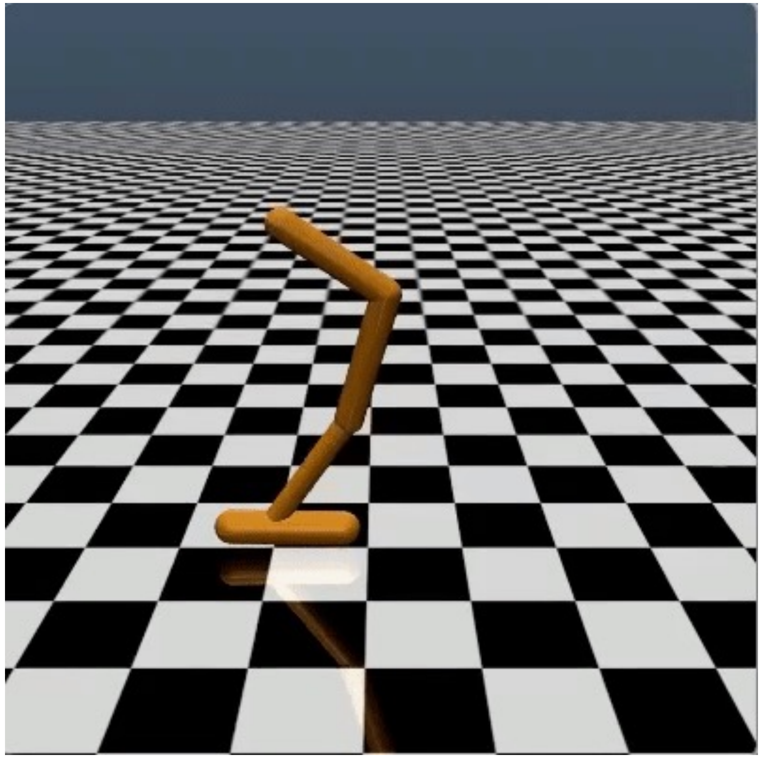
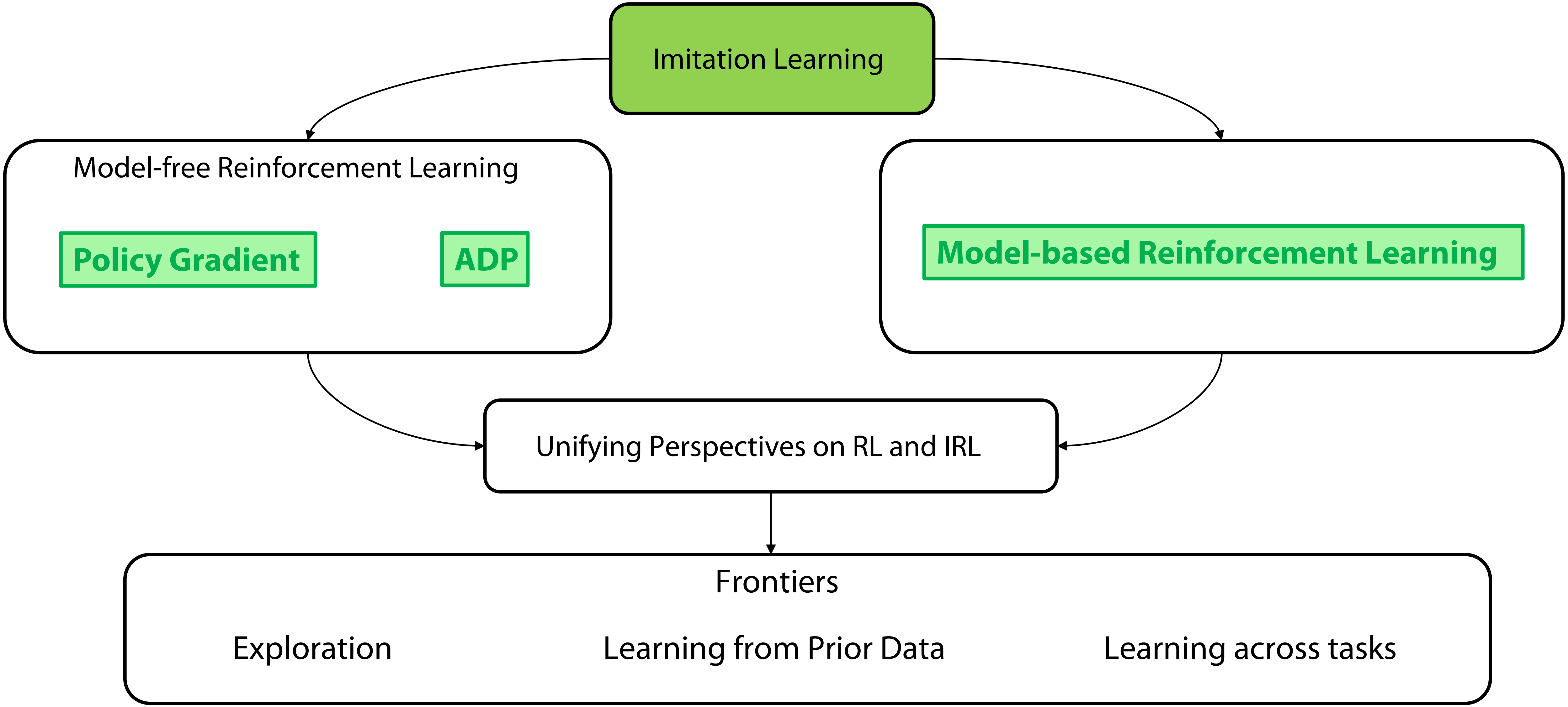
Reinforcement Learning
Spring 2024

Abhishek Gupta

TAs: Patrick Yin, Qiuyu Chen



Class Structure



Ok, let's talk about "optimality"

Optimal control problems aim to find the "max" reward policy

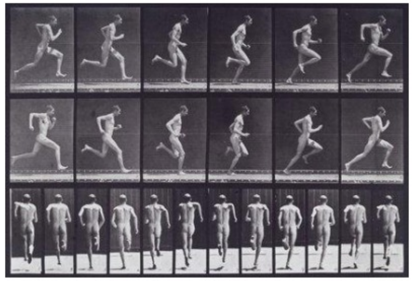
$$\arg \max_{a_0^j, a_1^j, \ldots, a_T^j} \sum_{t=0}^{T} r(\hat{s}_t^j, a_t^j)$$

$$\hat{s}_{t+1}^j \sim \hat{p}_\theta(.|\hat{s}_t^j, a_t^j)$$

$$\max_\theta \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T} r(s_t, a_t) \right]$$

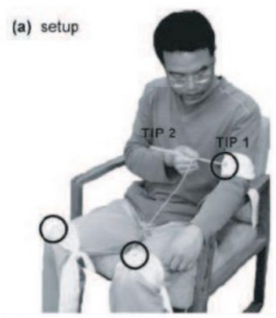
No notion of smooth suboptimality

People are not perfectly rational, "noisily" rational



Muybridge (c. 1870)

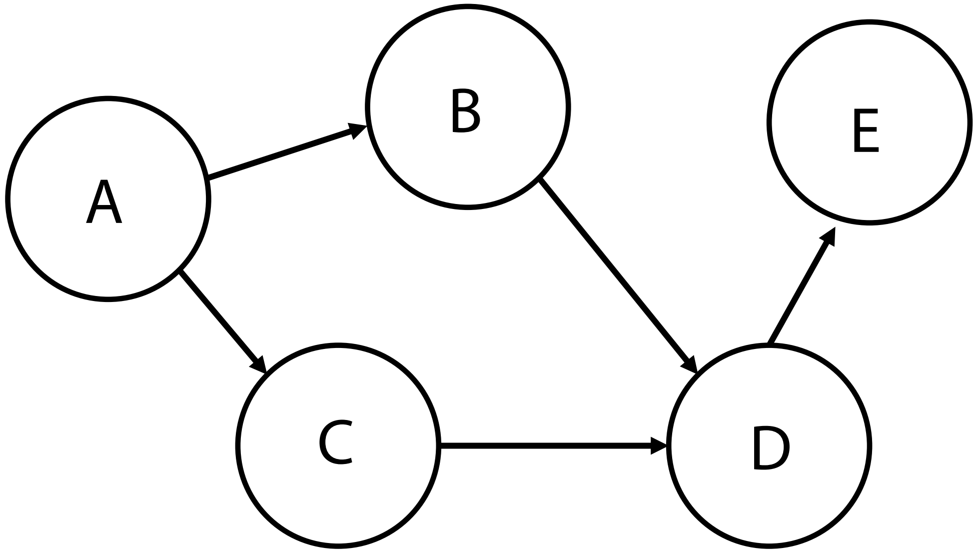
Mombaur et al. '09



Li & Todorov '06

Ziebart '08

Background: Probabilistic Graphical Models



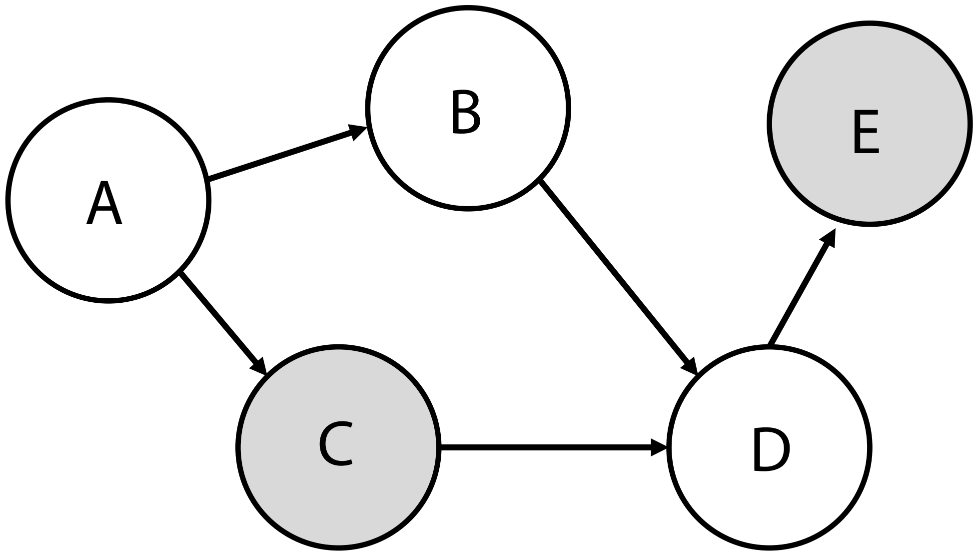
Convenient way to encode joint probability distribution

Encodes probabilities and conditional independences

$$P(A, B, \ldots,) = \Pi_X P(X|\text{Parents}(X))$$

$$P(A, B, \ldots,) = P(A)P(B|A)P(C|A)P(D|B, C)P(E|D)$$

Probabilistic Graphical Models



So what can you do with a probabilistic graphical model?

P(B|C, E)

P(A, B|C, E)

Answer posterior inference queries

What does this have to do with RL?

Isn't RL about maximizing expected reward?

Need to "eliminate" variables and use Bayes rule
→ Easy in discrete space, challenging in continuous

