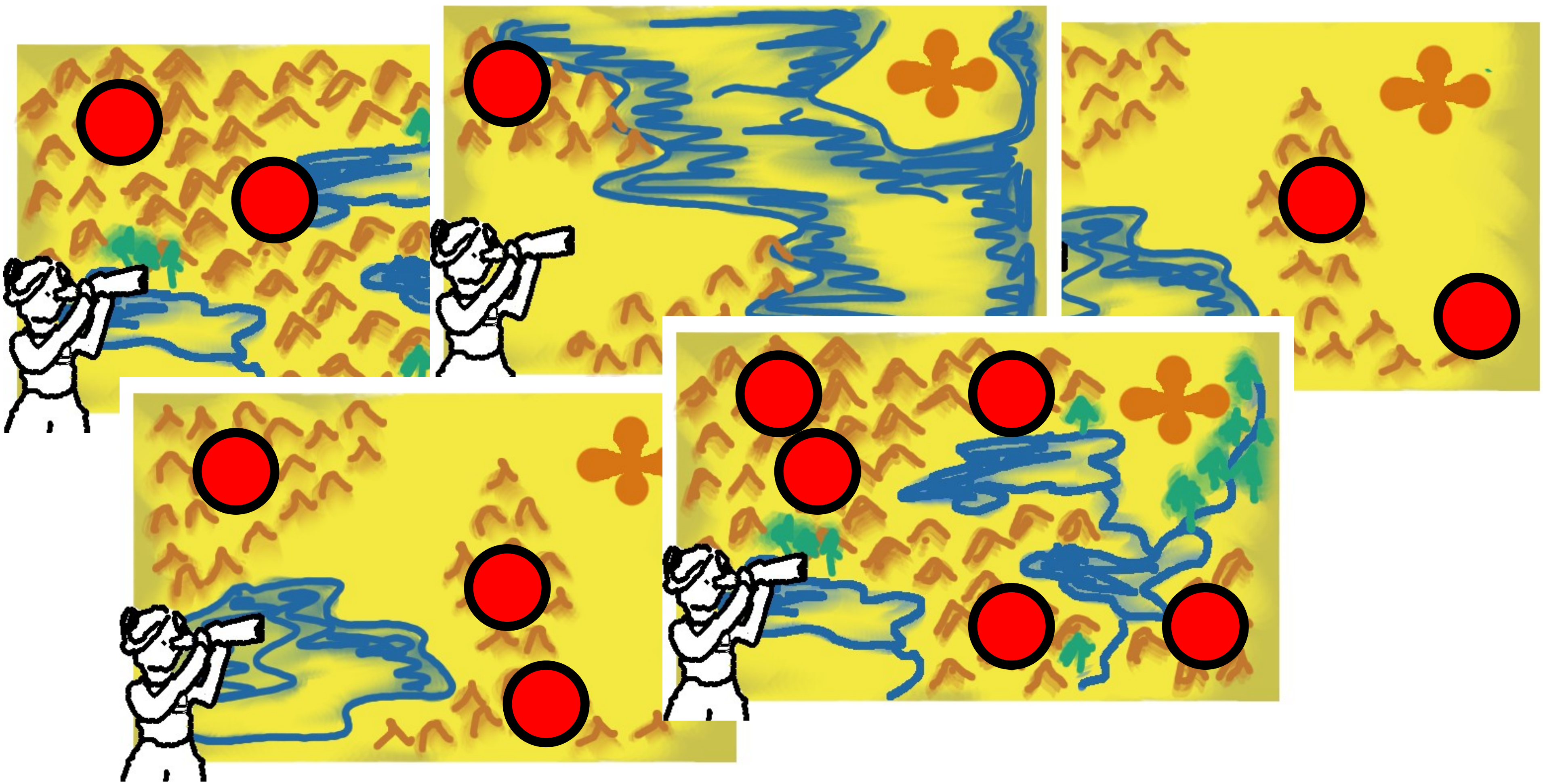
Being (more) Bayesian

If the world representation are unknown, can keep distributions over those too.



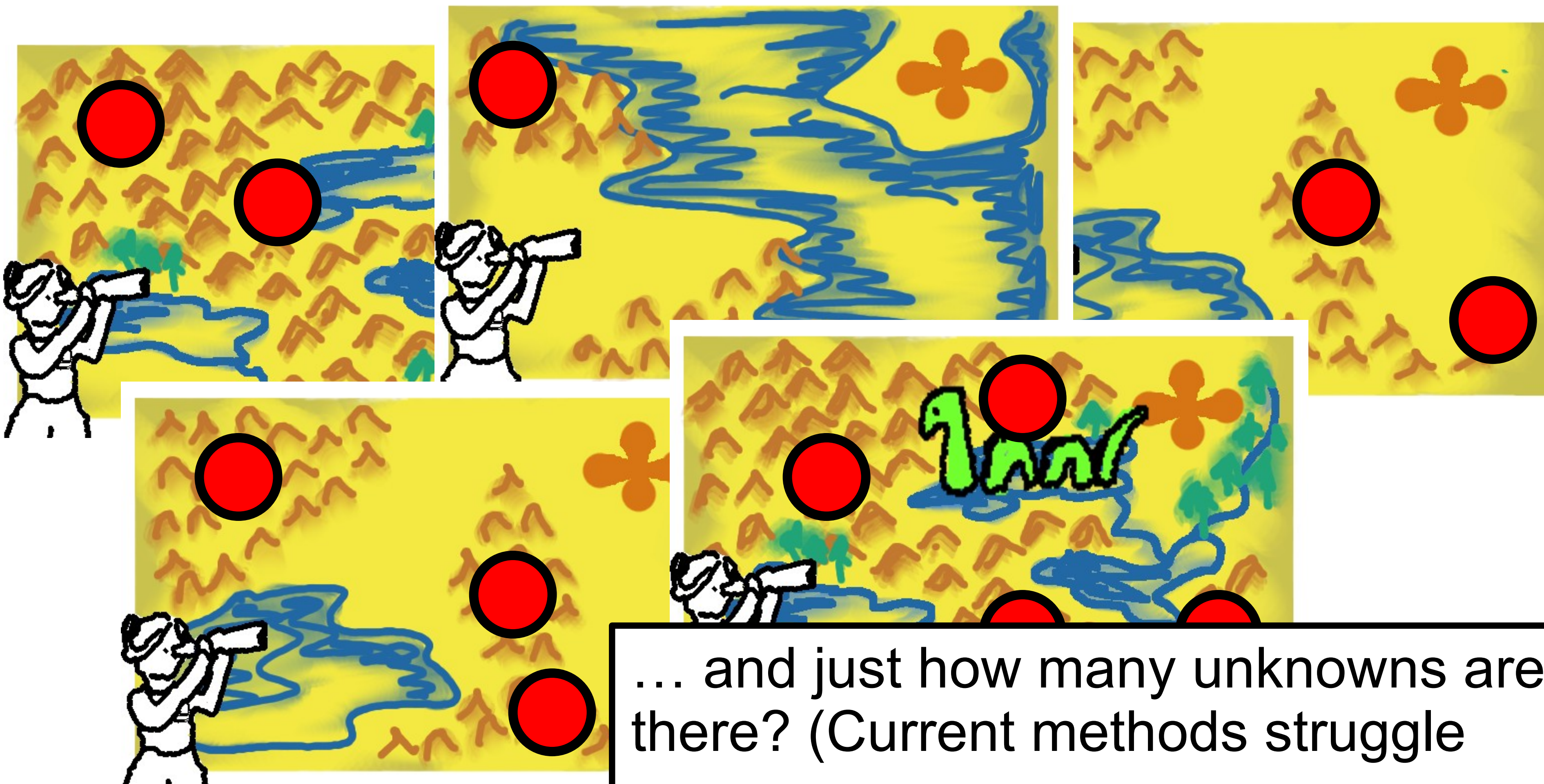
Why is this problem hard?

Lots of unknowns to reason about!



Why is this problem hard?

Lots of unknowns to reason about!



… and just how many unknowns are there? (Current methods struggle reasoning about too many unknowns.)

Why is this problem hard?

Lots of unknowns to reason about!

… and how many unknowns are there? Current methods struggle trying to reason about too many unknowns.

We'll address these challenges via

Bayesian Nonparametric Techniques

Bayesian models on an **infinite-dimensional** parameter space

We'll address these challenges via

Bayesian Nonparametric Techniques

Bayesian models on an **infinite-dimensional** parameter space

Already talked about keeping distributions p(m | h) over representations

We'll address these challenges via **Bayesian Nonparametric Techniques**

Bayesian models on an **infinite-dimensional** parameter space

Already talked about keeping distributions
p(m | h)
over representations

Now place the prior p(m) over representations with an infinite number of parameters in m

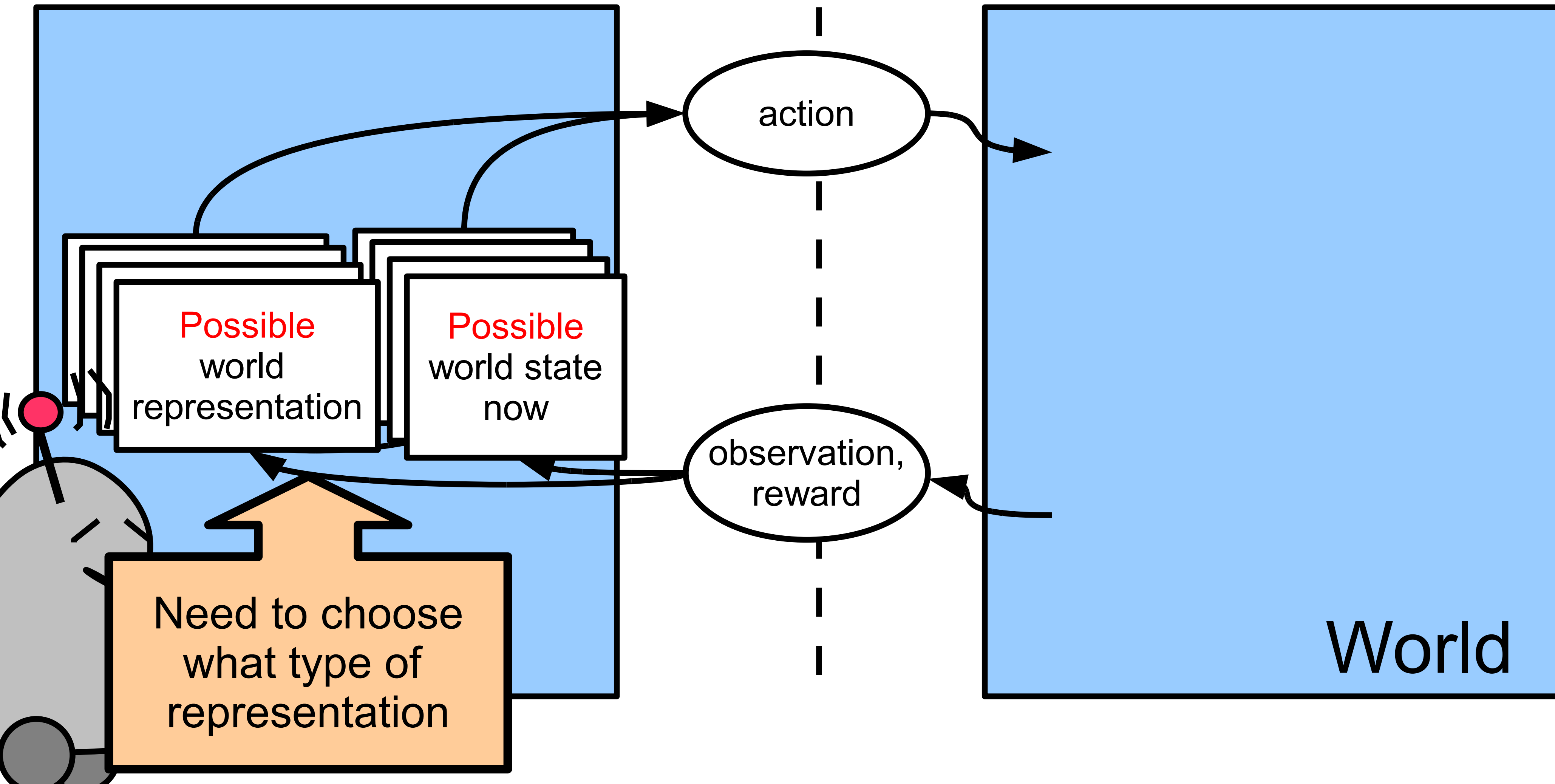
Outline

- Introduction: The partially-observable reinforcement learning setting
- Framework: Bayesian reinforcement learning
- Applying nonparametrics:
  - Infinite Partially Observable Markov Decision Processes*
  - Infinite State Controllers
  - Infinite Dynamic Bayesian Networks
- Conclusions and Continuing Work

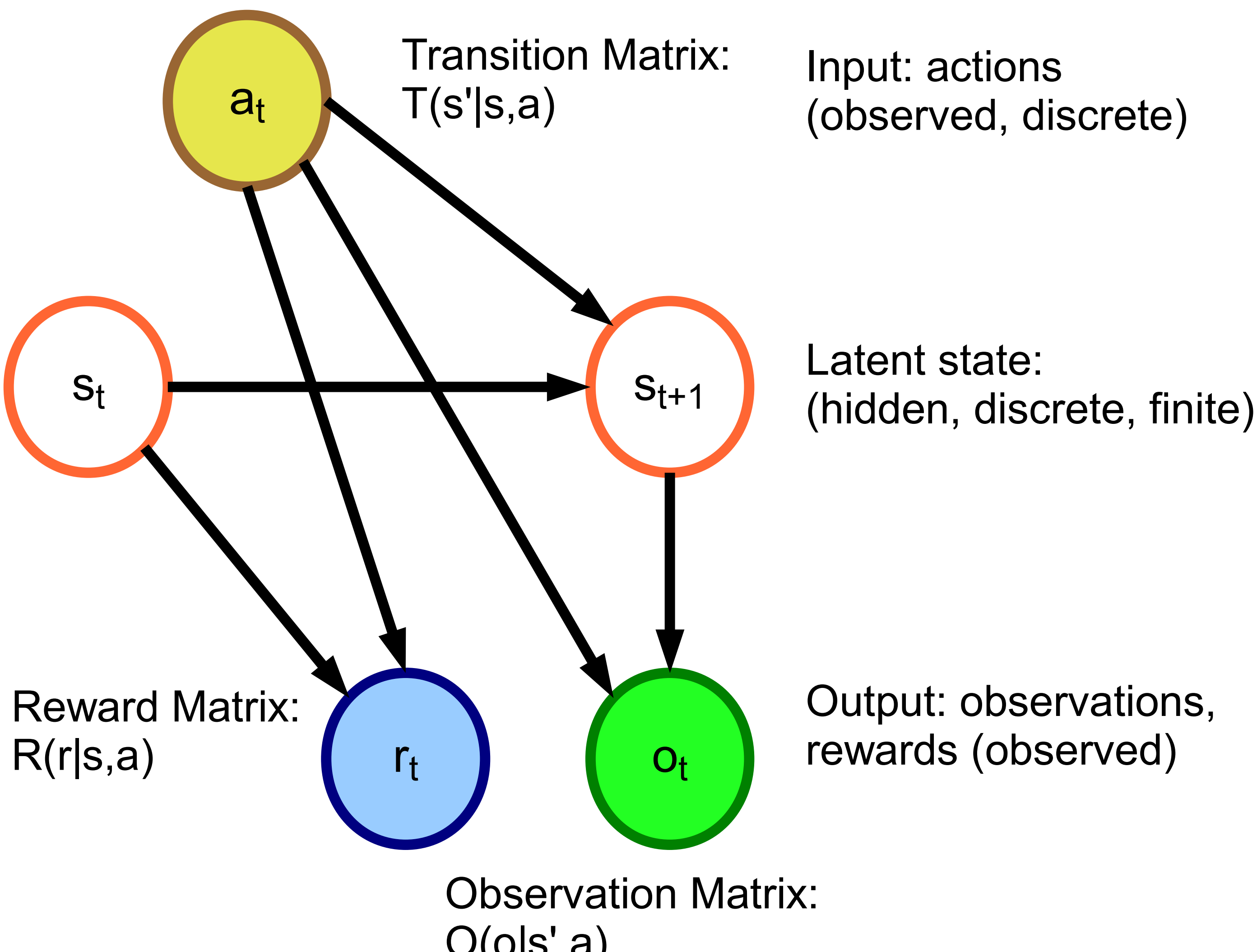
* Doshi-Velez, NIPS 2009

Being (more) Bayesian

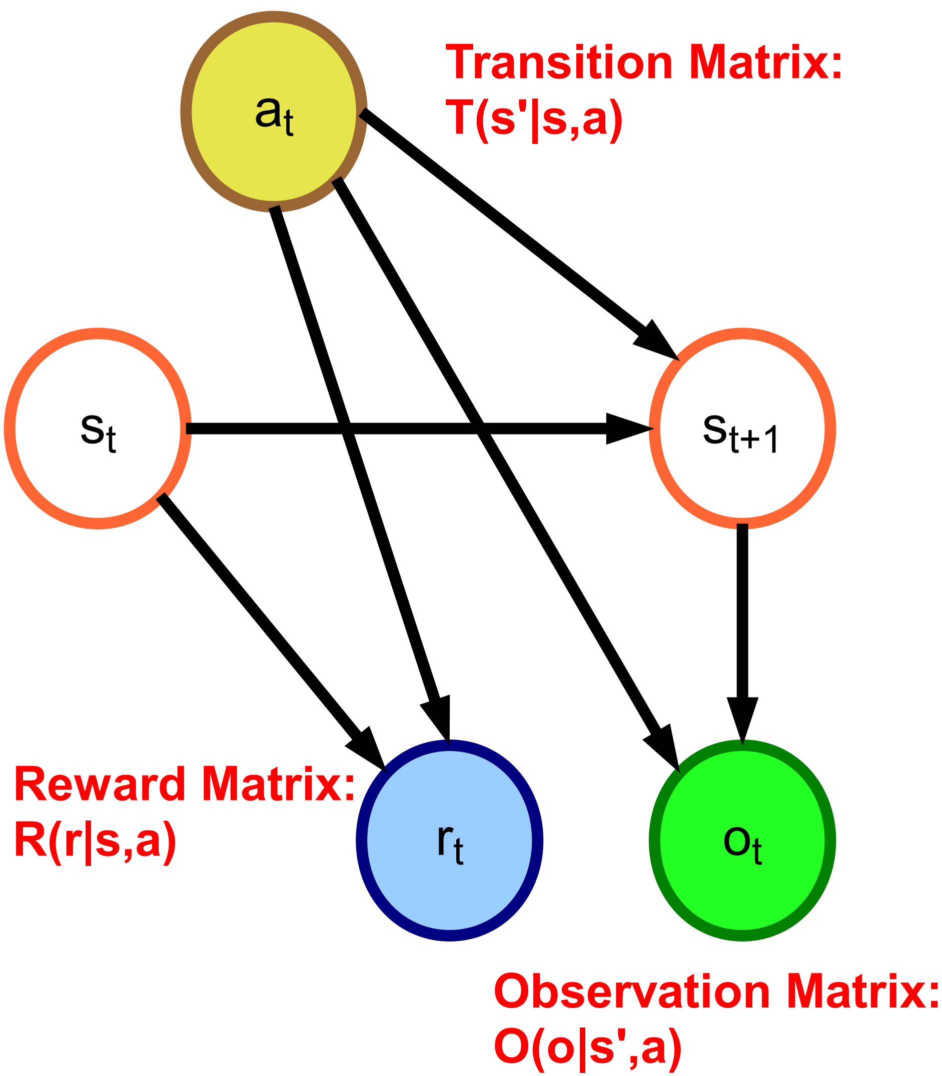
If the world representation are unknown, can keep distributions over those too.

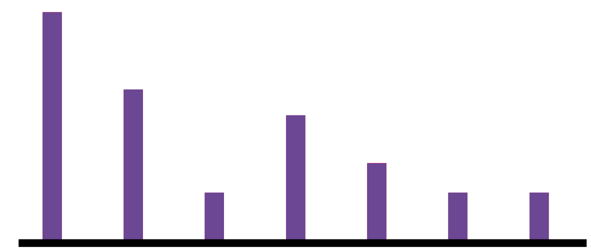


We represent the world as a partially observable Markov Decision Process (POMDP)



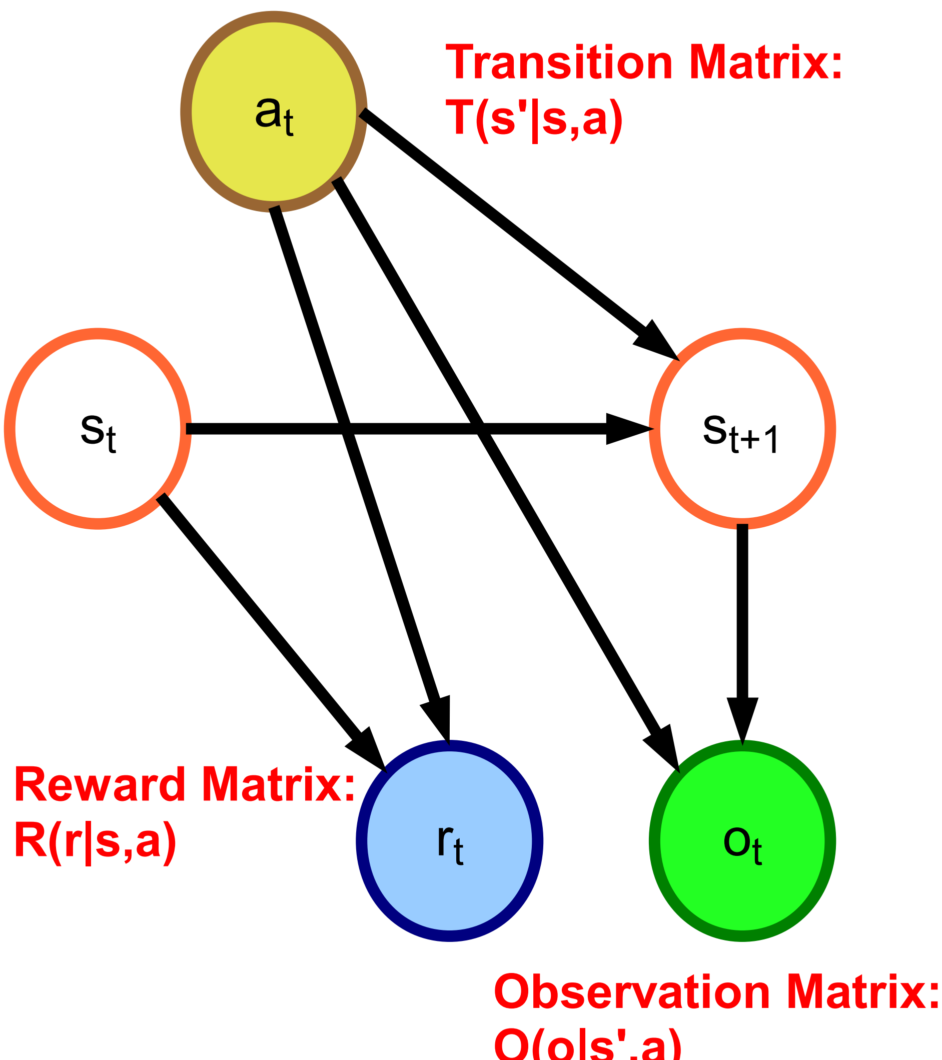
“Learning” a POMDP means learning the parameter values

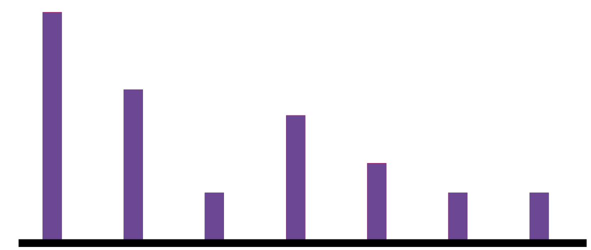




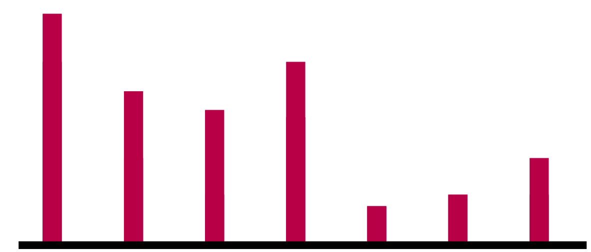
Ex.: T(◌|s,a) is a vector (*multinomial*)

Being Bayesian means putting distributions over the parameters





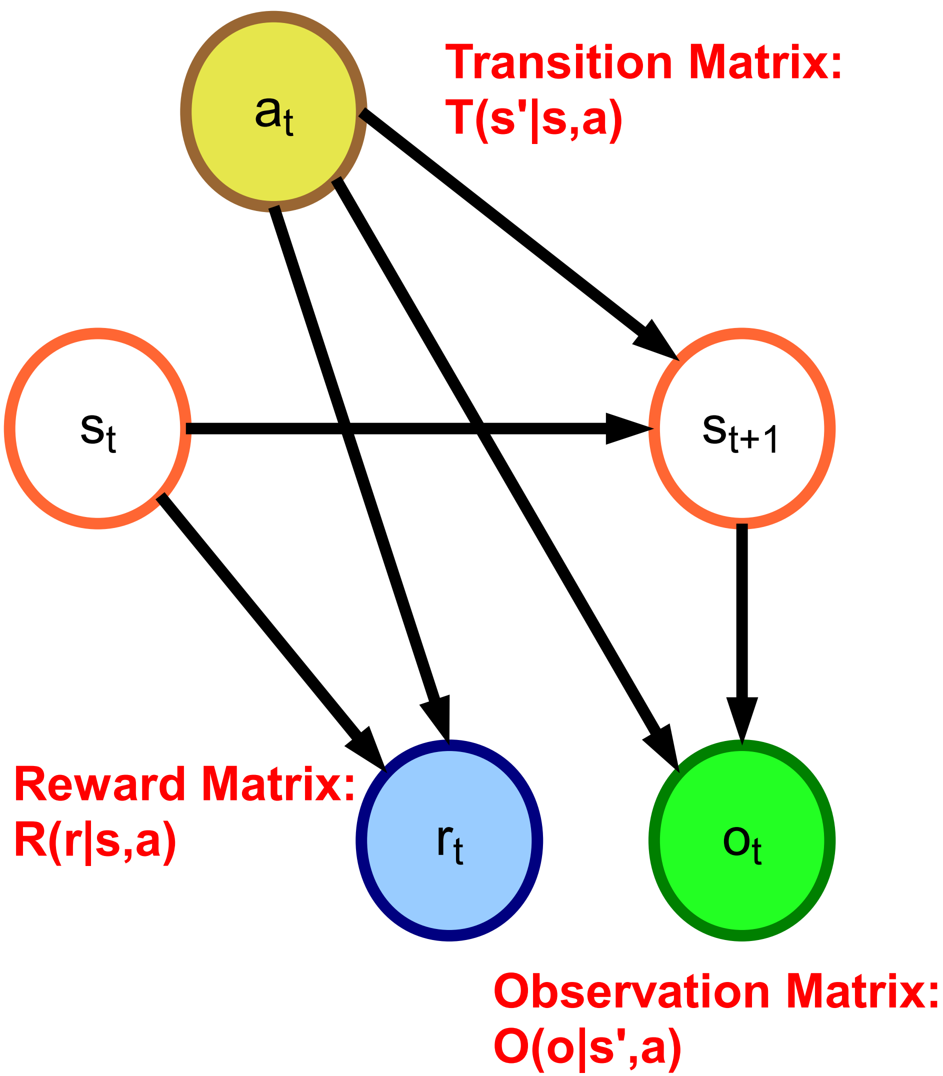
Ex.: T(◌|s,a) is a vector (*multinomial*)

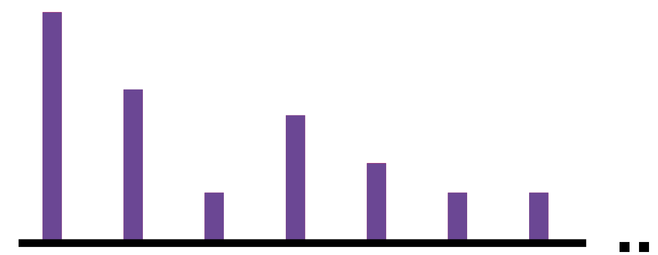


The conjugate prior p(T(◌|s,a)) is a *Dirichlet distribution*

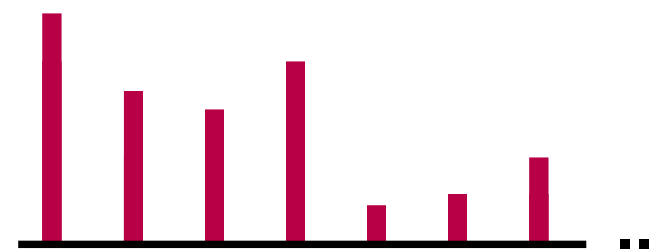
Making things nonparametric: the Infinite POMDP

(built from the HDP-HMM)

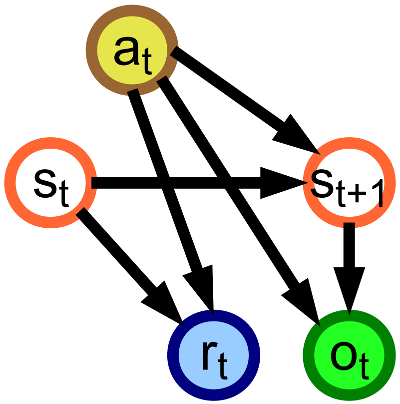




Ex.: $T(\cdot|s,a)$ is a vector (***infinite*** *multinomial*)



The conjugate prior $p(\, T(\cdot|s,a) \,)$ is a *Dirichlet* ***process***



Generative Process

(based on the HDP-HMM)

1. Sample the base transition distribution β:

   $\beta \sim \mathrm{Stick}(\ \gamma\ )$

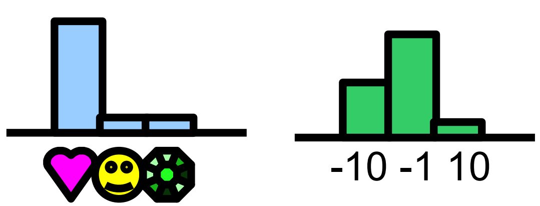
2. Sample the transition matrix in rows $T(\cdot|s,a)$:

   $T(\cdot|s,a) \sim DP(\ \beta\ ,\ \alpha\ )$

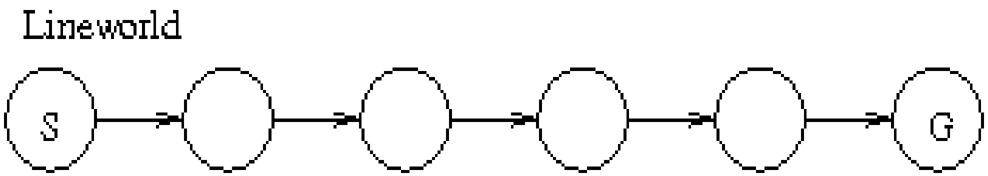
3. For each state-action pair, sample observation and reward distributions from a base distribution:

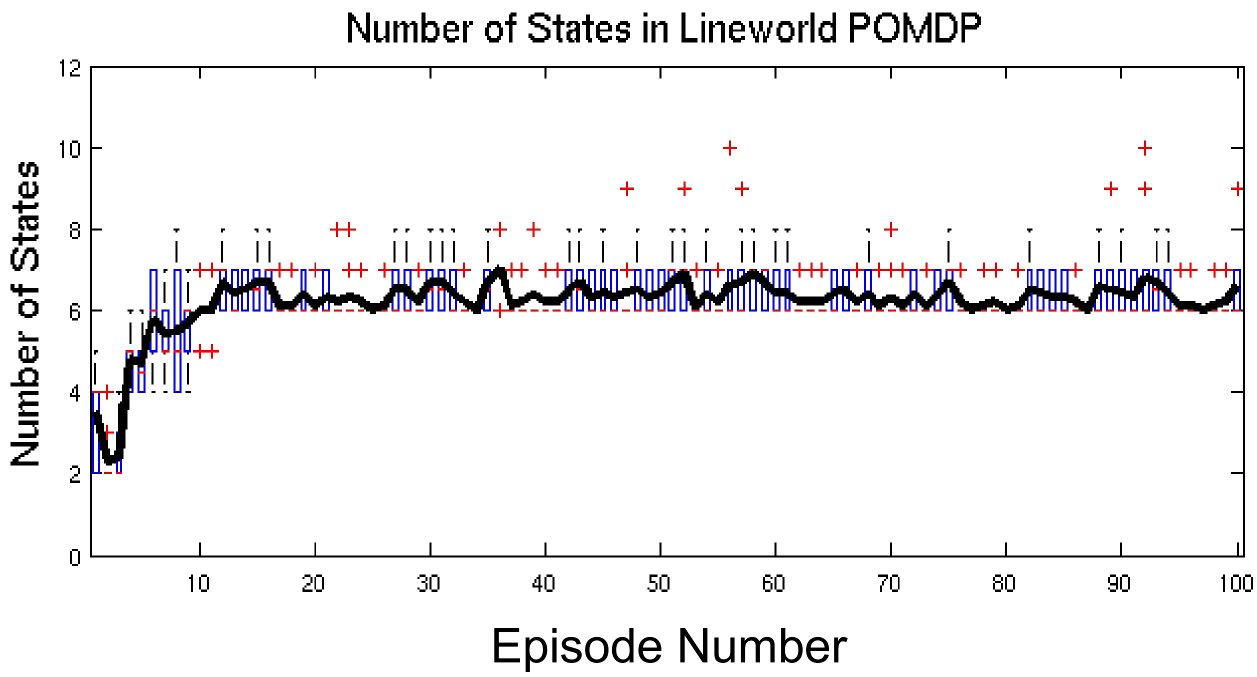
   $\Omega(o|s,o) \sim HO$

   $R(r|s,o) \sim HR$



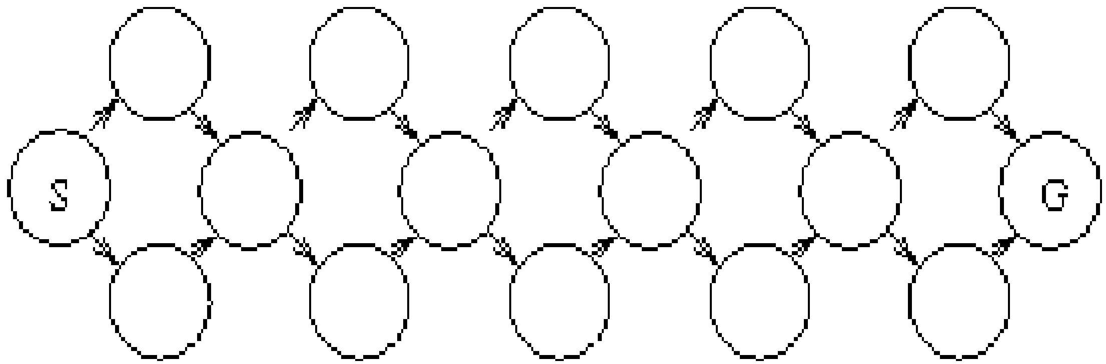
Model Complexity Grows with Data: Lineworld Example

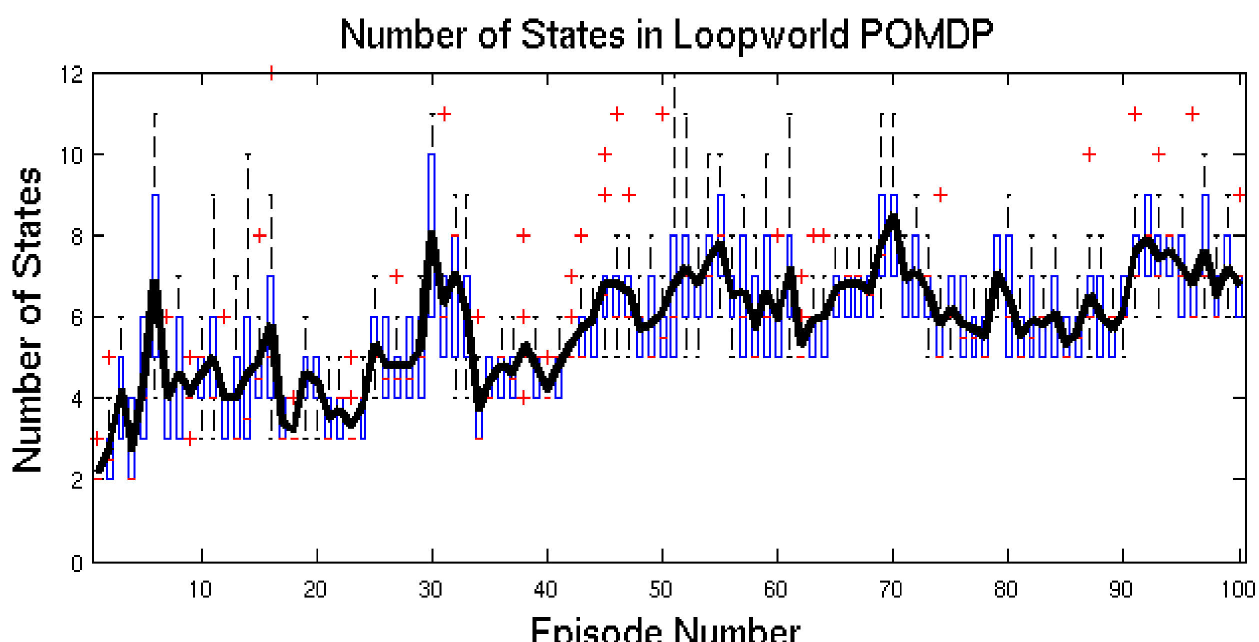




I

Model Complexity Grows with Data: Loopworld Example





Incorporating Data and Choosing Actions

All Bayesian reinforcement learning approaches alternate between two stages, belief monitoring and action selection.

