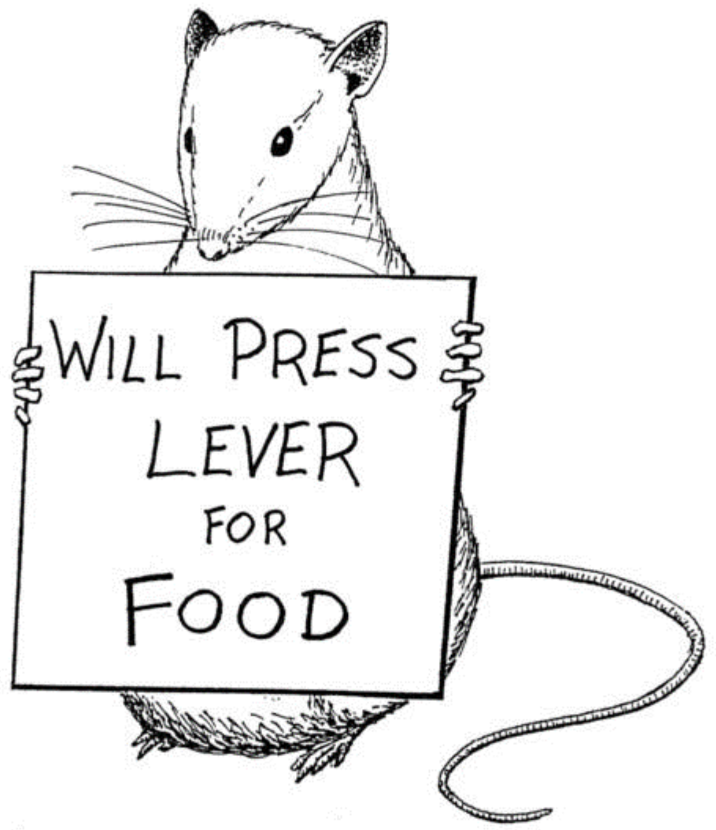
CS 277: Control and Reinforcement Learning

Winter 2022

Lecture 15: Bounded RL (cont.)

Roy Fox
Department of Computer Science
Bren School of Information and Computer Sciences
University of California, Irvine



Logistics

evaluations

- Course evaluations due end of next week, March 13

assignments

- Assignment 4 due Friday

Today's lecture

Bounded RL

Bounded RL methods

Abstractions

Bounded optimality

- Bounded optimizer = trades off value and divergence from prior $\pi_0(a \mid s)$

$$\max_\pi \mathbb{E}_{(s,a)\sim p_\pi}[r(s,a)] - \tau \mathbb{D}[\pi \| \pi_0] = \max_\pi \mathbb{E}_{(s,a)\sim p_\pi}\left[\beta r(s,a) - \log \frac{\pi(a \mid s)}{\pi_0(a \mid s)}\right]$$

- $\beta = \frac{1}{\tau}$ is the tradeoff coefficient between value and relative entropy
  - Similar to the inverse-temperature in thermodynamics
  - As $\beta \to 0$, the agent will fall back to the prior $\pi \to \pi_0$
  - As $\beta \to \infty$, the agent will be a perfect value optimizer $\pi \to \pi^*$
- We'll see reasons to have finite β

Simplifying assumption

- MaxEnt IRL was approximate because it violated dynamical constraints
  - $p_\pi(\xi) \propto \pi_0(\xi)\exp(R(\xi))$, regardless of trajectory feasibility
- For simplicity, let's do the same for RL
  - Suppose the environment is fully controllable $s_{t+1} = a_t$
  - Bellman equation:

$$V^*_\beta(s) = \max_\pi \mathbb{E}_{(s'|s)\sim\pi}\left[r(s) - \frac{1}{\beta}\log\frac{\pi(s'|s)}{\pi_0(s'|s)} + \gamma V^*_\beta(s')\right]$$

$$= r(s) - \frac{1}{\beta}\min_\pi \mathbb{D}\left[\pi \,\middle\|\, \frac{\pi_0(s'|s)\exp(\beta\gamma V^*_\beta(s'))}{Z'_\beta(s)}\right] + \frac{1}{\beta}\log Z'_\beta(s)$$

Linearly-Solvable MDPs (LMDPs)

- Optimal policy for $V_\beta(s) = r(s) - \frac{1}{\beta} \min_\pi \mathbb{D}\left[\pi \,\middle\|\, \frac{\pi_0(s'|s)\exp(\beta\gamma V_\beta(s'))}{Z'_\beta(s)}\right] + \frac{1}{\beta}\log Z'_\beta(s)$:
  - Soft-greedy policy: $\pi_\beta(s'|s) \propto \pi_0(s'|s)\exp(\beta\gamma V_\beta(s'))$
- Value recursion: $V_\beta(s) = r(s) + \frac{1}{\beta}\log Z'_\beta(s) = r(s) + \frac{1}{\beta}\log \mathbb{E}_{(s'|s)\sim\pi_0}[\exp(\beta\gamma V_\beta(s'))]$

$$Z_\beta(s) = \exp(\beta V_\beta(s)) = \exp(\beta r(s)) Z'_\beta(s) = \exp(\beta r(s))\mathbb{E}_{(s'|s)\sim\pi_0}[Z^\gamma_\beta(s')]$$

- In the undiscounted case $\gamma = 1$, with $D = \mathrm{diag}(\exp \beta r)$: $z = DP_0 z$
- We can solve for z, and therefore π, by finding a right-eigenvector of DP_0

Z-learning

$$Z(s) = \exp(\beta r(s))\mathbb{E}_{(s'|s)\sim\pi_0}[Z^\gamma(s')]$$

- We can do the same model-free:
  - Given experience (s, r, s') sampled by the prior policy π_0
  - Update $Z(s) \to \exp(\beta r)Z^\gamma(s')$
- Full-controllability condition ($s_{t+1} = a_t$) can be relaxed to allow $\pi_0(s'|s) = 0$
  - But we still allow any transition distribution $\pi(s'|s)$ over the remaining support
  - Later: the general case, $p(s'|s) = \sum_a \pi(a|s)p(s'|s,a)$