Lecture outline

Control as Inference - Formulation

↓

Variational Inference

↓

Control as Inference to Derive Policy Gradient

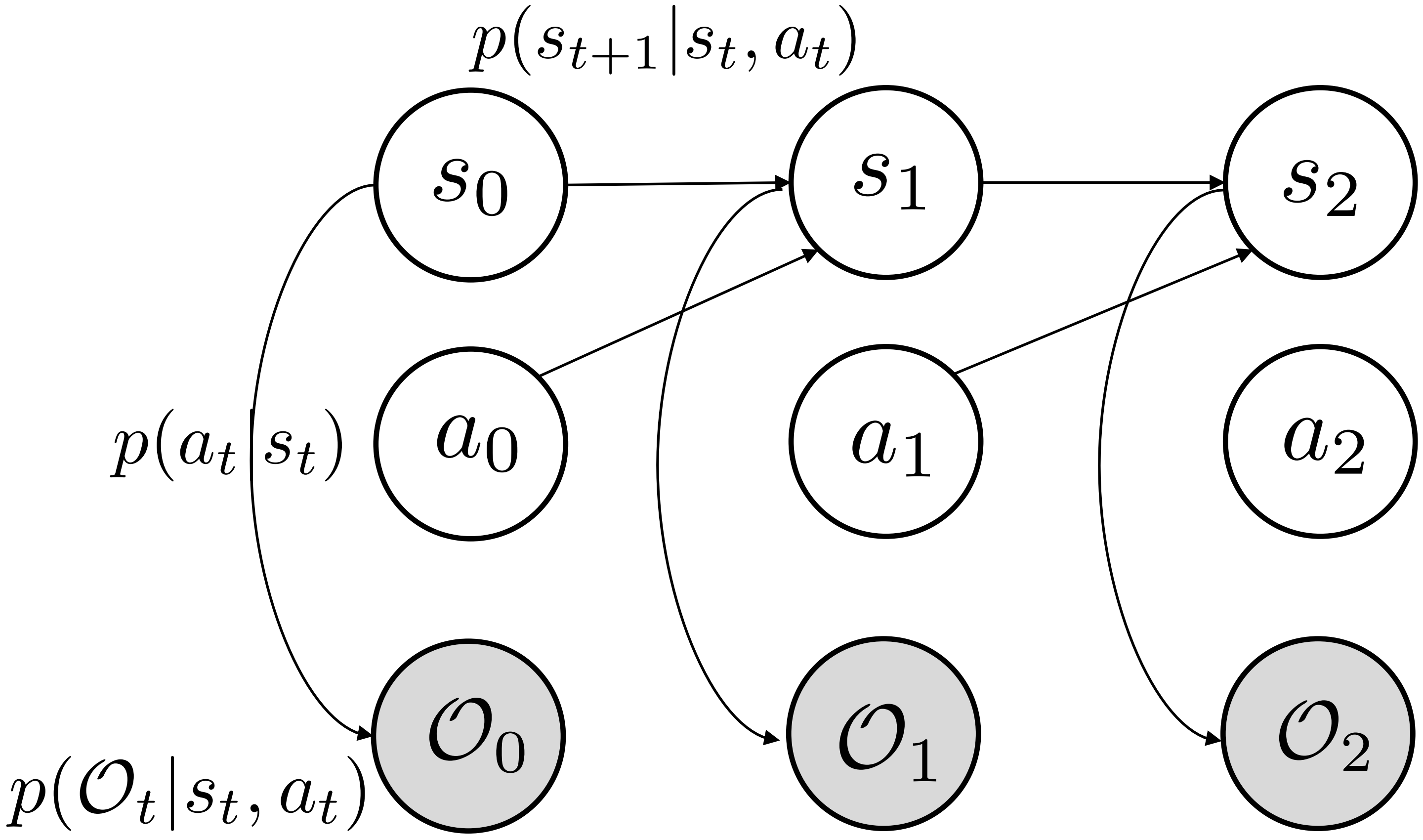
↓

Control as Inference to Derive Q-learning

↓

Control as Inference to Derive Model-Based RL

Using Probabilistic Graphical Models for Decision Making



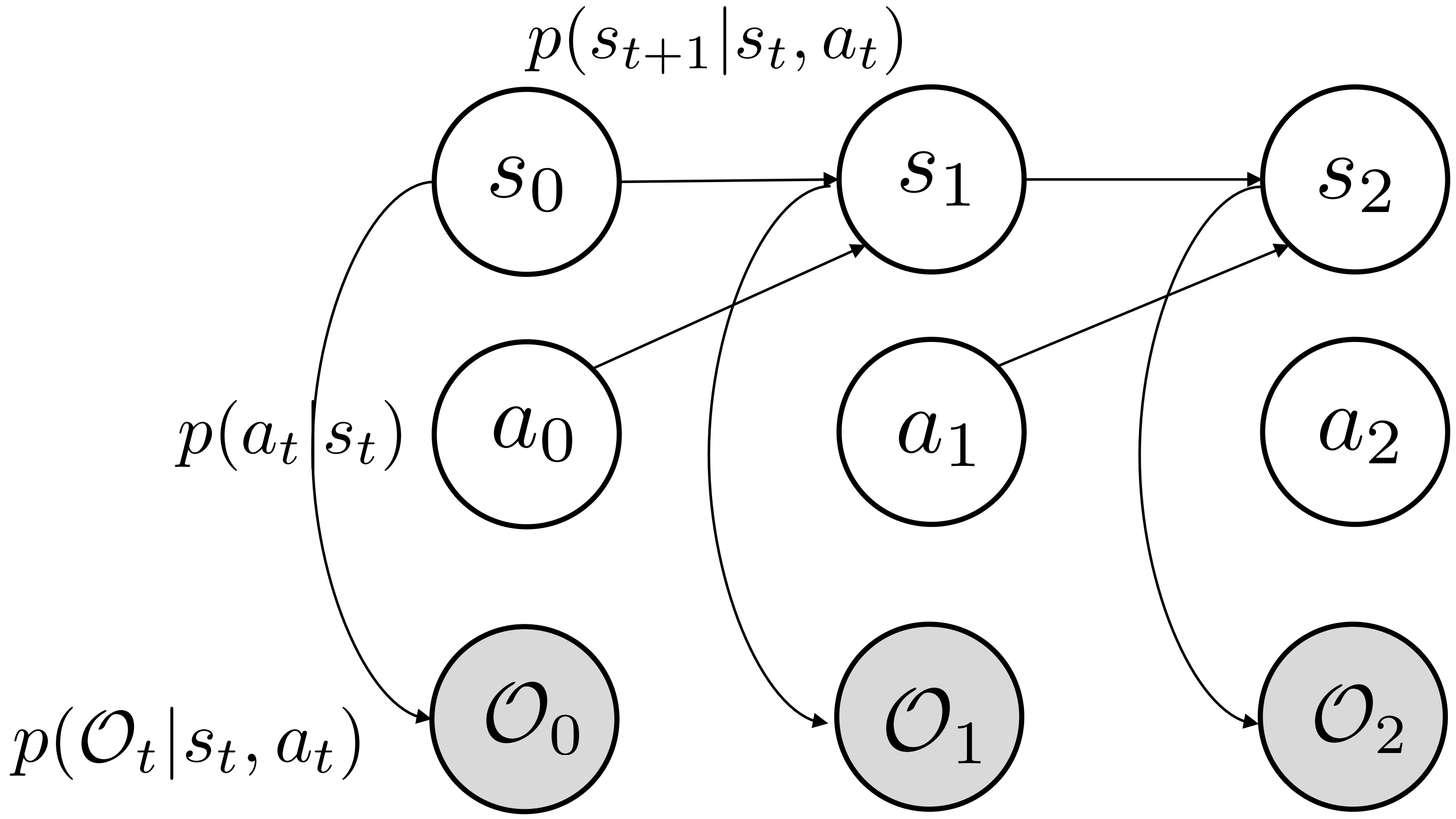
$$p(\mathcal{O}_t|s_t, a_t) = \exp(r(s_t, a_t))$$

Rewards must be negative
(subtract max reward WLOG)

Introduce binary "optimality" variables – optimal if O=1, suboptimal if O=0

Agents are observed to be **optimal**

Ok so how can we cast decision making as a PGM?



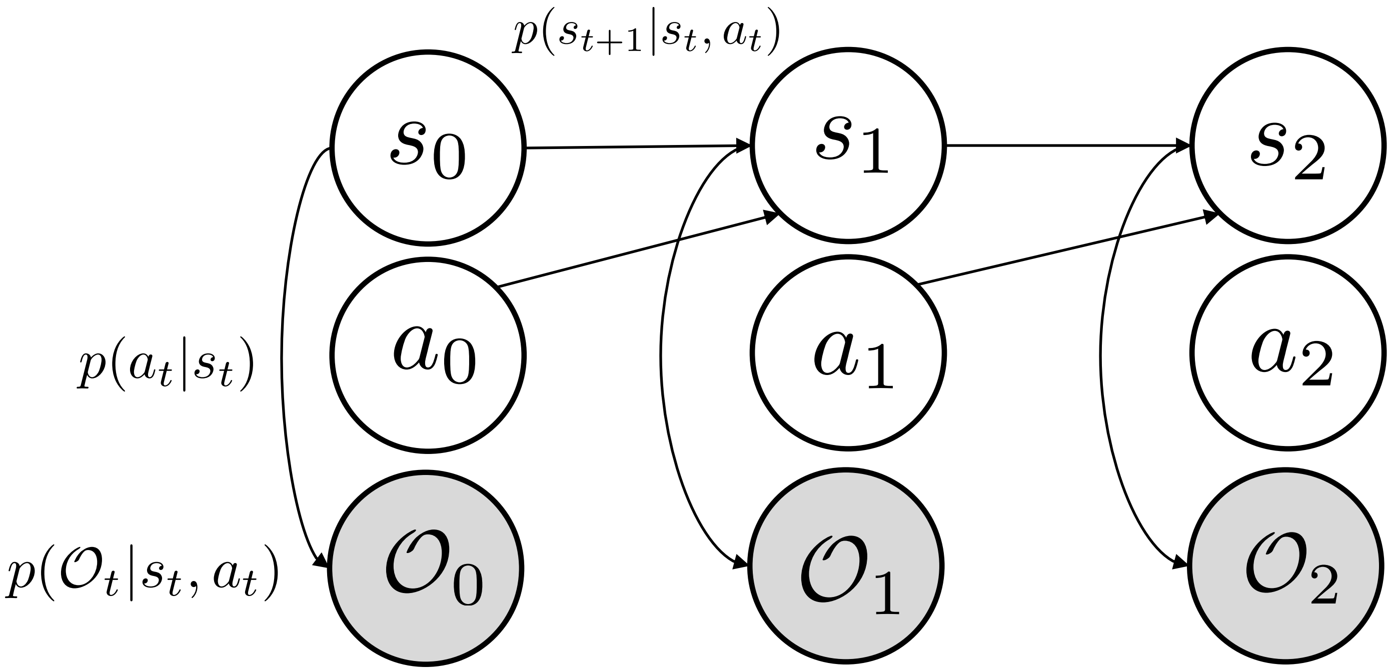
$$p(\mathcal{O}_t|s_t, a_t) = \exp(r(s_t, a_t))$$

$$p(\tau|\mathcal{O}_{0:T} = 1) \propto p(\tau)p(\mathcal{O}_{0:T}|\tau) = p(s_0)\prod_{t=0}^{T} p(s_{t+1}|s_t, a_t)p(a_t|s_t)p(\mathcal{O}_t|s_t, a_t)$$

$$= p(\tau)\exp(\sum_{t=0}^{T} r(s_t, a_t))$$

"Soft" optimality – higher return trajectories are higher likelihood

Ok big whoop, what do we do this?



$$p(\mathcal{O}_t|s_t, a_t) = \exp(r(s_t, a_t))$$

$$p(\tau|\mathcal{O}_{0:T} = 1) \propto p(\tau)\exp(\sum_{t=0}^{T} r(s_t, a_t))$$

Use case 1:

Derive soft RL algorithms

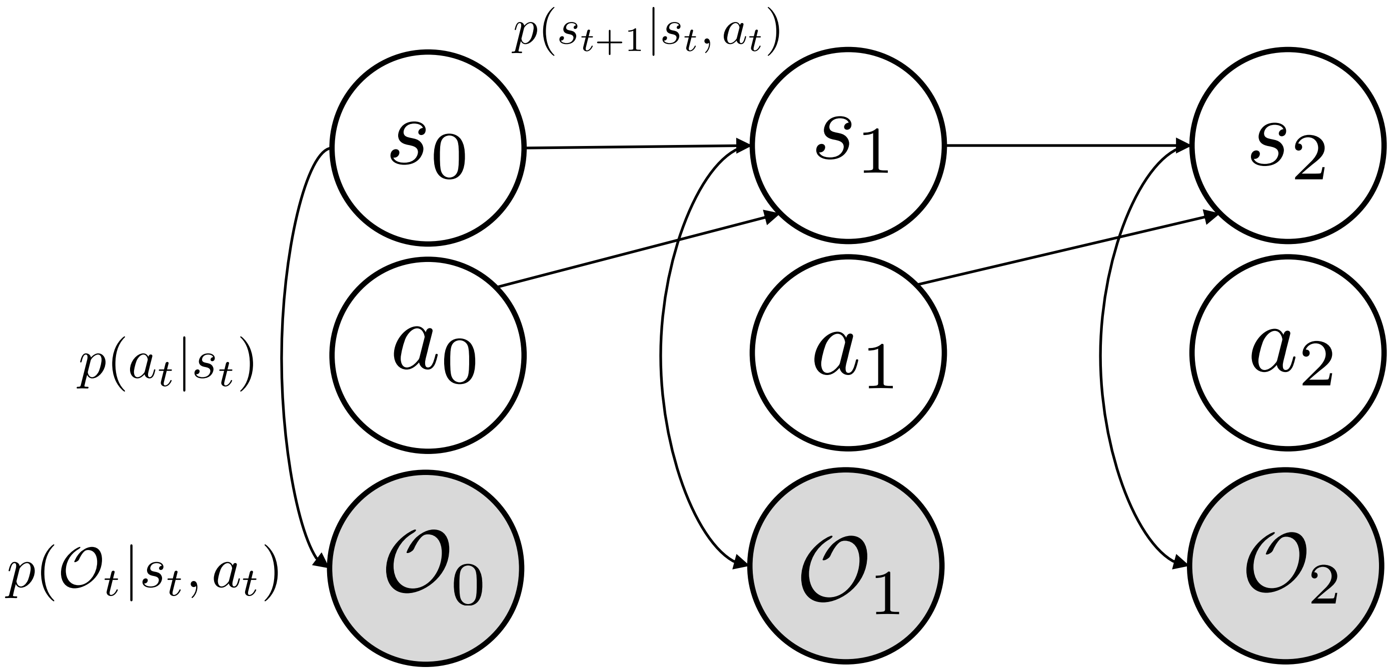
Use case 2:

Derive soft inverse RL algorithms

Use case 3:

Great algorithms for transfer

So what are we doing inference over?



$$p(\mathcal{O}_t|s_t, a_t) = \exp(r(s_t, a_t))$$

$$p(\tau|\mathcal{O}_{0:T} = 1) \propto p(\tau) \exp(\sum_{t=0}^{T} r(s_t, a_t))$$

Use case 1:

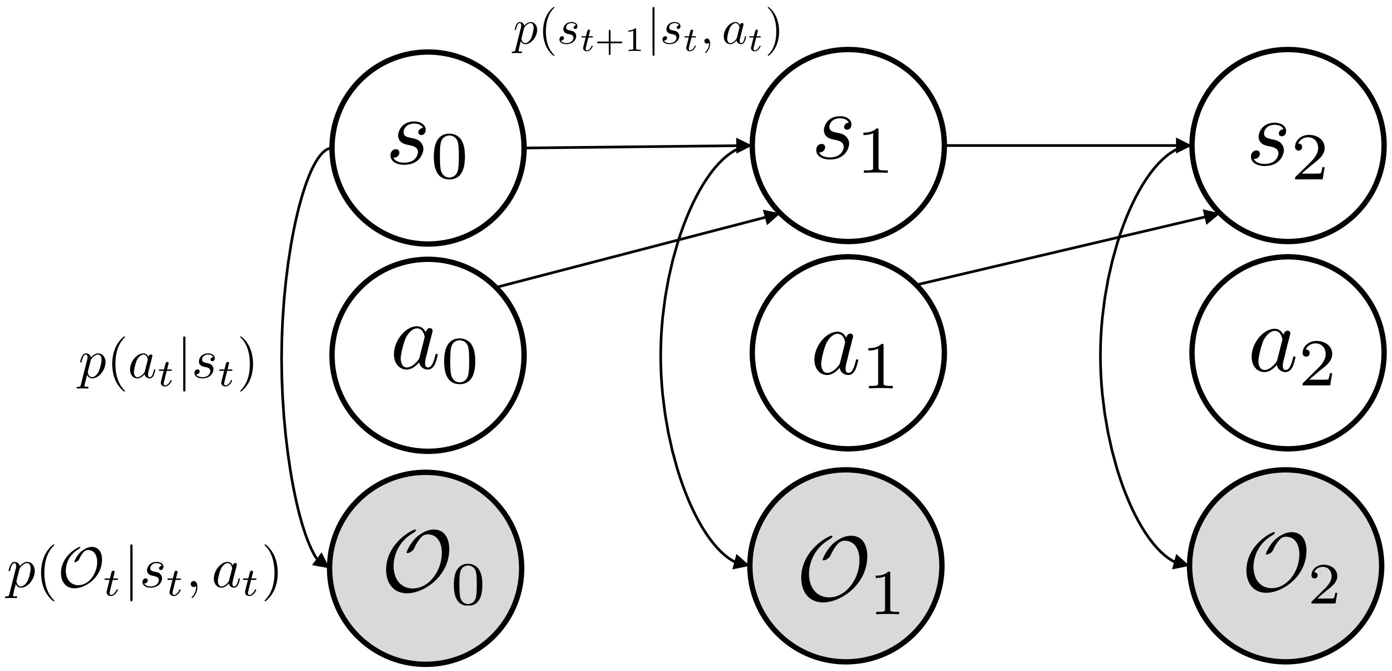
Derive soft RL algorithms

Insight: Computing optimal policy → posterior inference

$$p(a_t|s_t, \mathcal{O}_{t:T} = 1)$$

"Given that you are acting optimally, what is the likelihood of a particular action at a state"

Why isn't this trivial?



$$p(\mathcal{O}_t|s_t, a_t) = \exp(r(s_t, a_t))$$

$$p(\tau|\mathcal{O}_{0:T} = 1) \propto p(\tau)\exp(\sum_{t=0}^{T} r(s_t, a_t))$$