Incorporating Data and Choosing Actions

All Bayesian reinforcement learning approaches alternate between two stages, belief monitoring and action selection.

- Belief monitoring: maintain the posterior

$$b(s, m|h) = b(s|m, h)\, b(m|h)$$

Issue: we need a distribution over infinite models!

Key idea: only need to reason about parameters of states we've seen.

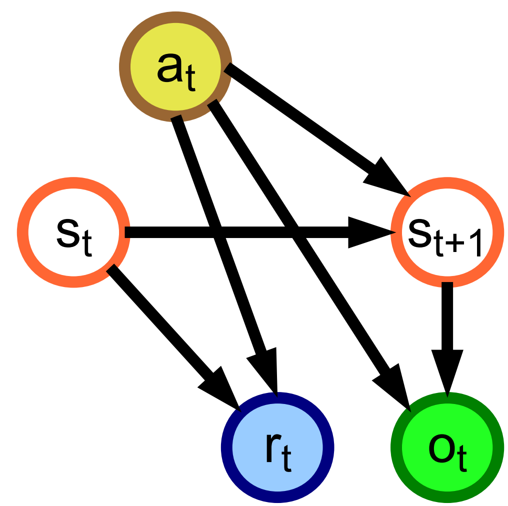
High-level Plan: Apply Bayes Rule

$$\underbrace{P(\text{model}|\text{data})} \propto \underbrace{P(\text{model}|\text{world})}\,\underbrace{P(\text{model})}$$

What's likely given the data?
Represent this complex distribution
by a set of samples from it...

How well do possible world
models match the data?

A priori, what models
do we think are likely?



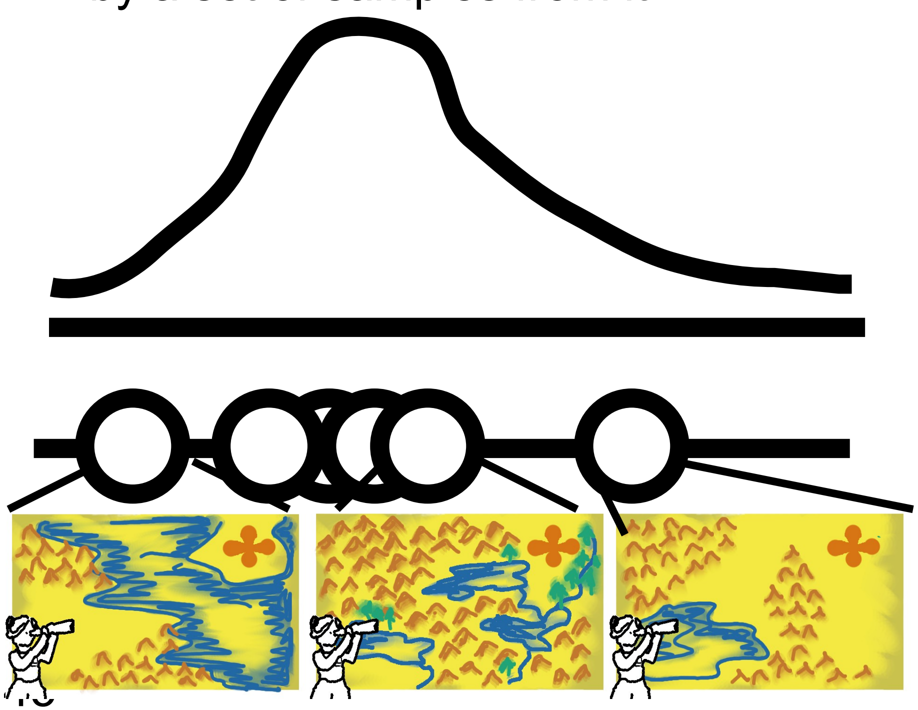
High-level Plan: Apply Bayes Rule

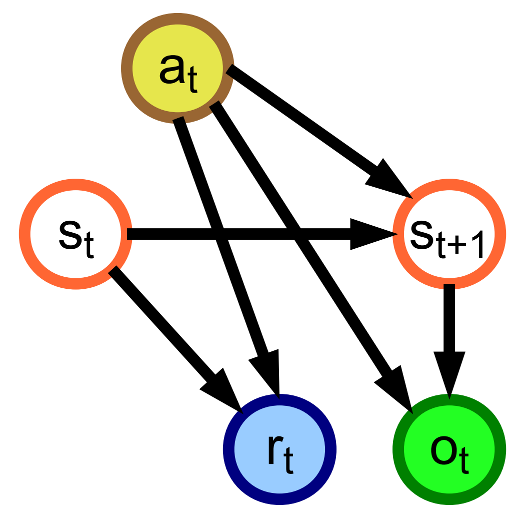
$$\underbrace{P(\text{model}|\text{data})}_{\text{What's likely given the data?}} \propto \underbrace{P(\text{model}|\text{world})}_{\text{How well do possible world models match the data?}} \underbrace{P(\text{model})}_{\text{A priori, what models do we think are likely?}}$$

What's likely given the data?
Represent this complex distribution
by a set of samples from it...

How well do possible world
models match the data?

A priori, what models
do we think are likely?





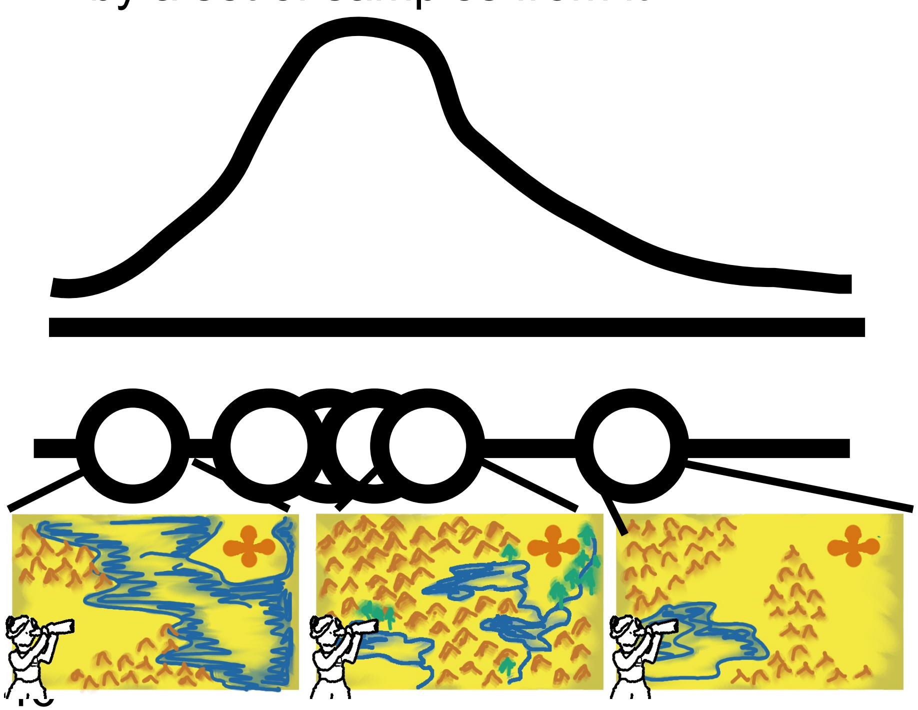
High-level Plan: Apply Bayes Rule

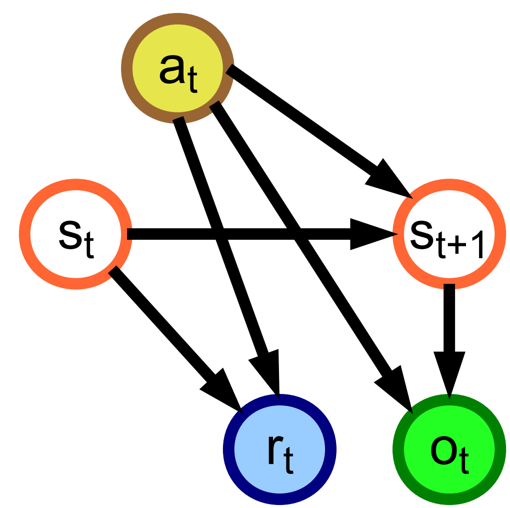
$$\underbrace{P(\text{model}|\text{data})}_{} \propto \underbrace{P(\text{model}|\text{world})}_{} \underbrace{P(\text{model})}_{}$$

What's likely given the data?
Represent this complex distribution
by a set of samples from it...

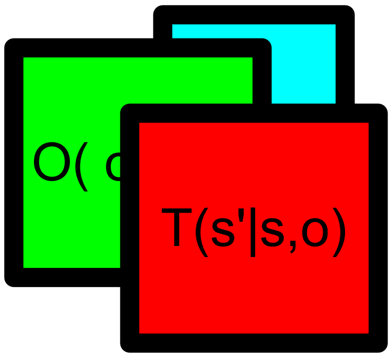
How well do possible world
models match the data?

A priori, what models
do we think are likely?

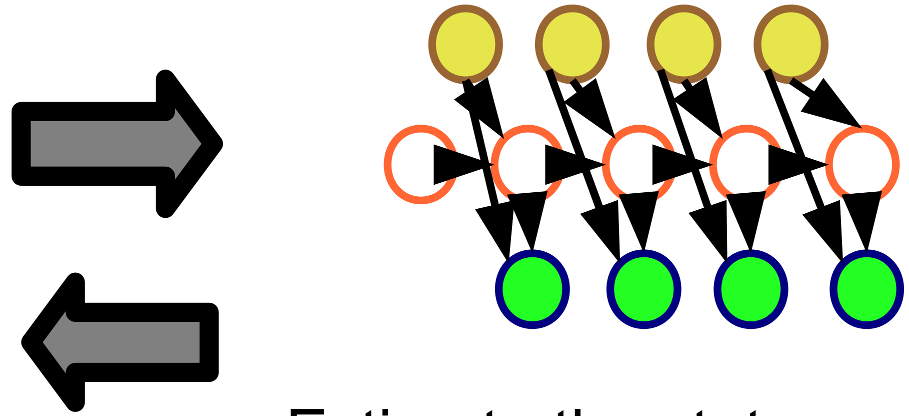




Inference: Beliefs over Finite Models

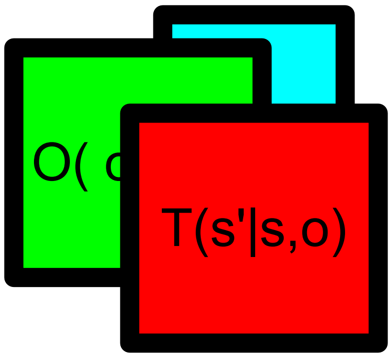


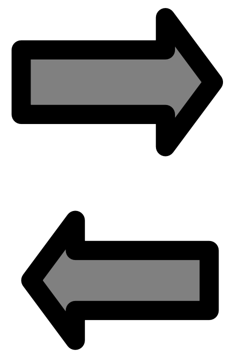
Estimate the parameters

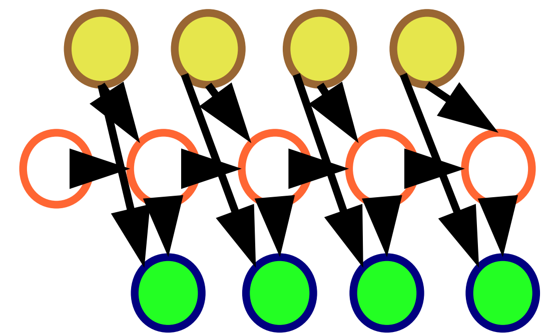


Estimate the state sequence:

Inference: Beliefs over Finite Models





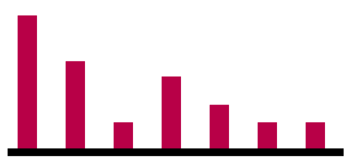


Estimate the parameters:

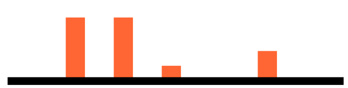
Estimate the state sequence:

Discrete case, use Dirichlet-multinomial conjugacy:

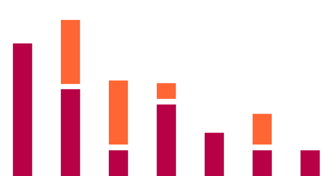
Transition Prior: β



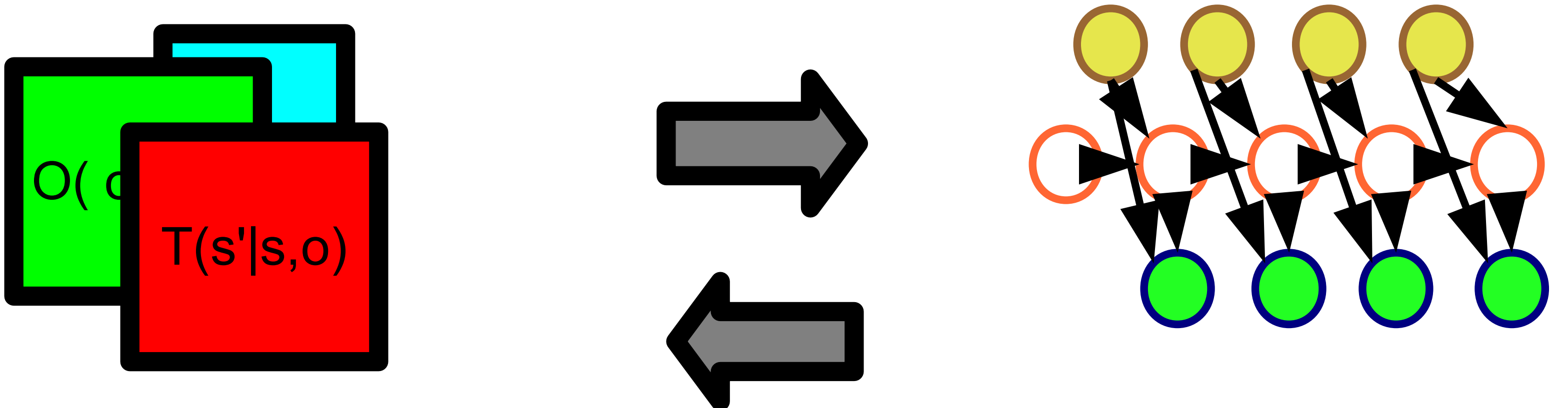
State-visit counts:



Posterior:



Inference: Beliefs over Finite Models



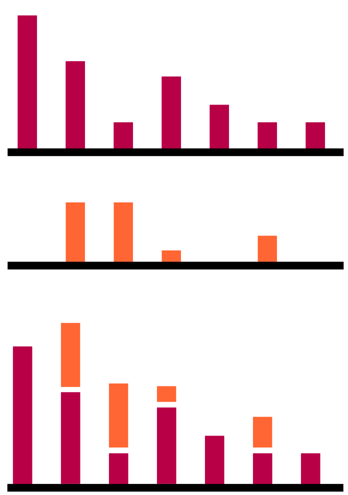
Estimate the parameters:

Discrete case, use Dirichlet-multinomial conjugacy:

Transition Prior: β

State-visit counts:

Posterior:

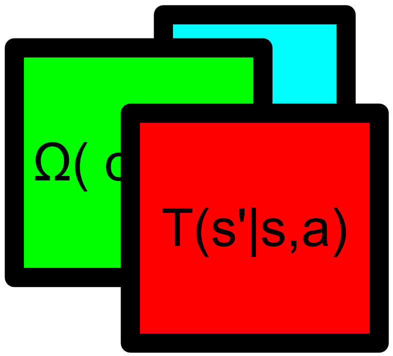


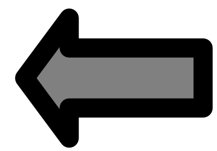
Estimate the state sequence:

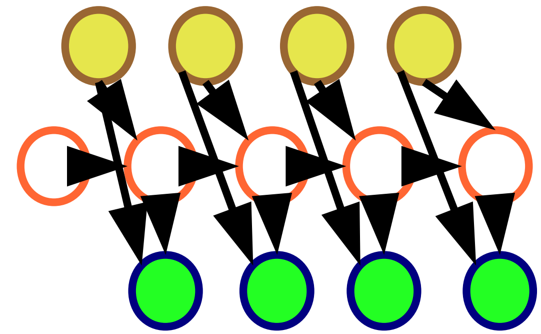
Forward filter (e.g. first part of Viterbi algorithm) to get marginal for the last state; backwards sample to get a state sequence.

Inference: Beliefs over Infinite Models

(Beam Sampling, Van Gael 2008)

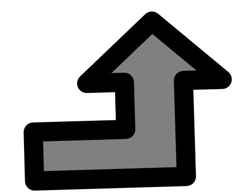


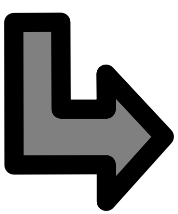




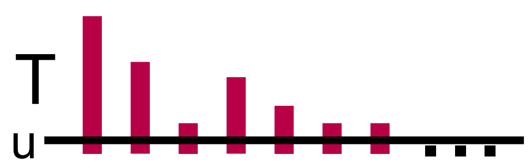
Estimate T,Ω,R for visited states

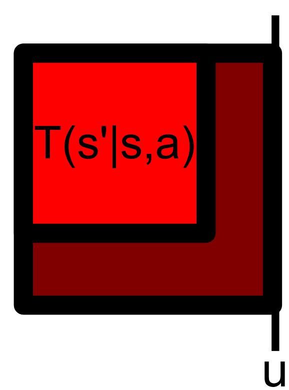
Estimate the state sequence





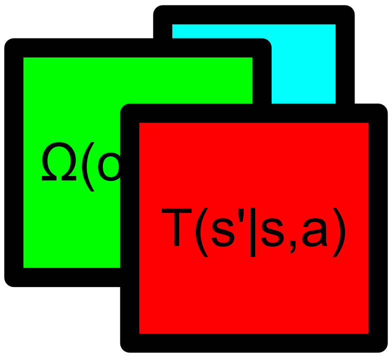
Pick a slice variable u to cut infinite model into a finite model.

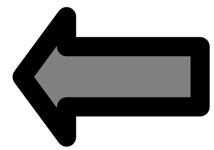


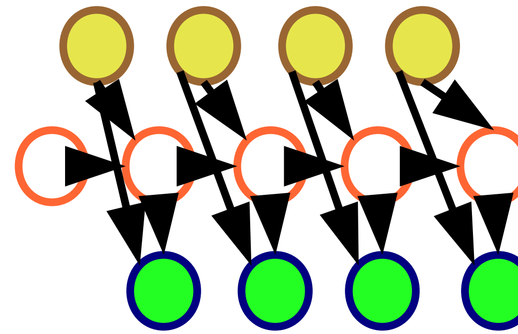


Inference: Beliefs over Infinite Models

(Beam Sampling, Van Gael 2008)

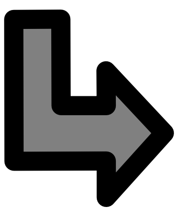




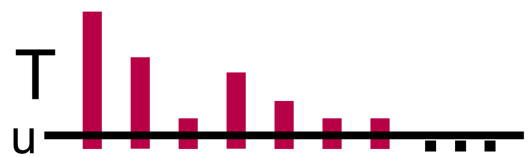


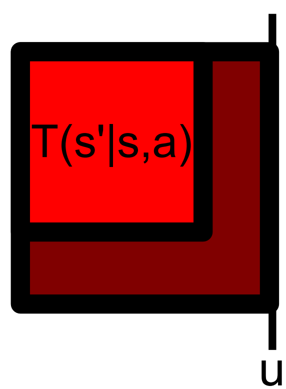
Estimate T,Ω,R for visited states

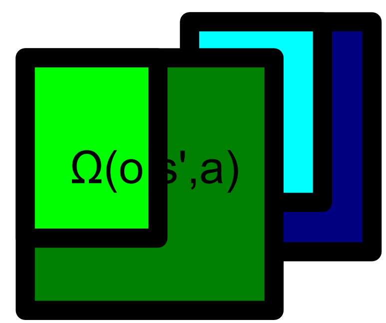
Estimate the state sequence

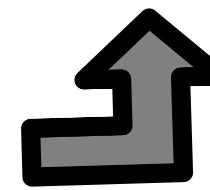


Pick a slice variable u to cut infinite model into a finite model.









Incorporating Data and Choosing Actions

All Bayesian reinforcement learning approaches alternate between two stages, belief monitoring and action selection.

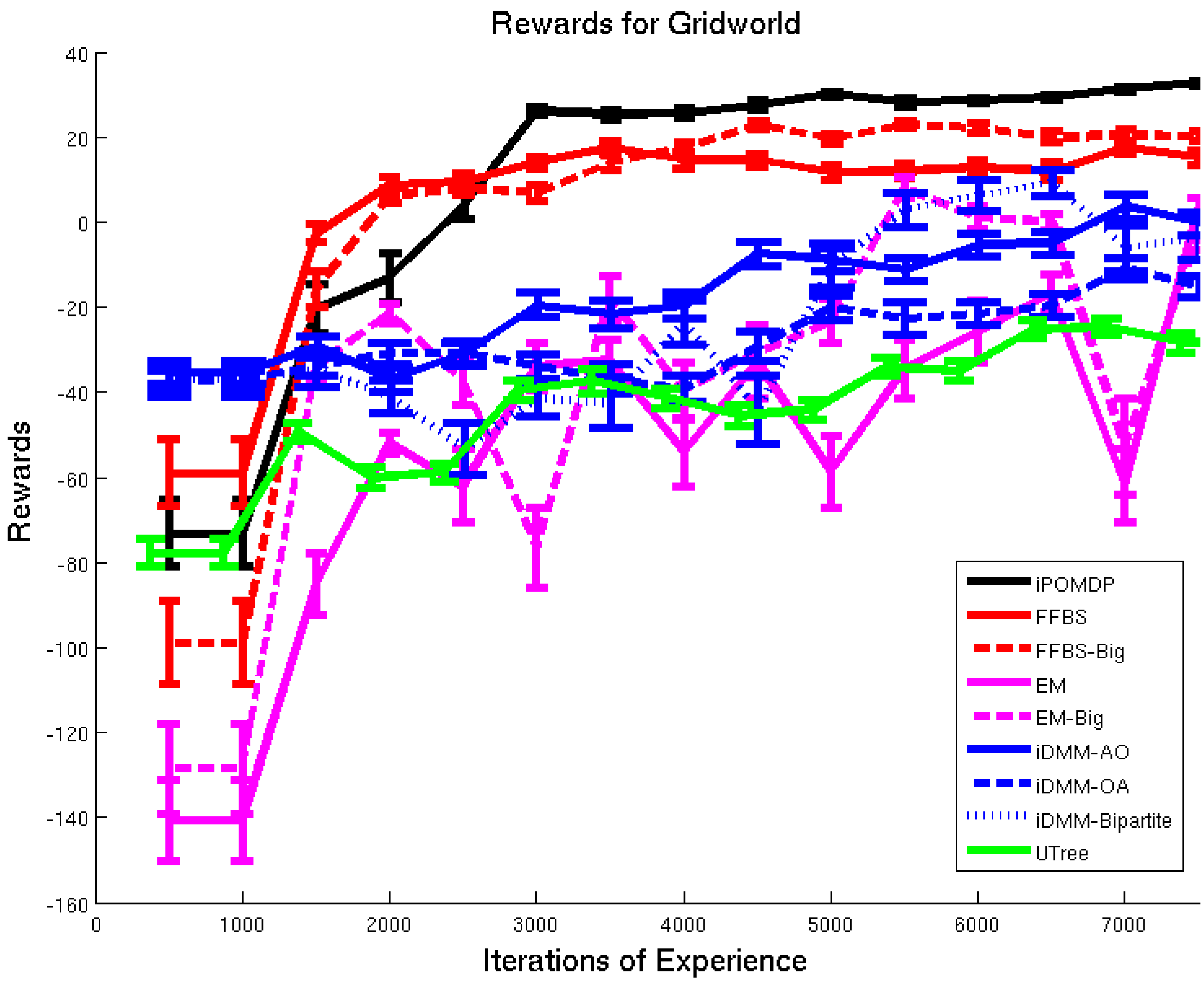
- Belief monitoring: maintain the posterior

$$b(s, m|h) = b(s|m, h)\, b(m|h)$$

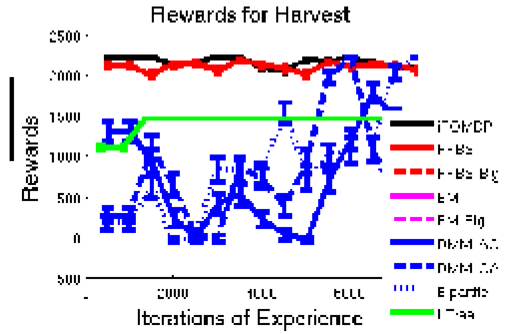
Issue: we need a distribution over infinite models!
Key idea: only need to reason about parameters of states we've seen.

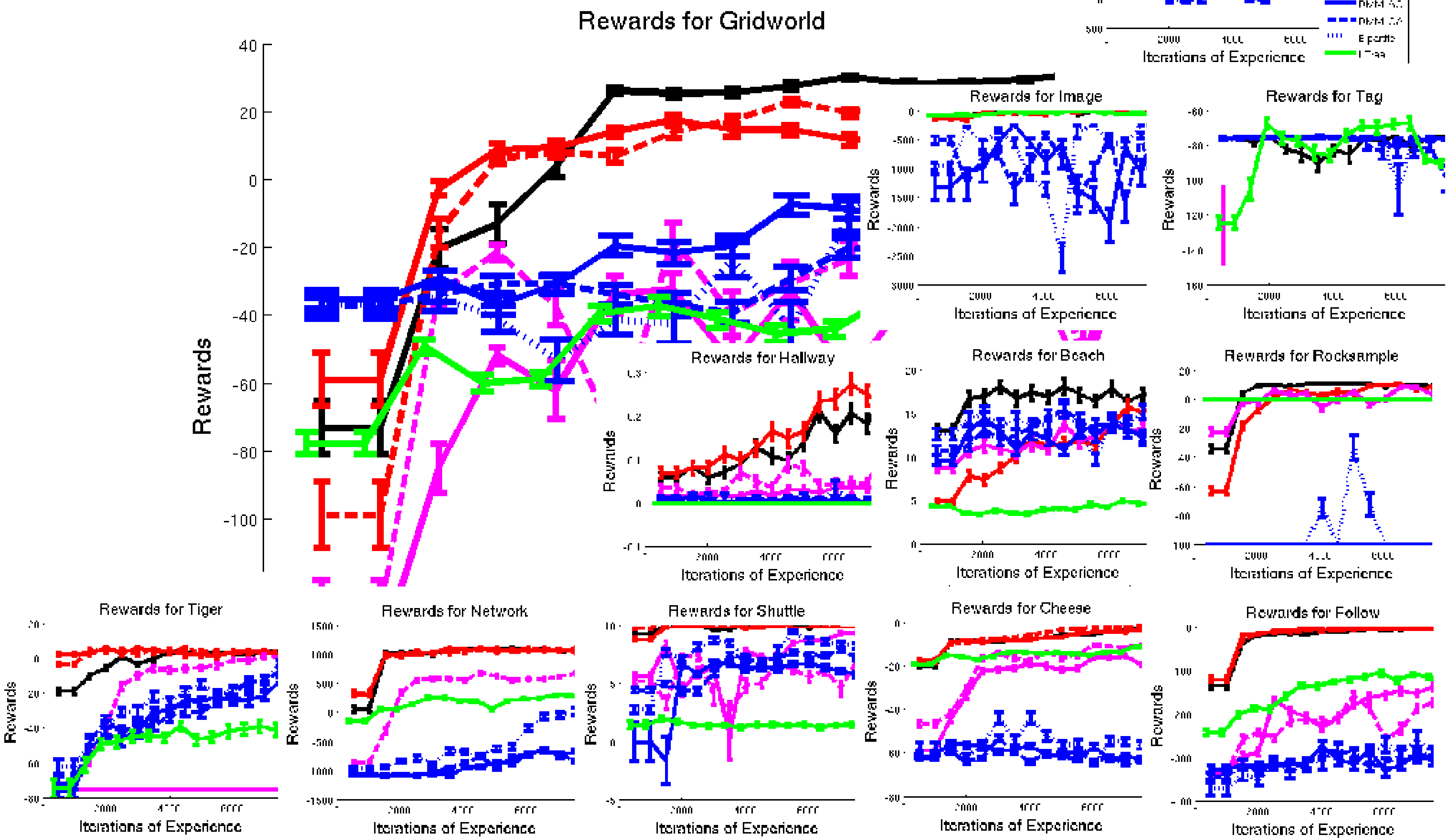
- Action selection: use a basic stochastic forward search (we'll get back to this...)