Duality between value and log prob

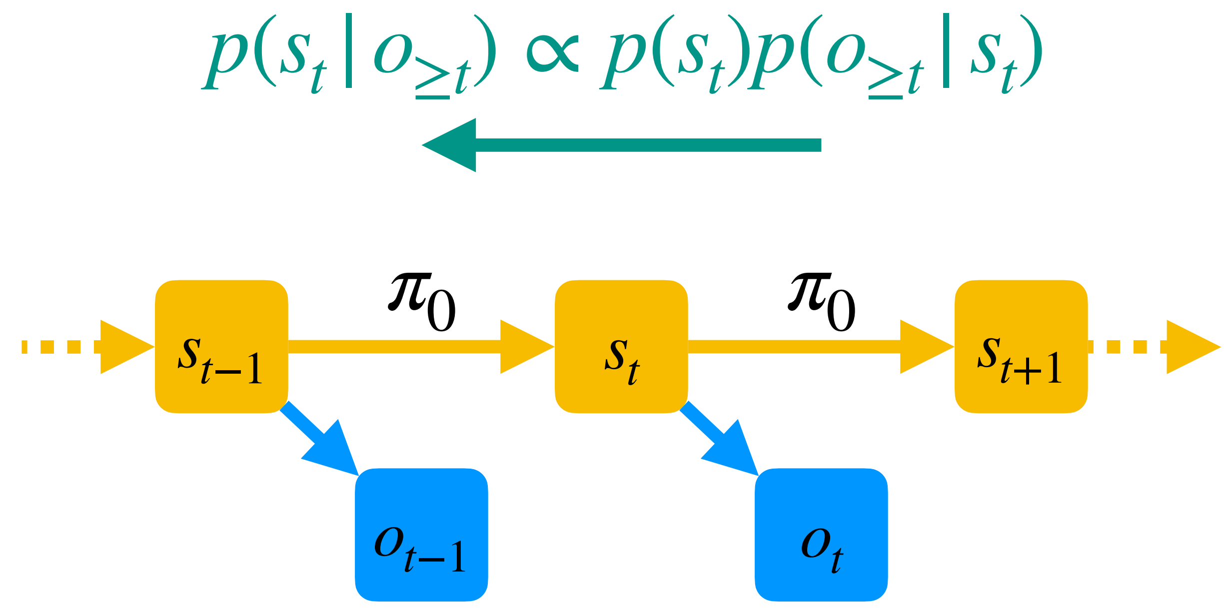
- We've seen many cases where log-probs play the role of reward / value
  - Or values the role of logits (unnormalized log-probs)
- Examples:
  - In LQG, $\log p(x \mid \hat{x}) = -\frac{1}{2} x^\intercal \Sigma x + \text{const}$; costs / values are quadratic
  - In value-based algorithms, good exploration policy: $\pi(a \mid s) = \operatorname{softmax}_a \beta Q(s, a)$
  - Imitation Learning can be viewed as RL with $r(s, a) = \log p^*(a \mid s)$
  - In IRL, a reward function can be viewed as a discriminator $D(s) = \exp(-r(s))$

Full-controllability duality

- Bounded control in LMDP: $Z(s) = \exp(\beta r(s))\mathbb{E}_{(s'|s)\sim\pi_0}[Z^\gamma(s')]$

- Backward filtering in a partially observable system with dynamics $\pi_0(s'|s)$

$$p(o_{\geq t}|s_t) = p(o_t|s_t)\mathbb{E}_{(s_{t+1}|s_t)\sim\pi_0}[p(o_{\geq t+1}|s_{t+1})]$$



- Equivalent if $Z(s) = p(o_{\geq t}|s_t)$ and $\exp(\beta r(s)) = p(o|s)$
  - Intuition: find states that give good reward ⇔ high likelihood of observations

- Exact equivalence only in the fully-controllable case
  - Partially controllable case takes more nuanced analysis

Bounded RL

- Back to the general case: $\max_\pi \mathbb{E}_{(s,a)\sim p_\pi}[\beta r(s,a)] - \mathbb{D}[\pi \| \pi_0]$

- Define an entropy-regularized Bellman optimality operator

$$\mathscr{T}[V](s) = \max_\pi \mathbb{E}_{(a|s)\sim\pi}\left[r(s,a) - \frac{1}{\beta}\log\frac{\pi(a|s)}{\pi_0(a|s)} + \gamma\mathbb{E}_{(s'|s,a)\sim p}[V(s')]\right]$$

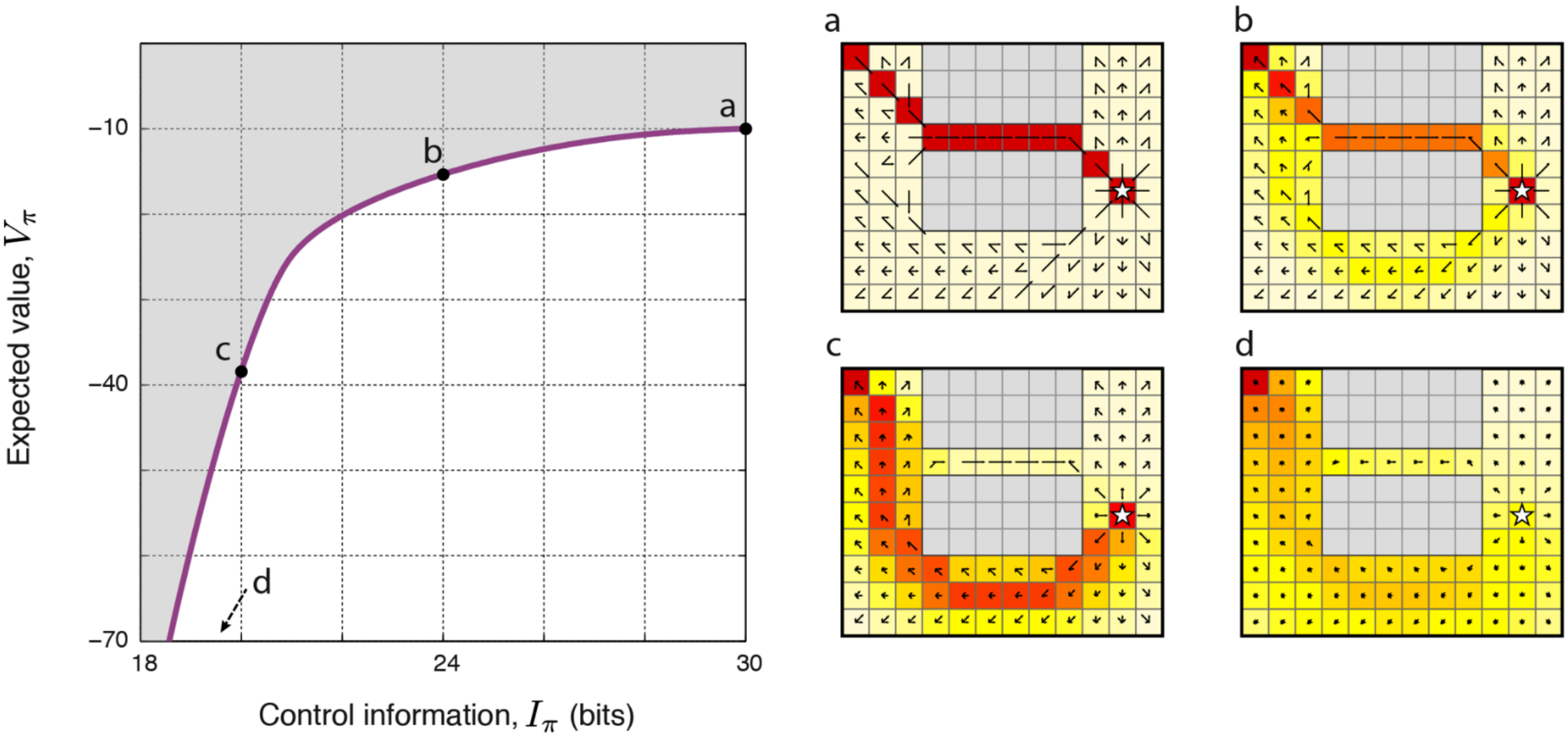
  - As in the unbounded case $\beta \to \infty$, this operator is contracting