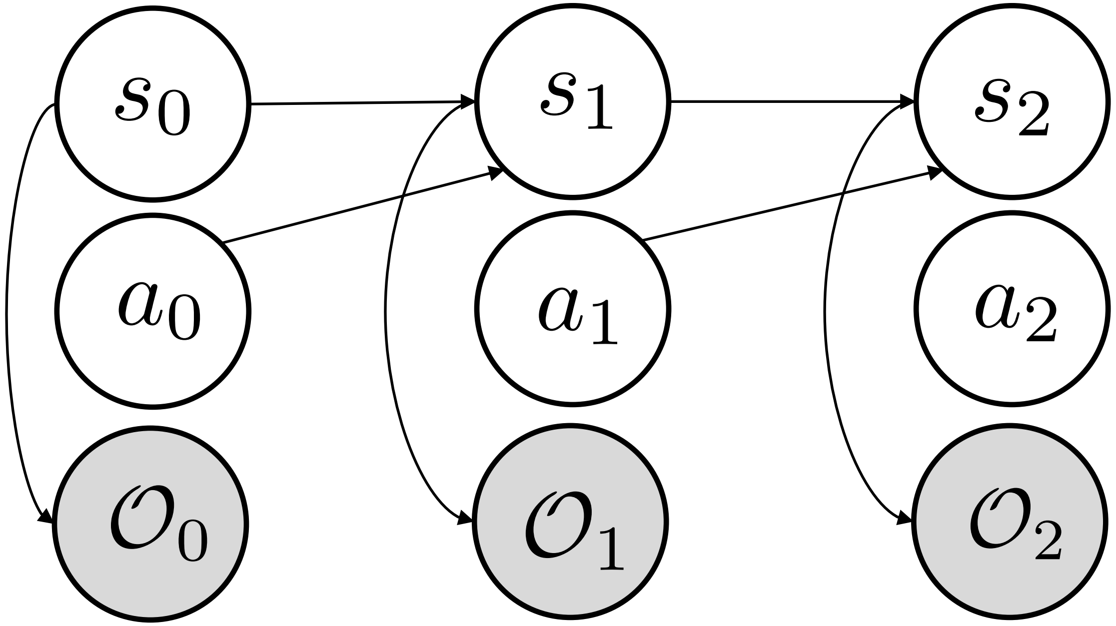
Optimal Policy → Posterior Inference

$$p(a_t|s_t, \mathcal{O}_{t:T} = 1) = \frac{p(a_t, \mathcal{O}_{t:T} = 1|s_t)}{p(\mathcal{O}_{t:T} = 1|s_t)} = \frac{\int\int\cdots\int p(a_{t:T}, \mathcal{O}_{t:T} = 1, s_{t:T}) ds_{t+1:T} da_{t+1:T}}{\int\int\cdots\int p(a_{t:T}, \mathcal{O}_{t:T} = 1, s_{t:T}) ds_{t+1:T} da_{t:T}}$$

"Given that you are acting optimally, what is the likelihood of a particular action at a state"

Difficult/intractable to compute
→ Most RL algorithms are approximations to this

What makes this so cool?



Optimal Policy → Posterior Inference

$$p(a_t|s_t, \mathcal{O}_{t:T} = 1)$$

$$= \frac{p(a_t, \mathcal{O}_{t:T} = 1|s_t)}{p(\mathcal{O}_{t:T} = 1|s_t)}$$

$$= \frac{\int\int \cdots \int p(a_{t:T}, \mathcal{O}_{t:T} = 1, s_{t:T}) ds_{t+1:T} da_{t+1:T}}{\int\int \cdots \int p(a_{t:T}, \mathcal{O}_{t:T} = 1, s_{t:T}) ds_{t+1:T} da_{t:T}}$$

<u>Policy Gradient</u>

Variational Inference lower bound solved with Gradient Ascent

<u>Approximate DP</u>

Variational Inference lower bound solved with dynamic programming

<u>Model-Based RL</u>

Posterior Inference Approximated with Monte-Carlo Samples

Can derive old algorithms + new classes of algorithms from the same framework!

Lecture outline

Control as Inference - Formulation

↓

Variational Inference

↓

Control as Inference to Derive Policy Gradient

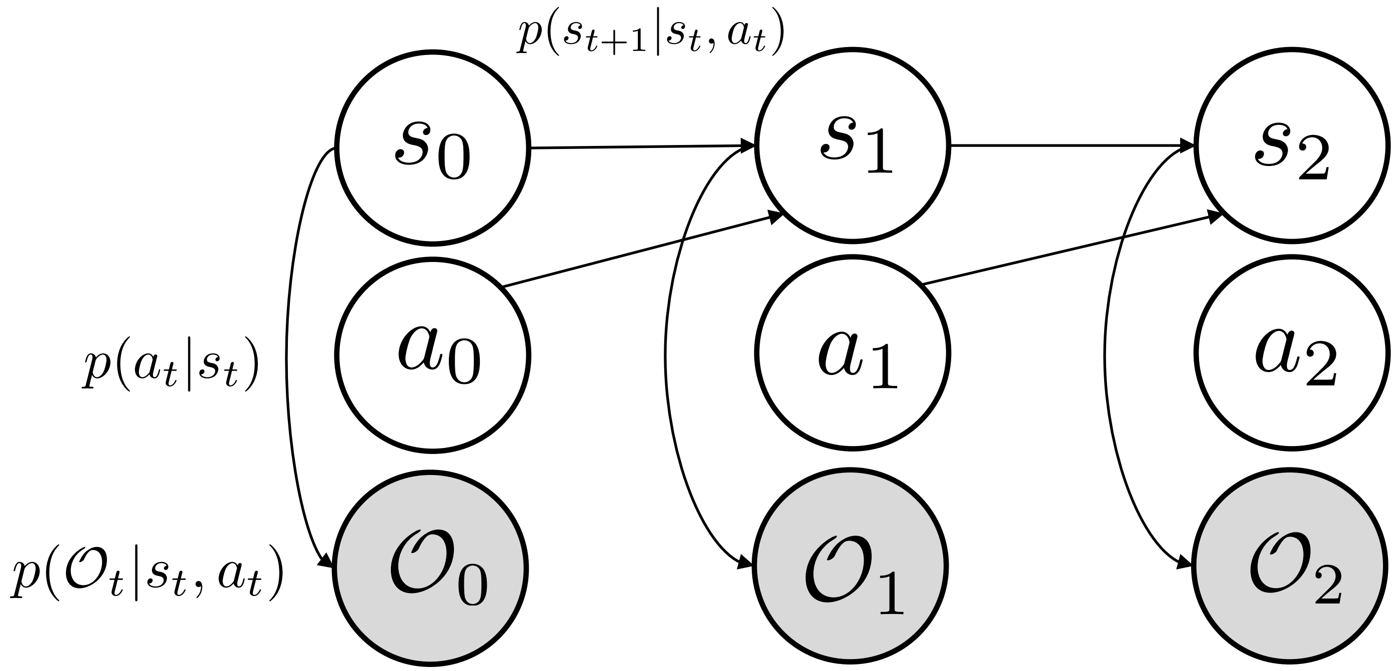
↓

Control as Inference to Derive Q-learning

↓

Control as Inference to Derive Model-Based RL

Why isn't this trivial?



$$p(\mathcal{O}_t|s_t, a_t) = \exp(r(s_t, a_t))$$

$$p(\tau|\mathcal{O}_{0:T} = 1) \propto p(\tau)\exp\left(\sum_{t=0}^{T} r(s_t, a_t)\right)$$

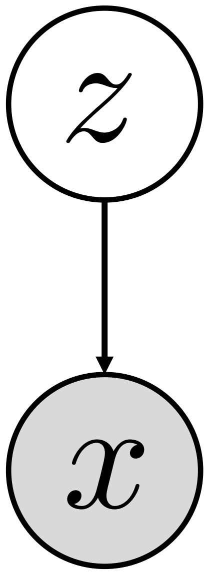
Optimal Policy → Posterior Inference

$$p(a_t|s_t, \mathcal{O}_{t:T} = 1) = \frac{p(a_t, \mathcal{O}_{t:T} = 1|s_t)}{p(\mathcal{O}_{t:T} = 1|s_t)} = \frac{\int\int\cdots\int p(a_{t:T}, \mathcal{O}_{t:T} = 1, s_{t:T})ds_{t+1:T}da_{t+1:T}}{\int\int\cdots\int p(a_{t:T}, \mathcal{O}_{t:T} = 1, s_{t:T})ds_{t+1:T}da_{t:T}}$$

"Given that you are acting optimally, what is the likelihood of a particular action at a state"

Difficult/intractable to compute
→ Most RL algorithms are approximations to this

Let's take the simplest possible example



Standard latent-variable model

Let us assume $p(x|z)$ is known, as is $p(z)$

Goal: Infer posterior $p(z|x)$

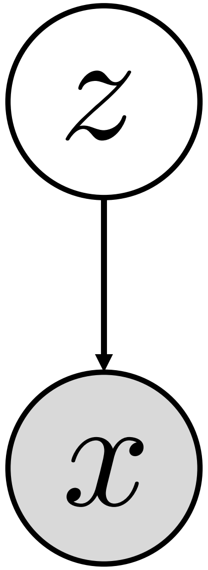
$$p(z|x) = \frac{p(x,z)}{p(x)} = \frac{p(x|z)p(z)}{p(x)}$$

$$= \frac{p(x|z)p(z)}{\int p(x|z)p(z)dz}$$

Challenging to compute efficiently with samples

Blei et al

So how can we solve this posterior inference problem?



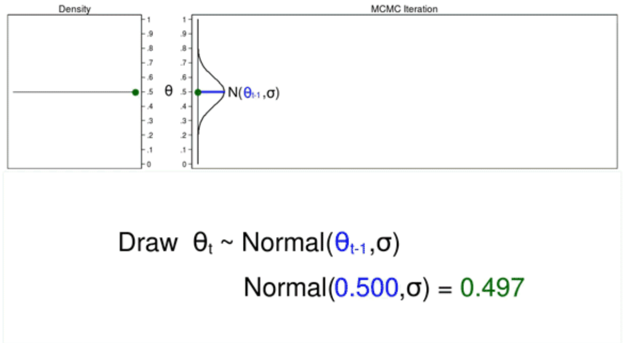
Let us assume $p(x|z)$ is known, as is $p(z)$

Goal: Infer posterior $p(z|x)$

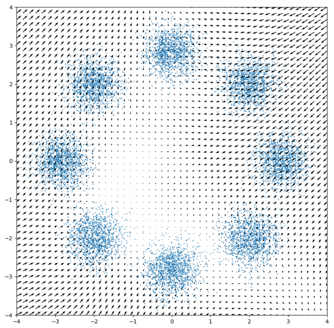
$$p(z|x) = \frac{p(x|z)p(z)}{\int p(x|z)p(z)dz}$$

Challenging to compute efficiently with samples

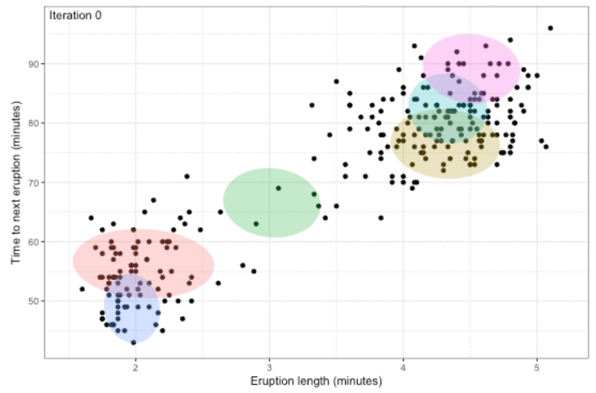
<u>MCMC</u>



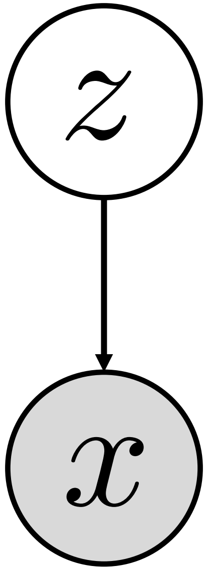
<u>EBMs and Score Matching</u>



<u>Variational Inference</u>

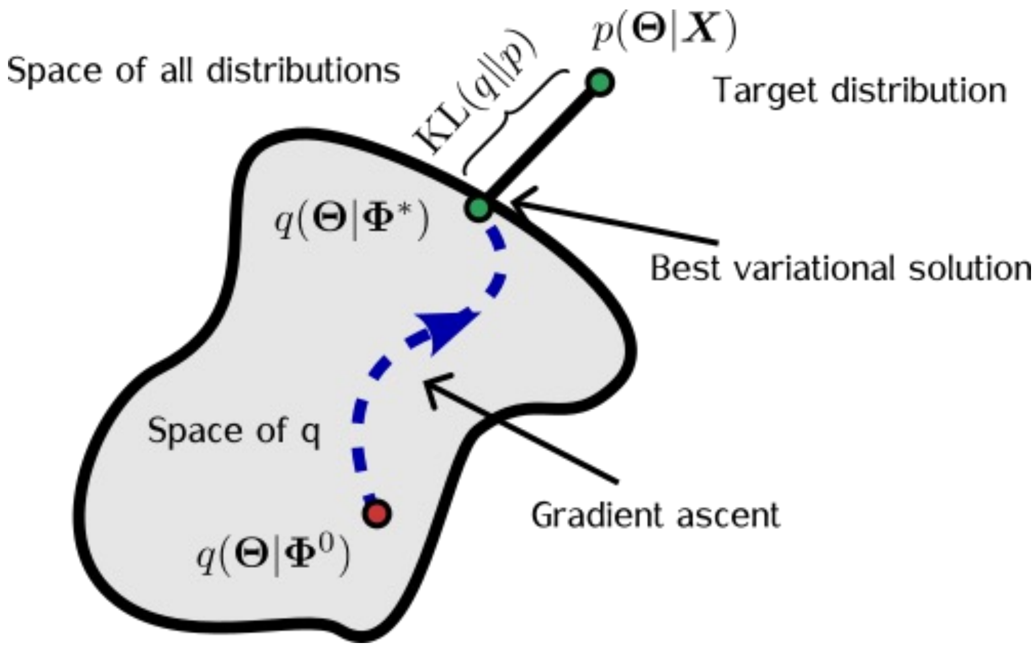


What is the key idea behind variational inference?



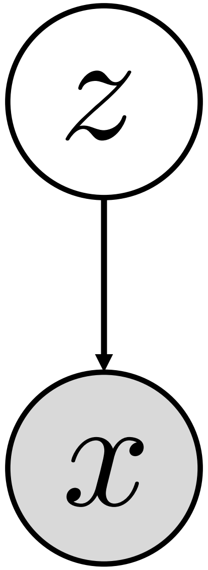
$$p(z|x) = \frac{p(x|z)p(z)}{\int p(x|z)p(z)dz}$$

Intractable!



Approximate challenging posterior with closest possible “tractable” posterior

Let's derive the Evidence Lower Bound



$$p(z|x) = \frac{p(x|z)p(z)}{\int p(x|z)p(z)dz}$$

Introduce a "tractable" approximation $q(z|x)$
e.g. Gaussian

Intractable!

Can choose **whatever** variational family you want
→ it's an approximation! 🤷🏾‍♂️

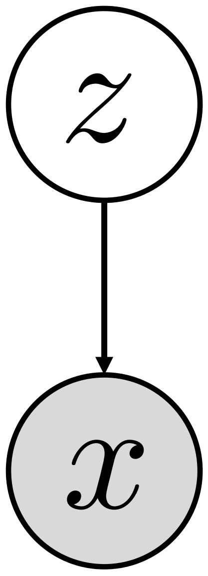
$$\phi^* \leftarrow \arg\min_{\phi} D_{KL}(q_\phi(z|x)||p(z|x))$$

Unknown

Known

How can we tractably approximate this objective?

Let's derive the Evidence Lower Bound



$$p(z|x) = \frac{p(x|z)p(z)}{p(x)}$$

Intractable!