Results on Standard Problems

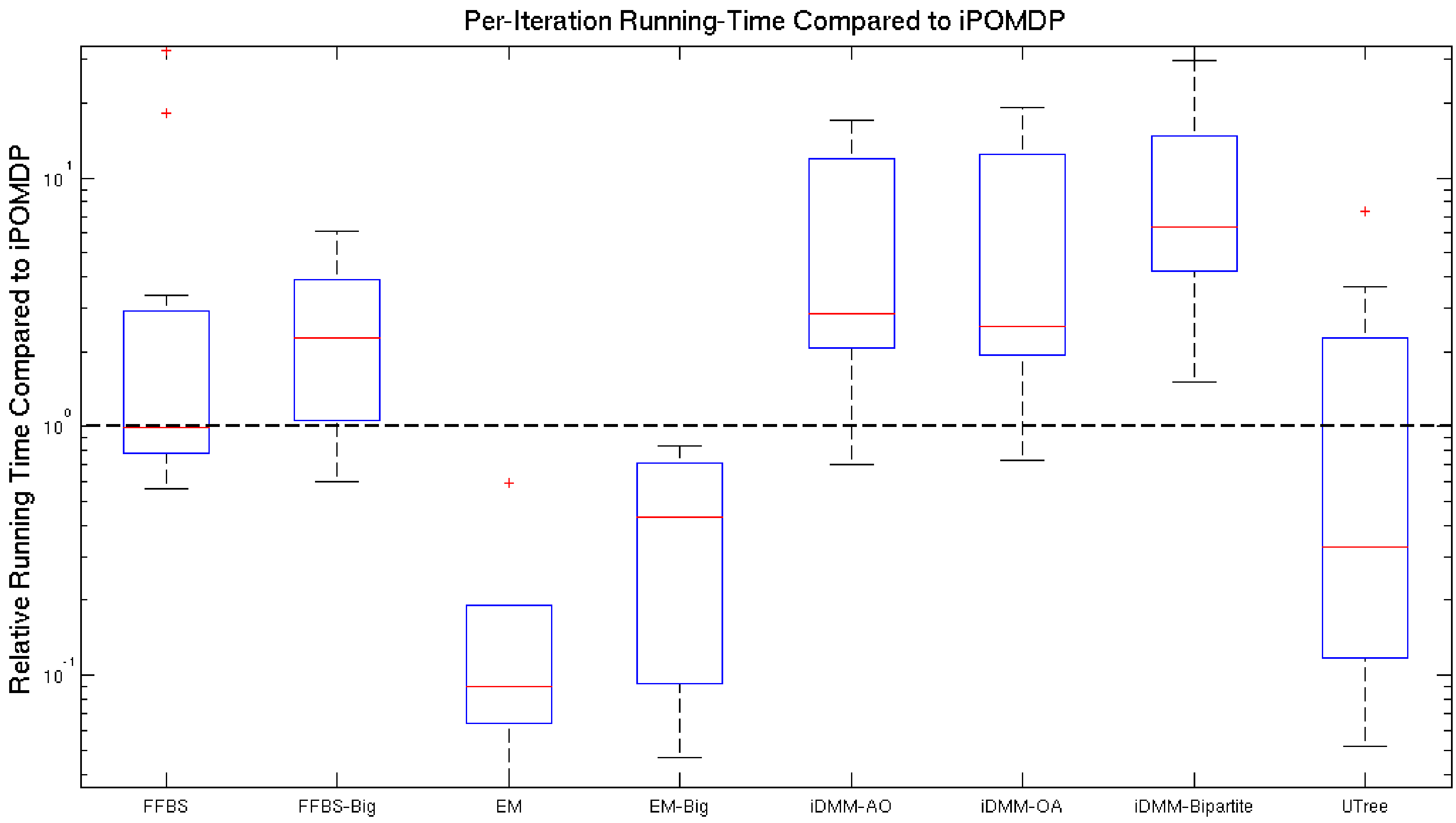


Results on Standard Pro

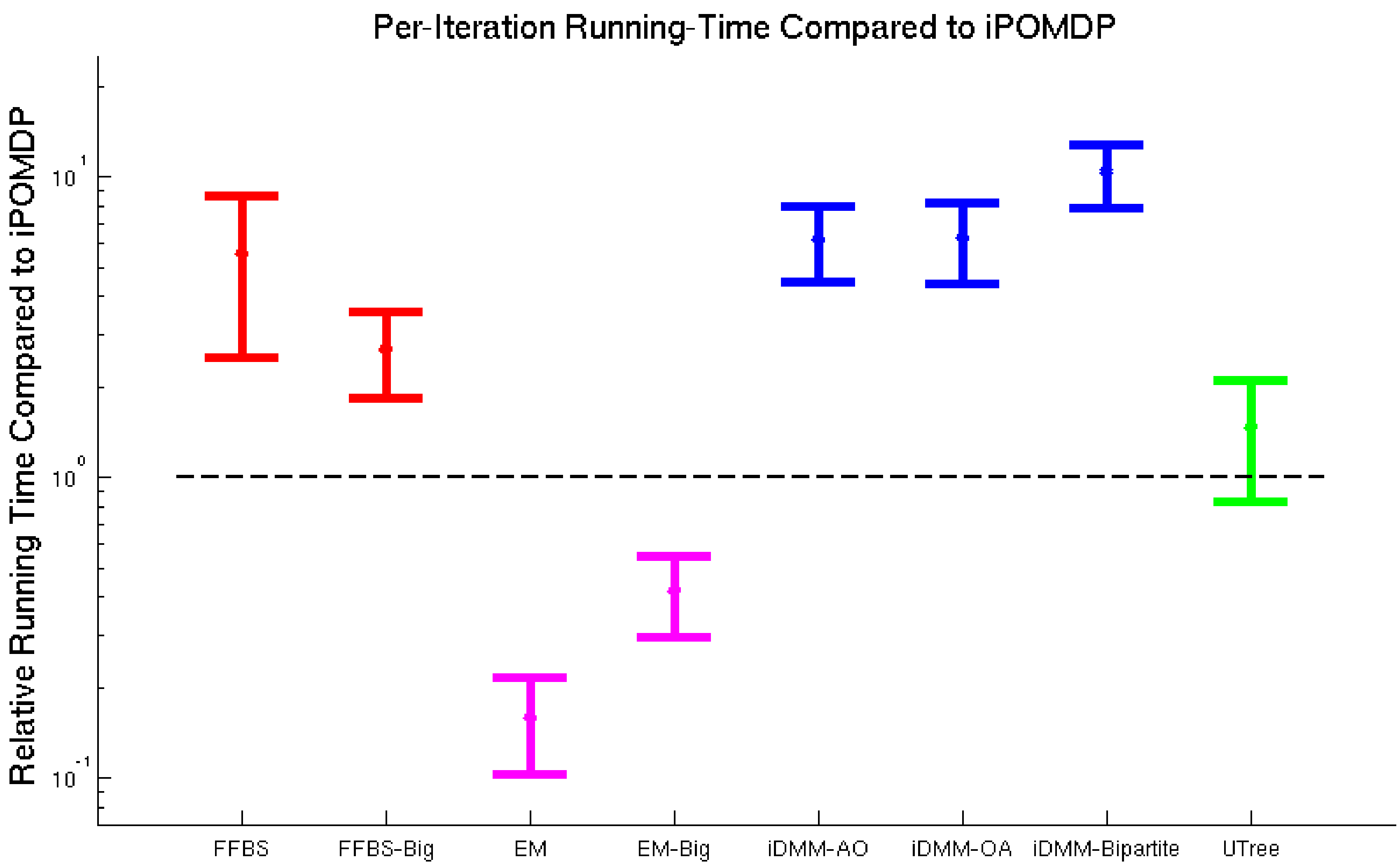




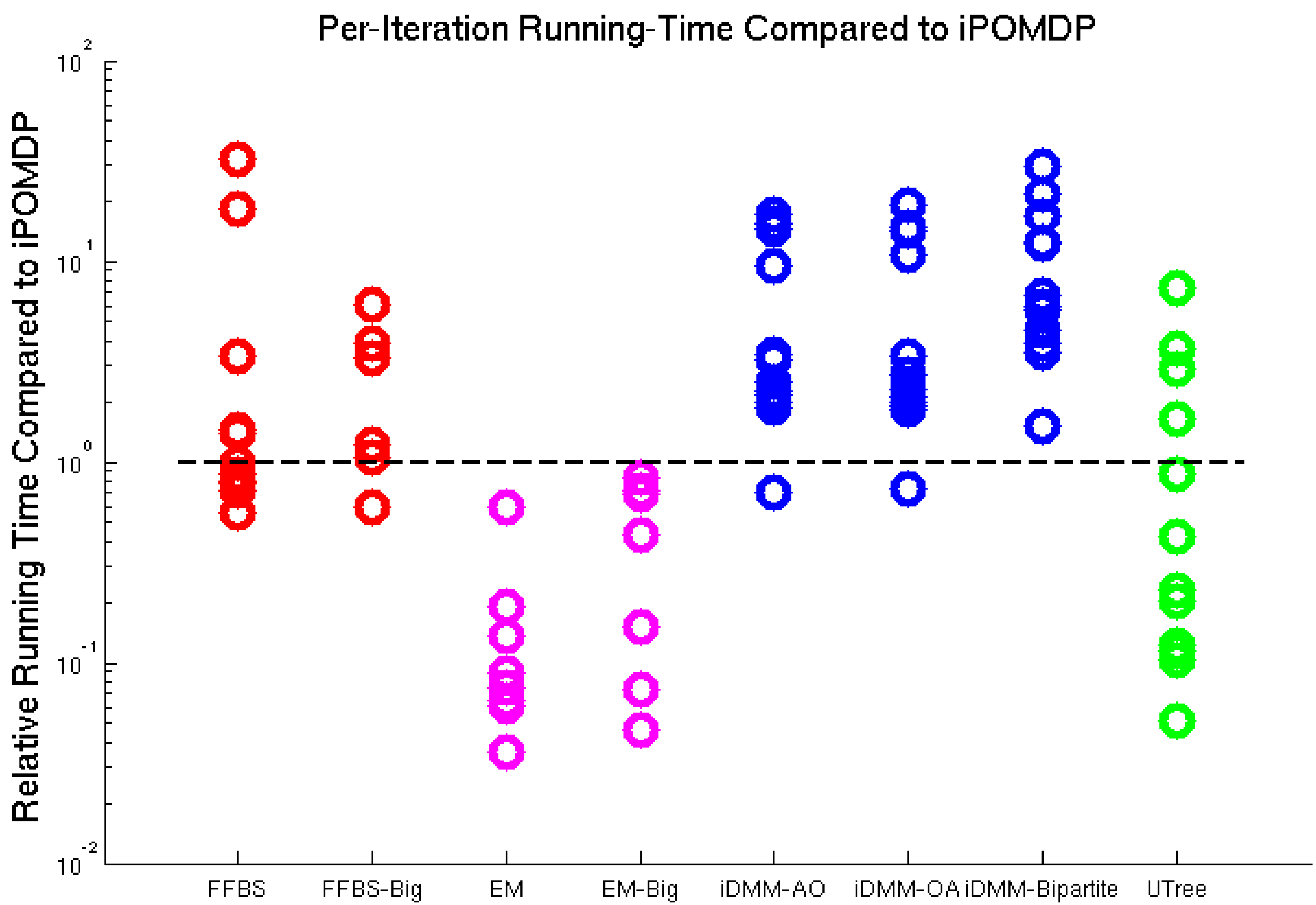
Results on Standard Problems



Results on Standard Problems



Results on Standard Problems



Outline

- Introduction: The partially-observable reinforcement learning setting
- Framework: Bayesian reinforcement learning
- Applying nonparametrics:
  - Infinite Partially Observable Markov Decision Processes
  - Infinite State Controllers*
  - Infinite Dynamic Bayesian Networks
- Conclusions and Continuing Work

* Doshi-Velez, Wingate, Tenenbaum, and Roy, NIPS 2010

Leveraging Expert Trajectories

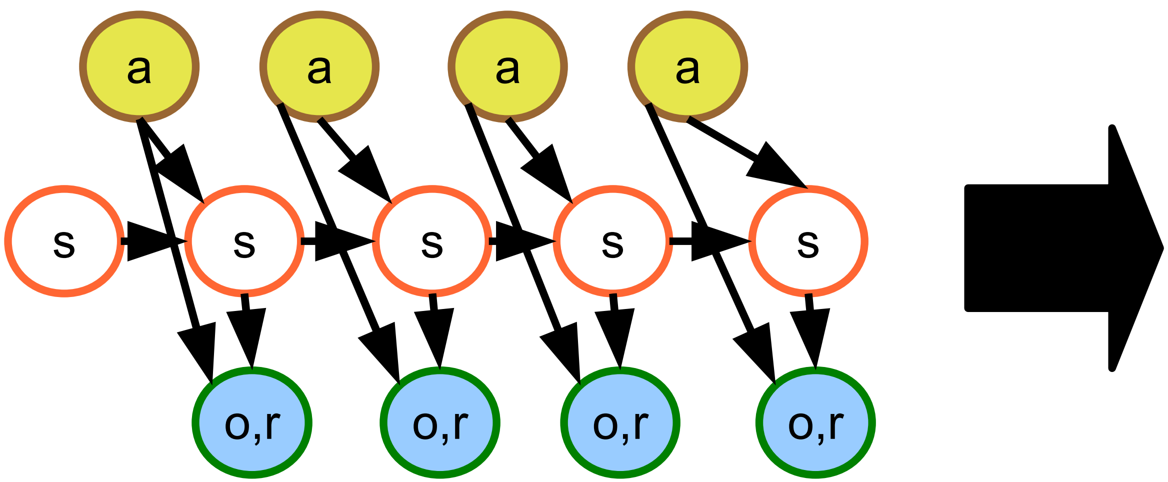
Often, an expert (could be another planning algorithm) can provide near-optimal trajectories.

However, combining expert trajectories with data from self-exploration is challenging:

- Experience provides direct information about the dynamics, which indirectly suggests a policy.
- Experts provide direct information about the policy, which indirectly suggests dynamics.

Policy Priors

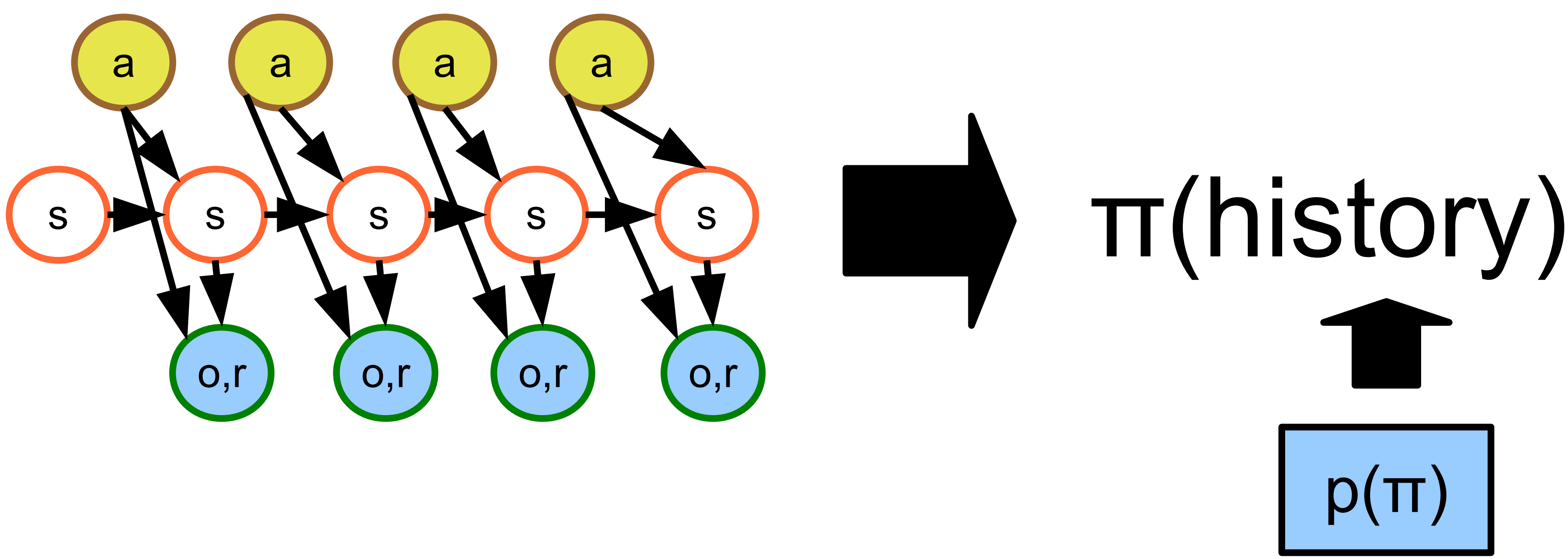
Suppose we're turning data from an expert's demo into a policy...



π(history)

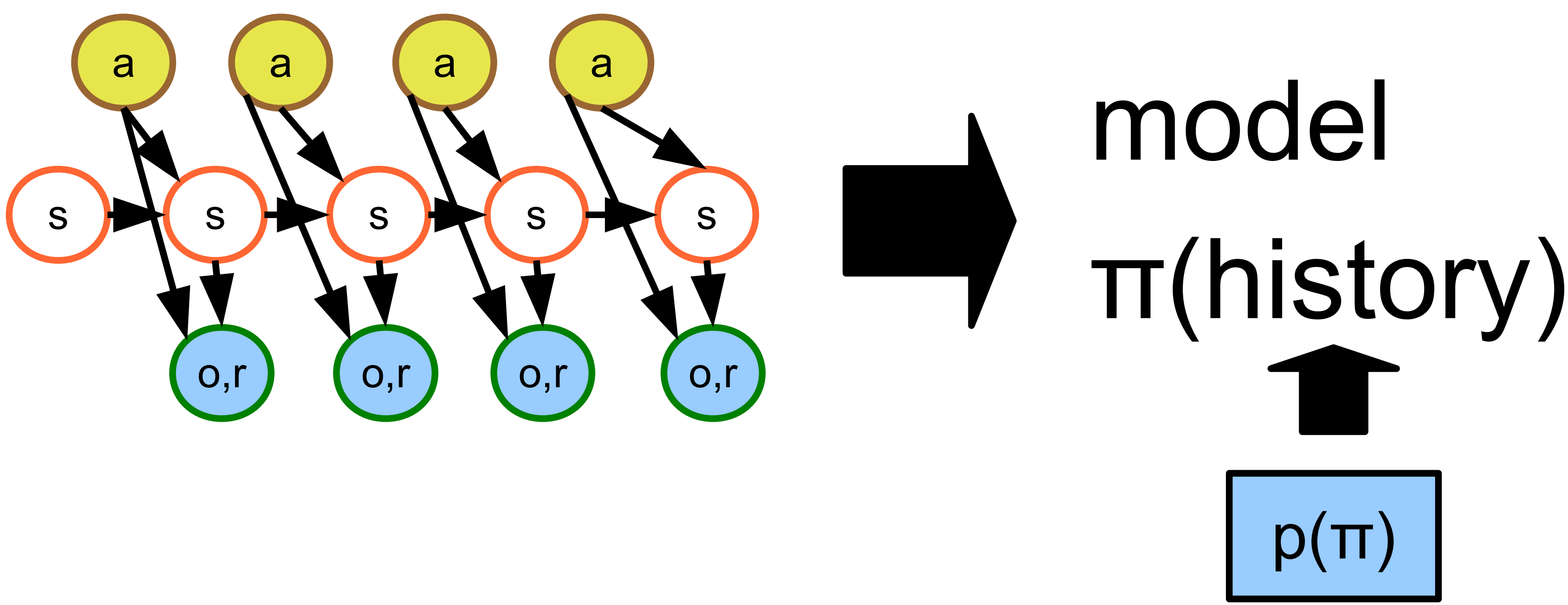
Policy Priors

Suppose we're turning data from an expert's demo into a policy... a prior over policies can help avoid overfitting...



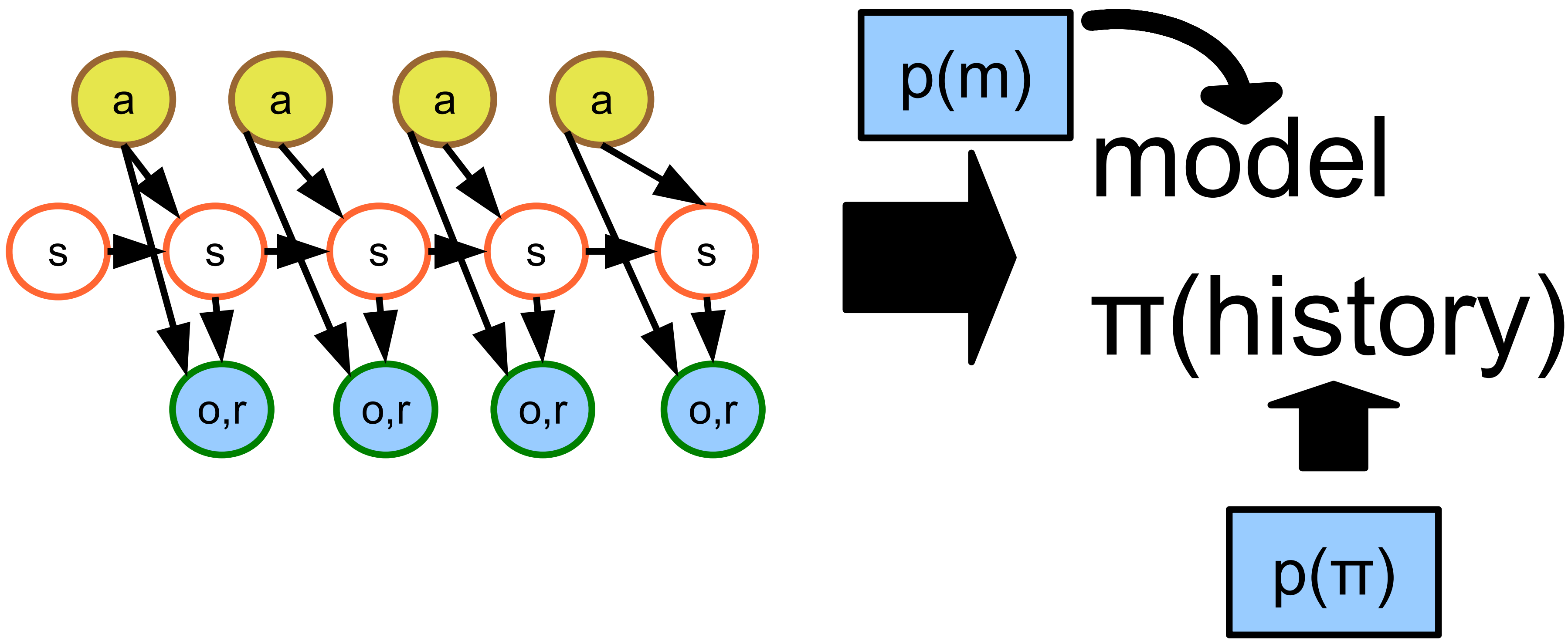
Policy Priors

Suppose we're turning data from an expert's demo into a policy... a prior over policies can help avoid overfitting... but the demo also provides information about the model



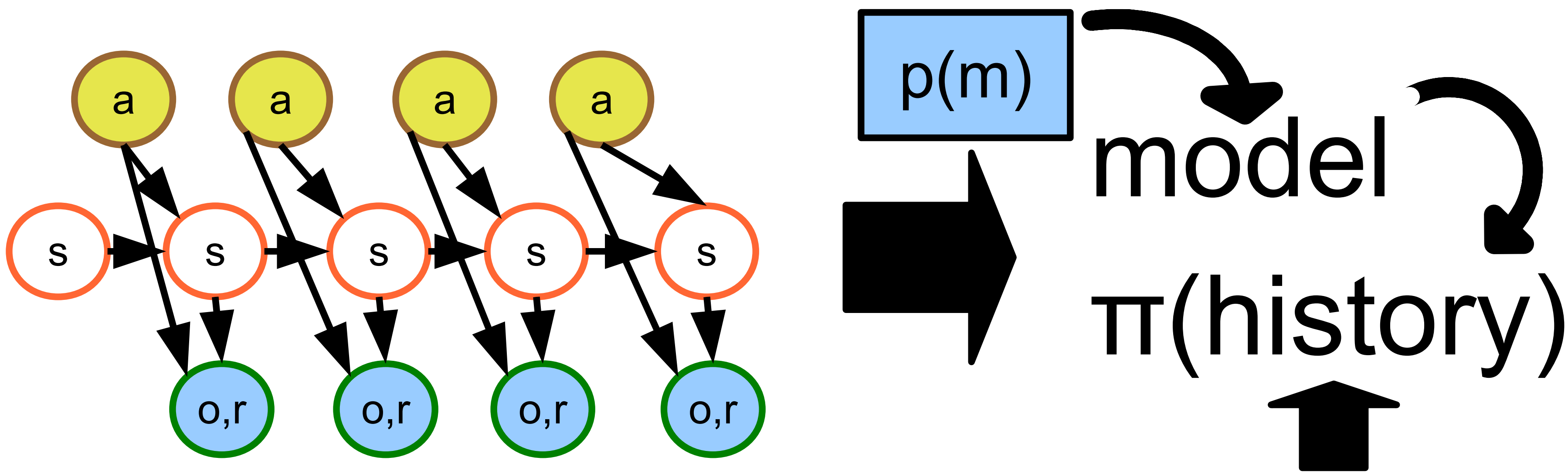
Policy Priors

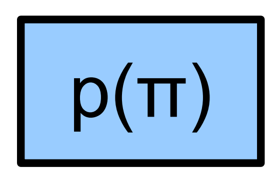
Suppose we're turning data from an expert's demo into a policy... a prior over policies can help avoid overfitting... but the demo also provides information about the model



Policy Priors

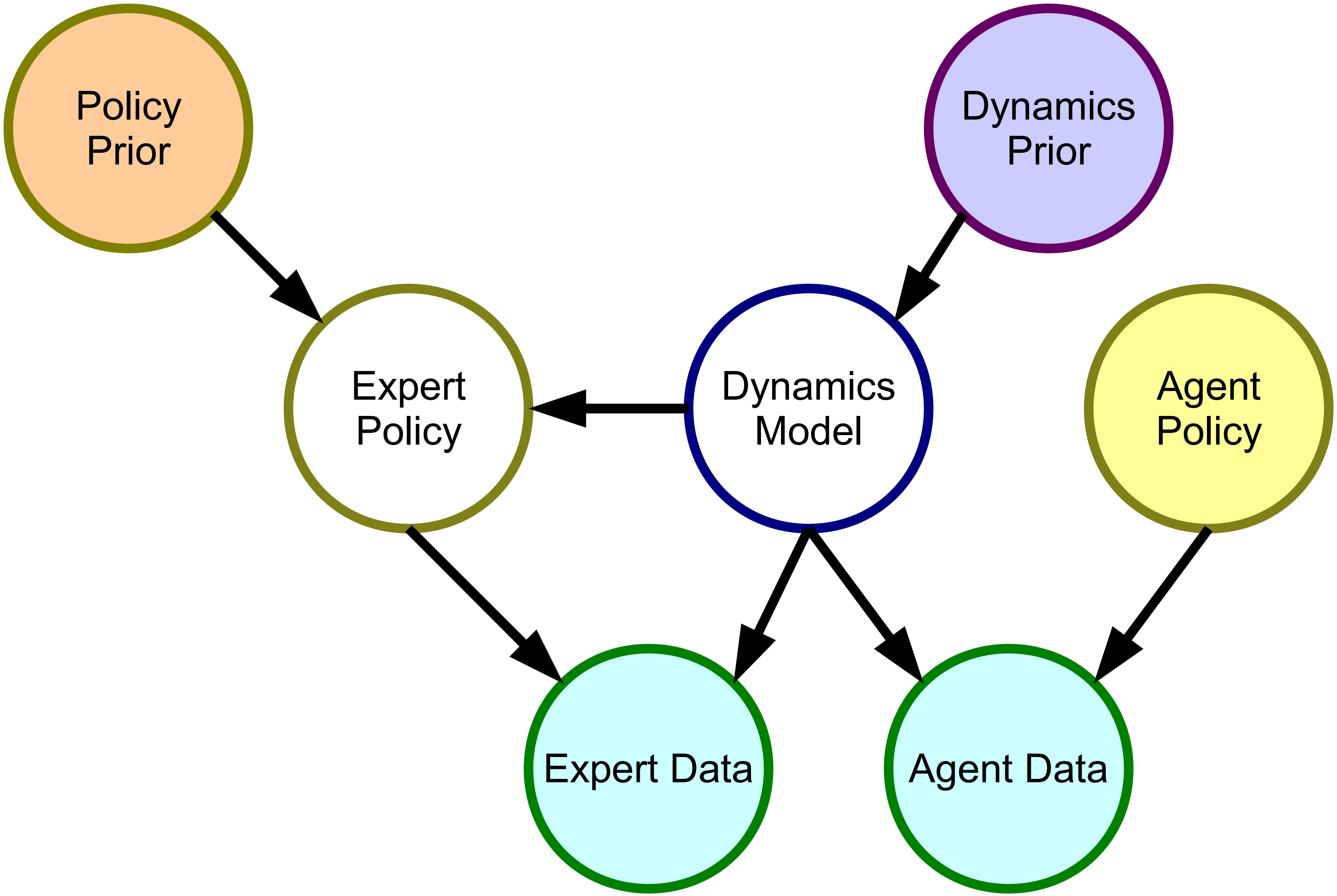
Suppose we're turning data from an expert's demo into a policy... a prior over policies can help avoid overfitting... but the demo also provides information about the model





But if we assume the expert acts near optimally with respect to the model, **don't** want to regularize!

Policy Prior Model

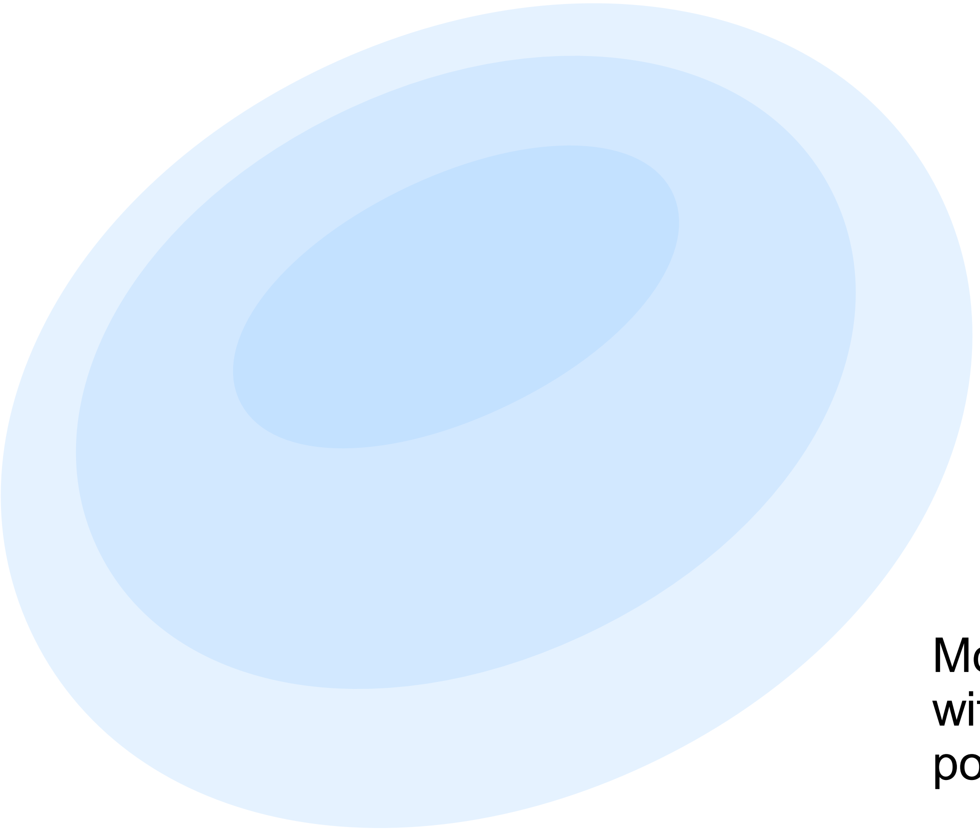


Policy Prior: What it means

Models
with simple
dynamics

Model Space

Policy Prior: What it means

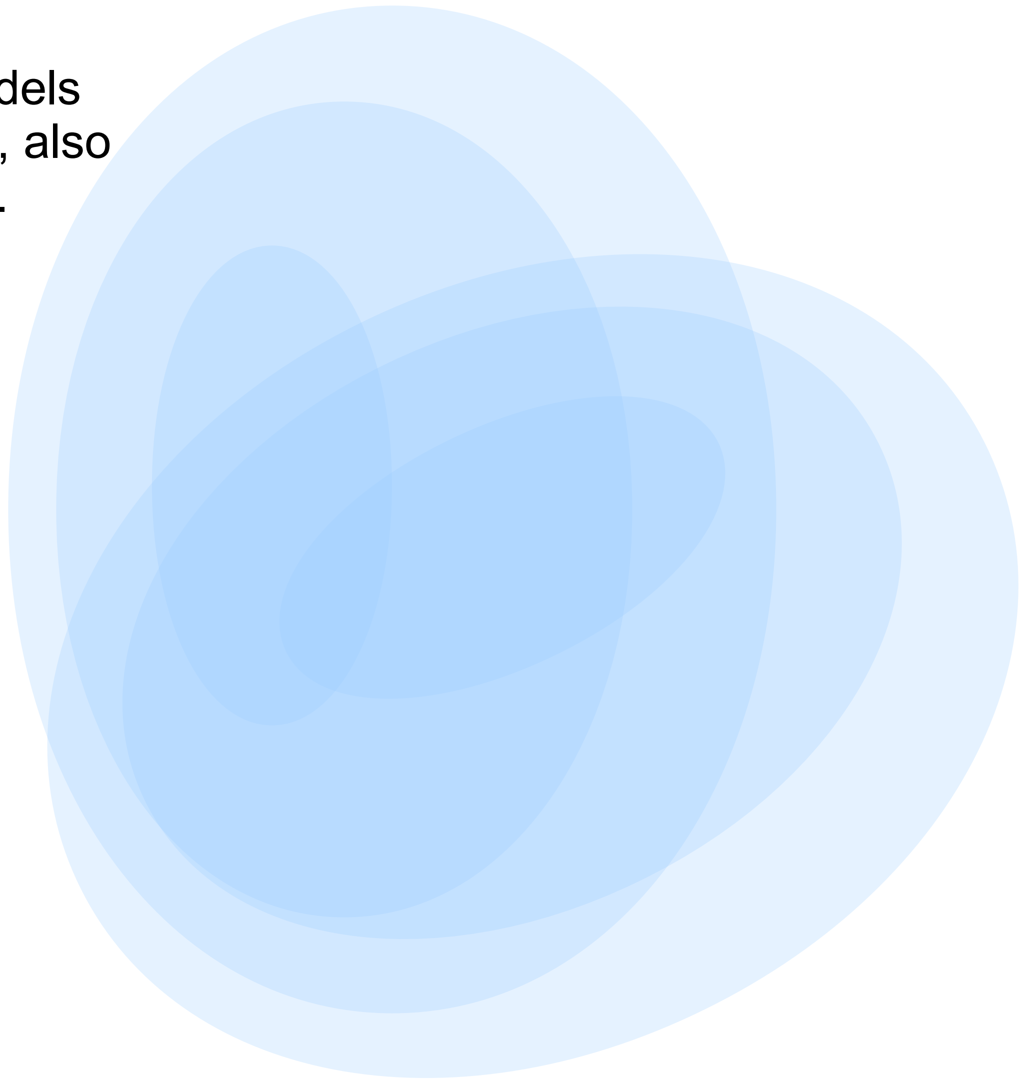


Models
with simple control
policies.