- Soft-optimal policy:

$$\pi(a|s) \propto \pi_0(a|s)\exp\beta(r(s,a) + \gamma\mathbb{E}_{(s'|s,a)\sim p}[V(s')]) = \pi_0(a|s)\exp\beta Q(s,a)$$

- Soft-optimal value recursion: $V(s) = \frac{1}{\beta}\log Z(s) = \frac{1}{\beta}\log\mathbb{E}_{(a|s)\sim\pi_0}[Q(s,a)]$

Value–RelEnt curve



[Rubin et al., 2012]

Today's lecture

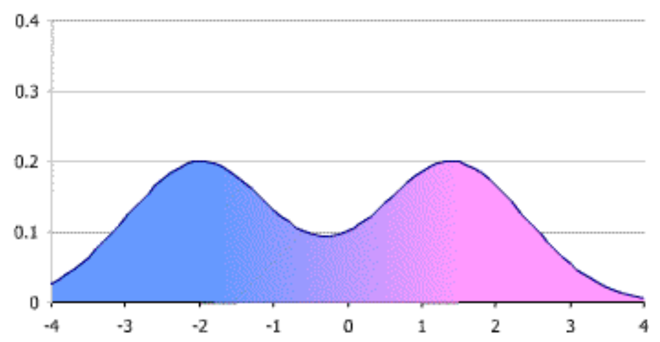
Bounded RL

Bounded RL methods

Abstractions

Exact and approximate inference

- Suppose we want to max log-likelihood of a dataset $\max_\theta \mathbb{E}_{x\sim D}[\log p_\theta(x)]$
  - And easier to compute with latent intermediate variable $p_\theta(z)p_\theta(x|z)$



- Expectation–Gradient (EG): $\nabla_\theta \log p_\theta(x) = \mathbb{E}_{(z|x)\sim p_\theta}[\nabla_\theta \log p_\theta(z,x)]$
- But what if sampling from the exact posterior $p_\theta(z|x)$ is also hard?
- Let's do importance sampling from any approximate posterior $q_\phi(z|x)$

$$\log p_\theta(x) = \log \mathbb{E}_{(z|x)\sim q_\phi}\left[\frac{p_\theta(z)}{q_\phi(z|x)} p_\theta(x|z)\right] \geq \mathbb{E}_{(z|x)\sim q_\phi}\left[\log \frac{p_\theta(z,x)}{q_\phi(z|x)}\right]$$

Variational Inference (VI): Evidence Lower Bound (ELBO)

- Two ways of decomposing $p_\theta(z, x)$:

$$\log p_\theta(x) \geq -\mathbb{D}[q_\phi(z|x) \| p_\theta(z, x)]$$

$$= \log p_\theta(x) + \mathbb{E}_{(z|x)\sim q_\phi}\left[\log \frac{p_\theta(z|x)}{q_\phi(z|x)}\right]$$

$$= \mathbb{E}_{(z|x)\sim q_\phi}\left[\log \frac{p_\theta(z)}{q_\phi(z|x)} + \log p_\theta(x|z)\right]$$

- Bounding gap: $\mathbb{D}[q_\phi(z|x) \| p_\theta(z|x)] \geq 0$
  - Smaller the better the guide $q_\phi(z|x)$ approximates $p_\theta(z|x)$
- Bound (RHS) can be computed efficiently as a proxy for our objective

Control as inference

- Consider soft "success" indicators (assuming $r \leq 0$)

$$p(v_t = 1 \mid s_t, a_t) = \exp \beta r(s_t, a_t)$$

- What is the log-probability that an entire trajectory ξ "succeeds"?