Unknown

$$\phi^* \leftarrow \arg\min_{\phi} D_{KL}(q_\phi(z|x)||p(z|x))$$

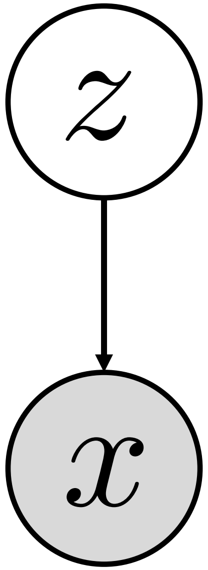
Known

$$D_{KL}(q_\phi(z|x)||p(z|x)) = \int q(z|x) \log \frac{q(z|x)}{p(z|x)} dz = \int q(z|x) \log \frac{q(z|x)p(x)}{p(x|z)p(z)} dz$$

$$= \int q(z|x) \log \frac{q(z|x)}{p(z)} dz - \int q(z|x) \log p(x|z) dz + \log p(x)$$

$$= D_{KL}(q(z|x)||p(z)) - \mathbb{E}_{z\sim q(z|x)} [\log p(x|z)] + \log p(x)$$

Let's derive the Evidence Lower Bound



$$p(z|x) = \frac{p(x|z)p(z)}{p(x)}$$

Intractable!

Unknown

$$\phi^* \leftarrow \arg\min_{\phi} D_{KL}(q_\phi(z|x)||p(z|x))$$

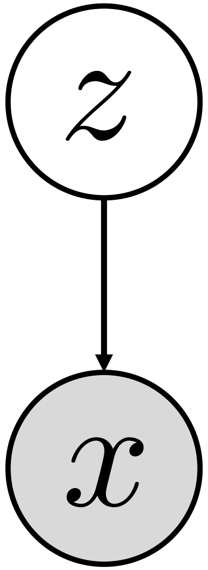
Known

$$D_{KL}(q_\phi(z|x)||p(z|x)) = D_{KL}(q(z|x)||p(z)) - \mathbb{E}_{z\sim q(z|x)}\left[\log p(x|z)\right] + \log p(x)$$

View 1: Find best posterior

View 2: Maximize marginal likelihood

Evidence Lower Bound: Best Posterior



$$p(z|x) = \frac{p(x|z)p(z)}{p(x)}$$

Intractable!

View 1: Find best posterior

$$D_{KL}(q_\phi(z|x)||p(z|x))$$

$$= D_{KL}(q(z|x)||p(z)) - \mathbb{E}_{z\sim q(z|x)}\left[\log p(x|z)\right] + \log p(x)$$

Likelihood/prior known – posterior hard to compute

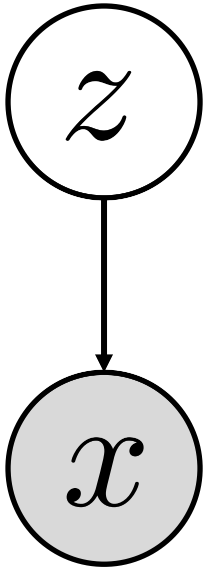
Maximum likelihood

Stay close to the prior

$$\max_q \mathbb{E}_{x\sim p(x)}\left[\mathbb{E}_{z\sim q(z|x)}\left[\log p(x|z)\right] - D_{KL}(q(z|x)||p(z))\right]$$

Learn a tractable posterior q(z|x) with known likelihood and sampling

Evidence Lower Bound: Max Marginal Likelihood



$$p(z|x) = \frac{p(x|z)p(z)}{p(x)}$$

Intractable!

View 2: Maximize marginal likelihood

$$D_{KL}(q_\phi(z|x)||p(z|x))$$

$$= D_{KL}(q(z|x)||p(z)) - \mathbb{E}_{z\sim q(z|x)}\left[\log p(x|z)\right] + \log p(x)$$

Likelihood unknown and posterior hard to compute

$$\log p(x) - D_{KL}(q(z|x)||p(z|x)) = \mathbb{E}_{z\sim q(z|x)}\left[\log p(x|z)\right] - D_{KL}(q(z|x)||p(z))$$

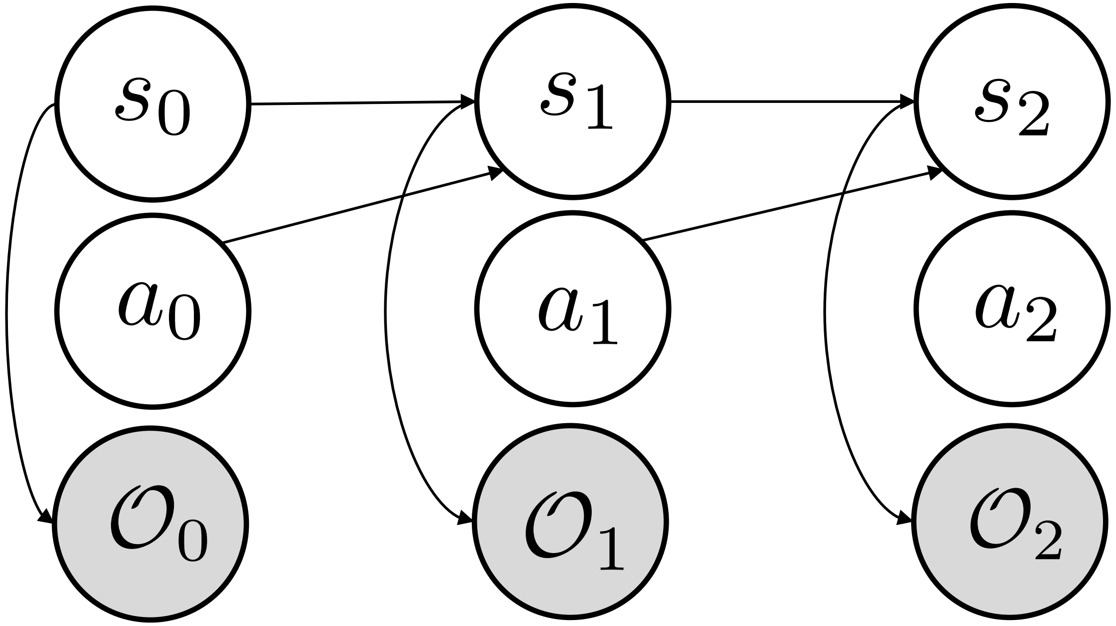
$$D_{KL}(p||q) \geq 0$$

$$\log p(x) \geq \mathbb{E}_{z\sim q(z|x)}\left[\log p(x|z)\right] - D_{KL}(q(z|x)||p(z))$$

Evidence **lower** bound – maximize to maximize likelihood

Learned

Lets revisit our original inference problem in control



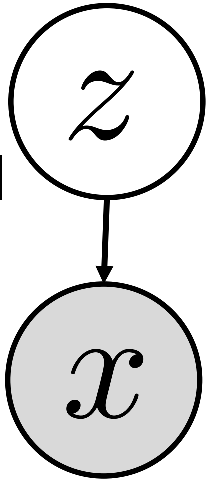
Optimal Policy → Posterior Inference

$$p(a_t|s_t, \mathcal{O}_{t:T} = 1)$$

$$= \frac{p(a_t, \mathcal{O}_{t:T} = 1|s_t)}{p(\mathcal{O}_{t:T} = 1|s_t)}$$

$$= \frac{\int\int\cdots\int p(a_{t:T}, \mathcal{O}_{t:T} = 1, s_{t:T}) ds_{t+1:T} da_{t+1:T}}{\int\int\cdots\int p(a_{t:T}, \mathcal{O}_{t:T} = 1, s_{t:T}) ds_{t+1:T} da_{t:T}}$$

Variational Inference



Approximate $p(a_t|s_t, \mathcal{O}_{t:T} = 1)$ by $q(a_t|s_t, \mathcal{O}_{t:T} = 1)$

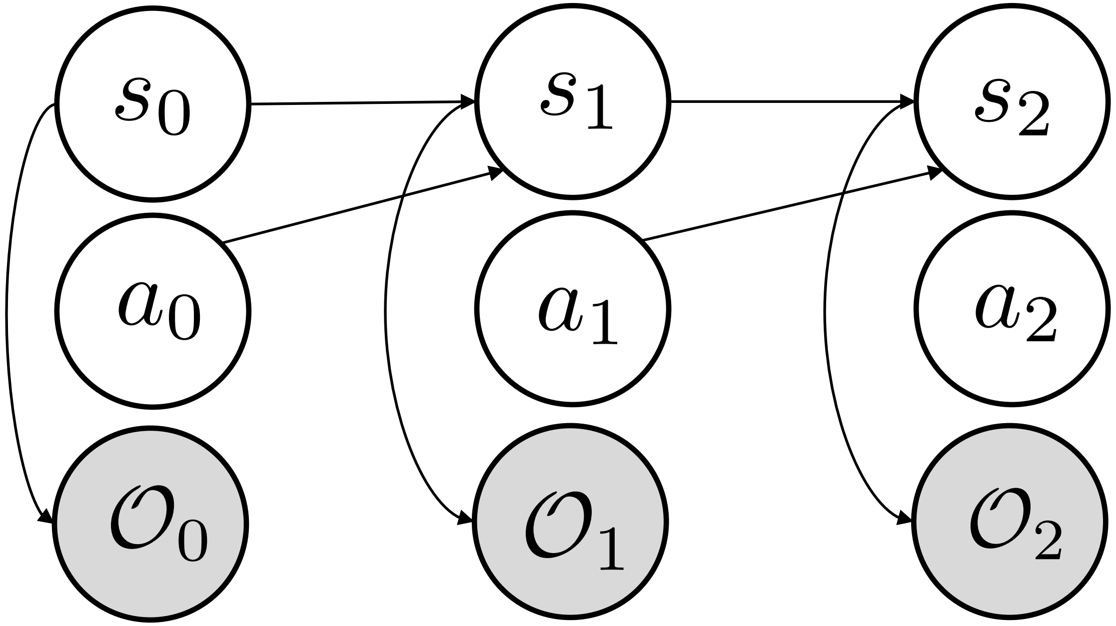
$$\max_q \mathbb{E}_{x\sim p(x)} \left[\mathbb{E}_{z\sim q(z|x)} [\log p(x|z)] - D_{KL}(q(z|x)||p(z))\right]$$

$x \updownarrow (\mathcal{O}_0, \mathcal{O}_1, \ldots, \mathcal{O}_T)$

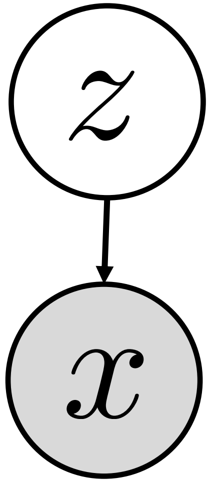
Tractable techniques for posterior policy computation

$z \updownarrow (s_0, a_0, s_1, a_1, \ldots, s_T, a_T)$

Lets revisit our original inference problem in control



Variational Inference



$$\max_q \mathbb{E}_{x\sim p(x)}\left[\mathbb{E}_{z\sim q(z|x)}\left[\log p(x|z)\right] - D_{KL}(q(z|x)||p(z))\right]$$

$$x \updownarrow (\mathcal{O}_0, \mathcal{O}_1, \ldots, \mathcal{O}_T)$$

$$z \updownarrow (s_0, a_0, s_1, a_1, \ldots, s_T, a_T)$$

Next –
derive ELBO and work out how to compute
Policy gradient/Actor-Critic

Lecture outline

Control as Inference - Formulation

↓

Variational Inference

↓

Control as Inference to Derive Policy Gradient

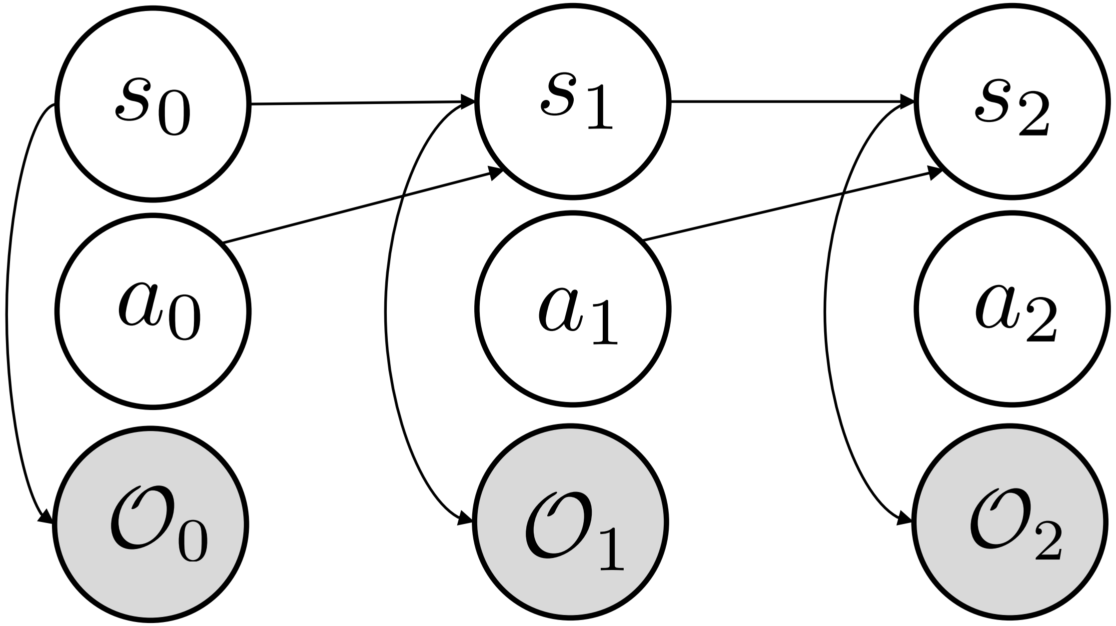
↓

Control as Inference to Derive Q-learning

↓

Control as Inference to Derive Model-Based RL

Lets revisit our original inference problem in control



Optimal Policy → Posterior Inference

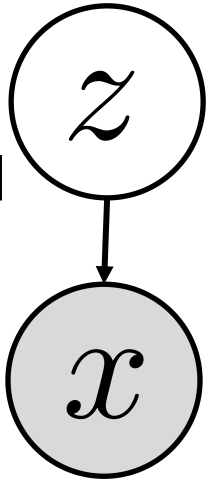
Approximate $p(a_t|s_t, \mathcal{O}_{t:T} = 1)$ by $q(a_t|s_t, \mathcal{O}_{t:T} = 1)$

$$\max_q \mathbb{E}_{x\sim p(x)} \left[\mathbb{E}_{z\sim q(z|x)} [\log p(x|z)] - D_{KL}(q(z|x)||p(z))\right]$$

$x \updownarrow (\mathcal{O}_0, \mathcal{O}_1, \ldots, \mathcal{O}_T)$

$z \updownarrow (s_0, a_0, s_1, a_1, \ldots, s_T, a_T)$

Variational Inference



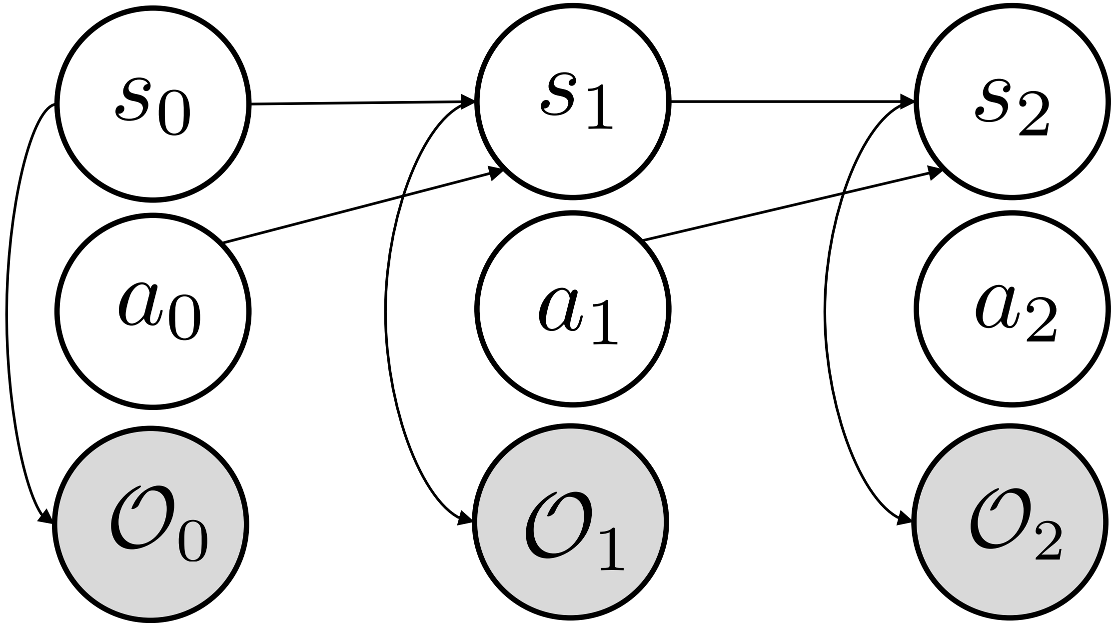
So what do we need to compute this?