Model Space

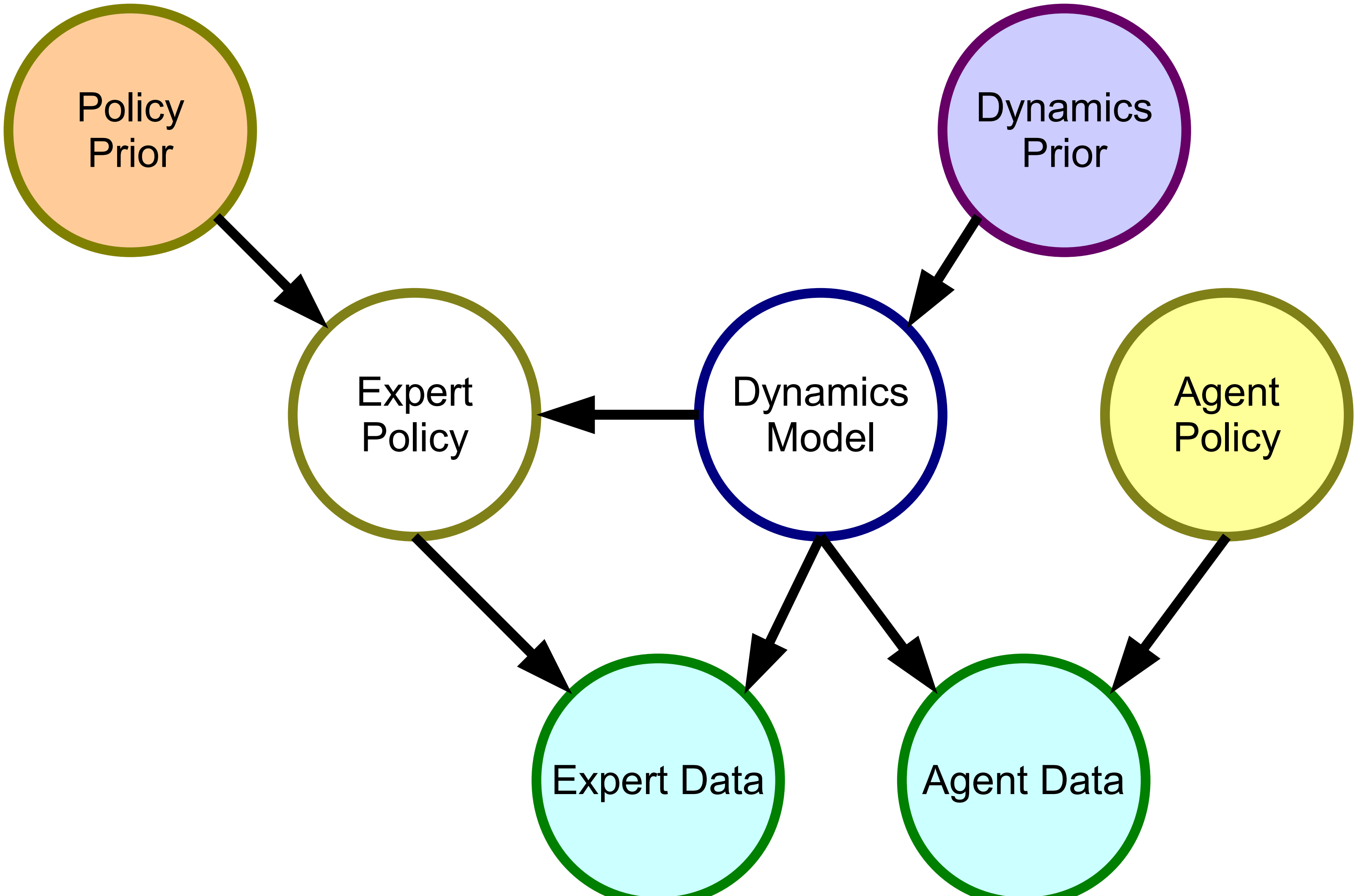
Policy Prior: What it means

Joint Prior: models with few states, also easy to control.



Model Space

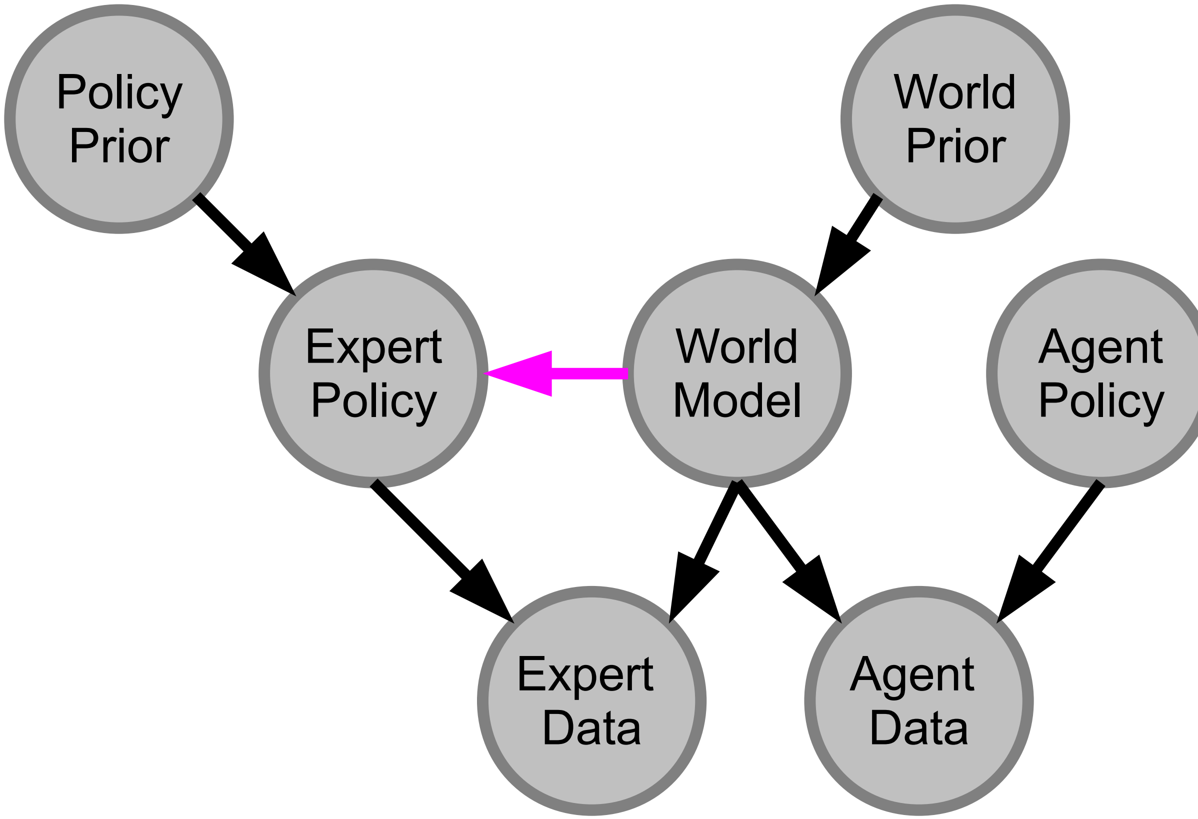
Policy Prior Model



Modeling the Model-Policy Link

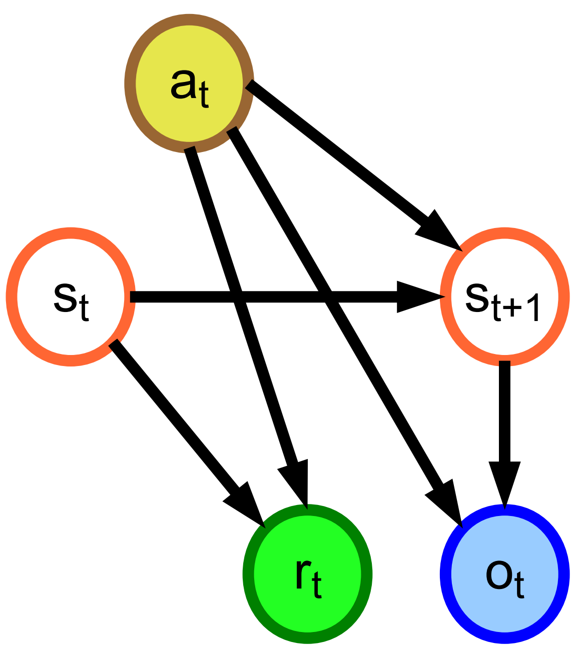
Apply a Factorization:
the probability of the expert policy π
p(π | m , policy prior)
is proportional to
f(π , m) g(π , policy prior)

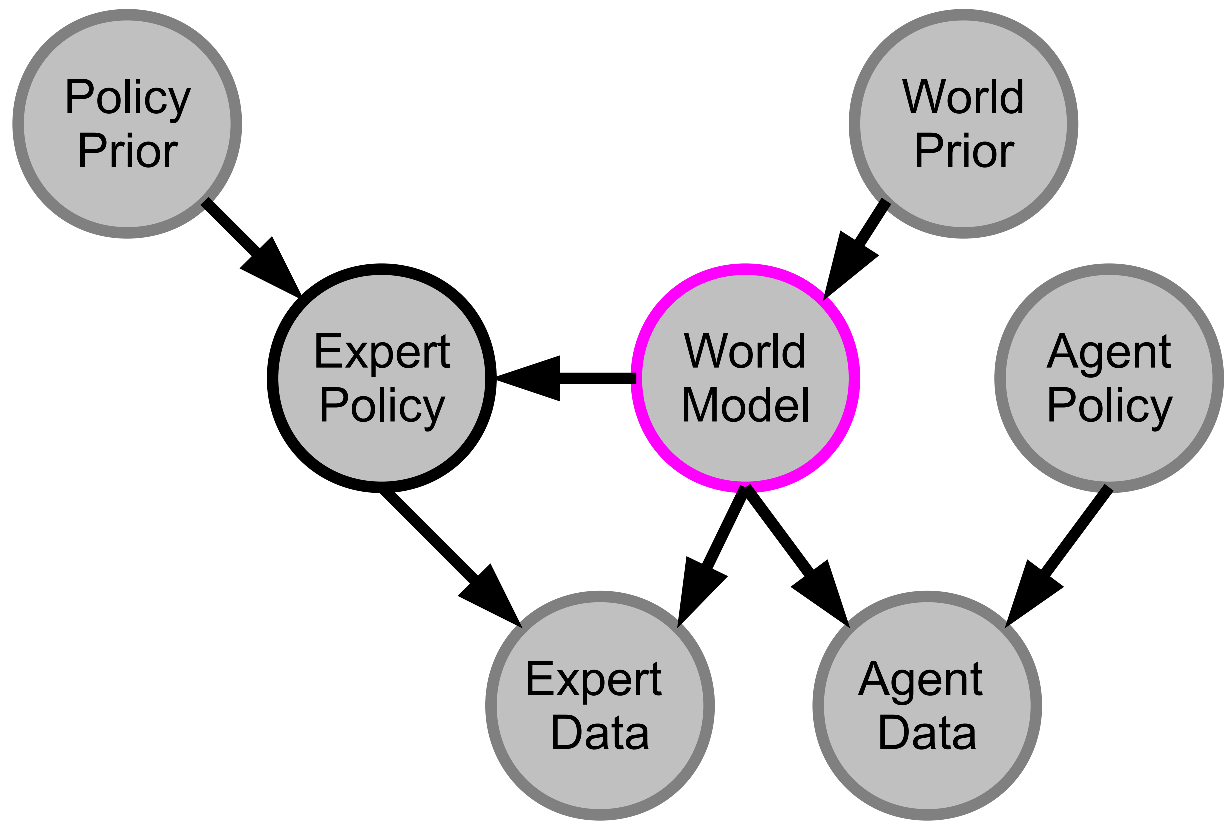
Many options for f(π , m), assume we want something like δ(π*,π) where π* is the optimal policy under m



Modeling the World Model

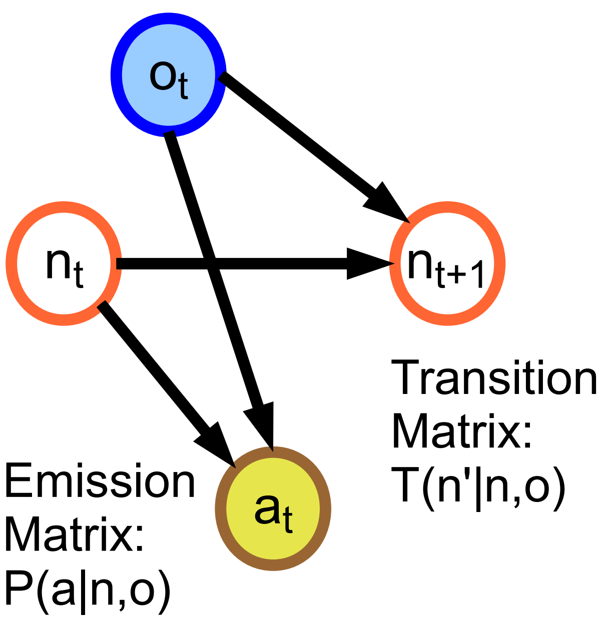
Represent the world model with an infinite POMDP

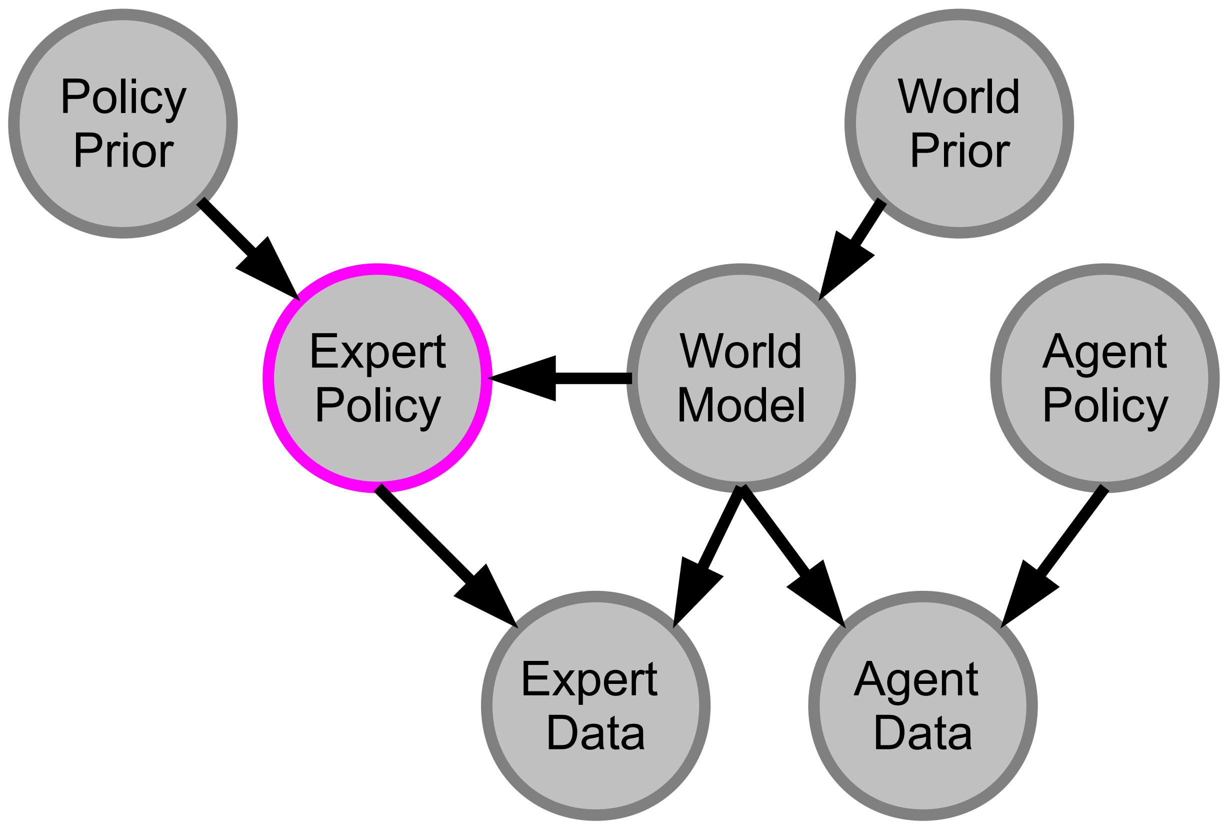




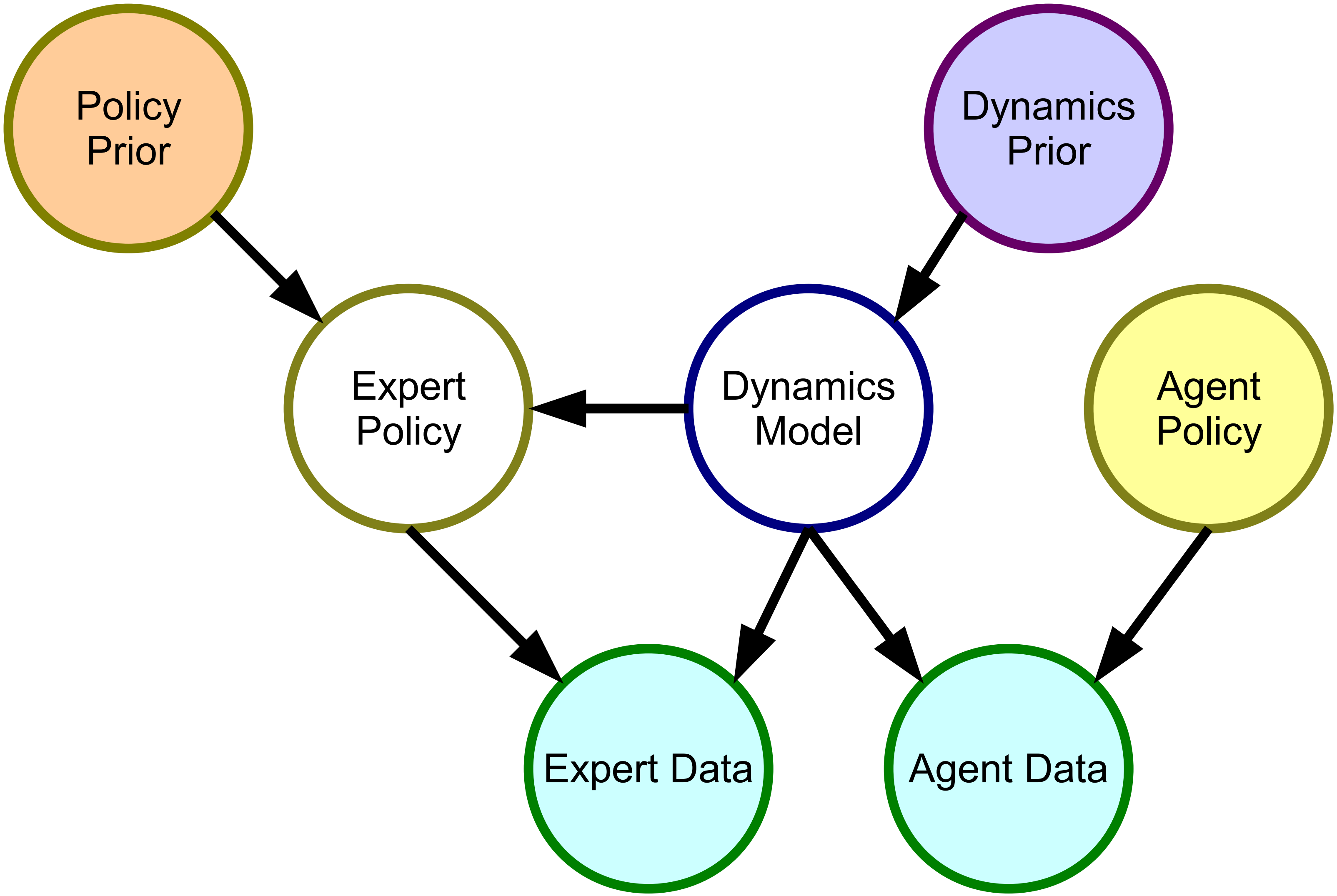
Modeling the Policy

Represent the policy as a (in)finite state controller:

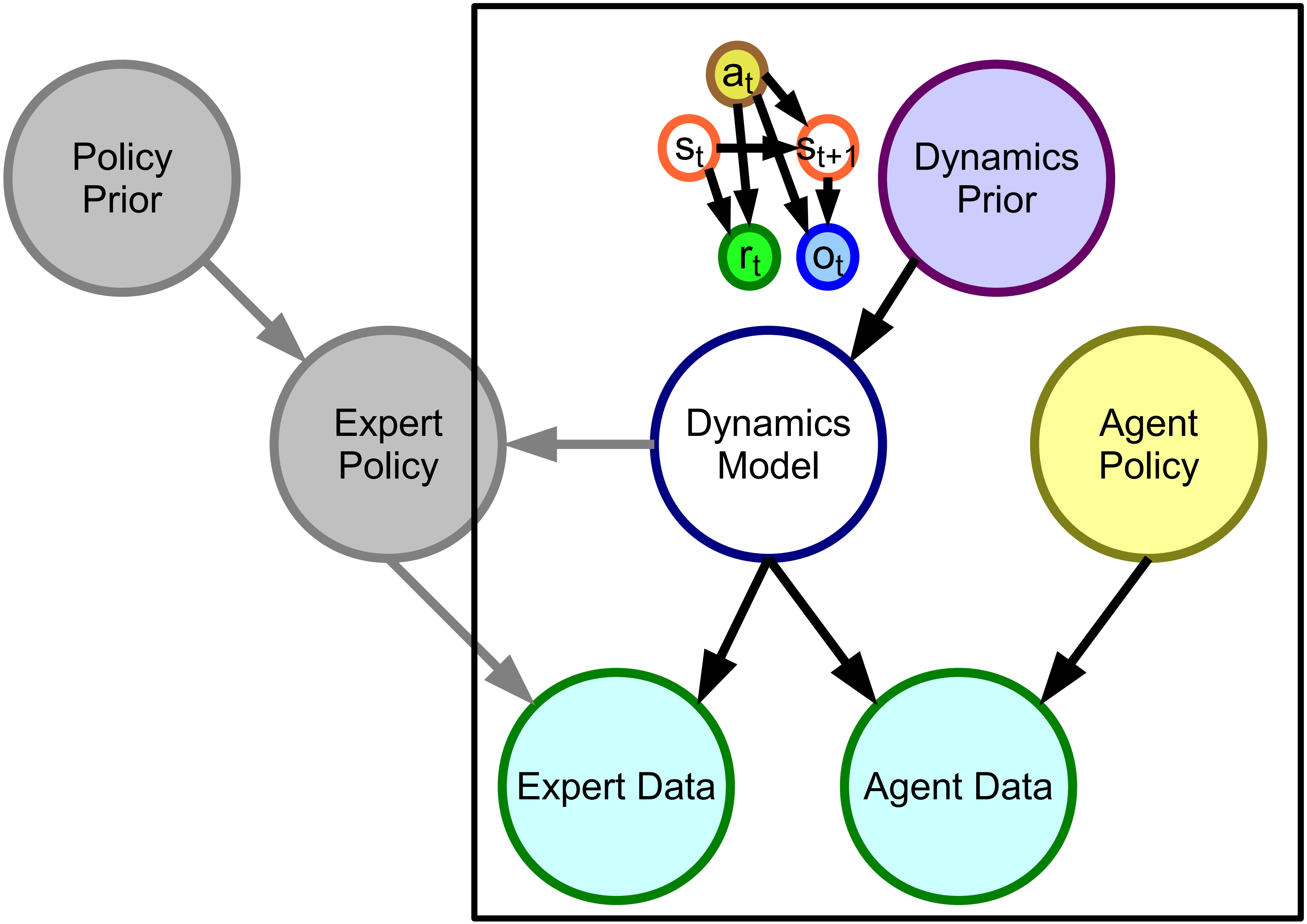




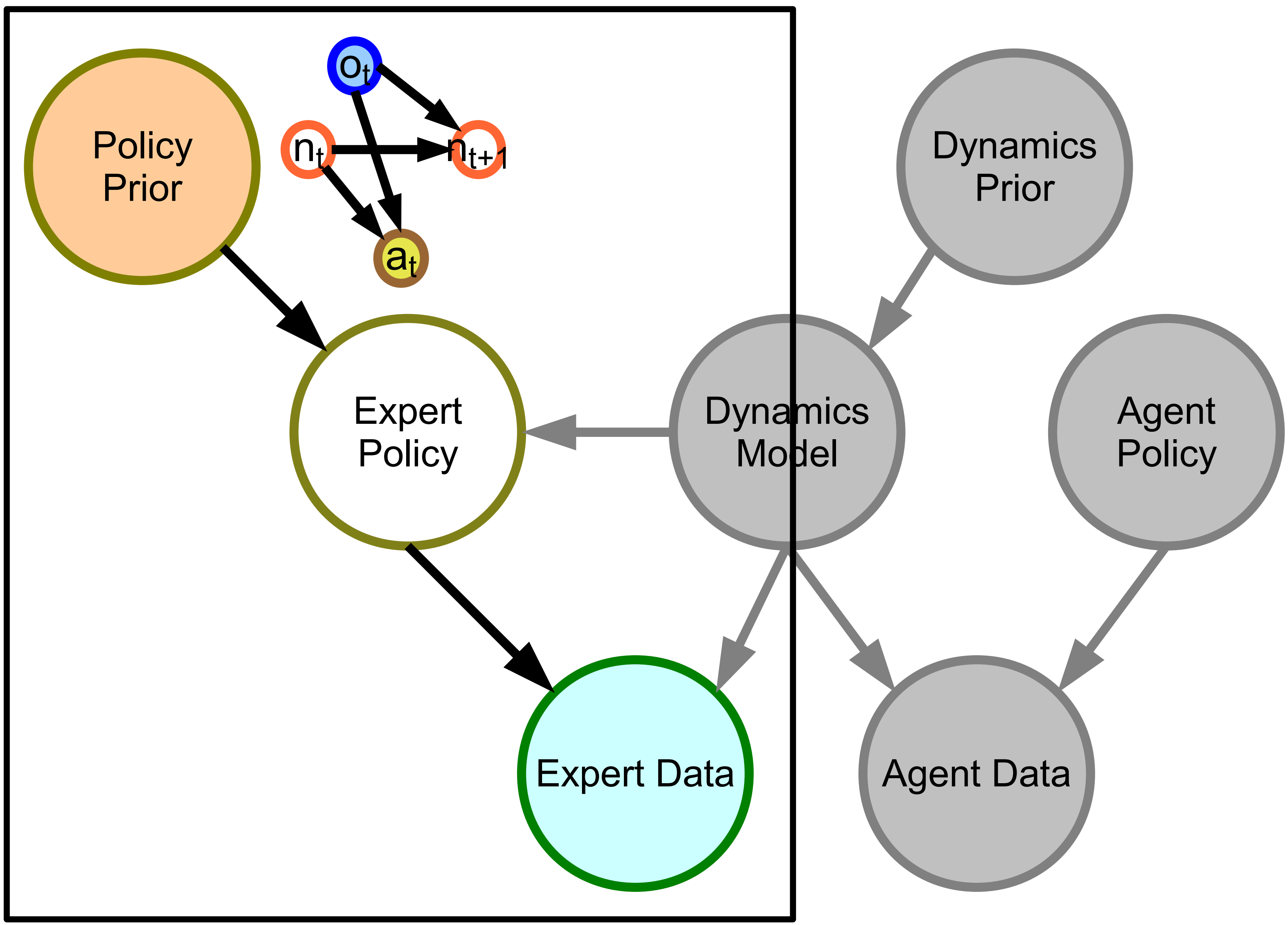
Doing Inference



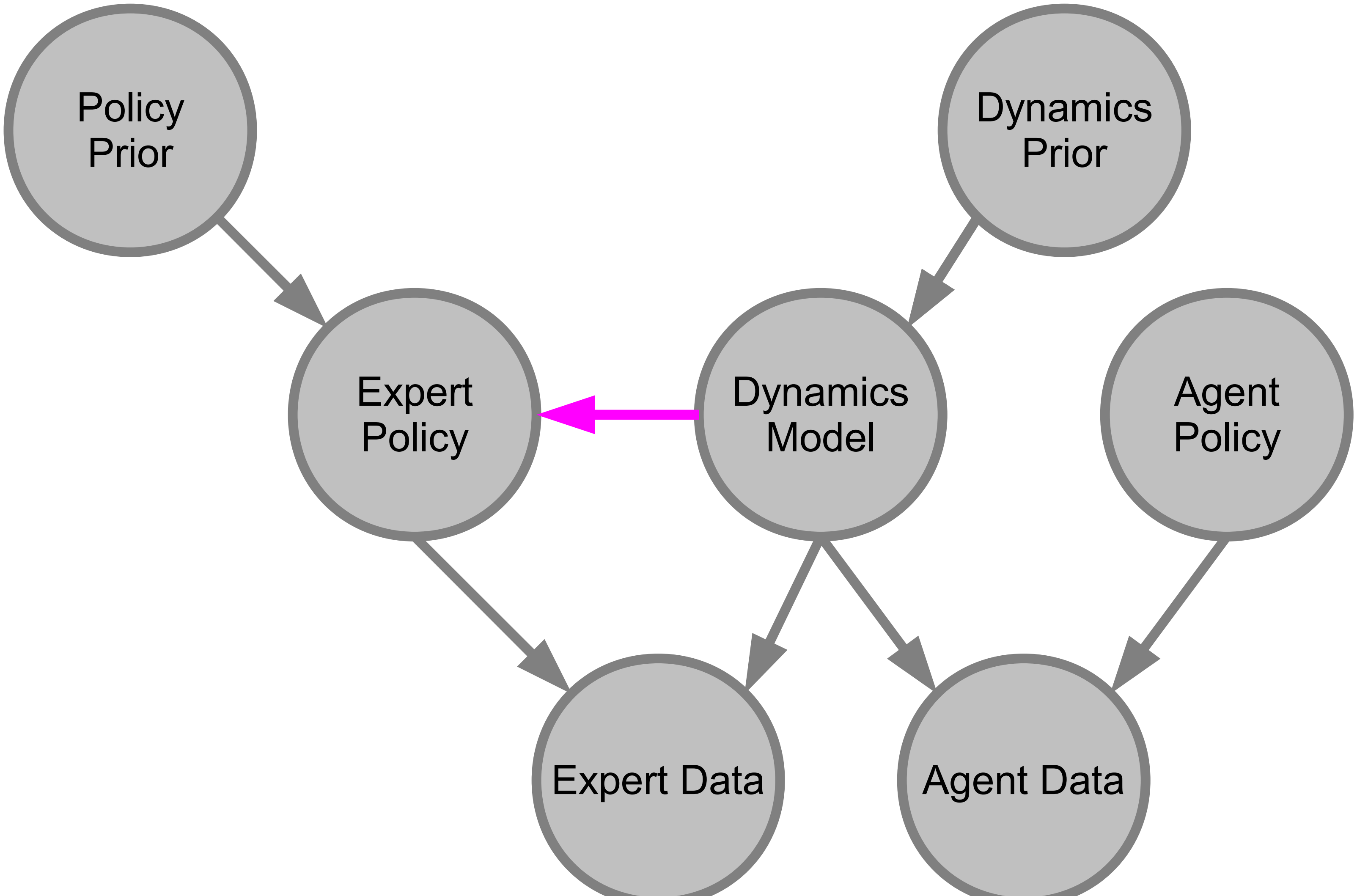
Some parts aren't too hard...



Some parts aren't too hard...