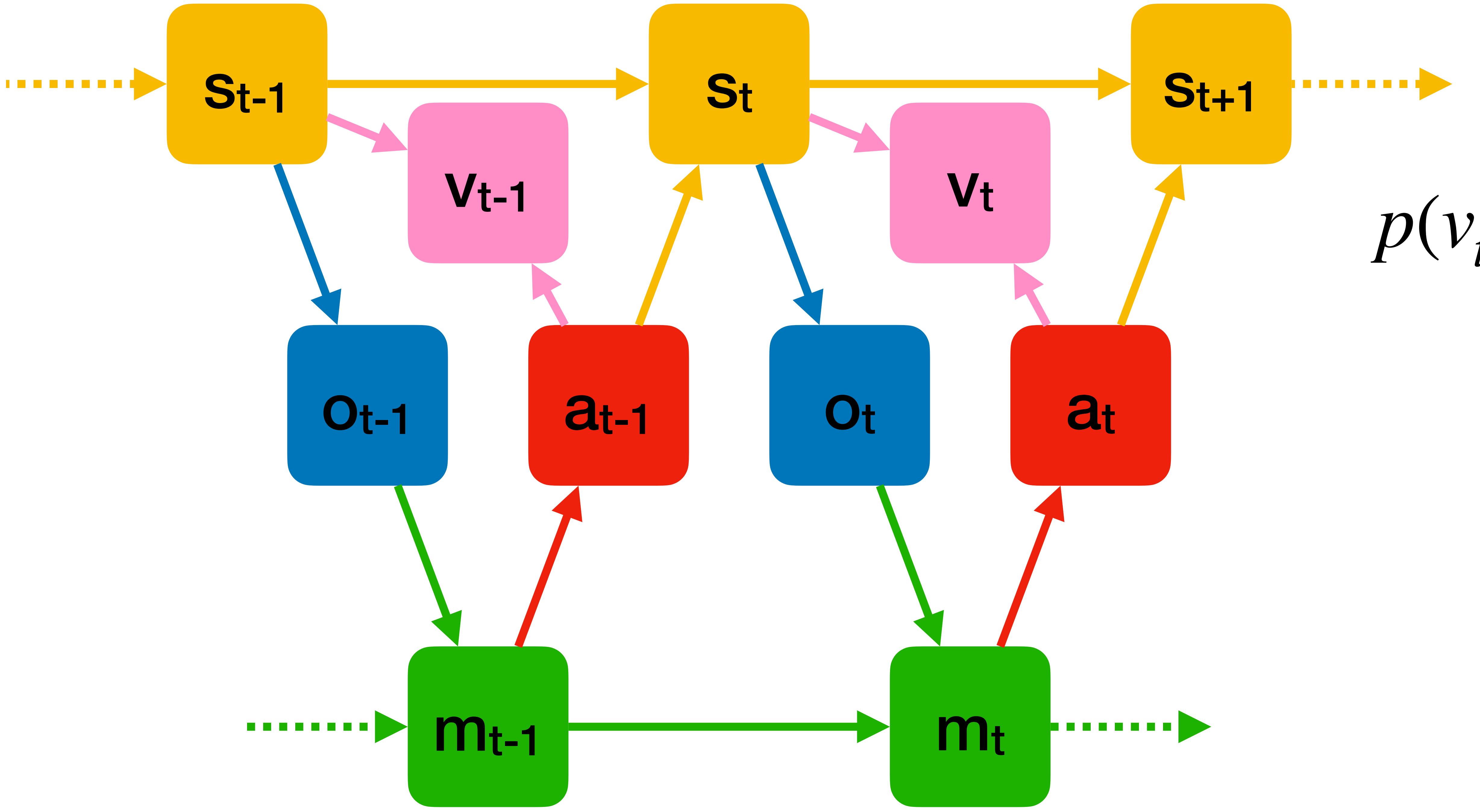
$$\log p(\mathscr{V} \mid \xi) = \sum_t \log p(v_t = 1 \mid s_t, a_t) = \beta \sum_t r(s_t, a_t) = \beta R(\xi)$$

- What is the posterior distribution over trajectories, given success?

$$p(\xi \mid \mathscr{V}) = \frac{p_0(\xi) p(\mathscr{V} \mid \xi)}{p_0(\mathscr{V})} = \frac{p_0(\xi) \exp \beta R(\xi)}{Z}$$

  - But this distribution is not realizable, due to dynamical constraints

Pseudo-observations



$$p(v_t = 1 \,|\, s_t, a_t) = \exp \beta r_t$$

General duality between VI and bounded RL

- In VI, take $x = \mathscr{V}$, $z = \xi$, and $p_\theta(\xi) = p_0(\xi)$ (fix generator to prior)
- Optimize the ELBO with a realizable guide distribution $q_\phi(\xi \,|\, \mathscr{V}) = p_{\pi_\phi}(\xi)$
- The ELBO becomes:

$$\mathbb{E}_{(\xi|\mathscr{V})\sim q_\phi}\left[\log p_0(\mathscr{V} \,|\, \xi) + \log \frac{p_0(\xi)}{q_\phi(\xi \,|\, \mathscr{V})}\right] = \mathbb{E}_{\xi\sim p_{\pi_\phi}}\left[\beta R(\xi) - \log \frac{p_{\pi_\phi}(\xi)}{p_0(\xi)}\right]$$

$$= \mathbb{E}_{(s,a)\sim p_{\pi_\phi}}\left[\beta r(s,a) - \log \frac{\pi_\phi(a \,|\, s)}{\pi_0(a \,|\, s)}\right]$$

  - Equivalent to the bounded RL problem! (a.k.a.: MaxEnt RL, energy-based RL)

Soft Q-Learning (SQL)

- MaxEnt Bellman operator:

$$\mathcal{T}[Q](s,a) = r(s,a) + \gamma \mathbb{E}_{(s'|s,a)\sim p} \max_{\pi} \left[-\frac{1}{\beta} \log \frac{\pi(a'|s')}{\pi_0(a'|s')} + Q(s',a') \right]$$

- Maximum achieved for soft-optimal policy, soft-optimal value recursion

- With tabular parametrization: $Q(s,a) \to r + \frac{\gamma}{\beta} \log \mathbb{E}_{(a'|s')\sim\pi_0}[\exp \beta Q(s',a')]$

- With differentiable parametrization:

$$L_\theta(s,a,r,s') = (r + \frac{\gamma}{\beta} \log \mathbb{E}_{(a'|s')\sim\pi_0}[\exp \beta Q_{\bar{\theta}}(s',a')] - Q_\theta(s,a))^2$$

  - As $\beta \to \infty$, this becomes (Deep) Q-Learning

Soft Actor–Critic (SAC)

- Optimally: $\pi(a\,|\,s) = \dfrac{\pi_0(a\,|\,s)\exp\beta Q(s,a)}{\exp\beta V(s)}$ $\qquad V(s) = Q(s,a) - \dfrac{1}{\beta}\log\dfrac{\pi(a\,|\,s)}{\pi_0(a\,|\,s)}$

- In continuous action spaces, we can't explicitly softmax $Q(s,a)$ over a

- We can train a critic off-policy

$$L_\phi(s,a,r,s',a') = \left(r + \gamma\left(Q_{\bar\phi}(s',a') - \frac{1}{\beta}\log\frac{\pi_\theta(a'\,|\,s')}{\pi_0(a'\,|\,s')}\right) - Q_\phi(s,a)\right)^2$$