→ Choice for $q(z|x)$-- $q(s_0, a_0, \ldots, s_T, a_T|\mathcal{O}_0, \ldots, \mathcal{O}_T)$

Key desiderata:

1) Can sample,
2) Compute KL Divergence

Choice of Variational Family for Approximate Inference



$$\max_q \mathbb{E}_{x\sim p(x)} \left[\mathbb{E}_{z\sim q(z|x)} [\log p(x|z)] - D_{KL}(q(z|x)||p(z))\right]$$

$x \updownarrow (\mathcal{O}_0, \mathcal{O}_1, \ldots, \mathcal{O}_T)$

$z \updownarrow (s_0, a_0, s_1, a_1, \ldots, s_T, a_T)$

So what do we need to compute this?
→ Choice for $q(s_0, a_0, \ldots, s_T, a_T|\mathcal{O}_0, \ldots, \mathcal{O}_T)$

Use temporal structure of the **true** dynamics in q

$$q(s_0, a_0, \ldots, s_T, a_T|\mathcal{O}_0, \ldots, \mathcal{O}_T) = p(s_0)p(s_1|s_0, a_0)q(a_0|s_0)p(s_2|s_1, a_1)q(a_1|s_1)\ldots q(a_T|s_T)$$

$$= p(s_0)\Pi_{t=0}^{T} p(s_{t+1}|s_t, a_t)q(a_t|s_t)$$

True dynamics and initial state

Approximate policy for $p(a_t|s_t, \mathcal{O}_{t:T} = 1)$

Does it satisfy:
1) Can sample,
2) Compute KL Divergence

Sampling from Variational Family

$$\max_q \mathbb{E}_{x\sim p(x)}\left[\mathbb{E}_{z\sim q(z|x)}\left[\log p(x|z)\right] - D_{KL}(q(z|x)||p(z))\right]$$

$x \updownarrow (\mathcal{O}_0, \mathcal{O}_1, \ldots, \mathcal{O}_T)$

$z \updownarrow (s_0, a_0, s_1, a_1, \ldots, s_T, a_T)$

$$q(s_0, a_0, \ldots, s_T, a_T|\mathcal{O}_0, \ldots, \mathcal{O}_T) = p(s_0)\Pi_{t=0}^{T} p(s_{t+1}|s_t, a_t)q(a_t|s_t)$$

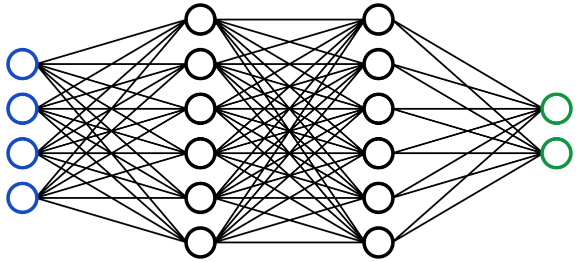
Does it satisfy:

1) Can sample

$\updownarrow$

Just sample from the policy q in the true environment p

Approximate q with neural network



Computing Evidence Lower Bound

$$x \quad (\mathcal{O}_0, \mathcal{O}_1, \ldots, \mathcal{O}_T) \qquad z \ (s_0, a_0, s_1, a_1, \ldots, s_T, a_T)$$

$$\max_q \mathbb{E}_{x \sim p(x)} \left[\mathbb{E}_{z \sim q(z|x)} [\log p(x|z)] - D_{KL}(q(z|x) || p(z)) \right]$$

$$q(s_0, a_0, \ldots, s_T, a_T | \mathcal{O}_0, \ldots, \mathcal{O}_T) = p(s_0) \Pi_{t=0}^T p(s_{t+1}|s_t, a_t) q(a_t|s_t)$$

1)

$$\mathbb{E}_{x \sim p(x), z \sim q(z|x)} [\log p(x, z) - \log q(z|x)]$$

2)

$$\mathbb{E}_{\substack{s_0 \sim p(s_0) \\ a_t \sim q(a_t|s_t) \\ s_{t+1} \sim p(s_{t+1}|s_t, a_t)}} \Big[\log p(s_0, \ldots, s_T, a_0, \ldots, a_T, \mathcal{O}_0, \ldots, \mathcal{O}_T) - \log q(s_0, a_0, \ldots, s_T, a_T | \mathcal{O}_0, \ldots, \mathcal{O}_T) \Big]$$

3)

$$\mathbb{E}_{\substack{s_0 \sim p(s_0) \\ a_t \sim q(a_t|s_t) \\ s_{t+1} \sim p(s_{t+1}|s_t, a_t)}} \left[\left[\log p(s_0) + \sum_t [\log p(a_t|s_t) + \log p(s_{t+1}|s_t, a_t) + \log p(\mathcal{O}_t|s_t, a_t)] \right] - \left[\log p(s_0) + \sum_t [\log q(a_t|s_t) + \log p(s_{t+1}|s_t, a_t)] \right] \right]$$

Computing Evidence Lower Bound

$$\max_q \mathbb{E}_{x \sim p(x)} \left[\mathbb{E}_{z \sim q(z|x)} \left[\log p(x|z) \right] - D_{KL}(q(z|x) || p(z)) \right]$$

Set to uniform

3) $$\mathbb{E}_{\substack{s_0 \sim p(s_0) \\ a_t \sim q(a_t|s_t) \\ s_{t+1} \sim p(s_{t+1}|s_t, a_t)}} \left[\left[\log p(s_0) + \sum_t [\log p(a_t|s_t) + \log p(s_{t+1}|s_t, a_t) + \log p(\mathcal{O}_t|s_t, a_t)] \right] - \left[\log p(s_0) + \sum_t [\log q(a_t|s_t) + \log p(s_{t+1}|s_t, a_t)] \right] \right]$$

4) $$\mathbb{E}_{\substack{s_0 \sim p(s_0) \\ a_t \sim q(a_t|s_t) \\ s_{t+1} \sim p(s_{t+1}|s_t, a_t)}} \left[\sum_t \log p(\mathcal{O}_t|s_t, a_t) - \log q(a_t|s_t) \right]$$

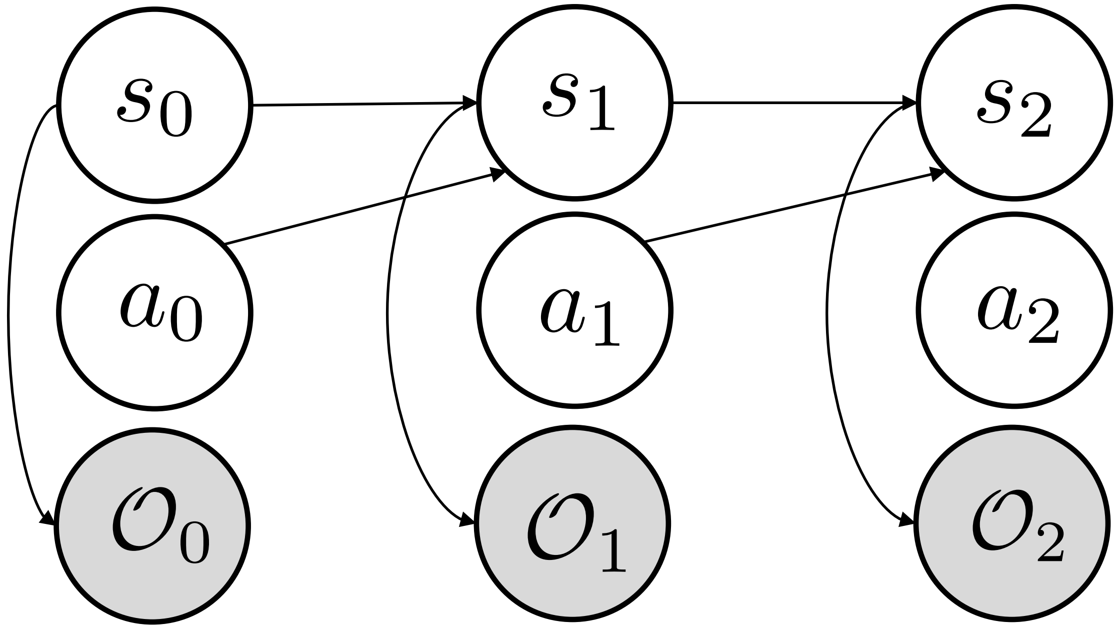
$$p(\mathcal{O}_t|s_t, a_t) = \exp(r(s_t, a_t))$$

5) $$\mathbb{E}_{\substack{s_0 \sim p(s_0) \\ a_t \sim q(a_t|s_t) \\ s_{t+1} \sim p(s_{t+1}|s_t, a_t)}} \left[\sum_t r(s_t, a_t) + \mathcal{H}(q(\cdot|s_t)) \right]$$

Maximum entropy RL

Gradient ascent = PG!

Ok so what did we show?

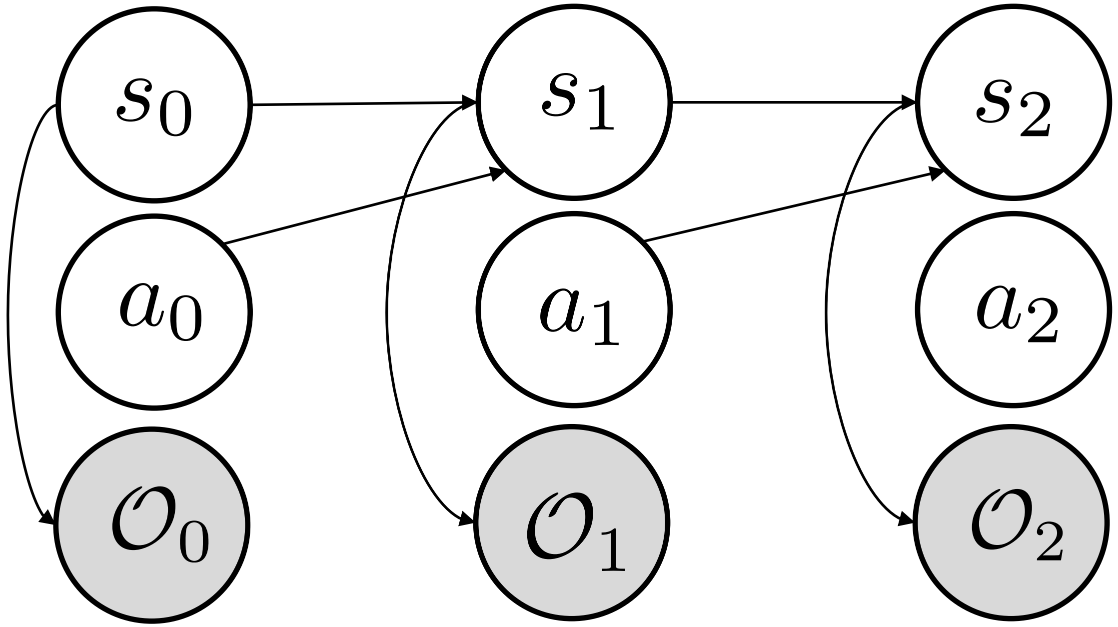


$$p(\mathcal{O}_t|s_t, a_t) = \exp(r(s_t, a_t))$$

Optimal Policy → Posterior Inference

Approximate $p(a_t|s_t, \mathcal{O}_{t:T} = 1)$ by $q(a_t|s_t, \mathcal{O}_{t:T} = 1)$

Ok so what did we show?



Optimal Policy → Posterior Inference

$$p(a_t | s_t, \mathcal{O}_{t:T} = 1)$$

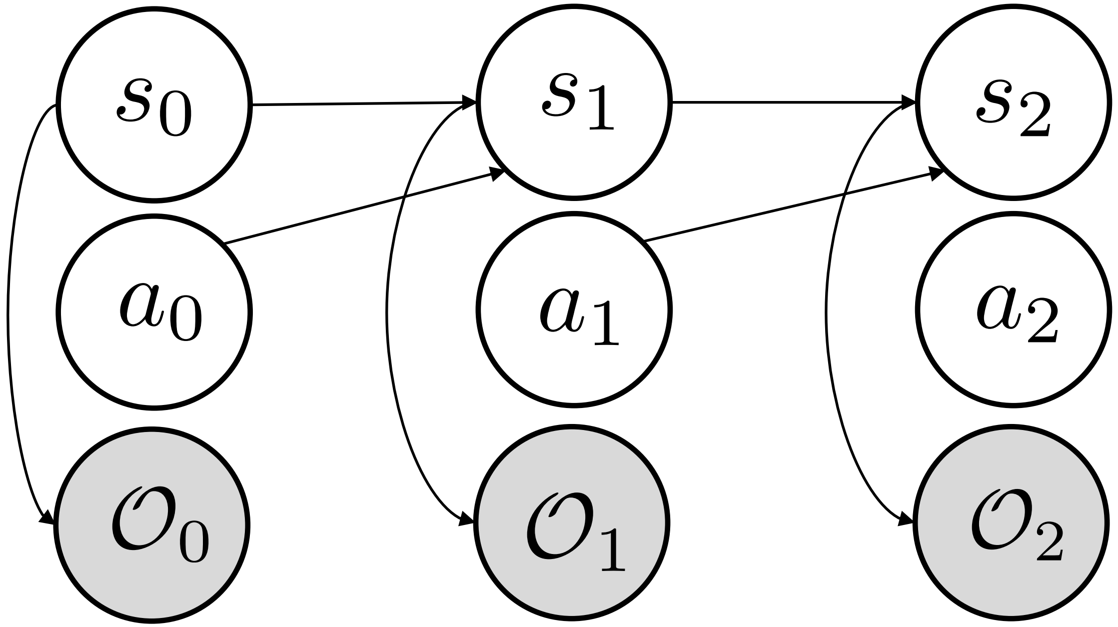
Difficult inference problem in closed form
→ use variational inference

Find approximate posterior q(z|x) by optimizing the ELBO

$$\max_q \mathbb{E}_{x \sim p(x)} \left[\mathbb{E}_{z \sim q(z|x)} [\log p(x|z)] - D_{KL}(q(z|x) || p(z)) \right]$$

$$x \updownarrow (\mathcal{O}_0, \mathcal{O}_1, \ldots, \mathcal{O}_T) \qquad z \updownarrow (s_0, a_0, s_1, a_1, \ldots, s_T, a_T)$$

Ok so what did we show?