But: Model-Policy Link is Hard



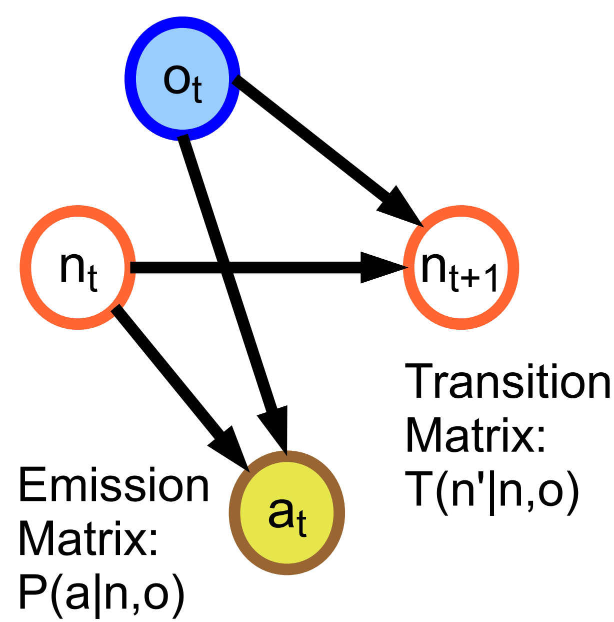
Sampling Policies Given Models

Suppose we choose f() and g() so that the probability of an expert policy, $p(\ \pi \mid m\ ,\ \text{data}\ ,\ \text{policy prior}\ )$ is proportional to

$$\underbrace{f(\ \pi\ ,\ m\ )}_{\delta(\pi^*,\pi)\text{ where }\pi^*\text{ is opt}(m)}\ \underbrace{g(\ \pi\ ,\ \text{data}\ ,\ \text{policy prior})}_{\text{iPOMDP prior + data}}$$

where the policy π is given by a set of

- transitions $T(n'|n,o)$
- emissions $P(a|n,o)$

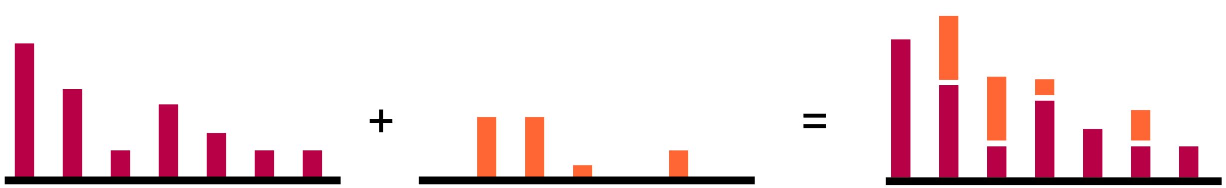


Looking at a Single T(n'|n,o)

Consider the inference update for a single distribution T(n'|n,o):

$$f(\pi, m)\ \boxed{g(\pi, \text{data}, \text{policy prior})}$$

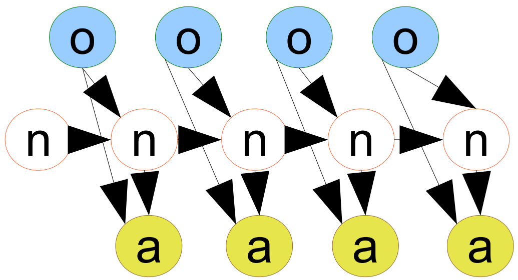
Easy with Beam Sampling if we have
Dirichlet-multinomial conjugacy
(data just adds counts to the prior)



β: prior/mean transition from the iPOMDP

node-visit counts: how often n' occurred after seeing o in n

posterior Dirichlet for T(n'|n,o)

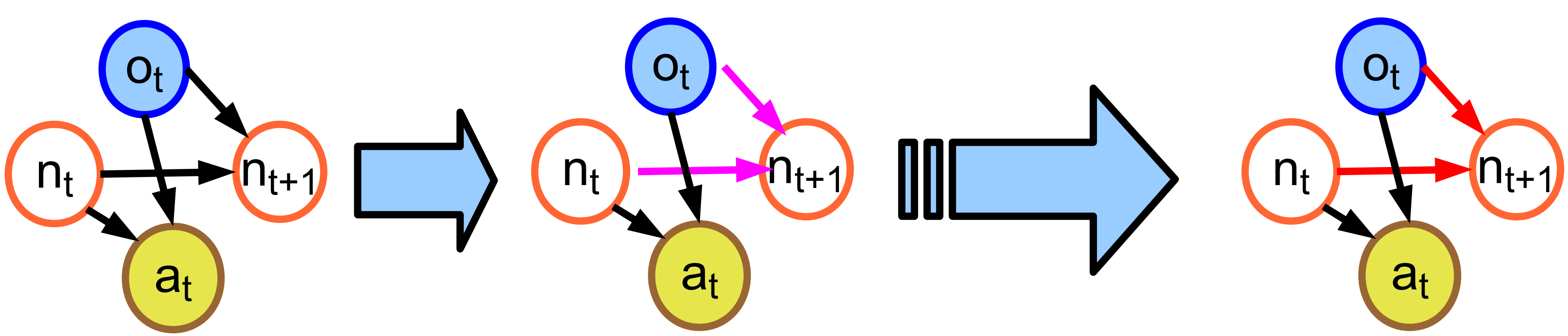


Looking at a Single T(n'|n,o)

Consider the inference update for a single distribution T(n'|n,o):

$$\underbrace{\boxed{f(\pi, m)}}\, g(\pi, \text{data}, \text{policy prior})$$

Approximate δ(π*,π)
with Dirichlet counts, using
Bounded Policy Iteration (BPI)
(Poupart and Boutilier, 2003)



Current policy has some value for T(n'|n,o)

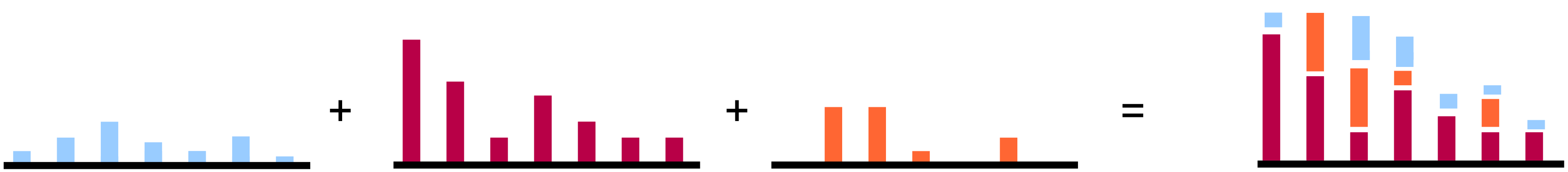
One step of BPI changes T'(n'|n,o) = T(n'|n,o) + a (keeps node alignment)

More steps of BPI change T*(n'|n,o) = T(n'|n,o) + a* (nodes still aligned)

Combine with a Tempering Scheme

Consider the inference update for a single distribution T(n'|n,o):

$$\underbrace{\boxed{f(\pi , m)}}_{\substack{\text{Approximate } \delta(\pi^*,\pi) \\ \text{with Dirichlet counts/BPI}}} g(\pi , \text{data} , \text{policy prior})$$

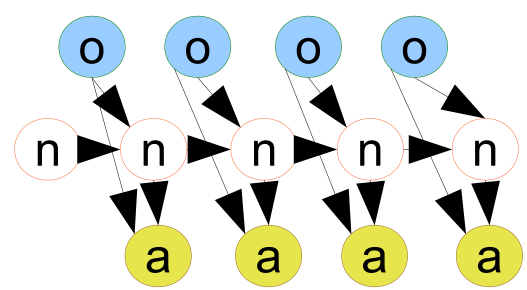


Distribution for T*(n|n,o) scaled by k

β: prior/mean transition from the iPOMDP

node-visit counts: how often n' occurred after seeing o in n

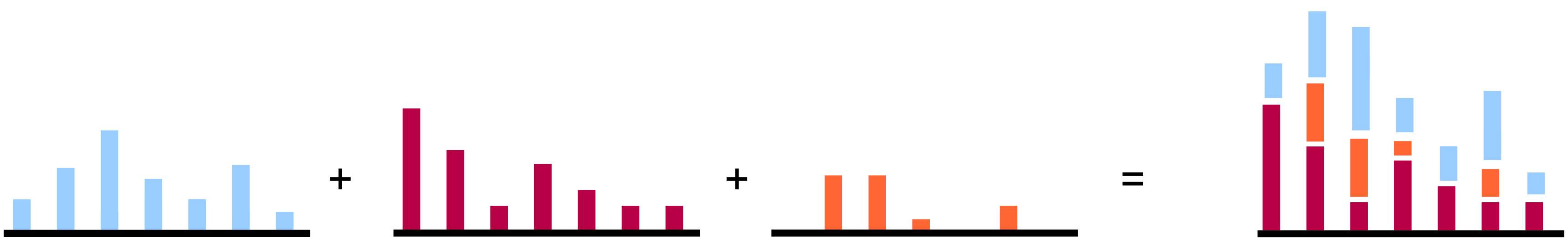
posterior Dirichlet for T(n'|n,o)



Combine with a Tempering Scheme

Consider the inference update for a single distribution T(n'|n,o):

$$\underbrace{\boxed{f(\pi, m)}}_{\substack{\text{Approximate } \delta(\pi^*,\pi) \\ \text{with Dirichlet counts/BPI}}} g(\pi, \text{data}, \text{policy prior})$$

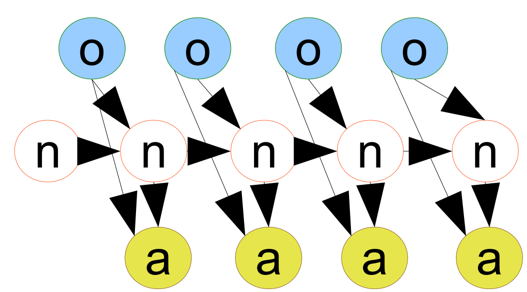


Distribution for T*(n|n,o) scaled by k

β: prior/mean transition from the iPOMDP

node-visit counts: how often n' occurred after seeing o in n

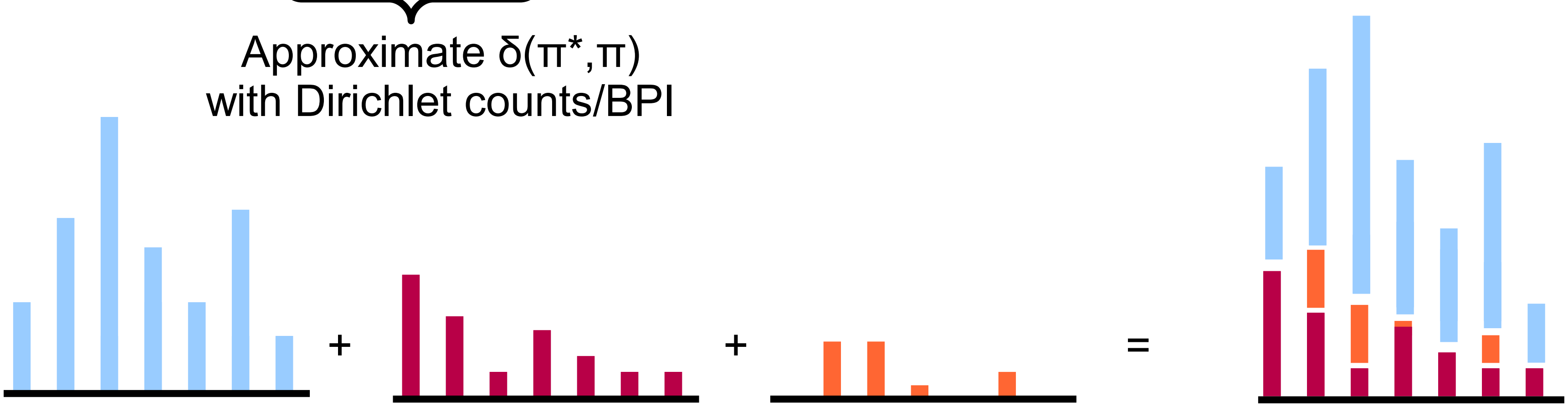
posterior Dirichlet for T(n'|n,o)



Combine with a Tempering Scheme

Consider the inference update for a single distribution T(n'|n,o):

$$\underbrace{\boxed{f(\pi , m)}}_{\substack{\text{Approximate } \delta(\pi^*,\pi) \\ \text{with Dirichlet counts/BPI}}} g(\pi , \text{data} , \text{policy prior})$$

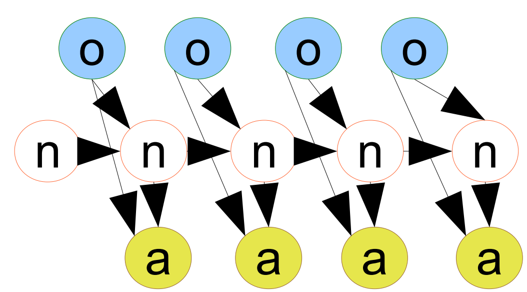


Distribution for T*(n|n,o) scaled by k

β: prior/mean transition from the iPOMDP

node-visit counts: how often n' occurred after seeing o in n

posterior Dirichlet for T(n'|n,o)



Sampling Models Given Policies

Apply Metropolis-Hastings Steps:

1. Propose a new model m' from q(m') = g(m | all data, prior)
2. Accept the new value with probability

$$min(1, \underbrace{\frac{f(\pi, m')\, g(m', D, p_M)}{f(\pi, m)\, g(m, D, p_M)}}_{\substack{\text{Likelihood ratio:} \\ p(m')/p(m)}} \cdot \underbrace{\frac{g(m, D, p_M)}{g(m', D, p_M)}}_{\substack{\text{Proposal ratio:} \\ q(m)/q(m')}}) = min(1, \frac{f(\pi, m')}{f(\pi, m)})$$

Sampling Models Given Policies

Apply Metropolis-Hastings Steps:

1. Propose a new model m' from q(m') = g(m | all data, prior)
2. Accept the new value with probability

$$min(1, \frac{f(\pi, m')g(m', D, p_M)\cdot g(m, D, p_M)}{f(\pi, m)g(m, D, p_M)\cdot g(m', D, p_M)}) = min(1, \frac{f(\pi, m')}{f(\pi, m)})$$

We still have a problem: If f() is strongly peaked, will never accept!

Sampling Models Given Policies

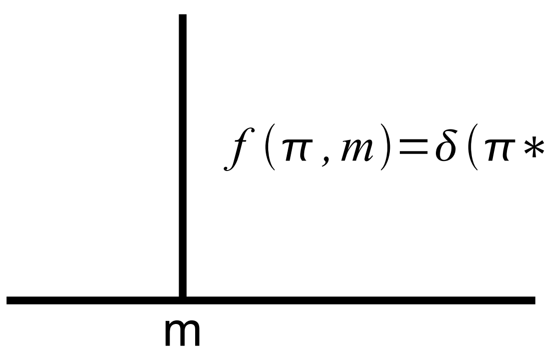
Apply Metropolis-Hastings Steps:

1. Propose a new model m' from q(m') = g(m | all data, prior)
2. Accept the new value with probability

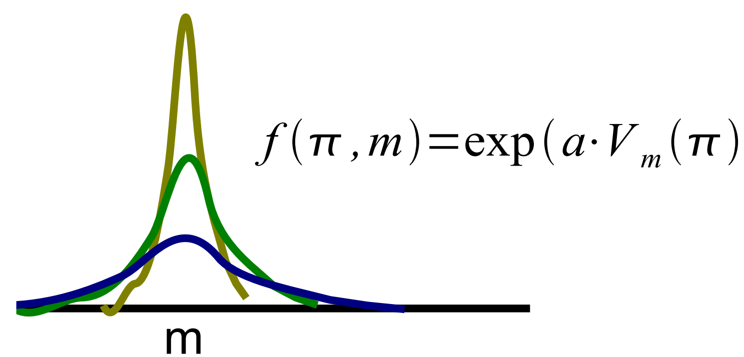
$$min(1, \frac{f(\pi, m')g(m', D, p_M)\cdot g(m, D, p_M)}{f(\pi, m)g(m, D, p_M)\cdot g(m', D, p_M)}) = min(1, \frac{f(\pi, m')}{f(\pi, m)})$$

We still have a problem: If f() is strongly peaked, will never accept!

Temper again...

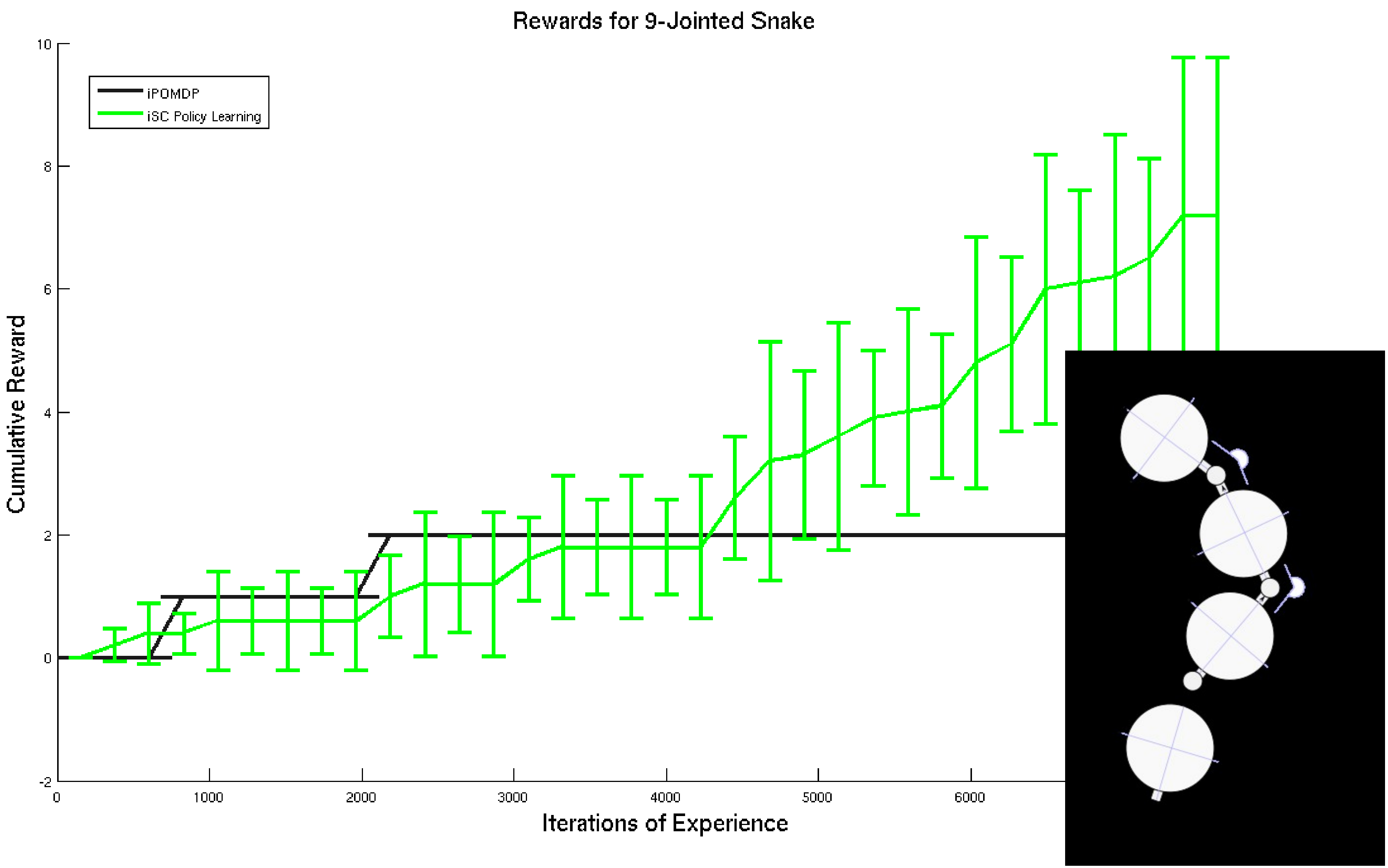


Desired link

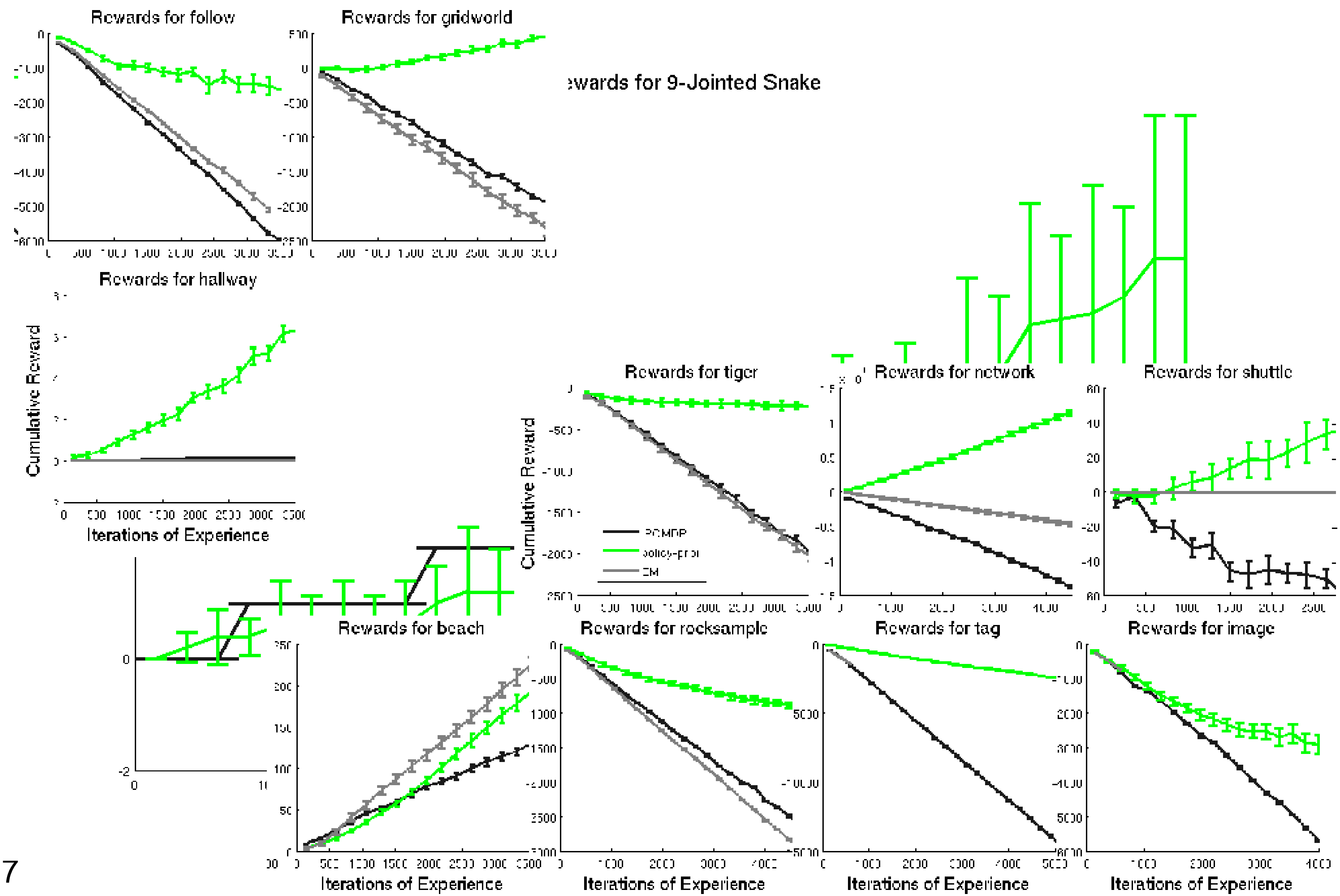


Intermediate functions

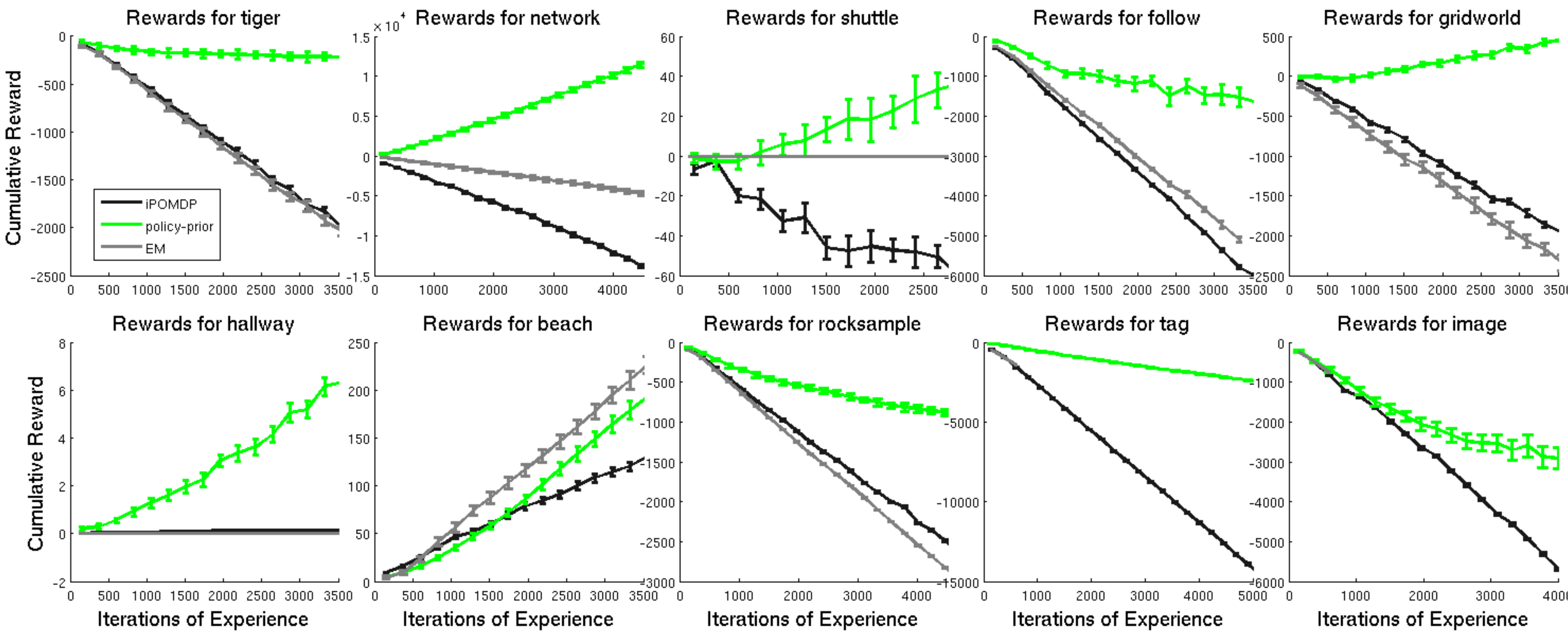
Example Result



Same trend for Standard Domains



Results on Standard Problems

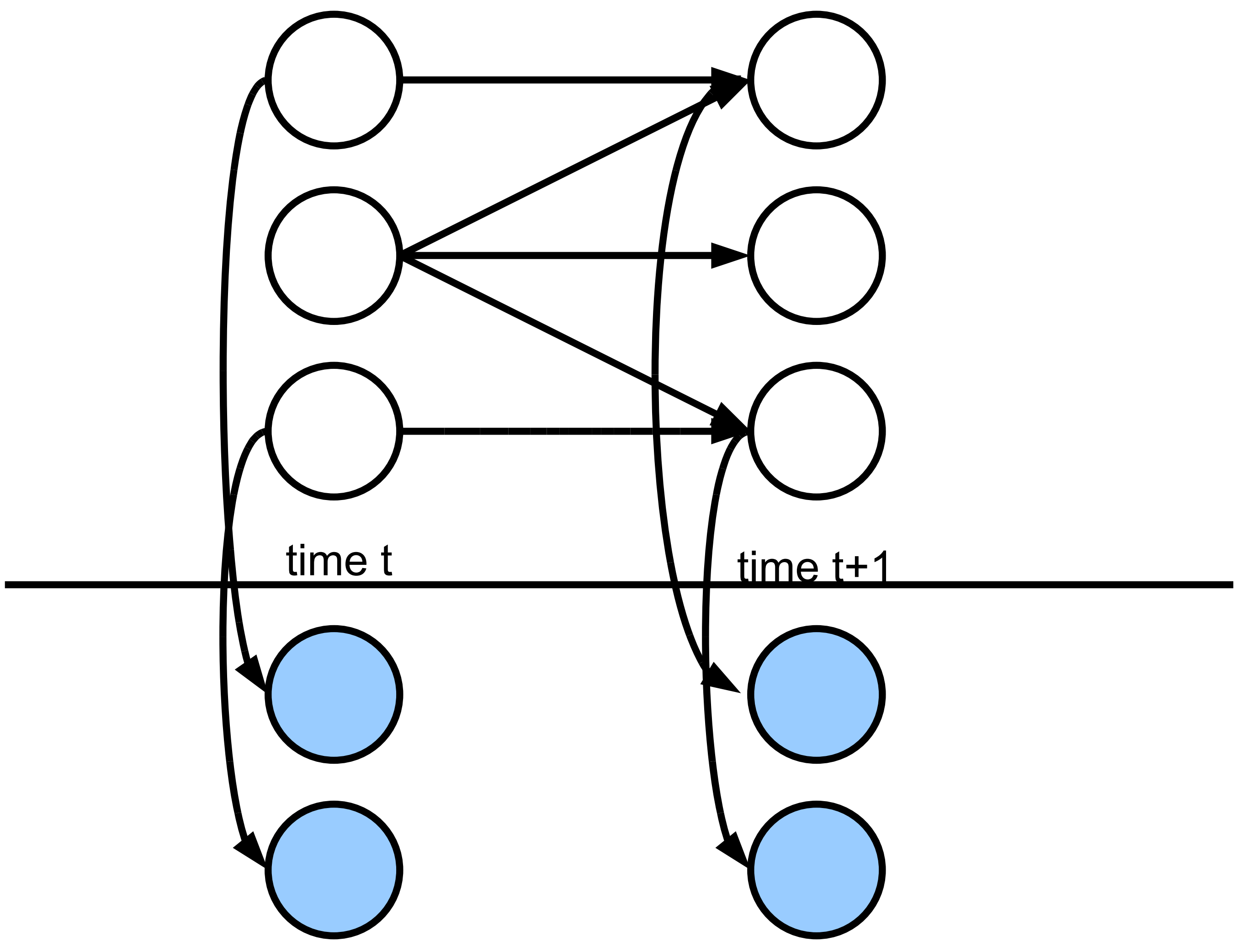


Outline

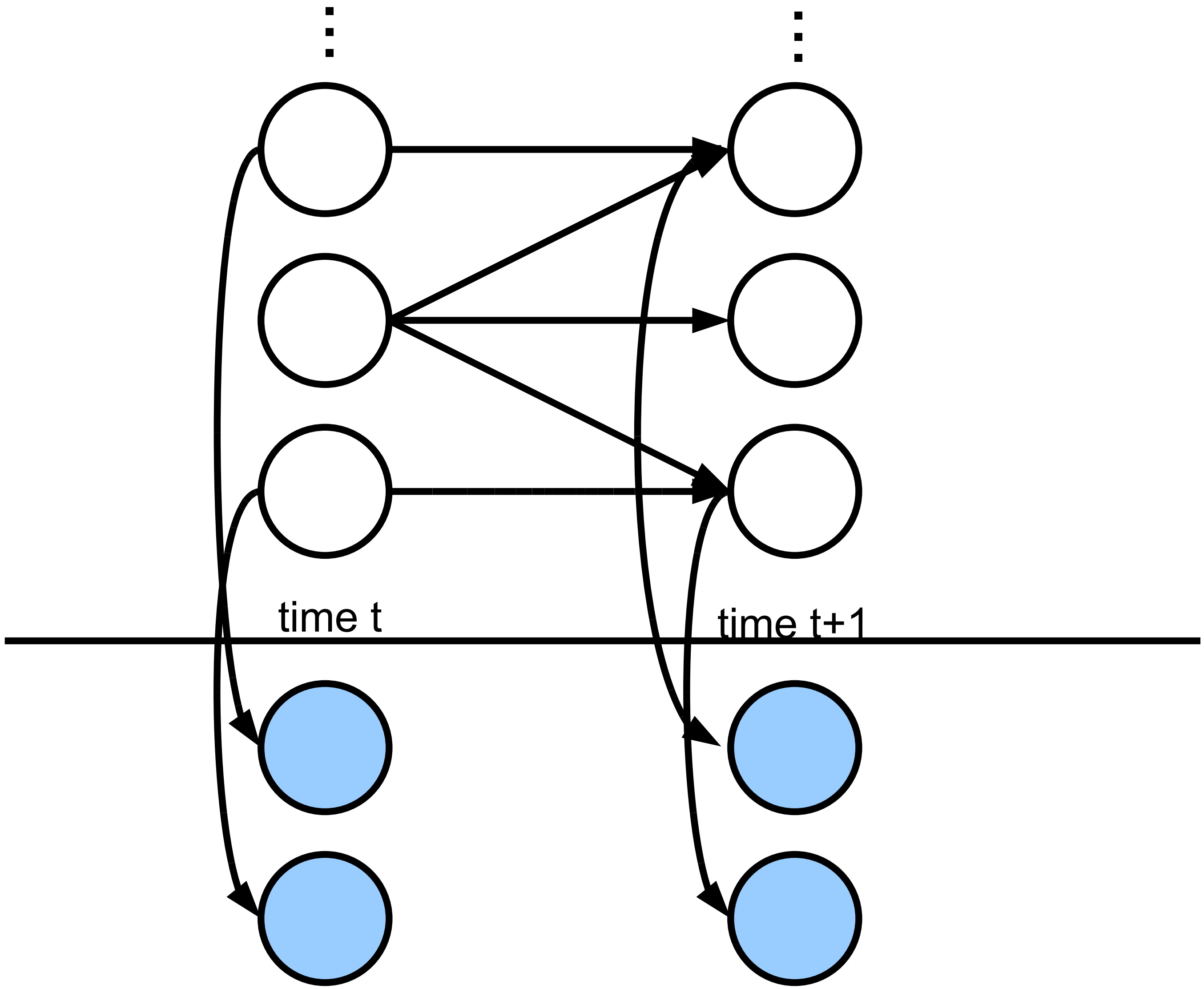
- Introduction: The partially-observable reinforcement learning setting
- Framework: Bayesian reinforcement learning
- Applying nonparametrics:
  - Infinite Partially Observable Markov Decision Processes
  - Infinite State Controllers
  - Infinite Dynamic Bayesian Networks*
- Conclusions and Continuing Work