- And a soft-greedy actor = imitate the critic

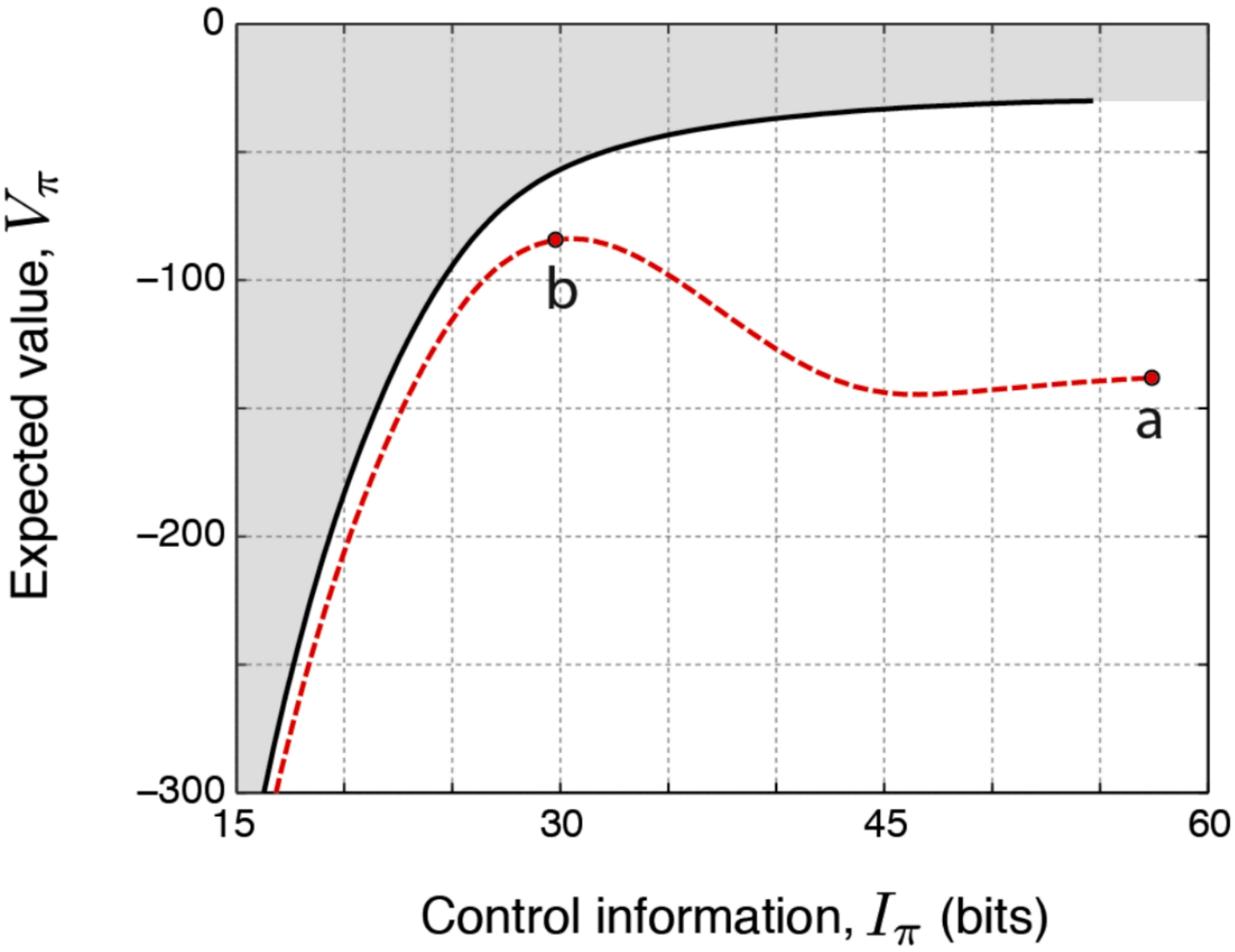
$$L_\theta(s) = \mathbb{E}_{(a|s)\sim\pi_\theta}[\log\pi_\theta(a\,|\,s) - \log\pi_0(a\,|\,s) - \beta Q_\phi(s,a)]$$

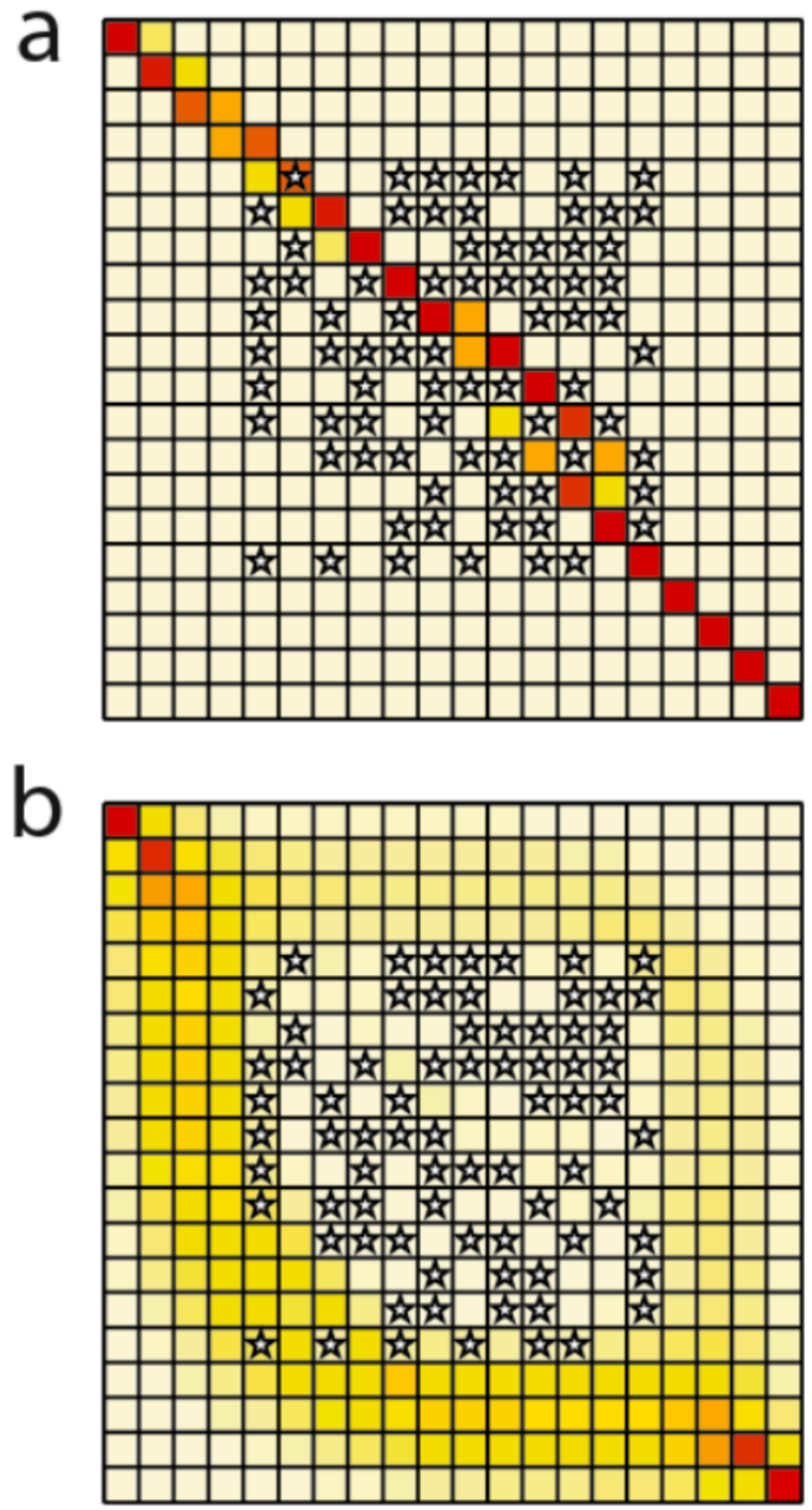
- Can optimize $\beta = \dfrac{1}{\tau}$ to match a target entropy $L_\tau(s,a) = -\tau\log\pi_\theta(a\,|\,s) - \tau H$