Find approximate posterior q(z|x) by optimizing the ELBO

$$\max_q \mathbb{E}_{x\sim p(x)}\left[\mathbb{E}_{z\sim q(z|x)}\left[\log p(x|z)\right] - D_{KL}(q(z|x)||p(z))\right]$$

$$x \Updownarrow (\mathcal{O}_0, \mathcal{O}_1, \ldots, \mathcal{O}_T) \qquad z \Updownarrow (s_0, a_0, s_1, a_1, \ldots, s_T, a_T)$$

$$q(s_0, a_0, \ldots, s_T, a_T|\mathcal{O}_0, \ldots, \mathcal{O}_T) = p(s_0)\Pi_{t=0}^T p(s_{t+1}|s_t, a_t)q(a_t|s_t)$$

$$\mathbb{E}_{\substack{s_0\sim p(s_0)\\ a_t\sim q(a_t|s_t)\\ s_{t+1}\sim p(s_{t+1}|s_t,a_t)}}\left[\sum_t \log p(\mathcal{O}_t|s_t, a_t) - \log q(a_t|s_t)\right] = \mathbb{E}_{\substack{s_0\sim p(s_0)\\ a_t\sim q(a_t|s_t)\\ s_{t+1}\sim p(s_{t+1}|s_t,a_t)}}\left[\sum_t r(s_t, a_t) + \mathcal{H}(q(\cdot|s_t))\right]$$

Maximize ELBO with SGD = policy gradient!

Lecture outline

Control as Inference - Formulation

↓

Variational Inference

↓

Control as Inference to Derive Policy Gradient

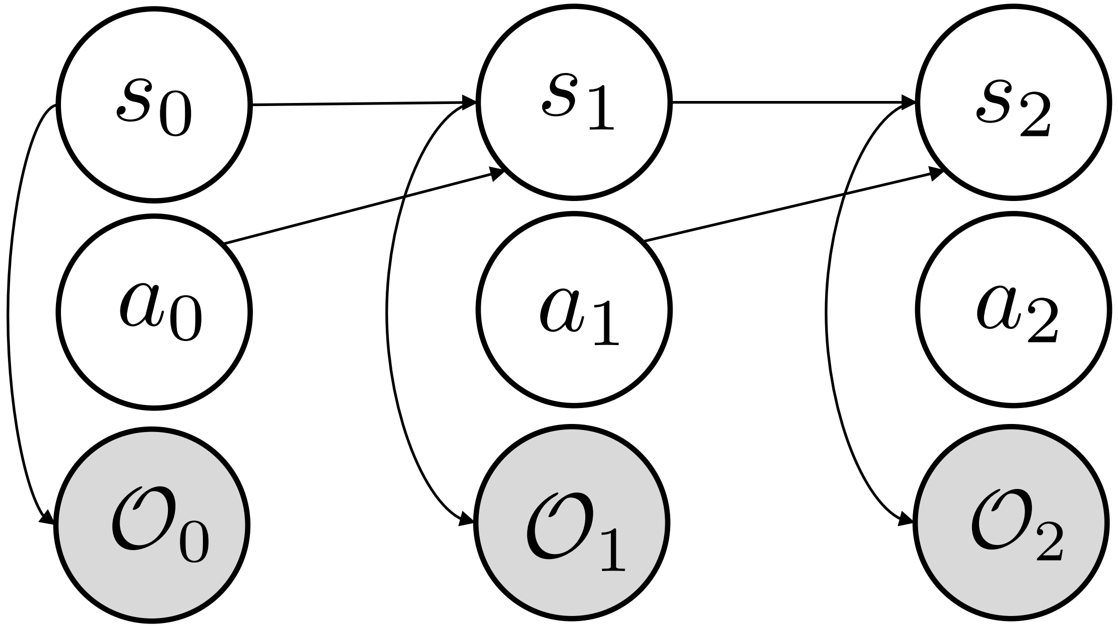
↓

Control as Inference to Derive Q-learning

↓

Control as Inference to Derive Model-Based RL

Can we derive (soft) Q-learning from the ELBO?



Find approximate posterior q(z|x) by optimizing the ELBO

$$\max_q \mathbb{E}_{x \sim p(x)} \left[\mathbb{E}_{z \sim q(z|x)} \left[\log p(x|z) \right] - D_{KL}(q(z|x) || p(z)) \right]$$

$$x \updownarrow (\mathcal{O}_0, \mathcal{O}_1, \ldots, \mathcal{O}_T) \qquad z \updownarrow (s_0, a_0, s_1, a_1, \ldots, s_T, a_T)$$

$$q(s_0, a_0, \ldots, s_T, a_T | \mathcal{O}_0, \ldots, \mathcal{O}_T) = p(s_0) \Pi_{t=0}^{T} p(s_{t+1}|s_t, a_t) q(a_t|s_t)$$

$$\mathbb{E}_{\substack{s_0 \sim p(s_0) \\ a_t \sim q(a_t|s_t) \\ s_{t+1} \sim p(s_{t+1}|s_t, a_t)}} \left[\sum_t \log p(\mathcal{O}_t|s_t, a_t) - \log q(a_t|s_t) \right] = \mathbb{E}_{\substack{s_0 \sim p(s_0) \\ a_t \sim q(a_t|s_t) \\ s_{t+1} \sim p(s_{t+1}|s_t, a_t)}} \left[\sum_t r(s_t, a_t) + \mathcal{H}(q(\cdot|s_t)) \right]$$

Maximize ELBO with DP = Soft Q learning!

Let's optimize the last step of the ELBO

$$\mathbb{E}_{\substack{s_0 \sim p(s_0) \\ a_t \sim q(a_t|s_t) \\ s_{t+1} \sim p(s_{t+1}|s_t, a_t)}} \left[\sum_t r(s_t, a_t) + \mathcal{H}(q(\cdot|s_t)) \right]$$

Consider the last time step

$$\max \; \mathbb{E}_{s_T \sim q(s_T)} \left[\mathbb{E}_{a_T \sim q(a_T|s_T)} \left[r(s_T, a_T) - \log q(a_T|s_T) \right] \right]$$

(log-exp)

$$= \mathbb{E}_{s_T \sim q(s_T)} \left[\mathbb{E}_{a_T \sim q(a_T|s_T)} \left[\log \exp(r(s_T, a_T)) - \log q(a_T|s_T) \right] \right]$$

$$= \mathbb{E}_{s_T \sim q(s_T)} \left[\mathbb{E}_{a_T \sim q(a_T|s_T)} \left[\log \exp(r(s_T, a_T)) - \log \int \exp(r(s_T, a_T)) da_T - \log q(a_T|s_T) \right] + \log \int \exp(r(s_T, a_T)) da_T \right]$$

(Add subtract to normalize)

$$= \mathbb{E}_{s_T \sim q(s_T)} \left[\mathbb{E}_{a_T \sim q(a_T|s_T)} \left[\log \frac{\exp(r(s_T, a_T))}{\int \exp(r(s_T, a_T)) da_T} - \log q(a_T|s_T) \right] \right]$$

$$= \mathbb{E}_{s_T \sim q(s_T)} \left[\mathbb{E}_{a_T \sim q(a_T|s_T)} \left[-\log \frac{q(a_T|s_T)}{\frac{\exp(r(s_T, a_T))}{\int \exp(r(s_T, a_T)) da_T}} \right] \right] = \mathbb{E}_{s_T \sim q(s_T)} \left[-D_{KL}\left(q(a_T|s_T) || \frac{\exp(r(s_T, a_T))}{\int \exp(r(s_T, a_T)) da_T}\right) \right]$$

(Definition of KL divergence)

Let's optimize the last step of the ELBO

$$\mathbb{E}_{\substack{s_0 \sim p(s_0) \\ a_t \sim q(a_t|s_t) \\ s_{t+1} \sim p(s_{t+1}|s_t, a_t)}} \left[\sum_t r(s_t, a_t) + \mathcal{H}(q(\cdot|s_t)) \right]$$

Consider the last time step

$$\max \; \mathbb{E}_{s_T \sim q(s_T)} \left[\mathbb{E}_{a_T \sim q(a_T|s_T)} \left[r(s_T, a_T) - \log q(a_T|s_T) \right] \right]$$

$$= \mathbb{E}_{s_T \sim q(s_T)} \left[-D_{KL}\left(q(a_T|s_T) \,\Big|\Big|\, \frac{\exp(r(s_T, a_T))}{\int \exp(r(s_T, a_T)) da_T}\right) \right]$$

KL-divergence D(p, q) is always non negative and is minimized when p == q

$$q(a_T|s_T) = \frac{\exp(r(s_T, a_T))}{\int \exp(r(s_T, a_T)) da_T}$$

Ok let's simplify

$$Q(s_T, a_T) = r(s_T, a_T)$$

$$V(s_T) = \log \int \exp(r(s_T, a_T)) da_T$$

$$\Rightarrow \quad q(a_T|s_T) = \exp(Q(s_T, a_T) - V(s_T))$$

Optimal policy is proportional to exponential advantage (soft-max)

Let's optimize the last step of the ELBO

$$\mathbb{E}_{\substack{s_0 \sim p(s_0) \\ a_t \sim q(a_t|s_t) \\ s_{t+1} \sim p(s_{t+1}|s_t, a_t)}} \left[\sum_t r(s_t, a_t) + \mathcal{H}(q(\cdot|s_t)) \right]$$

Consider the last time step

max $\mathbb{E}_{s_T \sim q(s_T)} \left[\mathbb{E}_{a_T \sim q(a_T|s_T)} \left[r(s_T, a_T) - \log q(a_T|s_T) \right] \right]$

$$= \mathbb{E}_{s_T \sim q(s_T)} \left[-D_{KL}\left(q(a_T|s_T) \middle\| \frac{\exp(r(s_T, a_T))}{\int \exp(r(s_T, a_T)) da_T}\right) + \log \int \exp(r(s_T, a_T)) da_T \right]$$

← (added back)

↑ Zero on optimal q

↑ Simply the value $V(s_T)$

$$= \mathbb{E}_{s_T \sim q(s_T)} \left[V(s_T) \right]$$

Simply the expected **value**

$$q(a_T|s_T) = \exp(Q(s_T, a_T) - V(s_T))$$

Optimal policy is proportional to exponential advantage (soft-max)

Let's optimize the step before

$$\mathbb{E}_{\substack{s_0 \sim p(s_0) \\ a_t \sim q(a_t|s_t) \\ s_{t+1} \sim p(s_{t+1}|s_t, a_t)}} \left[\sum_t r(s_t, a_t) + \mathcal{H}(q(\cdot|s_t)) \right]$$

Consider the second last time step

$$\arg\max_q \mathbb{E}_{s_{T-1} \sim q(s_{T-1})} \left[\mathbb{E}_{a_{T-1} \sim q(a_{T-1}|s_{T-1})} \left[r(s_{T-1}, a_{T-1}) - \log q(a_{T-1}|s_{T-1}) + \mathbb{E}_{\substack{s_T \sim p(s_T|s_{T-1}, a_{T-1}) \\ a_T \sim q(a_T|s_T)}} [r(s_T, a_T) - \log q(a_T|s_T)] \right] \right]$$

Exactly what we computed in the last step

$$\arg\max_q \mathbb{E}_{s_{T-1} \sim q(s_{T-1})} \left[\mathbb{E}_{a_{T-1} \sim q(a_{T-1}|s_{T-1})} \left[r(s_{T-1}, a_{T-1}) - \log q(a_{T-1}|s_{T-1}) + \mathbb{E}_{s_T \sim p(s_T|s_{T-1}, a_{T-1})} [V(s_T)] \right] \right]$$

From the last slide

Let us call this $Q(s_{T-1}, a_{T-1})$

$$Q(s_{T-1}, a_{T-1}) = r(s_{T-1}, a_{T-1}) + \mathbb{E}_{s_T \sim p(s_T|s_{T-1}, a_{T-1})} [V(s_T)]$$

(Looks like Bellman!)

$$\arg\max_q \mathbb{E}_{s_{T-1} \sim q(s_{T-1})} \left[\mathbb{E}_{a_{T-1} \sim q(a_{T-1}|s_{T-1})} \left[Q(s_{T-1}, a_{T-1}) - \log q(a_{T-1}|s_{T-1}) \right] \right]$$

Looks a lot like the previous time-step

Let's optimize the step before

$$\mathbb{E}_{\substack{s_0 \sim p(s_0) \\ a_t \sim q(a_t|s_t) \\ s_{t+1} \sim p(s_{t+1}|s_t, a_t)}} \left[\sum_t r(s_t, a_t) + \mathcal{H}(q(\cdot|s_t)) \right]$$

Consider the second last time step

$$\arg\max_q \mathbb{E}_{s_{T-1} \sim q(s_{T-1})} \left[\mathbb{E}_{a_{T-1} \sim q(a_{T-1}|s_{T-1})} \left[Q(s_{T-1}, a_{T-1}) - \log q(a_{T-1}|s_{T-1}) \right] \right]$$

$$Q(s_{T-1}, a_{T-1}) = r(s_{T-1}, a_{T-1}) + \mathbb{E}_{s_T \sim p(s_T|s_{T-1}, a_{T-1})} [V(s_T)]$$

Referring back to the last time step math and pattern matching

$$V(s_{T-1}) = \log \int \exp(Q(s_{T-1}, a_{T-1})) da_{T-1}$$

$$Q(s_{T-1}, a_{T-1}) = r(s_{T-1}, a_{T-1}) + \mathbb{E}_{s_T \sim p(s_T|s_{T-1}, a_{T-1})} [V(s_T)]$$

$$q(a_{T-1}|s_{T-1}) = \exp(Q(s_{T-1}, a_{T-1}) - V(s_{T-1}))$$

Optimal policy is proportional to exponential advantage (soft-max)

Let's make it recursive

This suggests a recursive dynamic programming algorithm!

$$Q(s_T, a_T) = r(s_T, a_T)$$
$$V(s_T) = \log \int \exp(r(s_T, a_T)) da_T$$

For t = T-1 to 1:

$$Q_t(s_t, a_t) = r(s_t, a_t) + \mathbb{E}_{s_{t+1} \sim p(s_{t+1}|s_t, a_t)} [V_{t+1}(s_{t+1})]$$ (Bellman update)

$$V_t(s_t) = \log \int \exp(Q(s_t, a_t)) da_t$$ (Soft-max)

$$q(a_t|s_t) = \exp(Q_t(s_t, a_t) - V_t(s_t))$$ (Soft-max)

Very similar to the "soft" (entropy) Q-learning procedure from earlier lectures!