* Doshi-Velez, Wingate, Tenenbaum, and Roy, ICML 2011

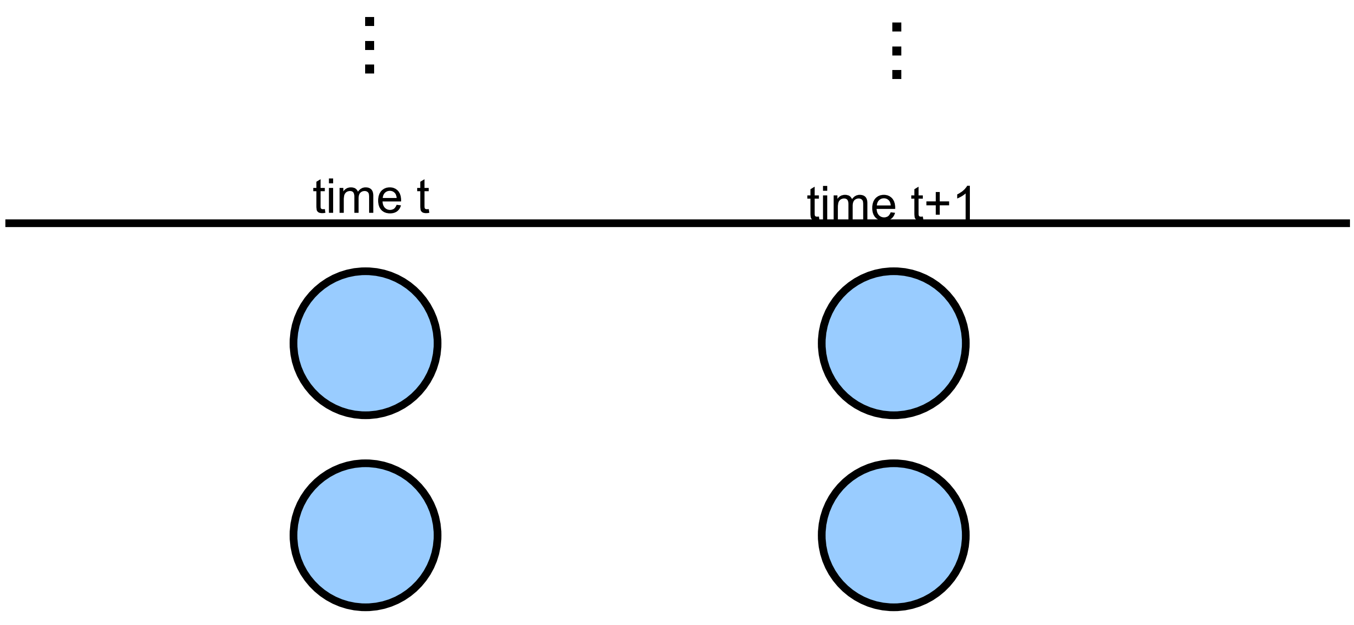
Dynamic Bayesian Networks



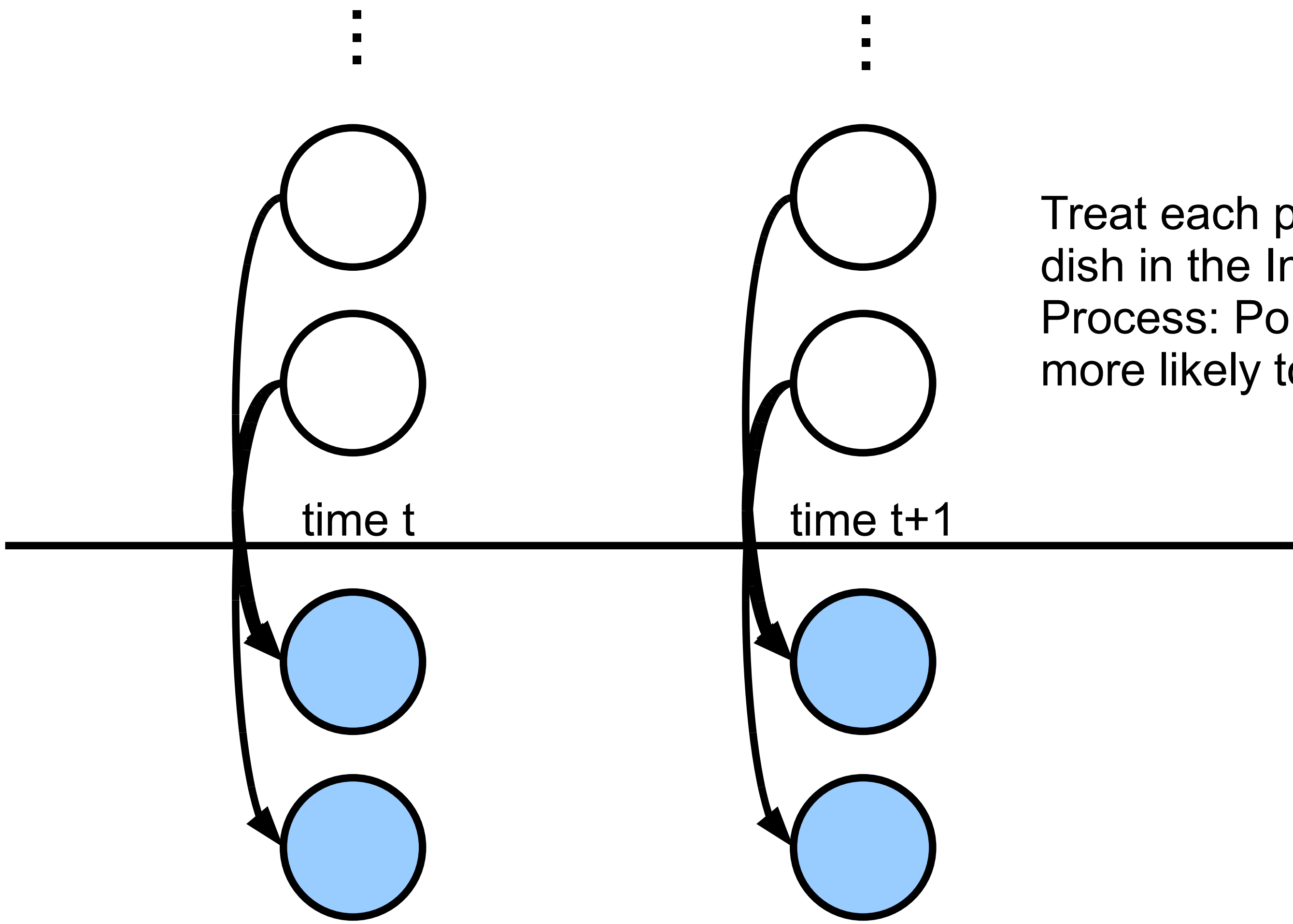
Making it infinite...



The iDBN Generative Process

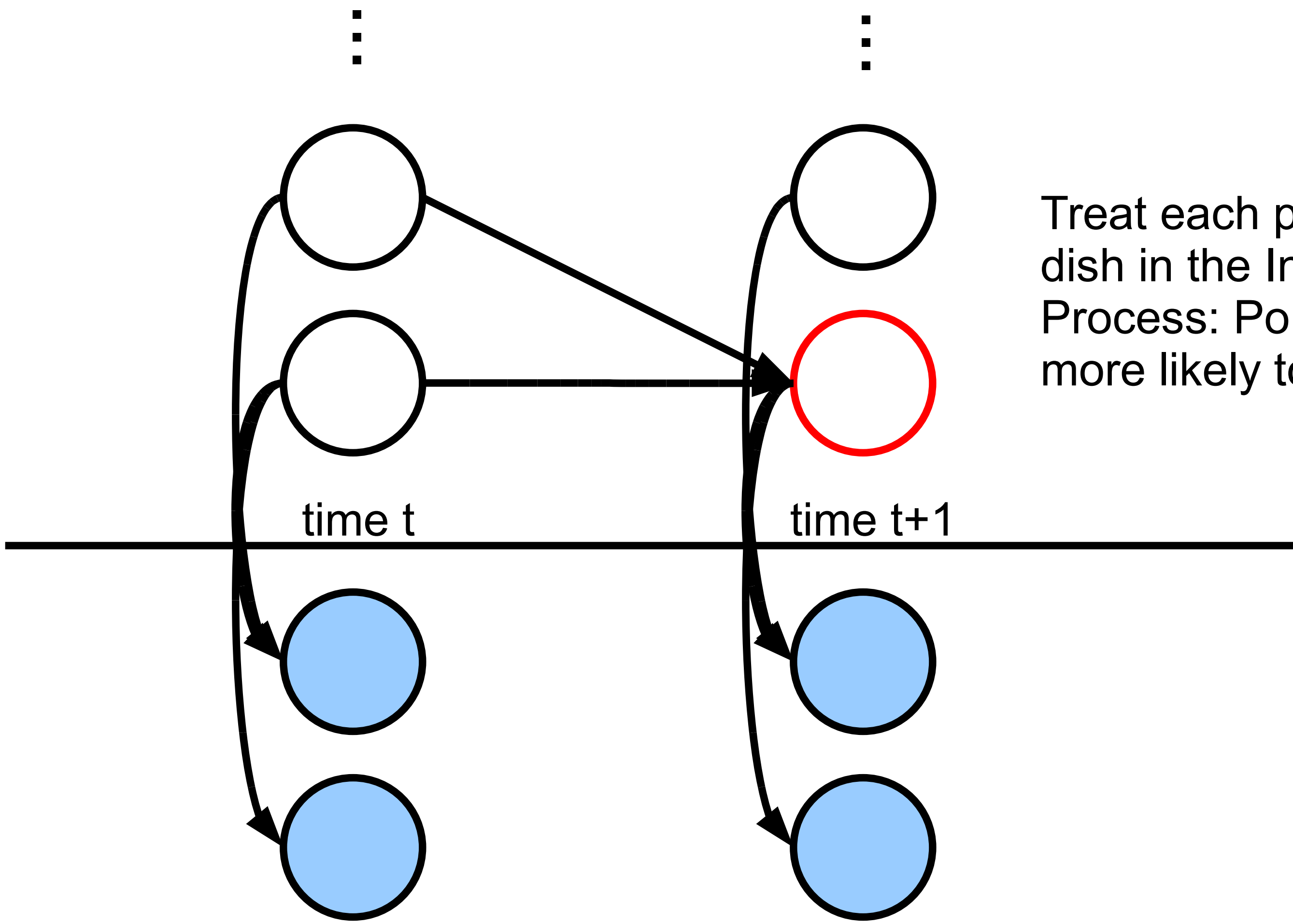


Observed Nodes Choose Parents



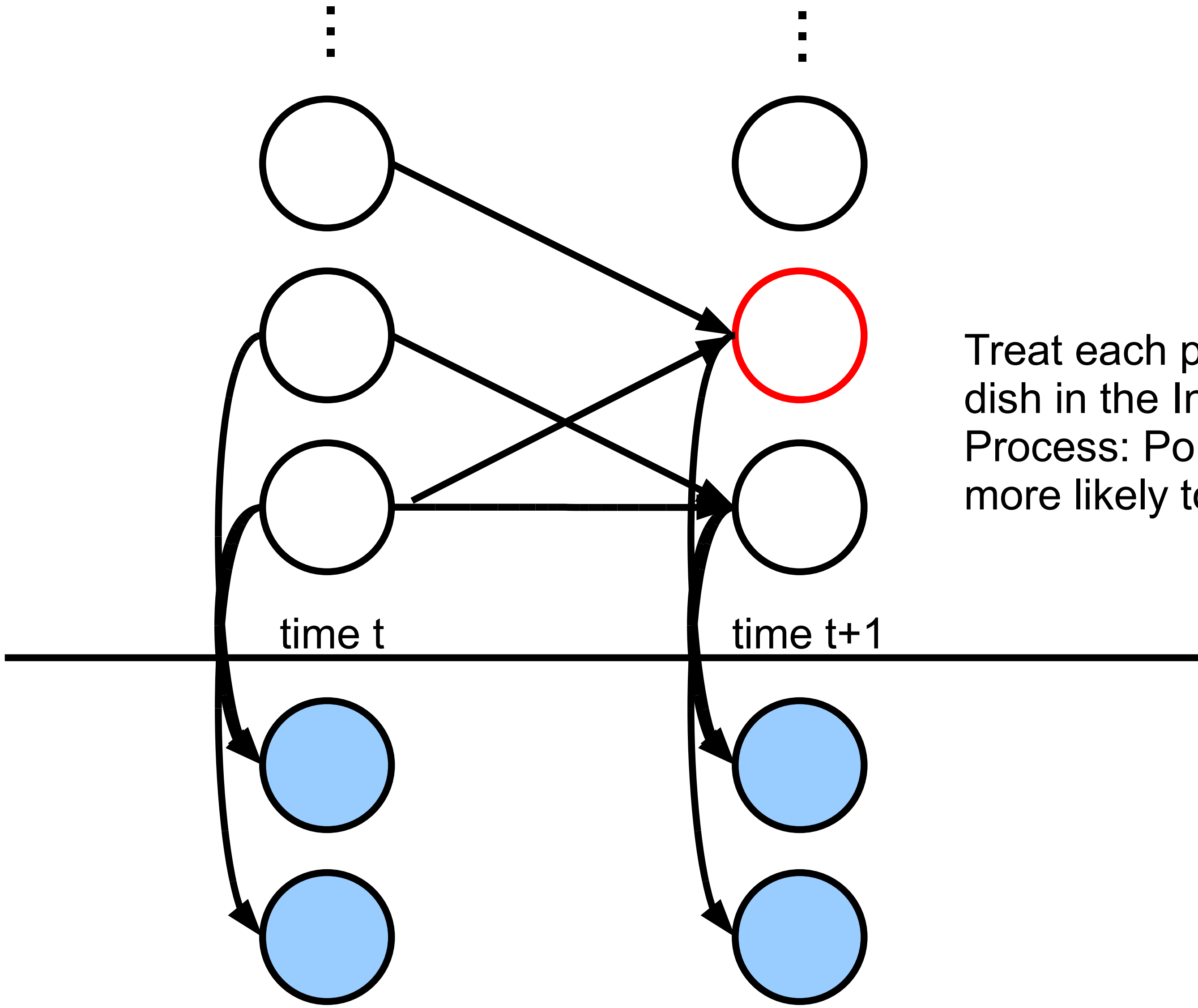
Treat each parent as a dish in the Indian Buffet Process: Popular parents more likely to be chosen.

Hidden Nodes Choose Parents



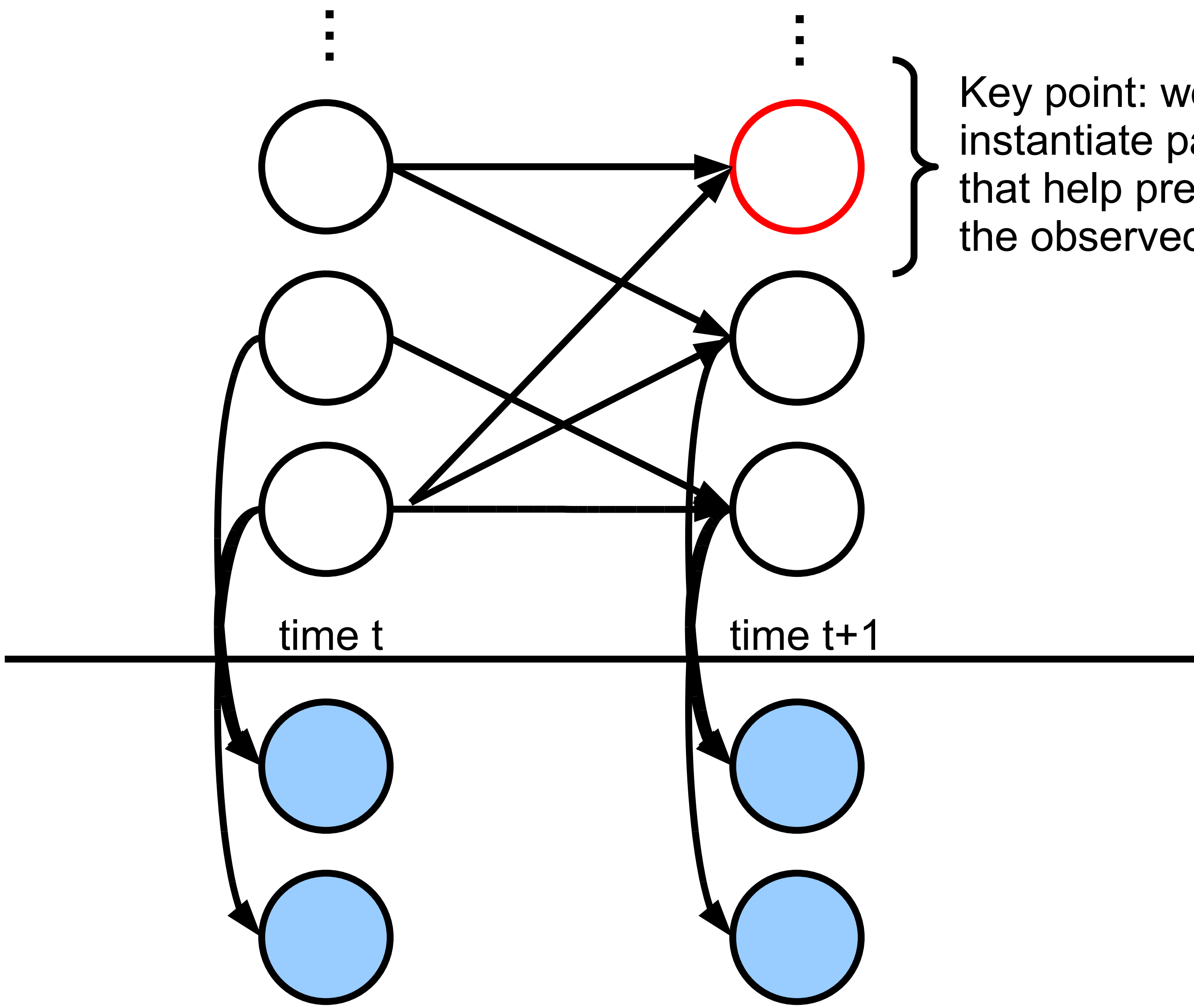
Treat each parent as a dish in the Indian Buffet Process: Popular parents more likely to be chosen.

Hidden Nodes Choose Parents

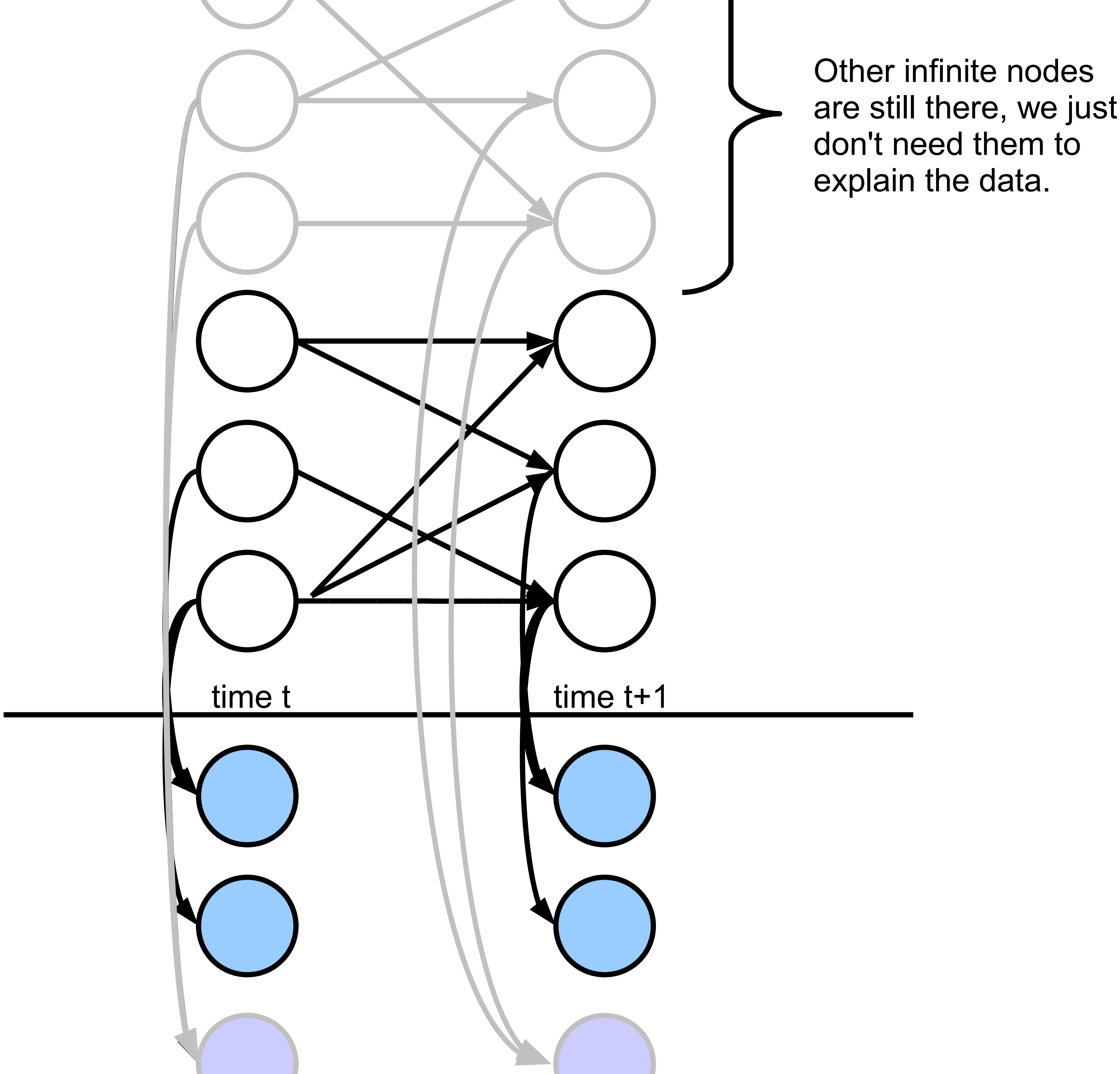


Treat each parent as a dish in the Indian Buffet Process: Popular parents more likely to be chosen.

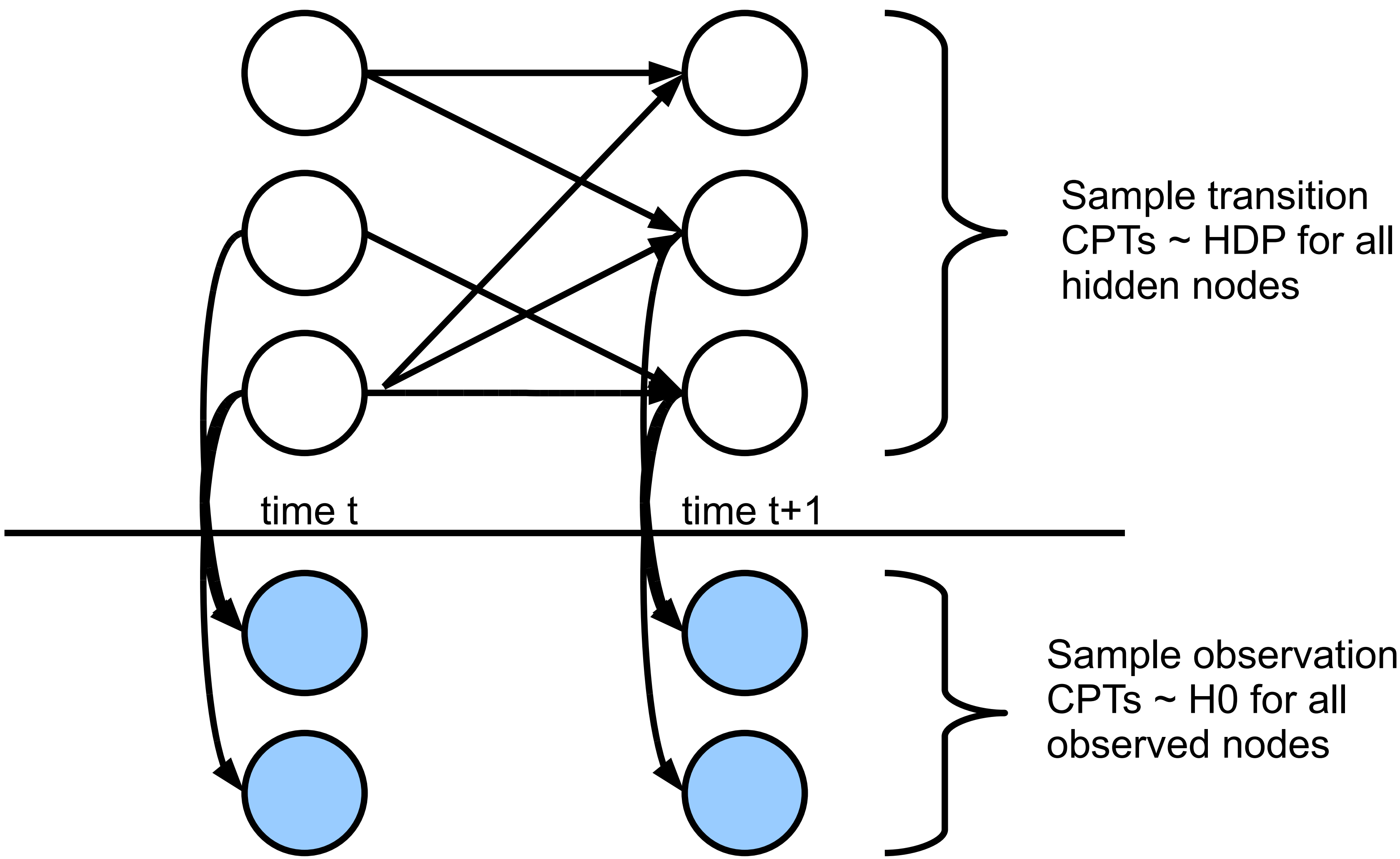
Hidden Nodes Choose Parents



Key point: we only need to instantiate parents for nodes that help predict values for the observed nodes.


Other infinite nodes are still there, we just don't need them to explain the data.
time t
time t+1

Instantiate Parameters



Inference

General Approach: Blocked Gibbs sampling with the usual tricks (tempering, sequential initialization,etc.)

Resample factor-factor connections	$p(P_{a_n} \| P_{a_k}, X, \beta)$	Gibbs sampling
Resample factor-observation connections	$p(P_{a_k} \| P_{a_n}, X, \beta)$	Gibbs sampling
Resample transitions	$p(T \| P_{a_k}, X, \beta)$	Dirichlet-multinomial
Resample observations	$p(\Omega \| P_{a_n}, X, \beta, Y)$	Dirichlet-multinomial
Resample state sequence	$p(X \| P_{a_n}, P_{a_k}, \beta, T, \Omega, Y)$	Factored frontier – Loopy BP
Add / delete factors	$p(P_{a_n} \| P_{a_k}, X, \beta)$	Metropolis-Hastings birth/death

Inference

General Approach: Blocked Gibbs sampling with (tempering, sequential initialization, etc.)

Common to all DBN inference

Resample factor-factor connections	$p(P_{a_n} \| P_{a_k}, X, \beta)$	Gibbs sampling
Resample factor-observation connections	$p(P_{a_k} \| P_{a_n}, X, \beta)$	Gibbs sampling
Resample transitions	$p(T \| P_{a_k}, X, \beta)$	Dirichlet-multinomial
Resample observations	$p(\Omega \| P_{a_n}, X, \beta, Y)$	Dirichlet-multinomial
Resample state sequence	$p(X \| P_{a_n}, P_{a_k}, \beta, T, \Omega, Y)$	Factored frontier – Loopy BP
Add / delete factors	$p(P_{a_n} \| P_{a_k}, X, \beta)$	Metropolis-Hastings birth/death

Inference

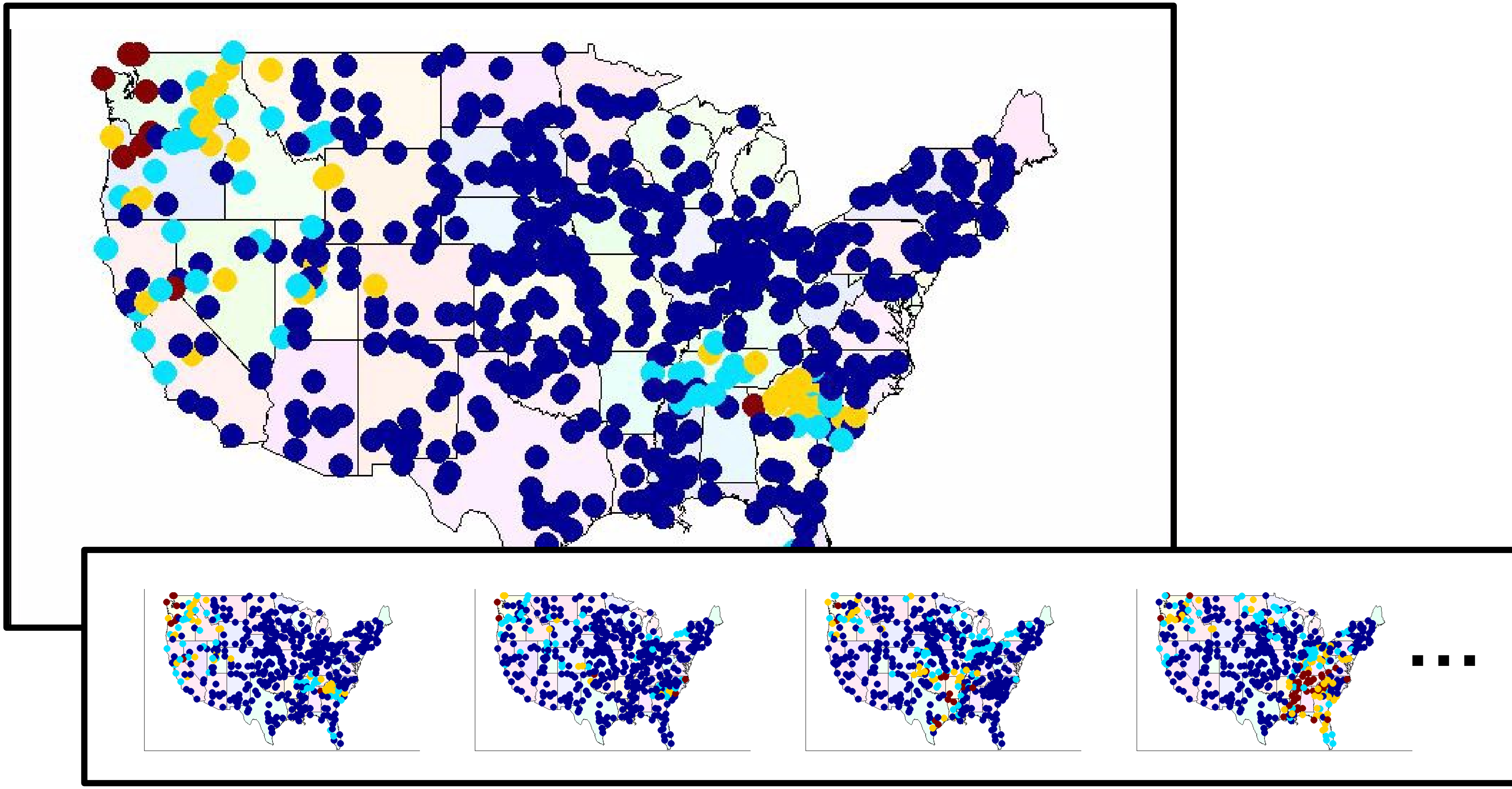
General Approach: Blocked Gibbs sampling with (tempering, sequential initialization, etc.)

Common to all DBN inference

Resample factor-factor connections	$p(P_{a_n} \| P_{a_k}, X, \beta)$	Gibbs sampling
Resample factor-observation connections	$p(P_{a_k} \| P_{a_n}, X, \beta)$	Gibbs sampling
Resample transitions	$p(T \| P_{a_k}, X, \beta)$	Dirichlet-multinomial
Resample observations	$p(\Omega \| P_{a_n}, X, \beta, Y)$	Dirichlet-multinomial
Resample state sequence	$p(X \| P_{a_n}, P_{a_k}, \beta, T, \Omega, Y)$	Factored frontier – Loopy BP
Add / delete factors	$p(P_{a_n} \| P_{a_k}, X, \beta)$	Metropolis-Hastings birth/death

Specific to iDBN
only 5% computational overhead!

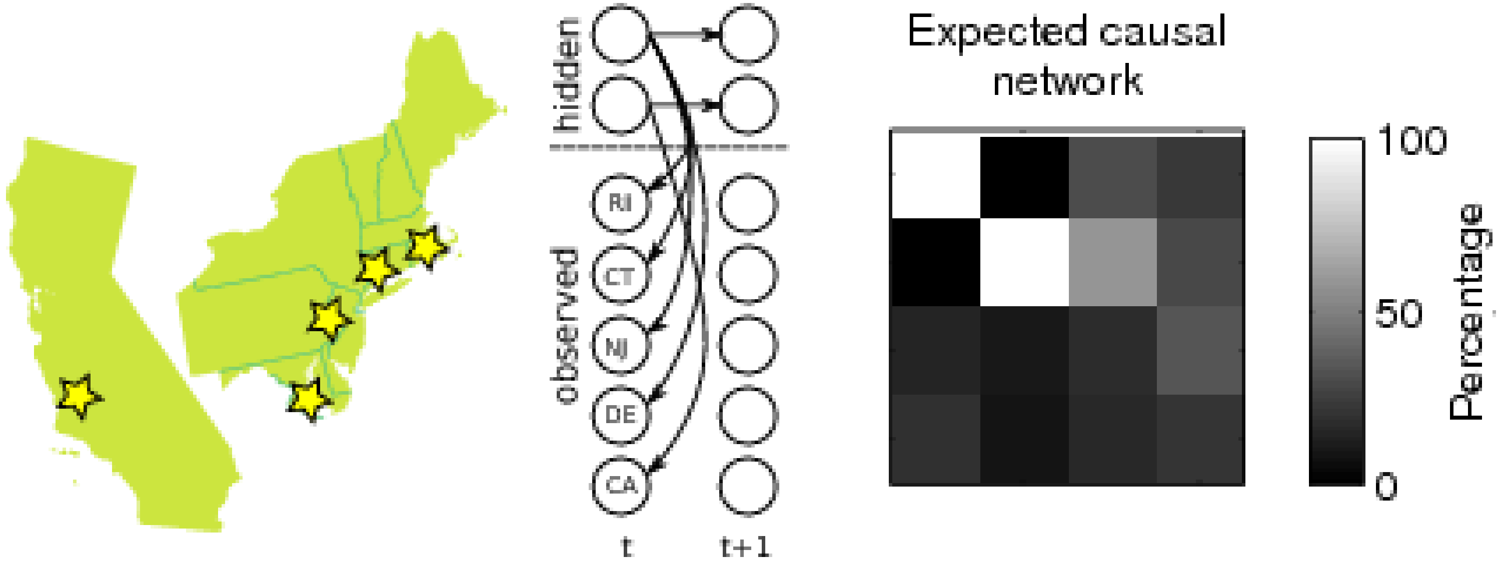
Example: Weather Data



Time series of US precipitation patterns...