Why use a finite β

- Model suboptimal agents / teachers
- Robustness to model misspecification / avoid overfitting
- With uncertainty in Q, eliminate bias due to winner's curse
  - For $\beta \to \infty$: positive bias $\mathbb{E}[\max_a Q(a)] \geq \max_a \mathbb{E}[Q(a)]$
  - For $\beta \to 0$: negative bias $\mathbb{E}[\mathbb{E}_{a \sim \pi_0}[Q(a)]] = \mathbb{E}_{a \sim \pi_0}[\mathbb{E}[Q(a)]] \leq \max_a \mathbb{E}[Q(a)]$
  - Somewhere in between there must be an unbiased β
- Robustness to non-stationary environment, multi-agent, etc.

Robustness to model uncertainty





Recap

- We can model bounded rationality with KL cost to diverge from prior π_0
- Equivalent to a form of variational inference
- Can be optimized with Soft Q-Learning (SQL)
  - In continuous action spaces, Soft Actor–Critic (SAC)
- Value–entropy trade-off coefficient β shouldn't be annealed too fast
  - Schedule with a target entropy or by other principles

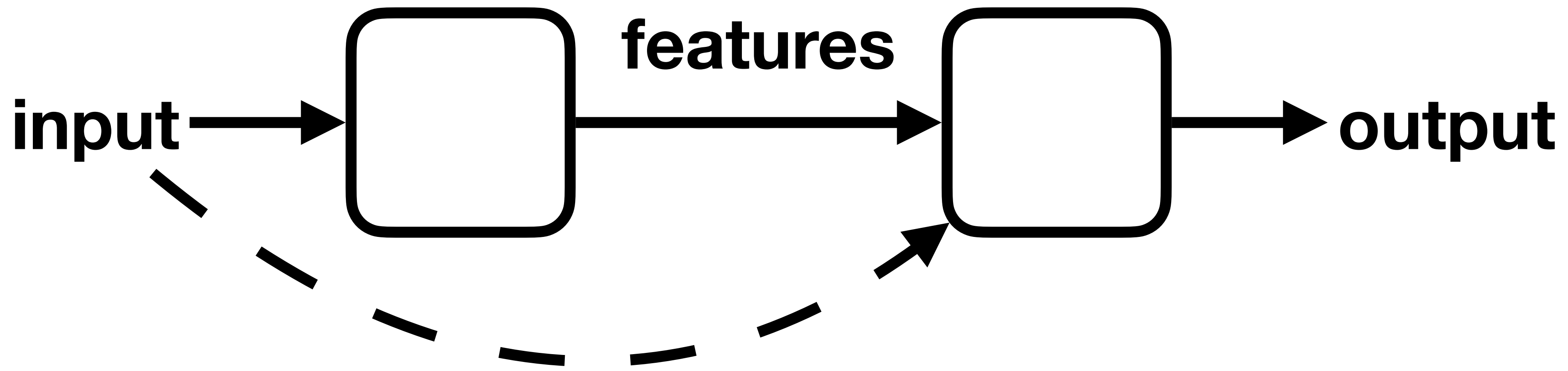
Today's lecture

Bounded RL

Bounded RL methods

Abstractions

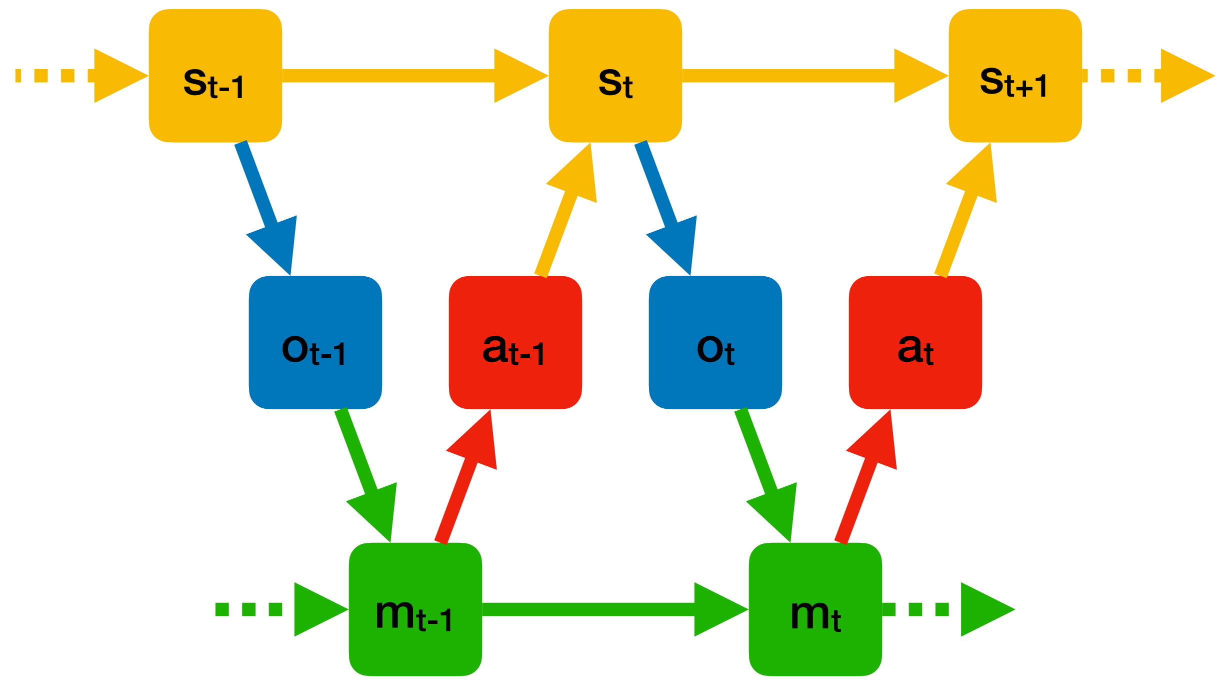
Abstractions in learning



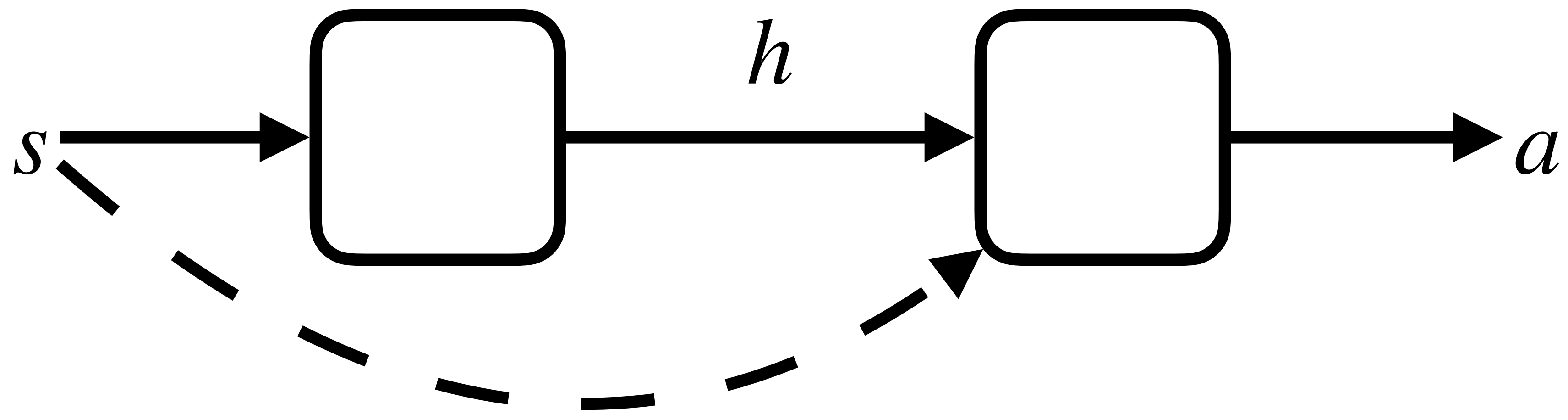
- Abstraction = succinct representation
  - Captures high-level features, ignores low-level
  - Can be programmed or learned
  - Can improve sample efficiency, generalization, transfer
- Input abstraction (in RL: state abstraction)
  - Allow downstream processing to ignore irrelevant input variation
- Output abstraction (in RL: action abstraction)
  - Allow upstream processing to ignore extraneous output details

Abstractions in sequential decision making

- Spatial abstraction: each decision has state / action abstraction
  - Easier to decide based on high-level state features (e.g. objects, not pixels)
  - Easier to make big decisions first, fill in the details later
- Temporal abstraction: abstractions can be remembered
  - No need to identify objects from scratch in every frame
    - High-level features can ignore fast-changing, short-term aspects
  - No need to make the big decisions again in every step
    - Focus on long-term planning, shorten the effective horizon