What does this suggest as an algorithm?

Optimize a "soft" Bellman equation

$$Q(s_t, a_t) \leftarrow r_t + \gamma \mathbb{E}_{s_{t+1} \sim p_s}\left[V(s_{t+1})\right]$$

$$V(s_t) \leftarrow \max_a Q(s_t, a)$$

$$\pi(a|s_t) \leftarrow \arg\max_a Q(s_t, a)$$

$$Q_{\text{soft}}(s_t, a_t) \leftarrow r_t + \gamma \mathbb{E}_{s_{t+1} \sim p_s}\left[V_{\text{soft}}(s_{t+1})\right]$$

$$V_{\text{soft}}(s_t) \leftarrow \alpha \log \int_{\mathcal{A}} \exp\left(\frac{1}{\alpha} Q_{\text{soft}}(s_t, a')\right) da'$$

$$\pi_{\text{soft}}(a|s_t) = \exp\left(\frac{1}{\alpha}\left(Q_{\text{soft}}(s_t, a) - V_{\text{soft}}(s_t)\right)\right)$$

Go from max to "softmax" (imagine if α goes to 0, it becomes a max)

Prevents premature collapse of exploration while smoothing out optimization landscape!

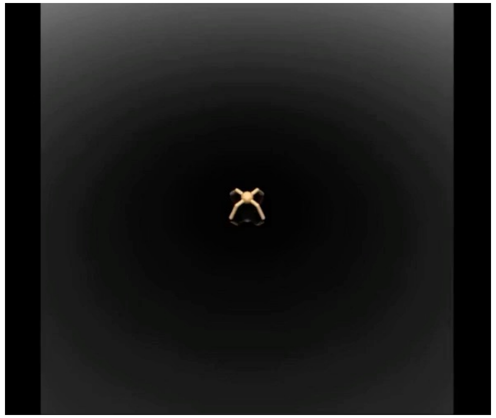
Why should we ever do soft-Q learning?

Optimization benefits

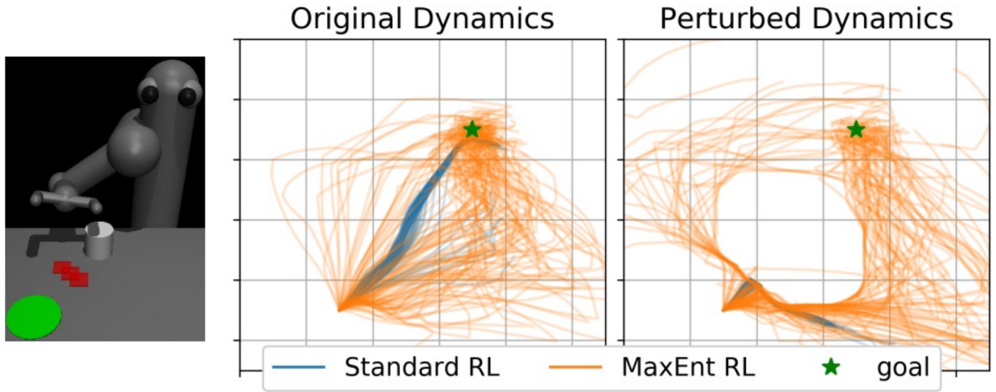
Corollary 5.1. *(Iteration complexity with log barrier regularization) Let* $\beta_\lambda := \frac{8\gamma}{(1-\gamma)^3} + \frac{2\lambda}{|\mathcal{S}|}$. *Starting from any initial* $\theta^{(0)}$, *consider the updates* (13) *with* $\lambda = \frac{\epsilon(1-\gamma)}{2\left\|\frac{d_\rho^{\pi^\star}}{\mu}\right\|_\infty}$ *and* $\eta = 1/\beta_\lambda$. *Then for all starting state distributions* ρ, *we have*

$$\min_{t<T} \left\{ V^\star(\rho) - V^{(t)}(\rho) \right\} \leq \epsilon \quad \textit{whenever} \quad T \geq \frac{320|\mathcal{S}|^2|\mathcal{A}|^2}{(1-\gamma)^6\,\epsilon^2} \left\| \frac{d_\rho^{\pi^\star}}{\mu} \right\|_\infty^2 .$$

Transfer



Deals better with misspecification



Ok so what did we show?

Find approximate posterior q(z|x) by optimizing the ELBO using dynamic programming

$$\max_q \mathbb{E}_{x\sim p(x)}\left[\mathbb{E}_{z\sim q(z|x)}\left[\log p(x|z)\right] - D_{KL}(q(z|x)||p(z))\right]$$

$$\mathbb{E}_{\substack{s_0\sim p(s_0)\\ a_t\sim q(a_t|s_t)\\ s_{t+1}\sim p(s_{t+1}|s_t,a_t)}}\left[\sum_t \log p(\mathcal{O}_t|s_t,a_t) - \log q(a_t|s_t)\right] = \mathbb{E}_{\substack{s_0\sim p(s_0)\\ a_t\sim q(a_t|s_t)\\ s_{t+1}\sim p(s_{t+1}|s_t,a_t)}}\left[\sum_t r(s_t,a_t) + \mathcal{H}(q(\cdot|s_t))\right]$$

Can derive a “soft” dynamic programming Q-learning update

For t = T-1 to 1:

$$Q_t(s_t,a_t) = r(s_t,a_t) + \mathbb{E}_{s_{t+1}\sim p(s_{t+1}|s_t,a_t)}\left[V_{t+1}(s_{t+1})\right]$$ (Bellman update)

$$V_t(s_t) = \log\int \exp(Q(s_t,a_t))da_t$$ (Soft-max)

$$q(a_t|s_t) = \exp(Q_t(s_t,a_t) - V_t(s_t))$$ (Soft-max)

Lecture outline

Control as Inference - Formulation

↓

Variational Inference

↓

Control as Inference to Derive Policy Gradient

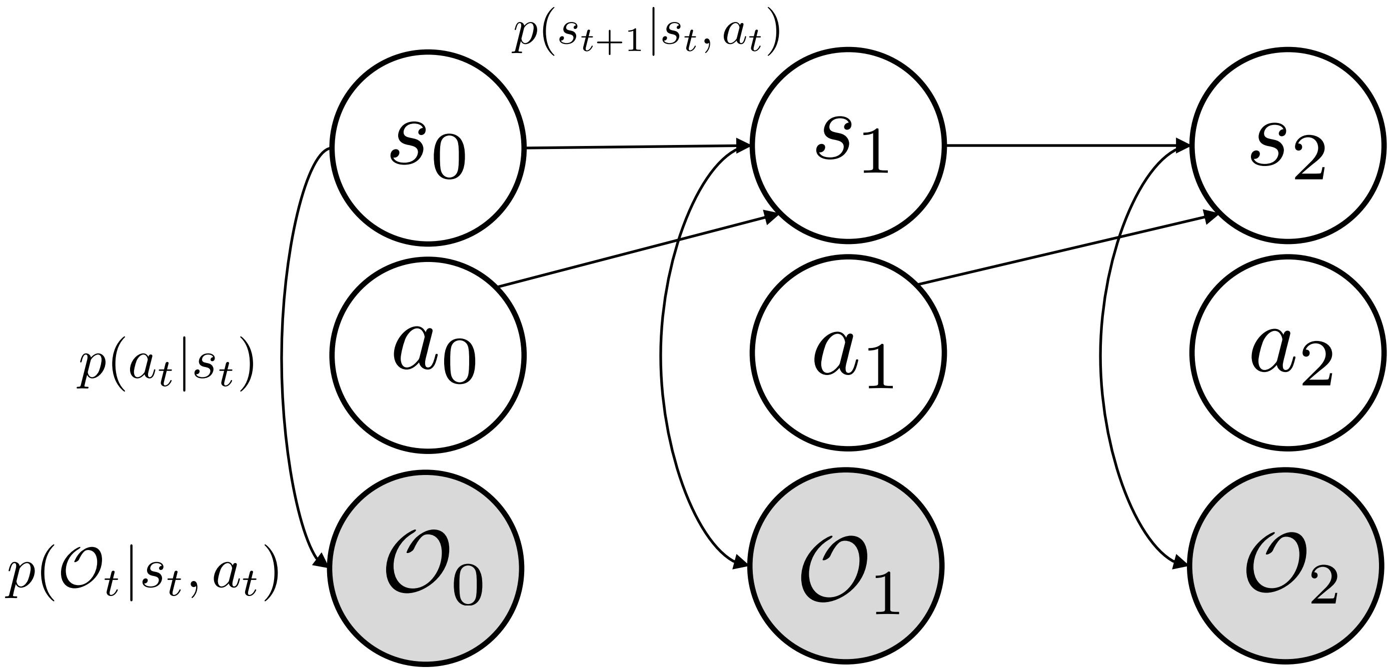
↓

Control as Inference to Derive Q-learning

↓

Control as Inference to Derive Model-Based RL

Let's back up from VI to max likelihood



$$p(\mathcal{O}_t|s_t, a_t) = \exp(r(s_t, a_t))$$

$$p(\tau|\mathcal{O}_{0:T} = 1) \propto p(\tau) \exp\left(\sum_{t=0}^{T} r(s_t, a_t)\right)$$

Let us assume we get a bunch of data of (s, a, s', r) from the true system p

We will try to learn a surrogate model $\widehat{p}$ to approximate p, use it for posterior inference

Model Learning via Maximum Likelihood

$$\min_{\hat{p}} D_{KL}(p(s_0, \ldots, s_T, a_0, \ldots, a_T, \mathcal{O}_0, \ldots, \mathcal{O}_T) || \hat{p}(s_0, \ldots, s_T, a_0, \ldots, a_T, \mathcal{O}_0, \ldots, \mathcal{O}_T))$$

↓ Definition of KLD

$$\max_{\hat{p}} \mathbb{E}_{p(s_0, \ldots, s_T, a_0, \ldots, a_T, \mathcal{O}_0, \ldots, \mathcal{O}_T)} [\log \hat{p}(s_0, \ldots, s_T, a_0, \ldots, a_T, \mathcal{O}_0, \ldots, \mathcal{O}_T)]$$

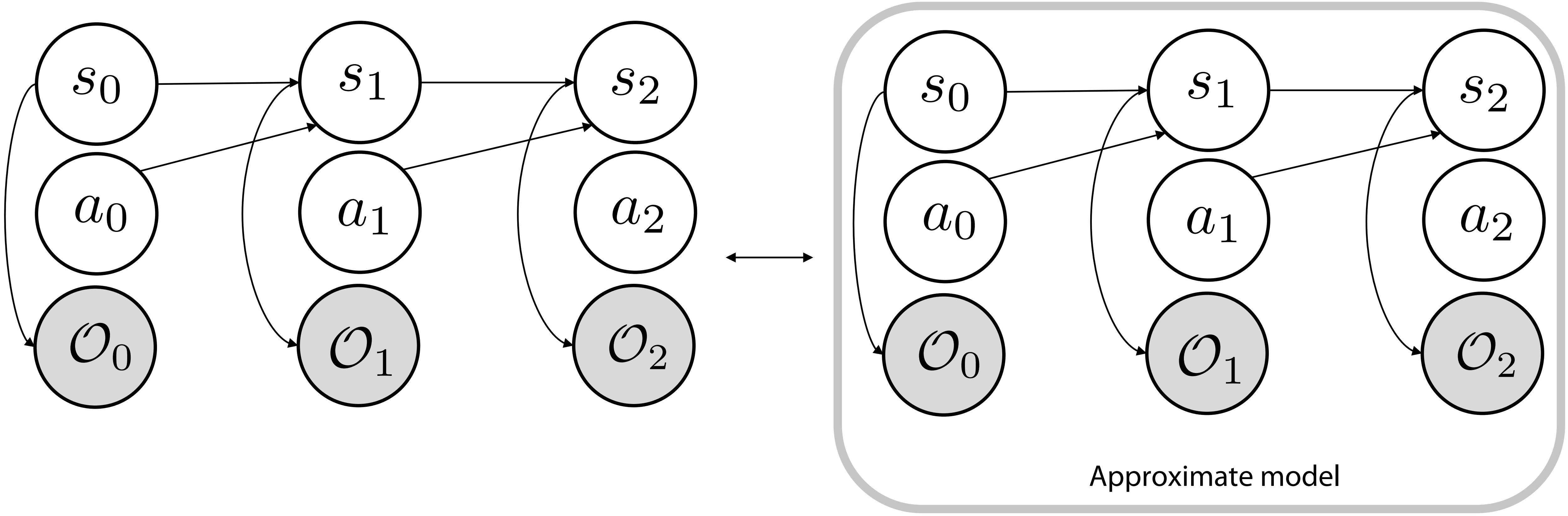
↓ Expansion of joint

$$\max_{\hat{p}} \mathbb{E}_{p(s_0, \ldots, s_T, a_0, \ldots, a_T, \mathcal{O}_0, \ldots, \mathcal{O}_T)} \left[\log \hat{p}(s_0) + \sum_t [\log \hat{p}(s_{t+1}|s_t, a_t) + \log \hat{p}(\mathcal{O}_t|s_t, a_t)] \right]$$

Model learning — Reward learning

Fitting $\hat{p}$ amounts to supervised learning on dynamics and rewards

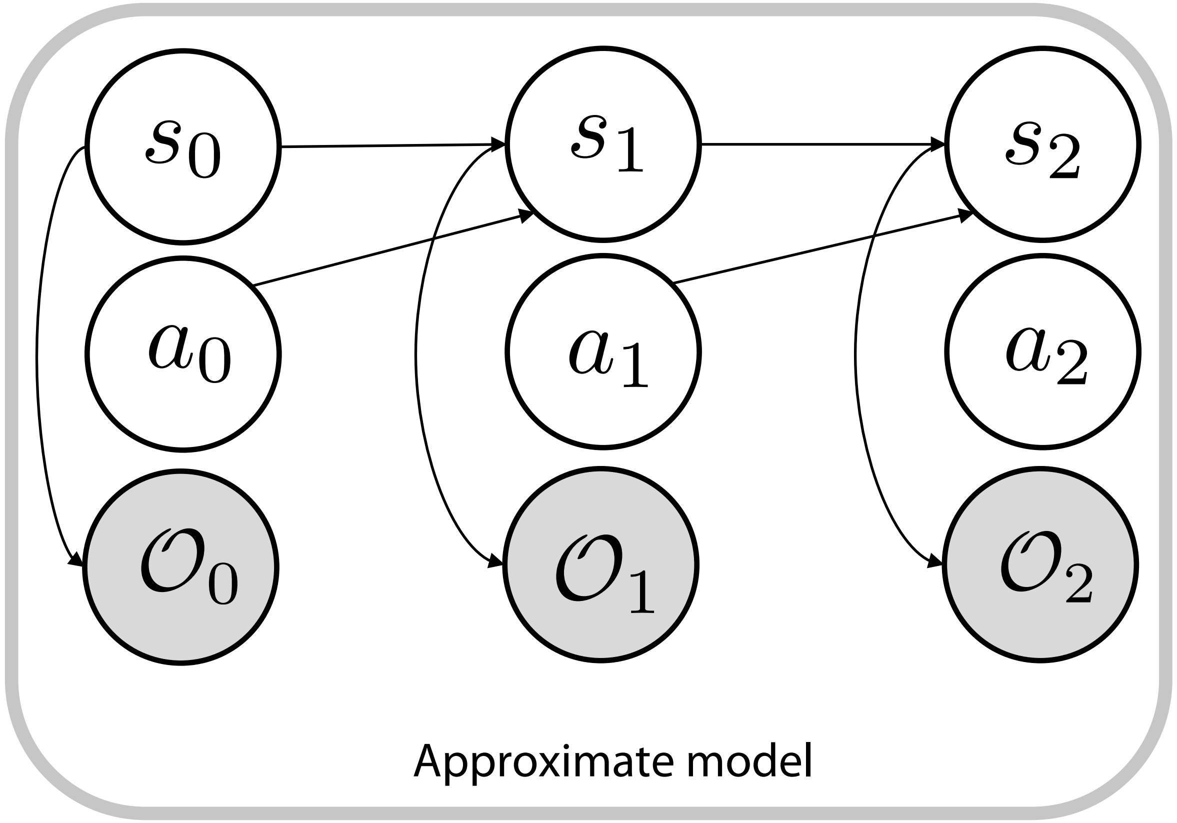
Model Learning via Maximum Likelihood



Fitting $\widehat{p}$ amounts to supervised learning on dynamics and rewards

How do we actually use this approximate model to obtain optimal actions?