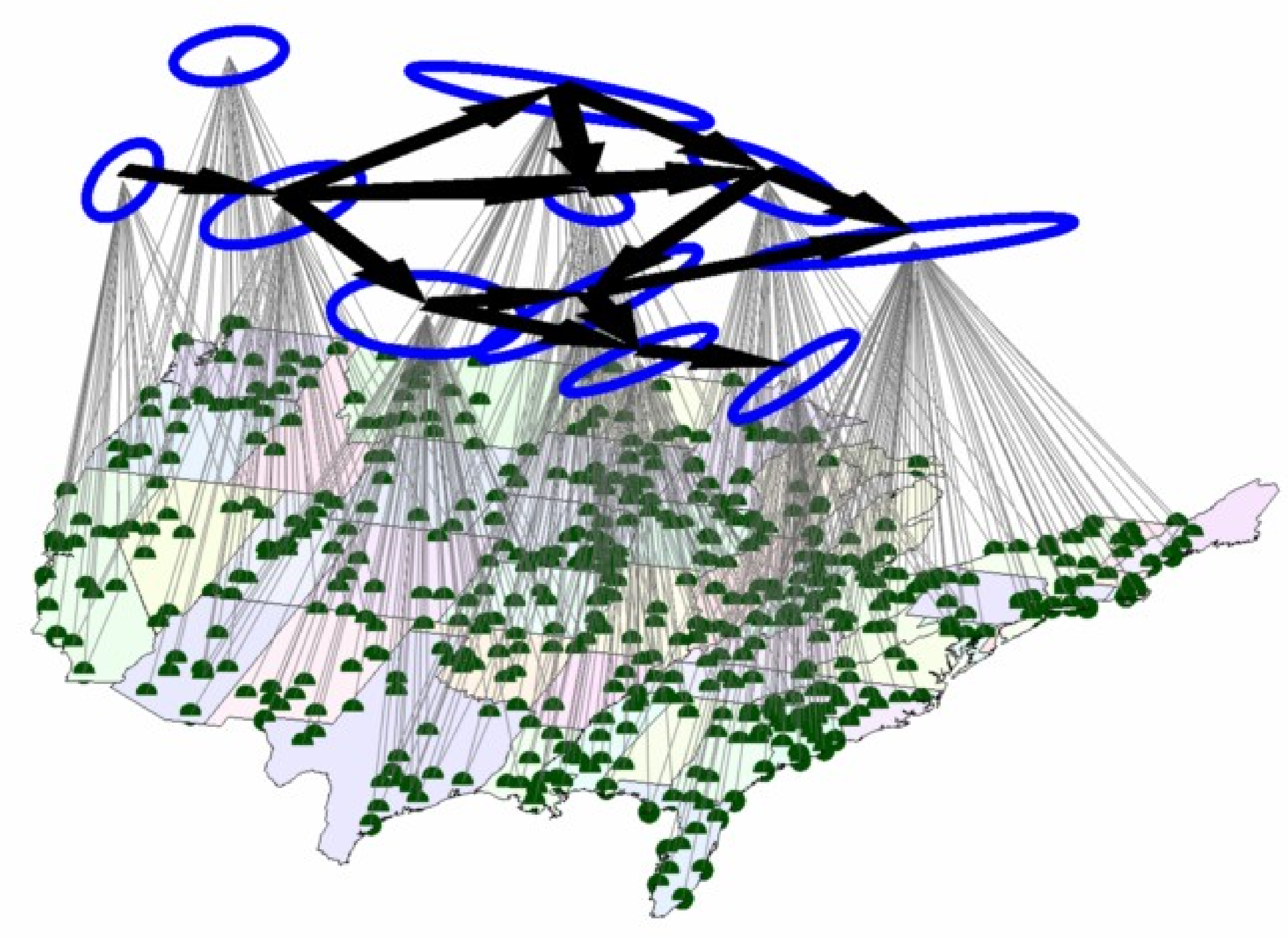
Weather Example: Small Dataset

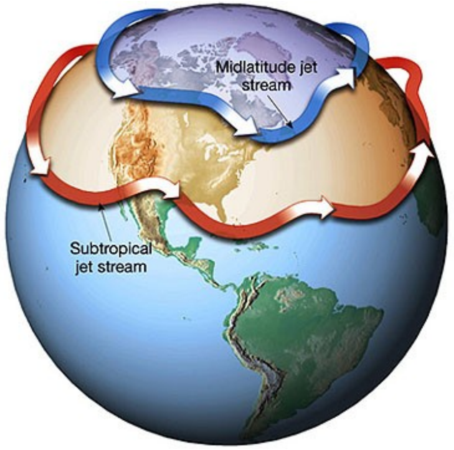
A model with just five locations quickly separates the east cost and the west coast data points.



Weather Example: Full Dataset

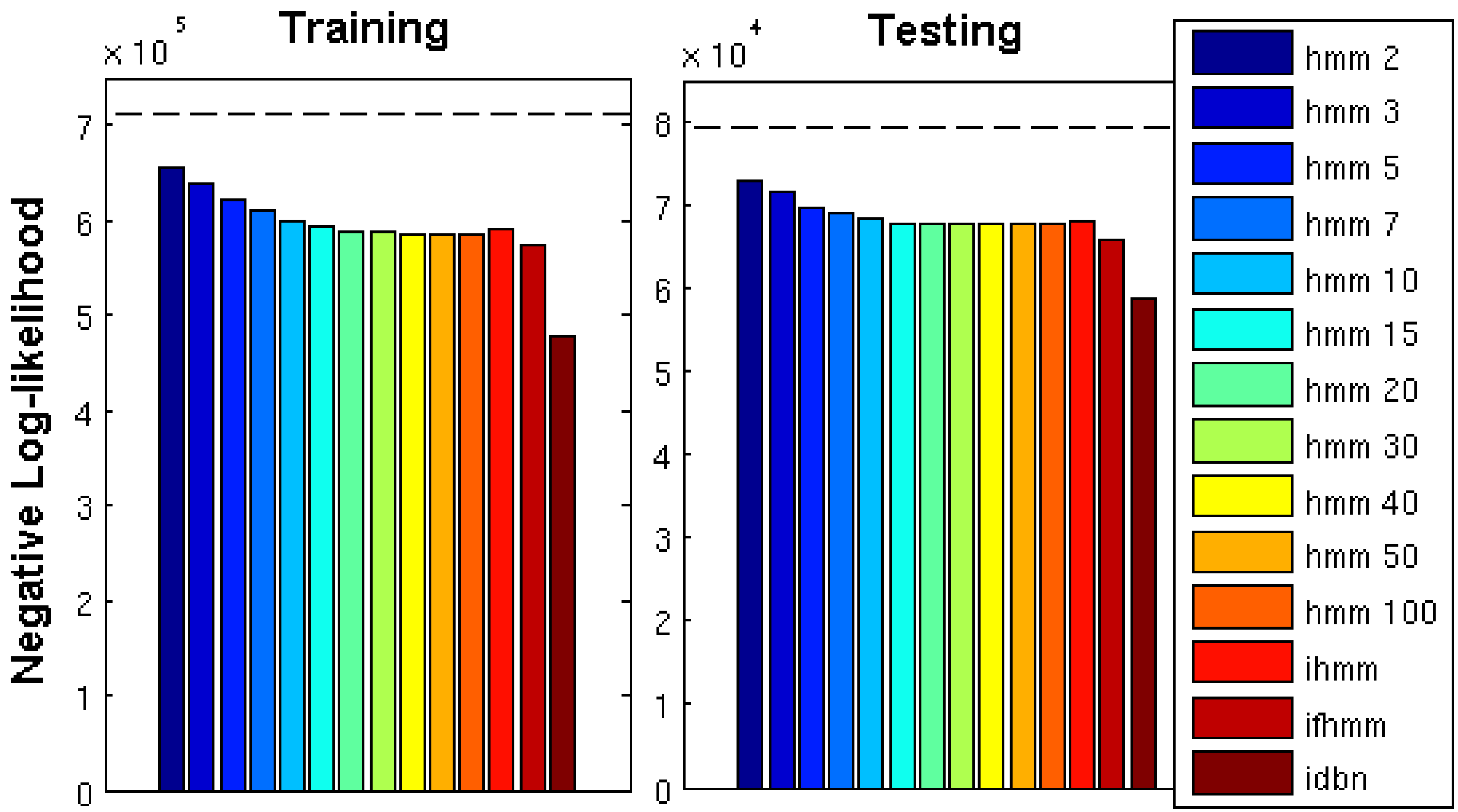
On the full dataset, we get regional factors with a general west-to-east pattern (the jet-stream).





Weather example: Full Dataset

Training and test performance (lower is better)



Outline

- Introduction: The partially-observable reinforcement learning setting
- Framework: Bayesian reinforcement learning
- Applying nonparametrics:
  - Infinite Partially Observable Markov Decision Processes
  - Infinite State Controllers
  - Infinite Dynamic Bayesian Networks
- Conclusions and Continuing Work

When should we use this?

(and what are the limitations?)

- Predictive accuracy is the priority.

  (learned representations aren't always interpretable, and they are not optimized for maximizing rewards)

- When the data is limited or fundamentally sparse... otherwise a history-based approach might be better.

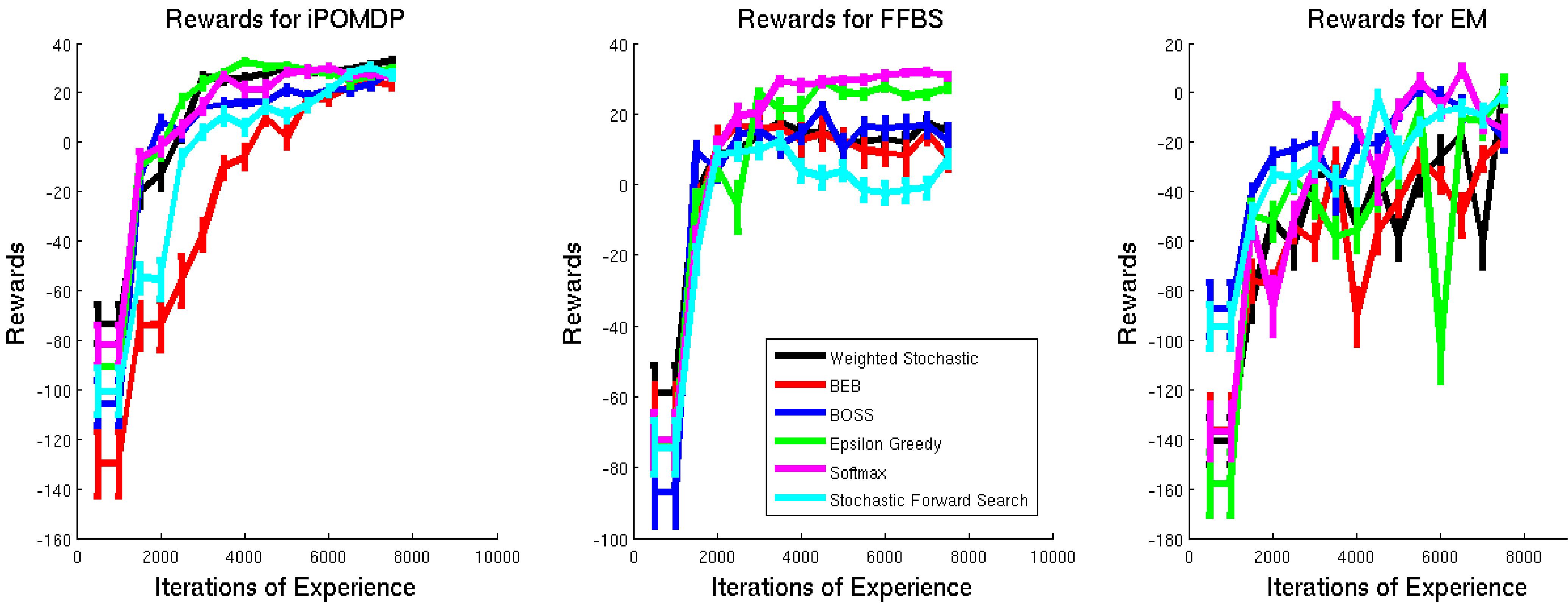
  (most reasonable methods perform well with lots of data, and Bayesian methods require more computation)

- When the "true" model is poorly understood... otherwise use calibration and system identification.

  (current priors are very general, not easy to combine with detailed system or parameter knowledge)

Continuing Work

- Action-selection: when do different strategies matter?



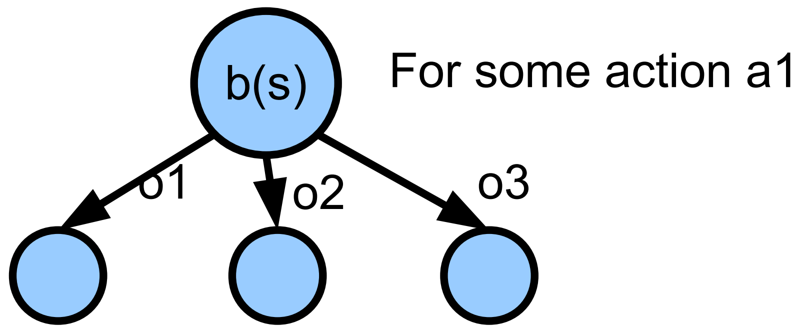
- Bayesian nonparametrics for history-based approaches: improving probabilistic-deterministic infinite automata
- Models that match realworld properties.

Summary

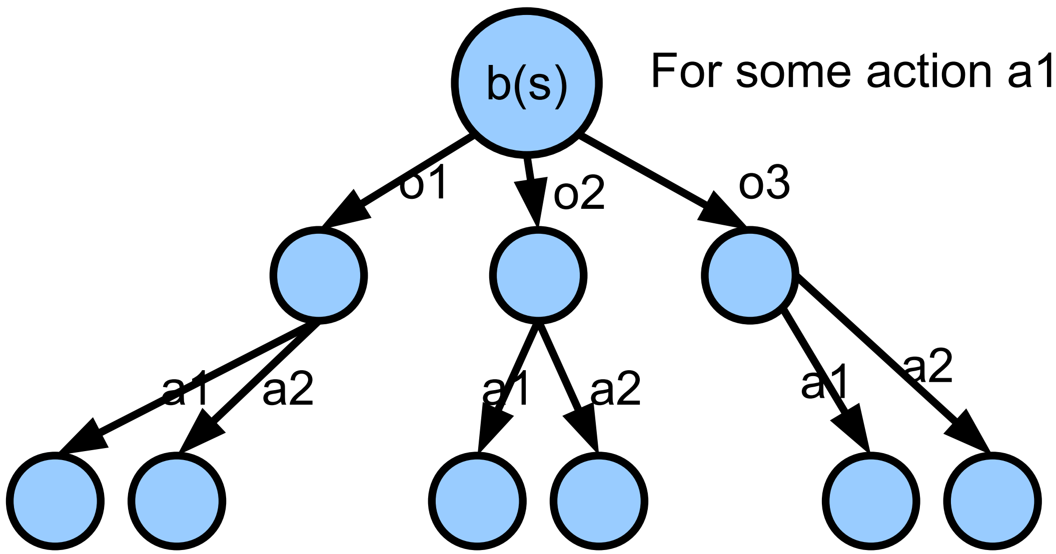
In this thesis, we introduced a novel approach to learning hidden-variable representations for partially-observable reinforcement learning using Bayesian nonparametric statistics. This approach allows for

- The representation to scale in sophistication with the complexity in the data
- Tracking uncertainty in the representation
- Expert trajectories to be incorporated
- Complex causal structures to be learned

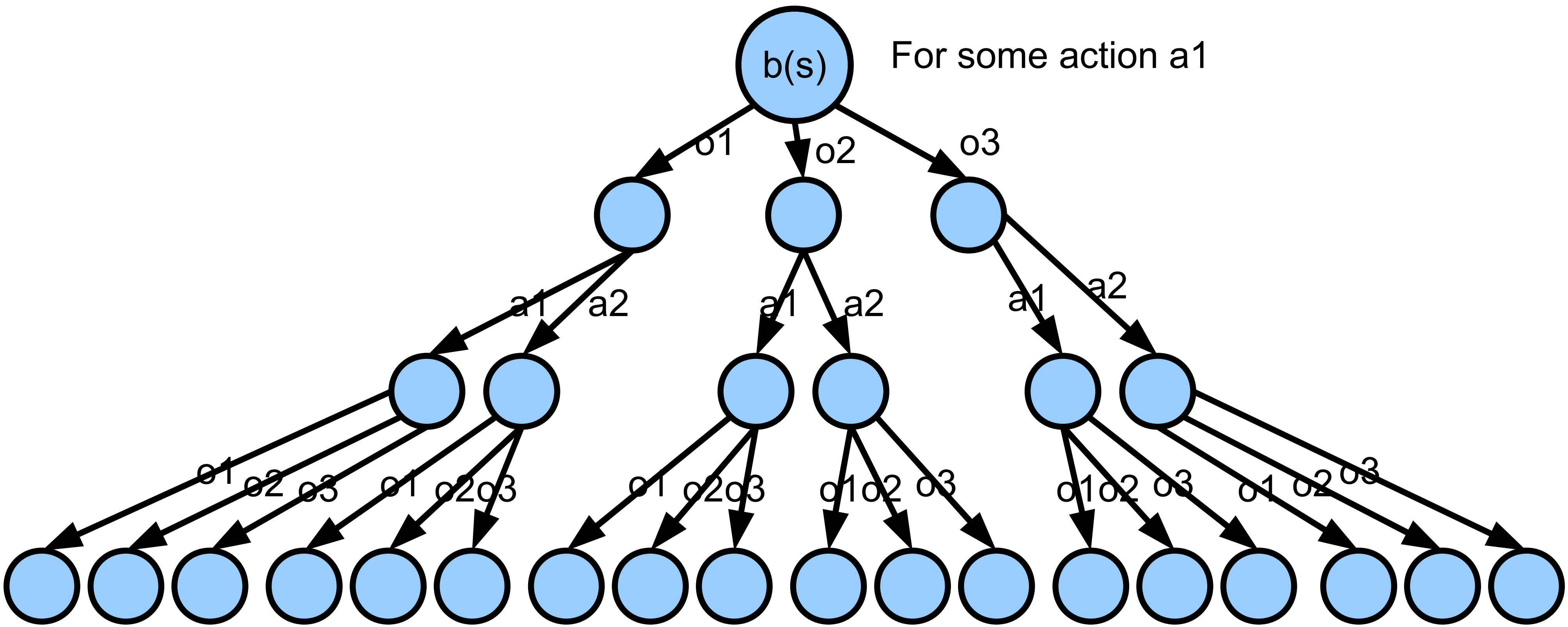
Standard Forward-Search to Determine the Value of an Action:



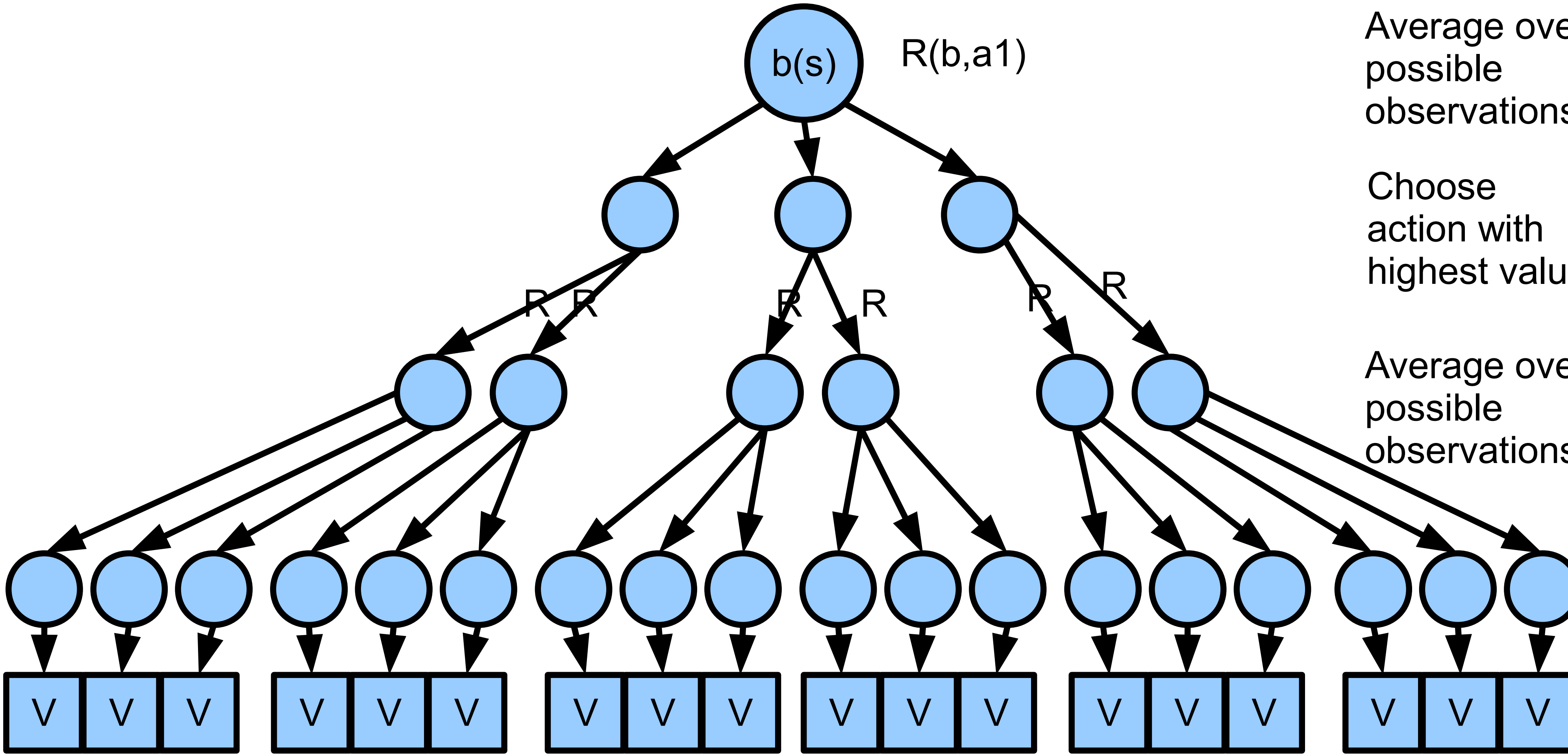
Consider what actions are possible after those observations ...



... and what observations are possible after those actions ...



Use highest-value branches to determine the action's value

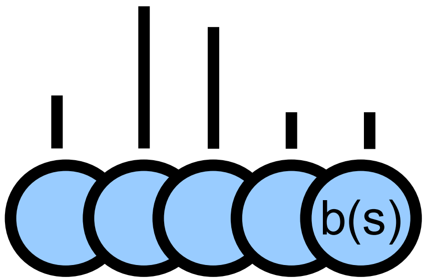


Average over possible observations

Choose action with highest value

Average over possible observations

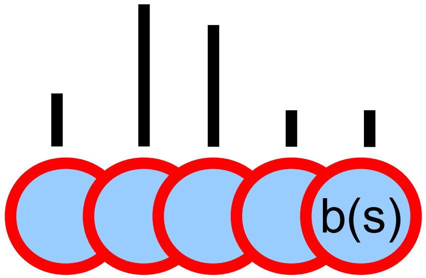
Forward-Search in Model Space



For some action a_1

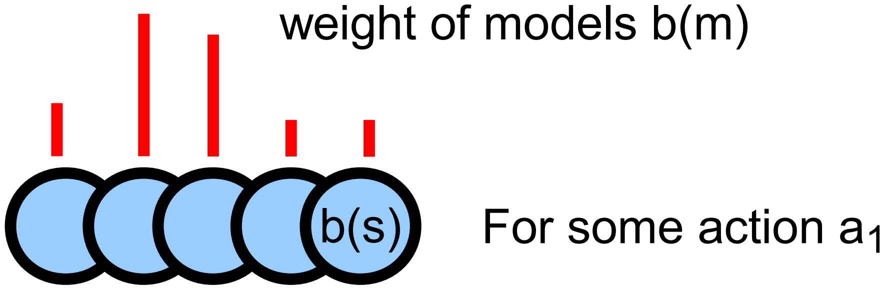
Forward-Search in Model Space

models

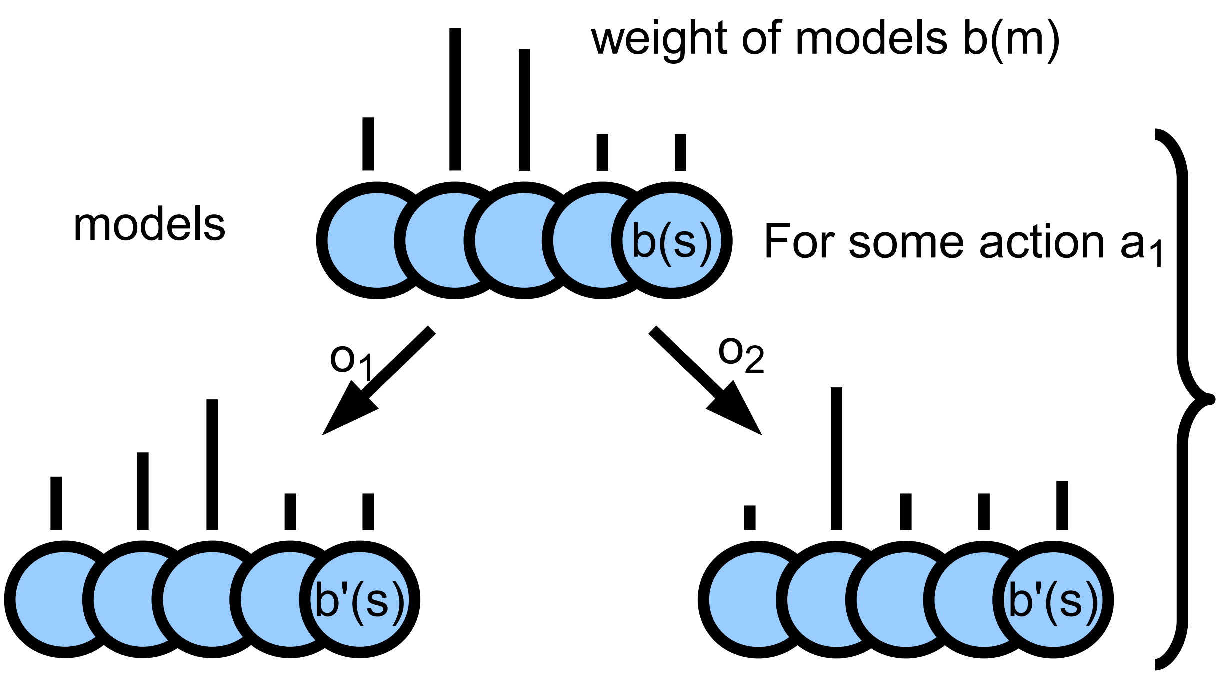


For some action a_1

Forward-Search in Model Space

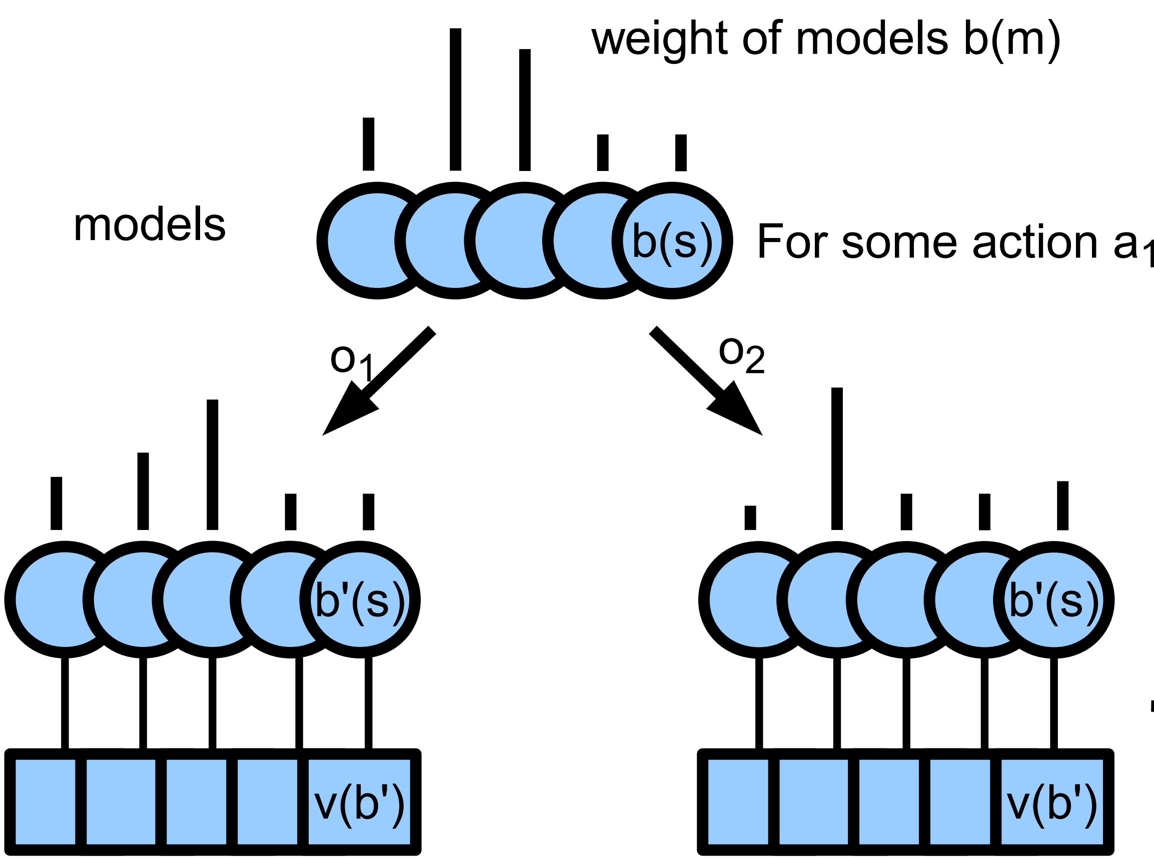


Forward-Search in Model Space (cartoon for a single action)



Each model updates its belief b(s) and weights over models b(m) also change.

Forward-Search in Model Space

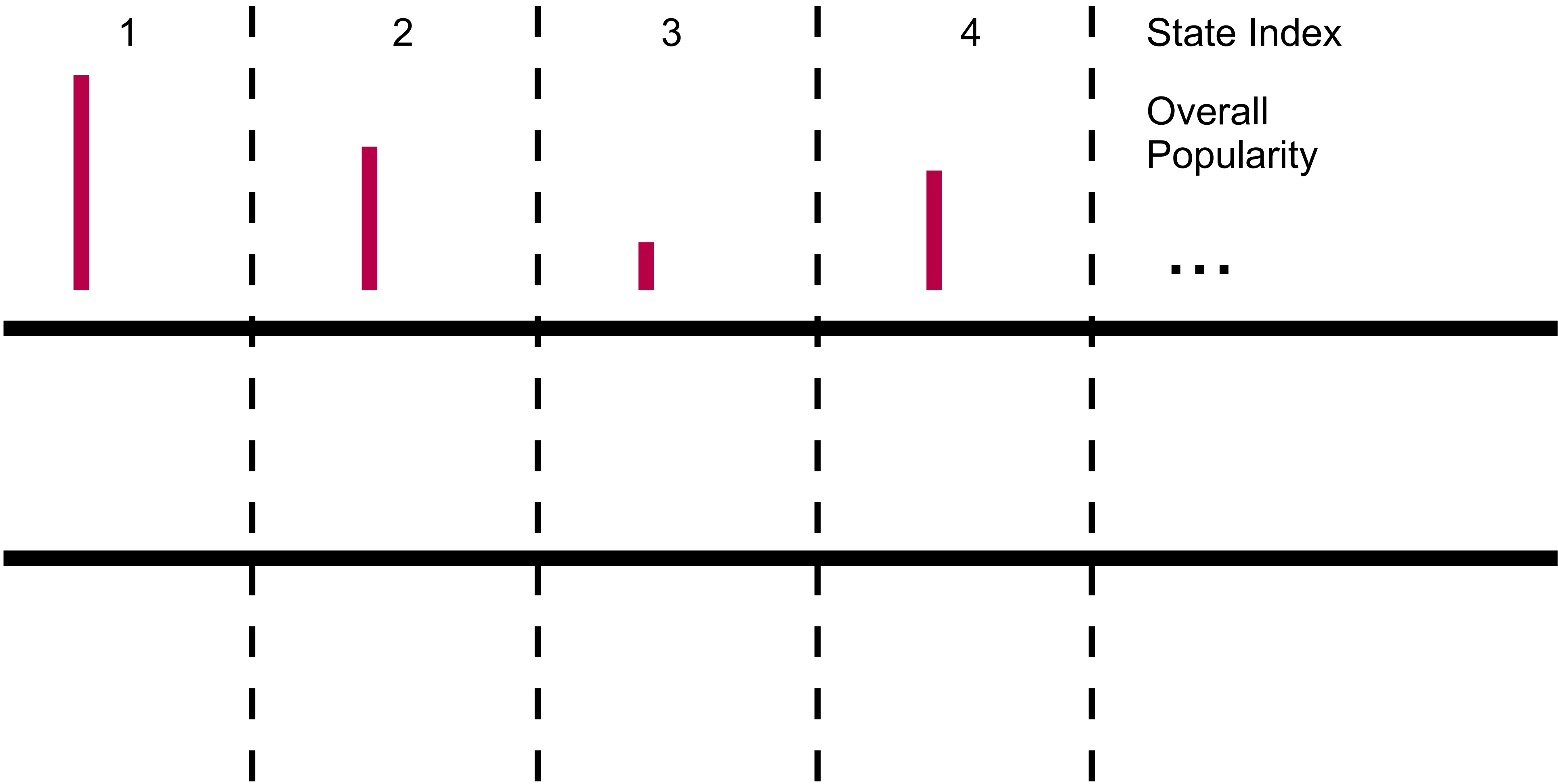


Each model updates its belief b(s) and weights over models b(m) also change..