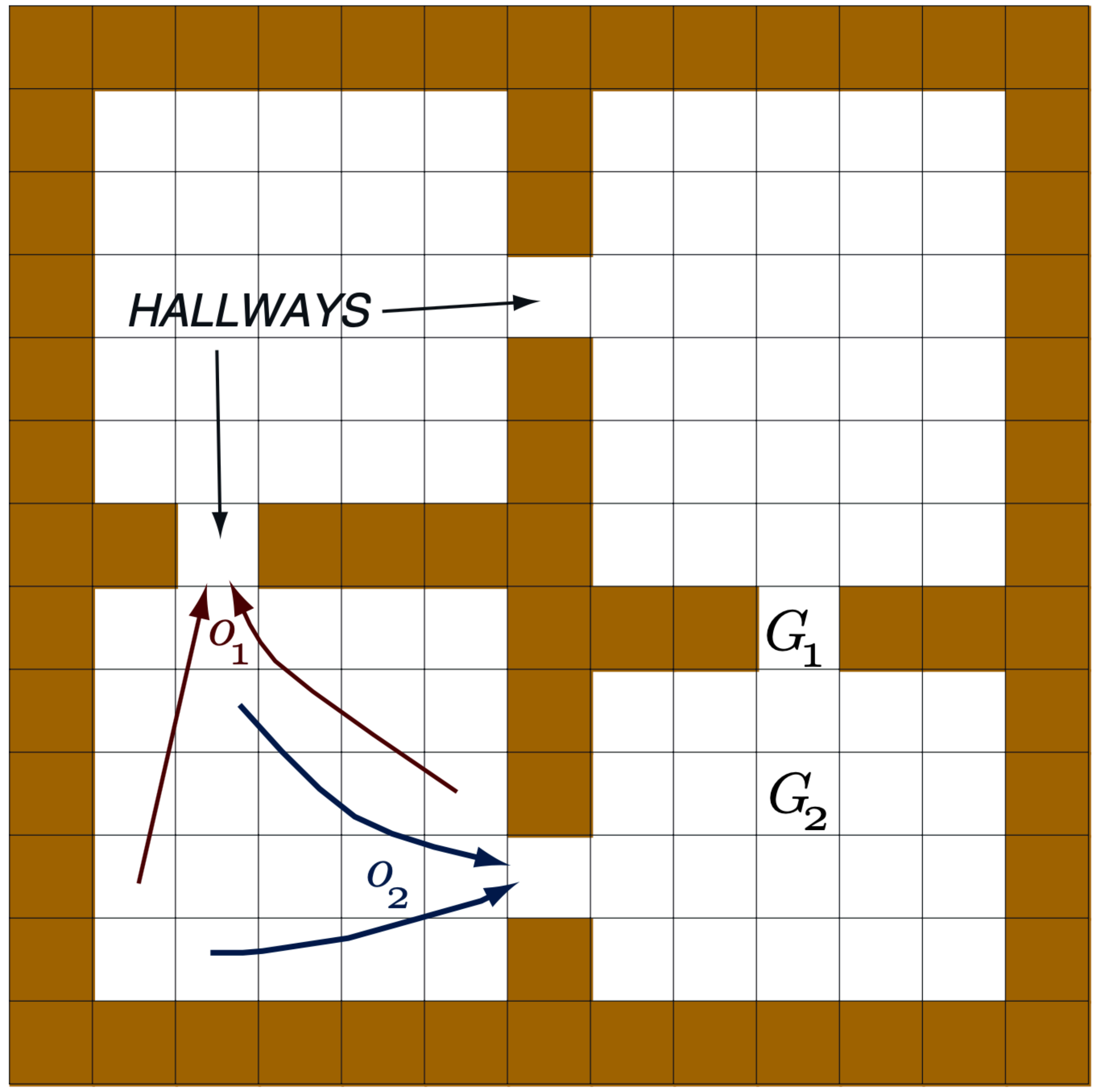


Options framework

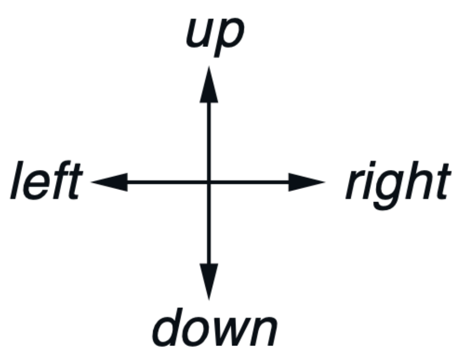


- Option = persistant action abstraction
  - High-level policy = select the active option $h \in \mathcal{H}$
  - Low-level option = "fills in the details", select action $\pi_h(a \mid s)$ every step
- When to switch the active option h?
  - Idea: option has some subgoal = postcondition it tries to satisfy
  - Option can detect when the subgoal is reached (or failed to be reached)
    - As part of deciding what action to take otherwise
  - $\implies$ the option terminates $\implies$ the high-level policy selects new option

Four-room example



4 stochastic primitive actions

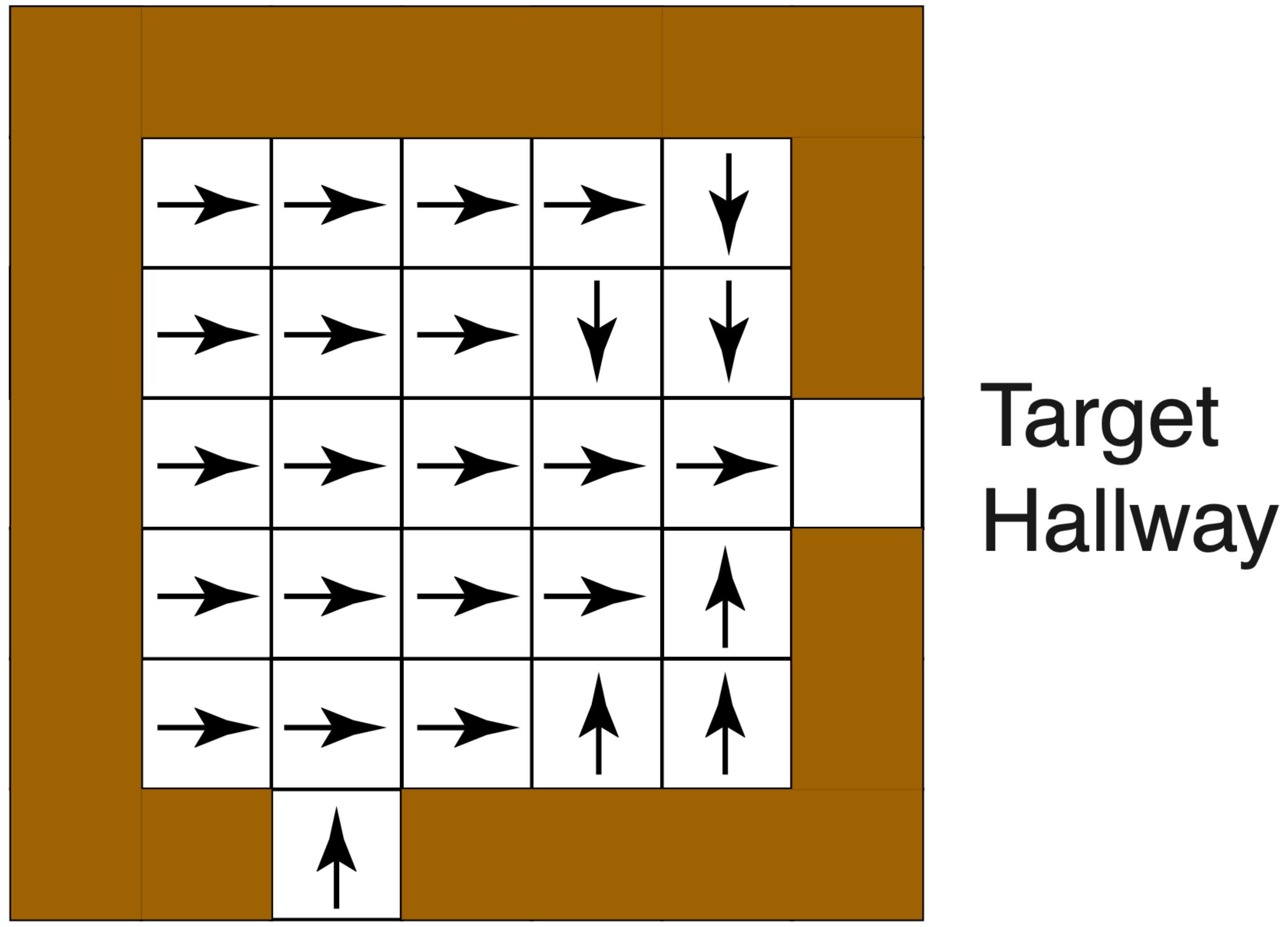


Fail 33% of the time

8 multi-step options

(to each room's 2 hallways)

one of the 8 options:



Options framework: definition

- Option: tuple $\langle \mathcal{I}_h, \pi_h, \beta_h \rangle$
  - The option can only be called in its initiation set $s \in \mathcal{I}_h$
  - It then takes actions according to policy $\pi_h(a|s)$
  - After each step, the policy terminates with probability $\beta_h(s)$
- Equivalently, define policy over extended action set $\pi_h : \mathcal{S} \to \Delta(\mathcal{A} \cup \{\bot\})$
- Initiation set can be folded into option-selection meta-policy $\pi_\bot : \mathcal{S} \to \Delta(\mathcal{H})$
- Together, $\pi_\bot$ and $\{\pi_h\}_{h \in \mathcal{H}}$ form the agent policy