Policy Extraction via Posterior Inference



Key idea: pretend that approximate model $\widehat{p}$ is the true model

↓

$$\hat{p}(a_t|s_t, \mathcal{O}_{t:T} = 1)$$

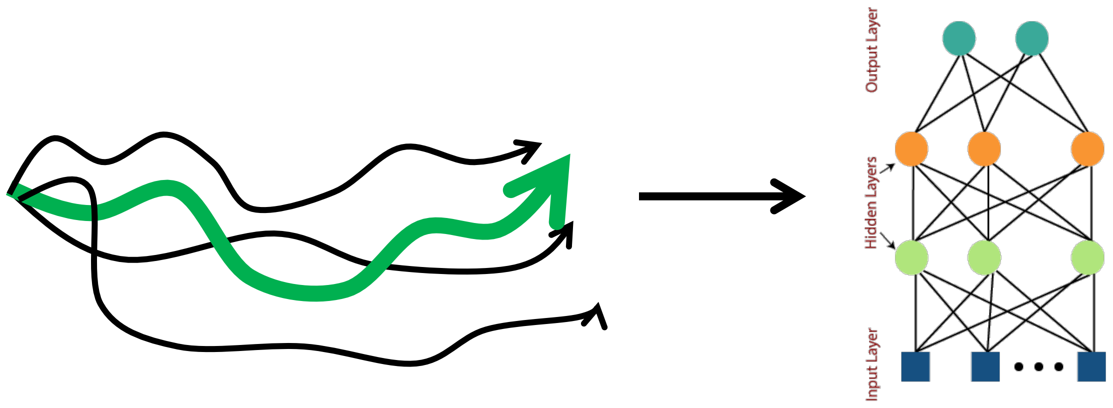
Just like in MFRL → perform posterior inference

Certainty equivalence ← But pretend that the model were true

Ok so how we do perform this inference?

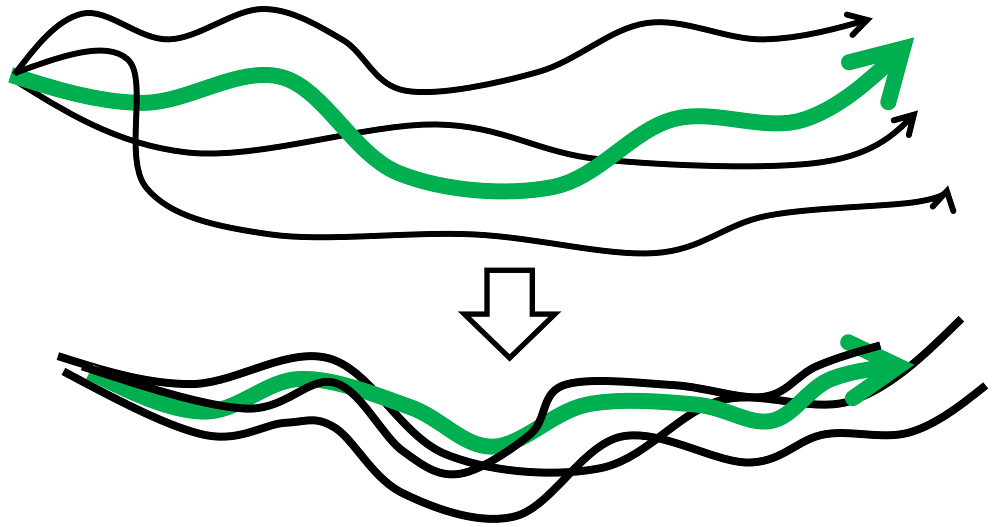
$$\hat{p}(a_t | s_t, \mathcal{O}_{t:T} = 1)$$

Idea 1: Variational inference in $\hat{p}$



Model-based policy optimization methods (Dyna ++)

Idea 2: Use Monte-Carlo Sampling for Inference



MPPI-style planning methods

Equivalence between posterior inference and MPPI

$$\hat{p}(a_t | s_t, \mathcal{O}_{t:T} = 1)$$

Let's expand out the nasty integrals with Bayes rule

$$= \frac{\hat{p}(a_t, s_t, \mathcal{O}_{t:T} = 1)}{\hat{p}(s_t, \mathcal{O}_{t:T} = 1)}$$

$$= \frac{\int \int \cdots \int \hat{p}(a_t, s_t, a_{t+1}, s_{t+1}, \ldots, a_T, s_T, \mathcal{O}_{t:T} = 1) ds_{t+1} da_{t+1} \ldots ds_T da_T}{\hat{p}(s_t, \mathcal{O}_{t:T} = 1)}$$

$$\propto \int \int \cdots \int \hat{p}(a_t, s_t, a_{t+1}, s_{t+1}, \ldots, a_T, s_T, \mathcal{O}_{t:T} = 1) ds_{t+1} da_{t+1} \ldots ds_T da_T$$

Equivalence between posterior inference and MPPI

$$\hat{p}(a_t|s_t, \mathcal{O}_{t:T} = 1)$$

$$\propto \int\int\cdots\int \hat{p}(a_t, s_t, a_{t+1}, s_{t+1}, \ldots, a_T, s_T, \mathcal{O}_{t:T} = 1) ds_{t+1} da_{t+1} \ldots ds_T da_T$$

Dynamics Action prior Optimality

$$\propto \int\int\cdots\int \hat{p}(s_0)\Pi_t \left[\hat{p}(s_{t+1}|s_t, a_t) p(a_t|s_t) p(\mathcal{O}_t|s_t, a_t)\right] ds_{t+1} da_{t+1} \ldots ds_T da_T$$

$$\propto \int\int\cdots\int \hat{p}(s_0)\Pi_t \left[\hat{p}(s_{t+1}|s_t, a_t) p(a_t|s_t)\right] \exp\left[\sum_t r(s_t, a_t)\right] ds_{t+1} da_{t+1} \ldots ds_T da_T$$

Substituting optimality definition $p(\mathcal{O}_t|s_t, a_t) = \exp(r(s_t, a_t))$

$$\propto \mathbb{E}_{\substack{s_0 \sim \hat{p}(s_0) \\ a_t \sim \hat{p}(a_t|s_t) \\ s_{t+1} \sim \hat{p}(s_{t+1}|s_t, a_t)}} \left[\exp\left[\sum_t r(s_t, a_t)\right]\right]$$

Just using definition of expectation

Equivalence between posterior inference and MPPI

$$\hat{p}(a_t|s_t, \mathcal{O}_{t:T} = 1)$$

$$\propto \mathbb{E}_{\substack{s_0 \sim \hat{p}(s_0) \\ a_t \sim \hat{p}(a_t|s_t) \\ s_{t+1} \sim \hat{p}(s_{t+1}|s_t, a_t)}} \left[\exp\left[\sum_t r(s_t, a_t) \right] \right]$$

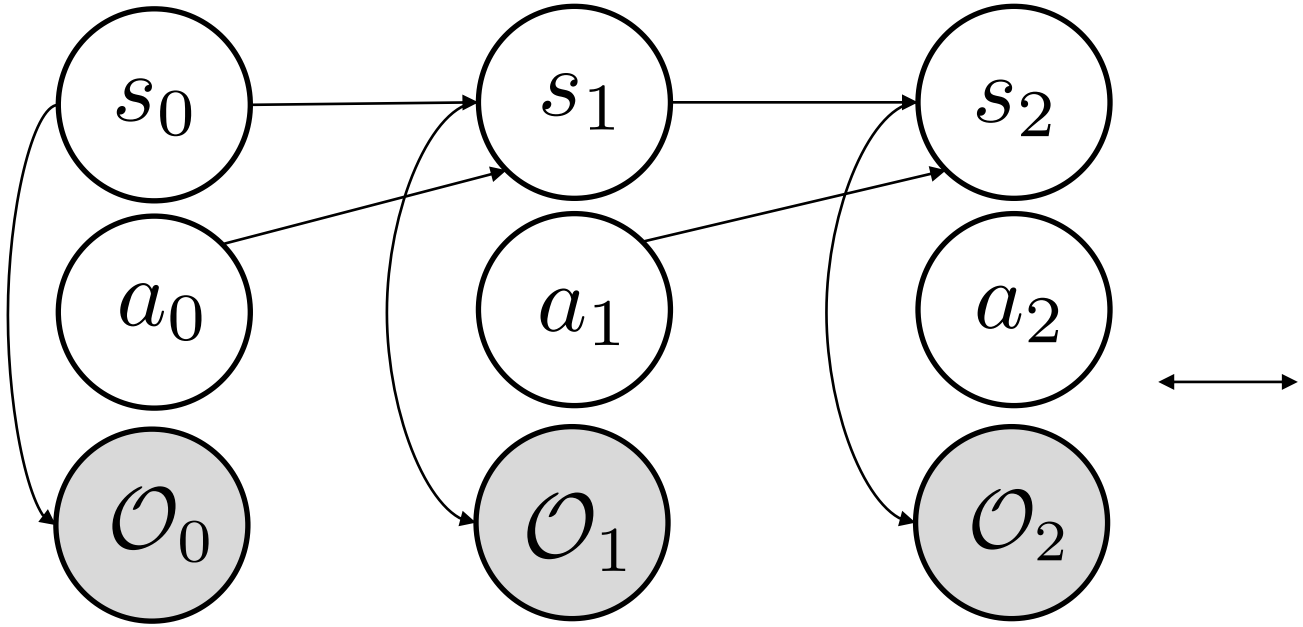
Taking a bunch of samples through model → choose actions proportional to the expected sum of rewards

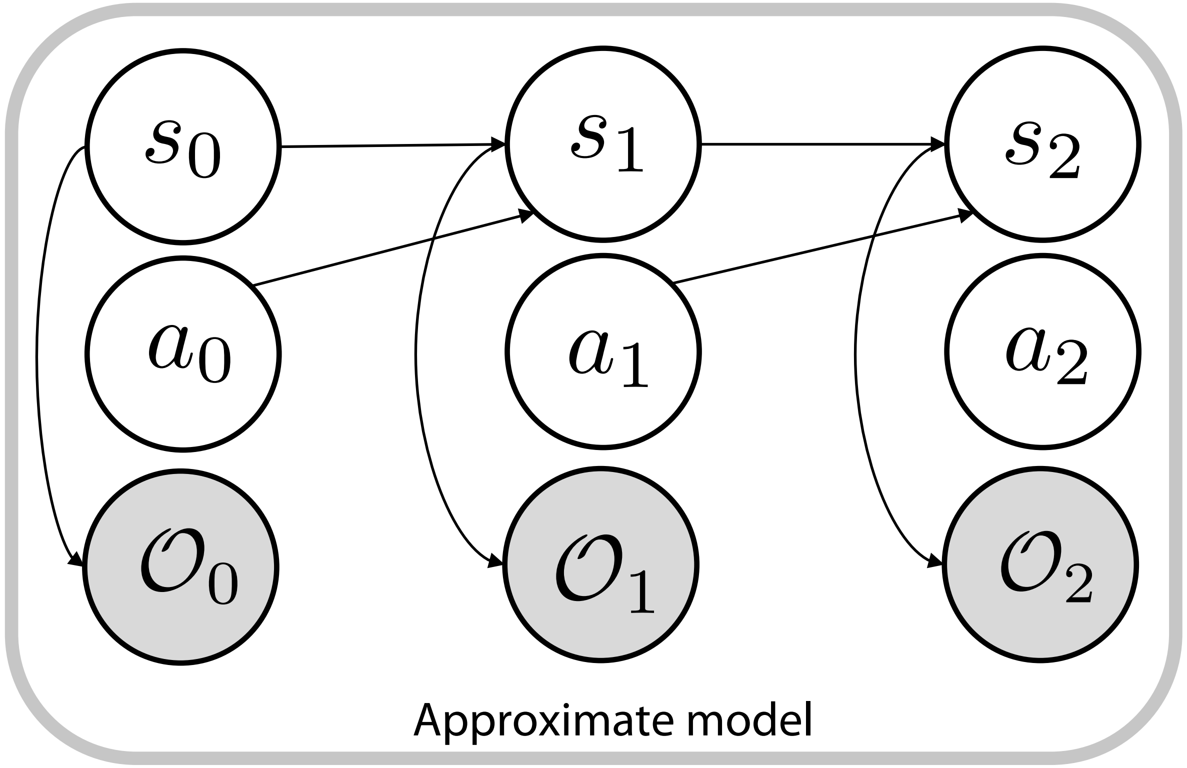
Can keep repeating with updated action prior

$$\propto \mathbb{E}_{\substack{s_0 \sim \hat{p}(s_0) \\ a_t \sim \hat{p}(a_t|s_t) \\ s_{t+1} \sim \hat{p}(s_{t+1}|s_t, a_t)}} \left[\exp\left[\sum_t r(s_t, a_t) \right] \right]$$

Can be thought of as a sampling-based Monte-Carlo approximation to posterior

Ok so what did we show?





Step 1: Learn model via min KL (supervised learning)

$$\max_{\hat{p}} \mathbb{E}_{p(s_0,\ldots,s_T,a_0,\ldots,a_T,\mathcal{O}_0,\ldots,\mathcal{O}_T)} \left[\log \hat{p}(s_0) + \sum_t \left[\log \hat{p}(s_{t+1}|s_t, a_t) + \log \hat{p}(\mathcal{O}_t|s_t, a_t)\right]\right]$$

Step 2: Obtain posterior actions via Monte-Carlo approximation (approx MPPI)

$$\propto \mathbb{E}_{\substack{s_0 \sim \hat{p}(s_0) \\ a_t \sim \hat{p}(a_t|s_t) \\ s_{t+1} \sim \hat{p}(s_{t+1}|s_t, a_t)}} \left[\exp\left[\sum_t r(s_t, a_t)\right]\right]$$

Lecture outline

Control as Inference - Formulation

↓

Variational Inference

↓

Control as Inference to Derive Policy Gradient

↓

Control as Inference to Derive Q-learning

↓

Control as Inference to Derive Model-Based RL

Class Structure