Leaves use values of beliefs for each model; sampled models are **small, quick to solve.**

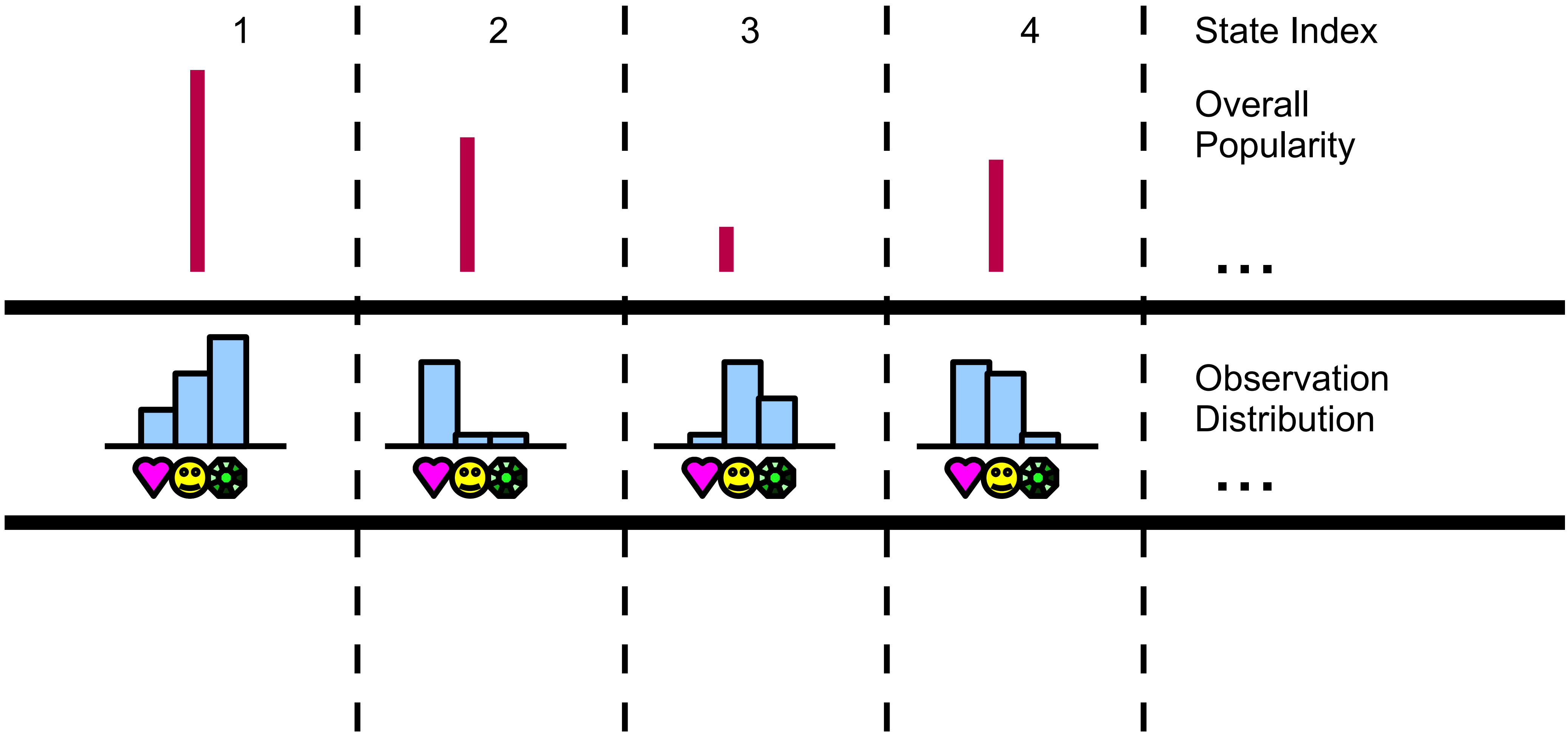
Generative Process

First, sample overall popularities, observation and reward distributions for each state.



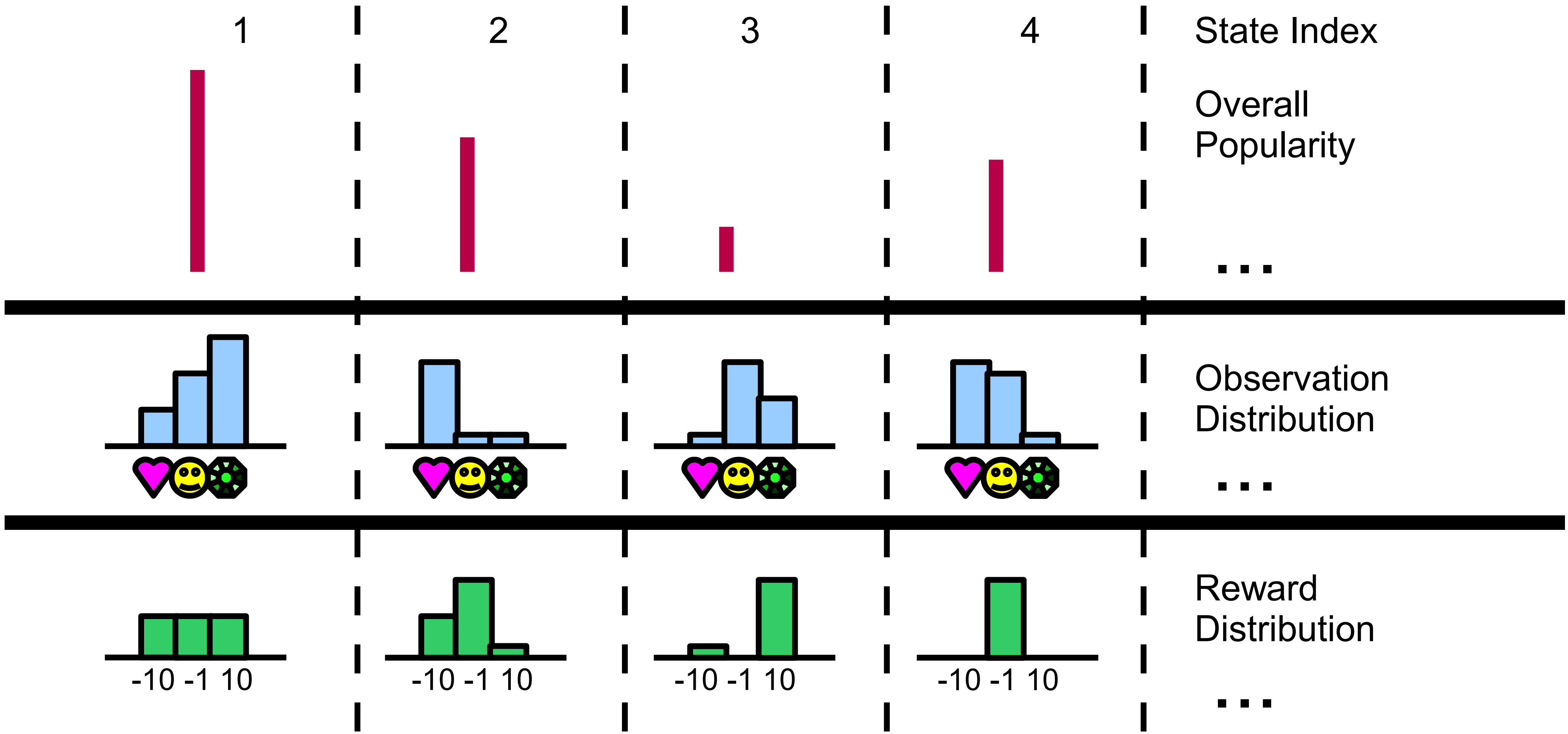
Generative Process

First, sample overall popularities, observation and reward distributions for each state-action.



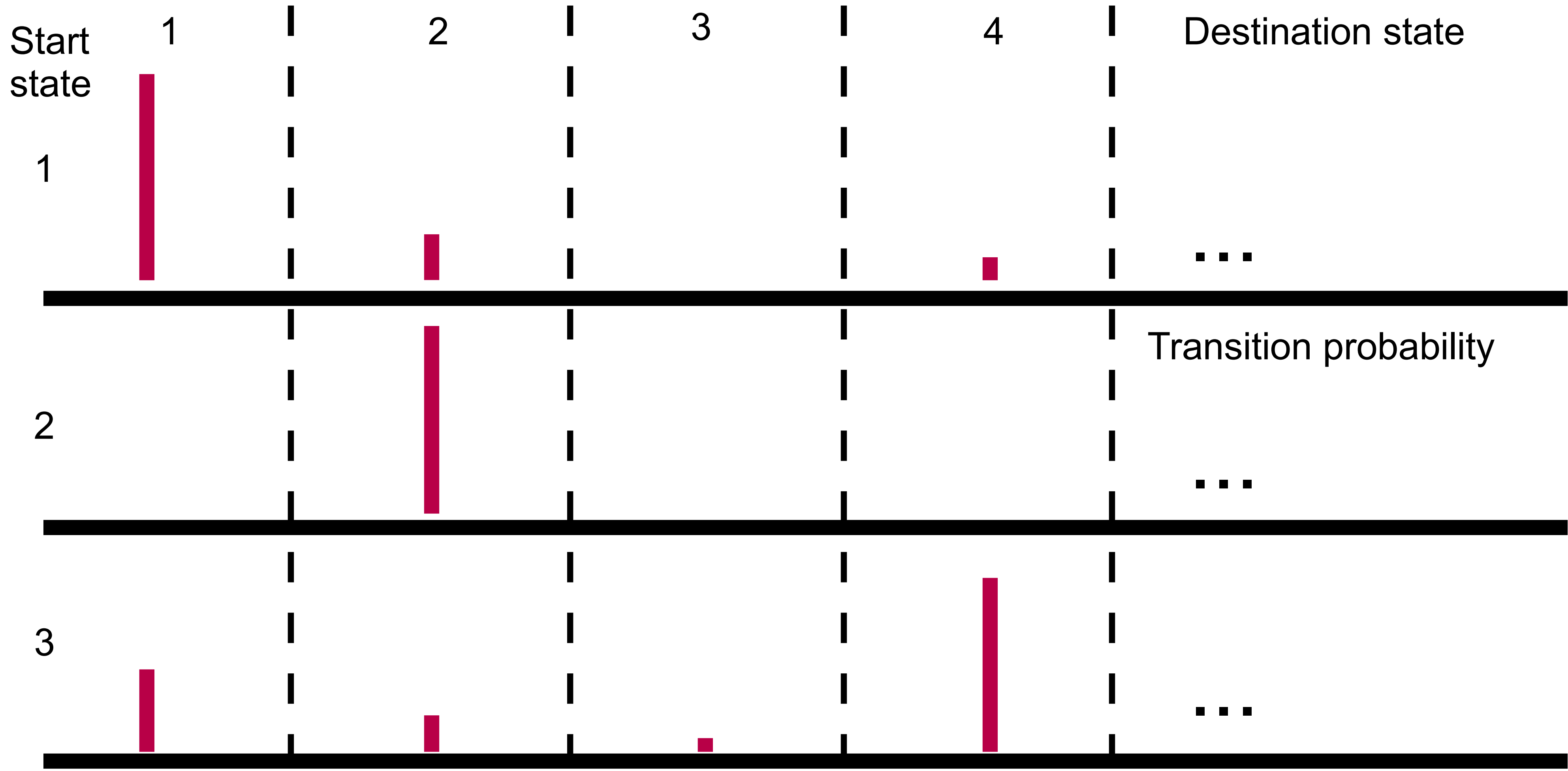
Generative Process

First, sample overall popularities, observation and reward distributions for each state-action.



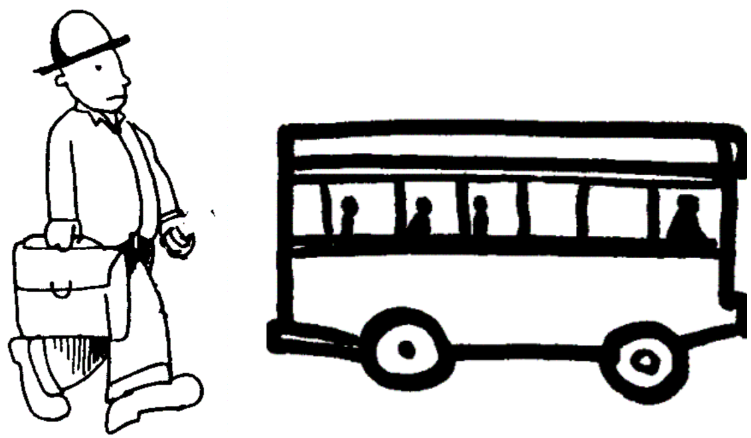
Generative Process

For each action, sample transition matrix using the state popularities as a base distribution.



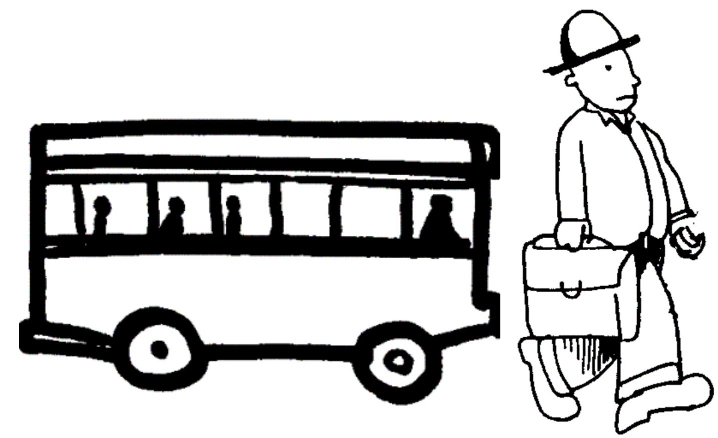
Thought Example: Ride or Walk?

o



Bus comes 90% of the time,
though sometimes 5 minutes late

B

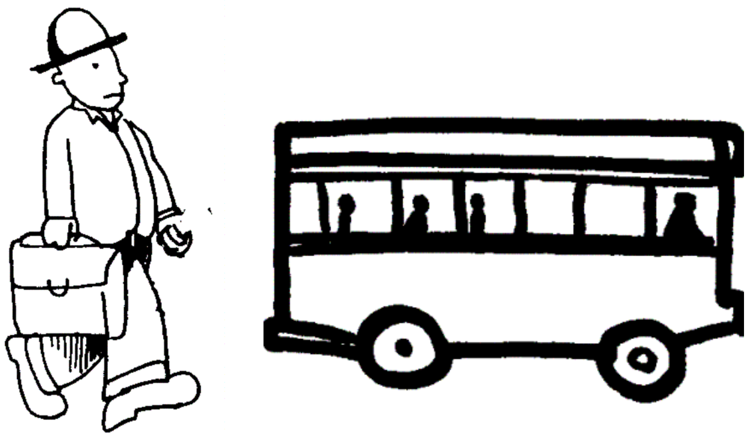


Bus comes 50% of the time,
though sometimes 5 minutes late

Suppose we initially think that each scenario is equally likely.

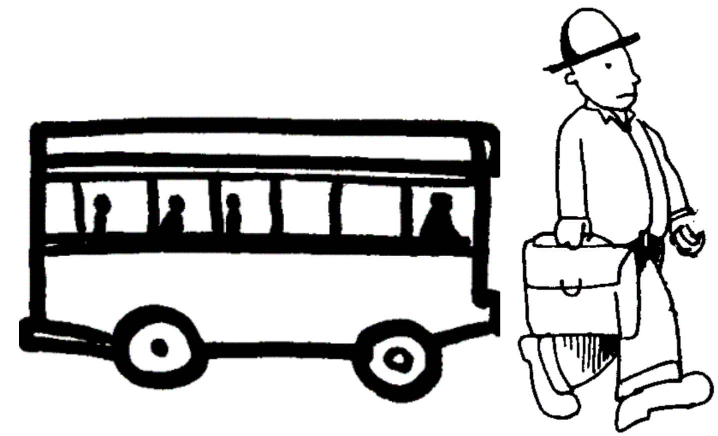
We gather some data...

o



Bus comes 90% of the time,
though sometimes 5 minutes late

B



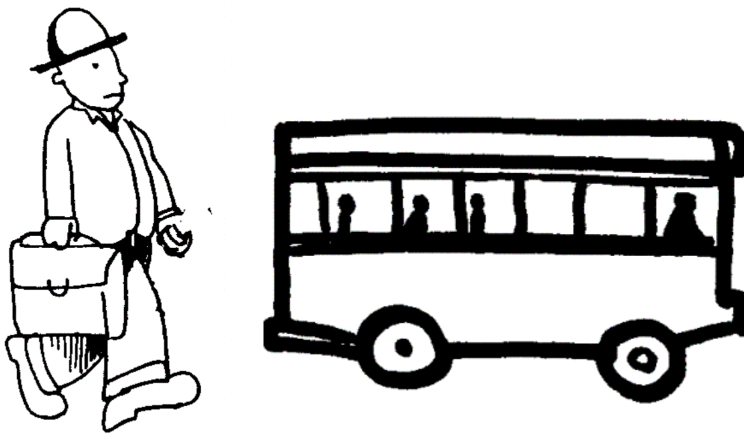
Bus comes 50% of the time,
though sometimes 5 minutes late

data: Bus come 1/2 times

It's now the time for the bus to arrive,
but it's not here. What do you do?

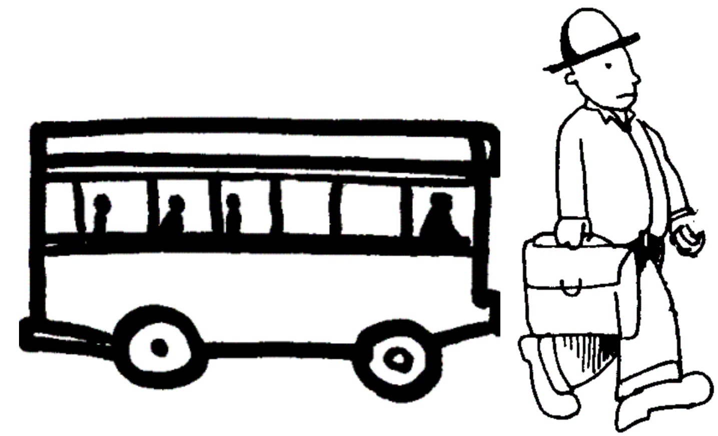
Compute the posterior on scenarios.

o



Bus comes 90% of the time,
though sometimes 5 minutes late

B

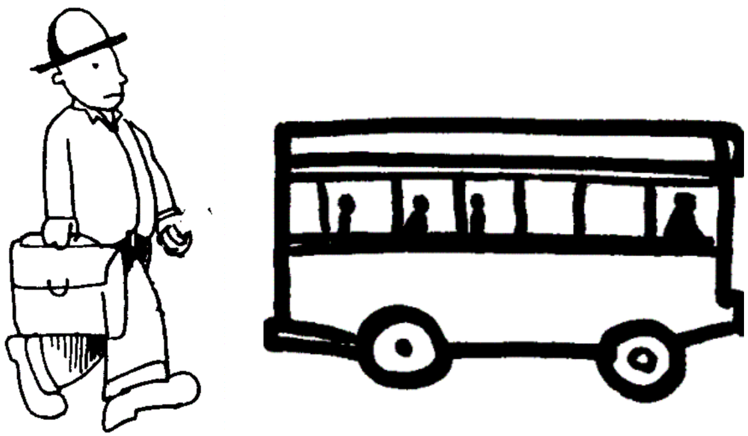


Bus comes 50% of the time,
though sometimes 5 minutes late

Bayesian reasoning tells us
B is ~3 times more likely than o

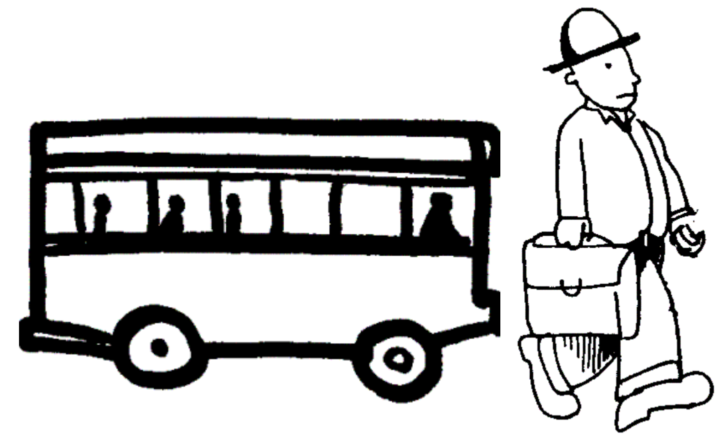
and decide if we're sure enough:

o



Bus comes 90% of the time,
though sometimes 5 minutes late

B



Bus comes 50% of the time,
though sometimes 5 minutes late

Bayesian reasoning tells us
B is ~3 times more likely than o
... but if we really prefer the bus,
we still might want more data.

Results on Some Standard Problems

Metric	States	
Problem	True	iPOMDP
Tiger	2	2.1
Shuttle	8	2.1
Network	7	4.36
Gridworld	26	7.36

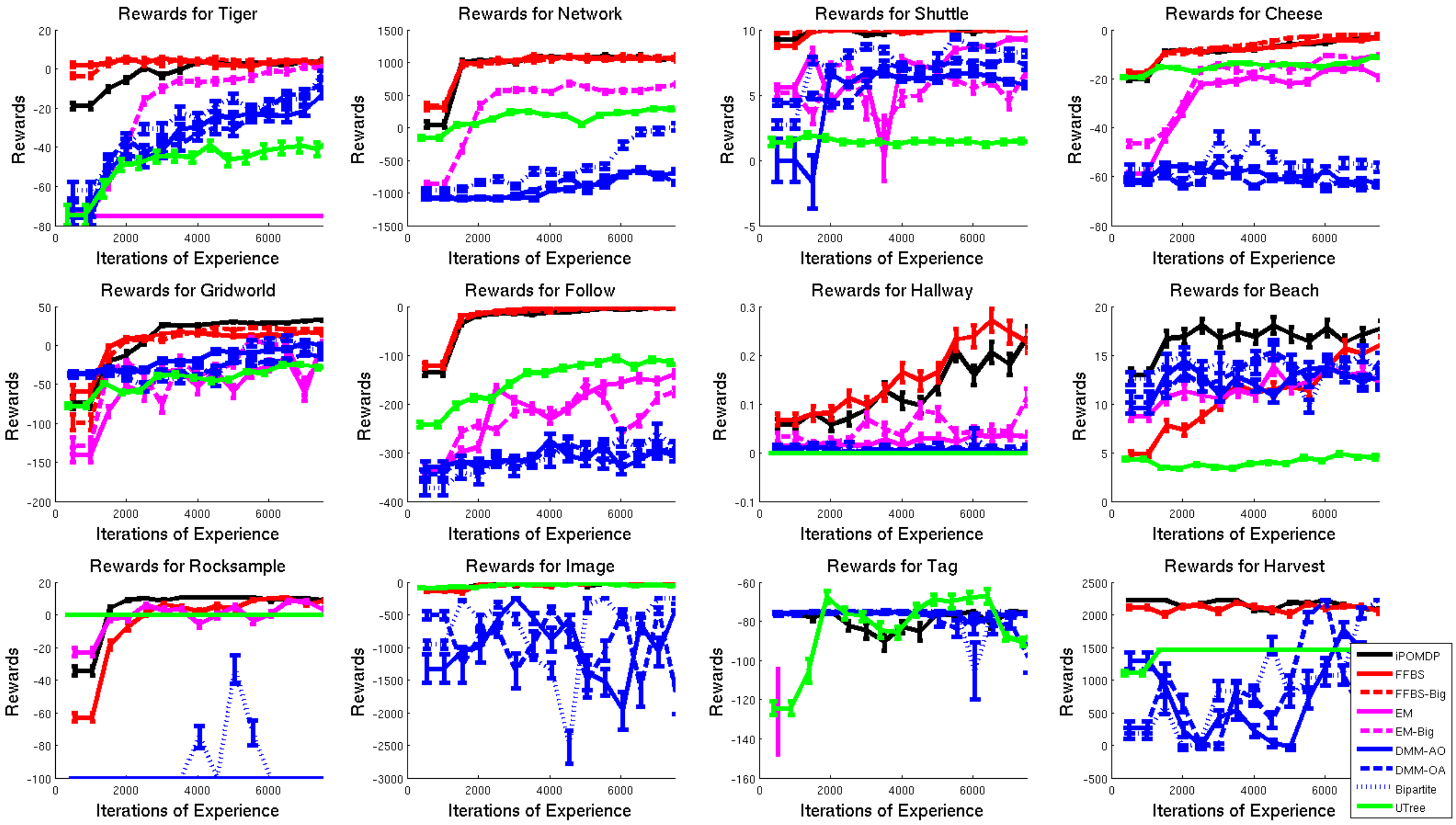
Results on Some Standard Problems

Metric	States		Relative Training Time		
Problem	True	iPOMDP	EM	FFBS	FFBS-big
Tiger	2	2.1	0.41	0.70	1.50
Shuttle	8	2.1	1.82	1.02	3.56
Network	7	4.36	1.56	1.09	4.82
Gridworld	26	7.36	3.57	2.48	59.1

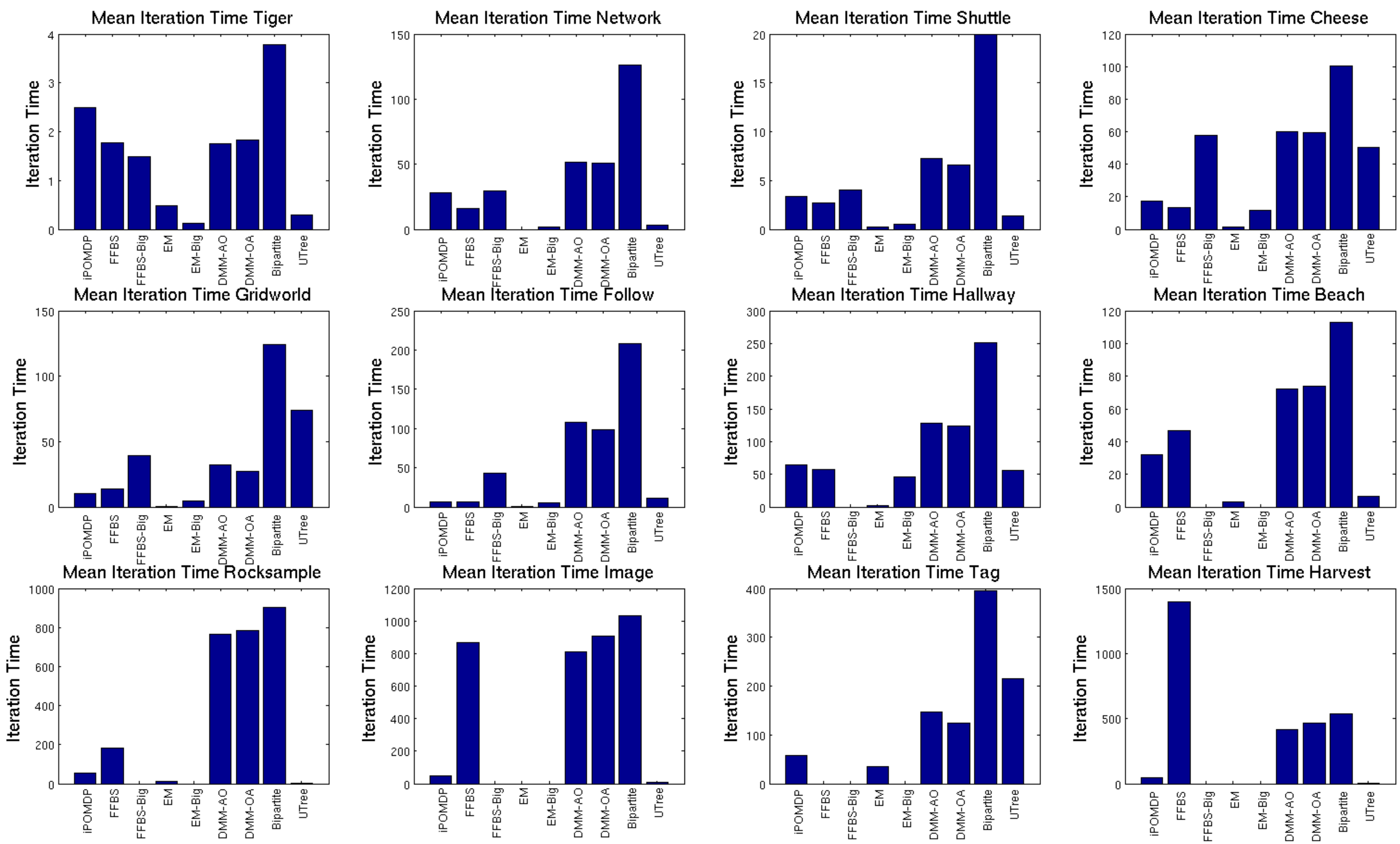
Results on Some Standard Problems

Metric	States		Relative Training Time			Test Performance			
Problem	True	iPOMDP	EM	FFBS	FFBS-big	EM	FFBS	FFBS-big	iPOMDP
Tiger	2	2.1	0.41	0.70	1.50	-277	0.49	4.24	4.06
Shuttle	8	2.1	1.82	1.02	3.56	10	10	10	10
Network	7	4.36	1.56	1.09	4.82	1857	7267	6843	6508
Gridworld	26	7.36	3.57	2.48	59.1	-25	-51	-67	-13

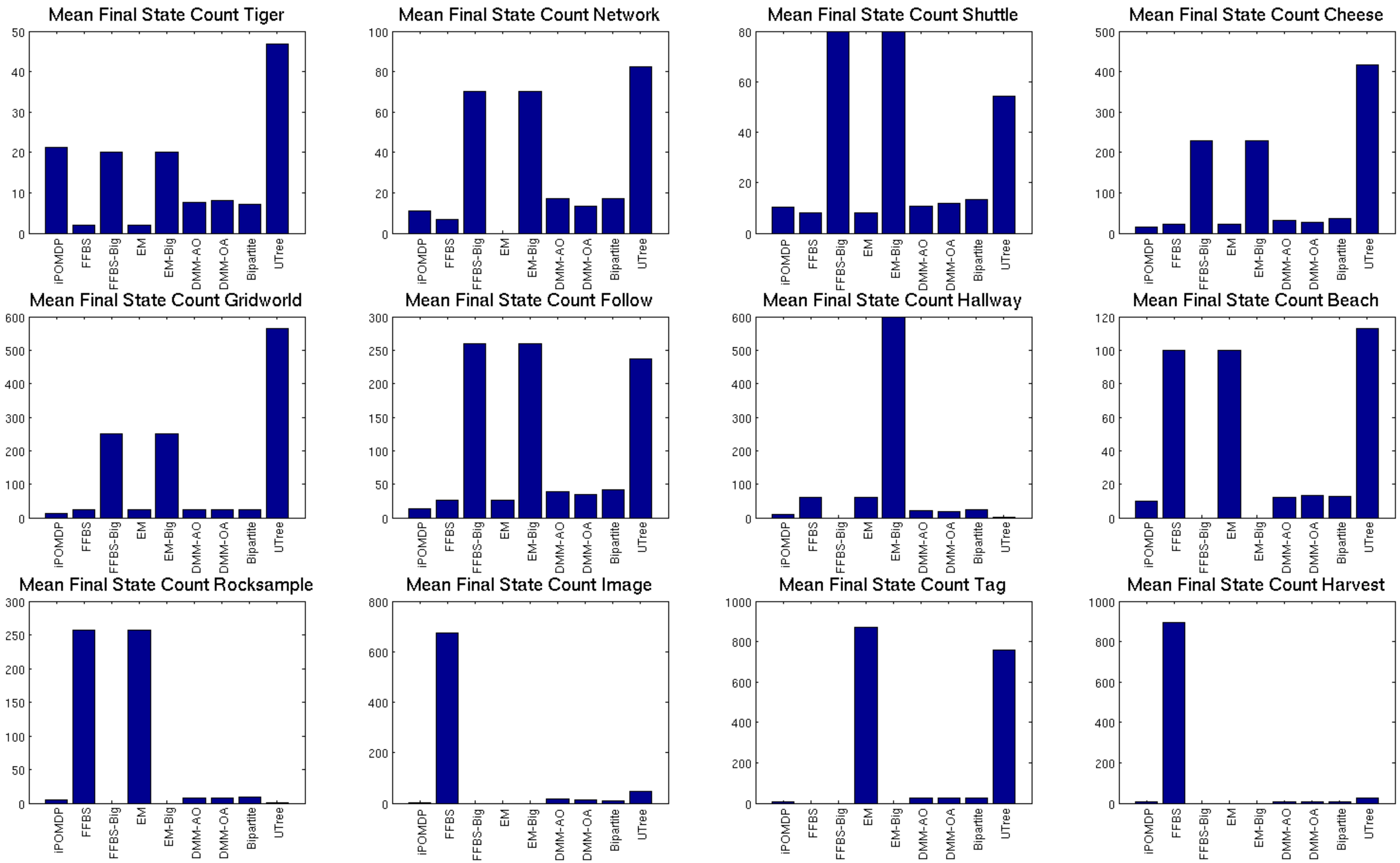
Results on Standard Problems



Results on Standard Problems



Results on Standard Problems



Summary of the Prior