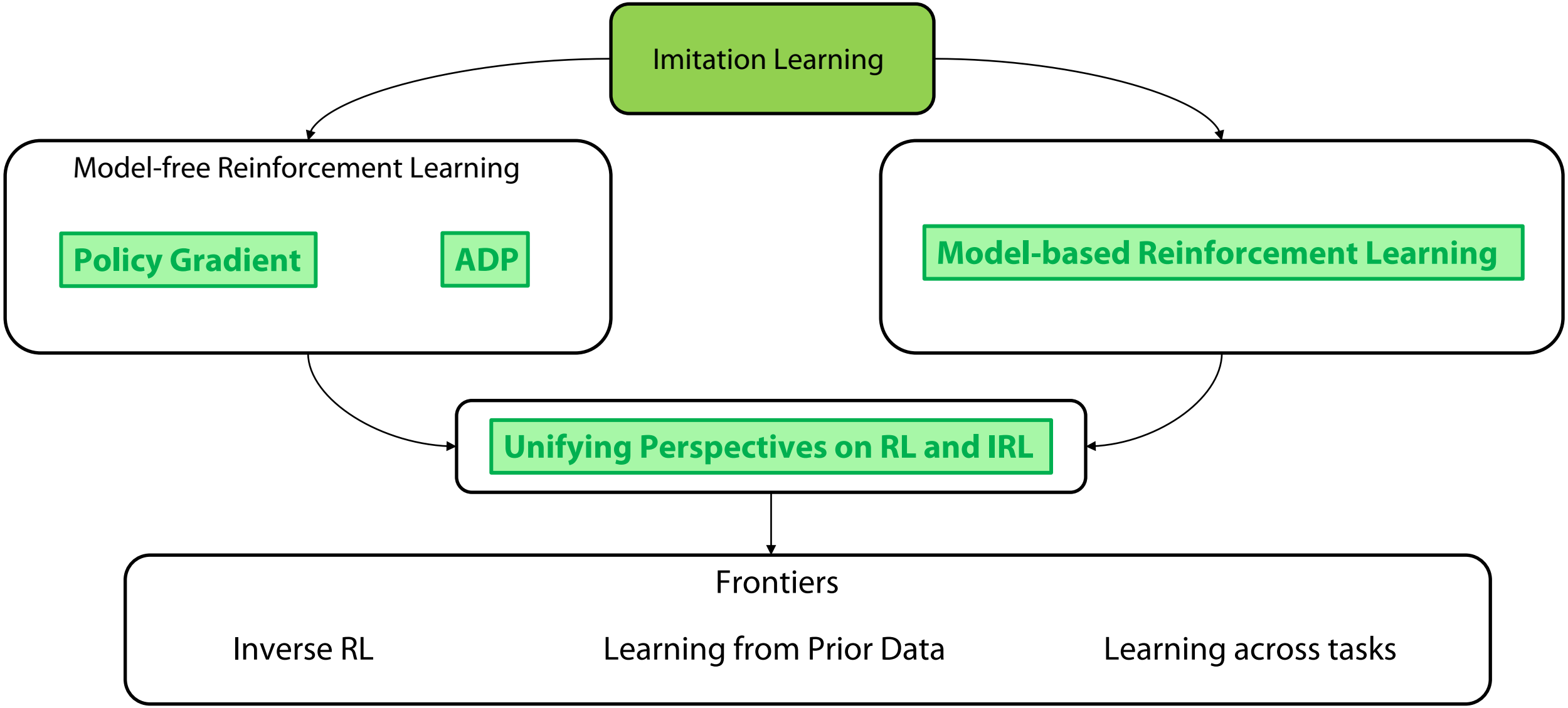
Reinforcement Learning
Spring 2026

Abhishek Gupta

TA: Mateo Guaman Castro

Class Structure



Lecture Outline

Why Imitation? + Problem formulation

↓

IRLv1 – max margin planning

↓

IRLv2 – max entropy IRL

↓

IRLv3 – partial policy optimization

↓

IRLv4 – adversarial IRL

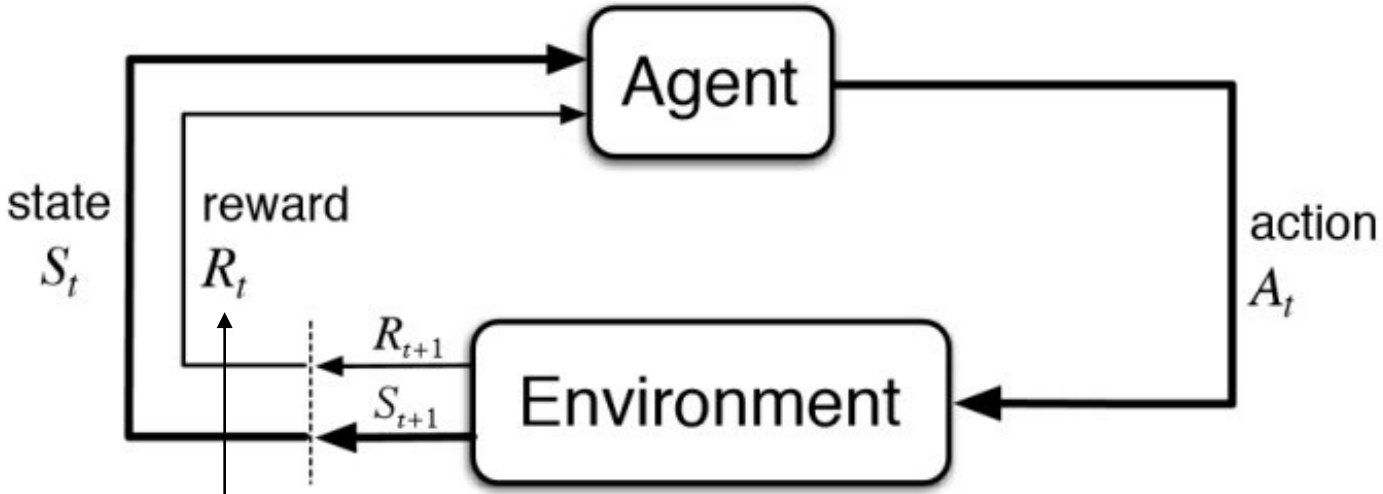
IRLv5 – non-adversarial IRL

Learning from Demonstrations

Avoid manual reward specification by learning from demos of optimal behavior

$\mathcal{D} = \{s_0^i, a_0^i, s_1^i, a_1^i, \ldots, s_T^i, a_T^i\}_{i=1}^N$

Demos of expert behavior

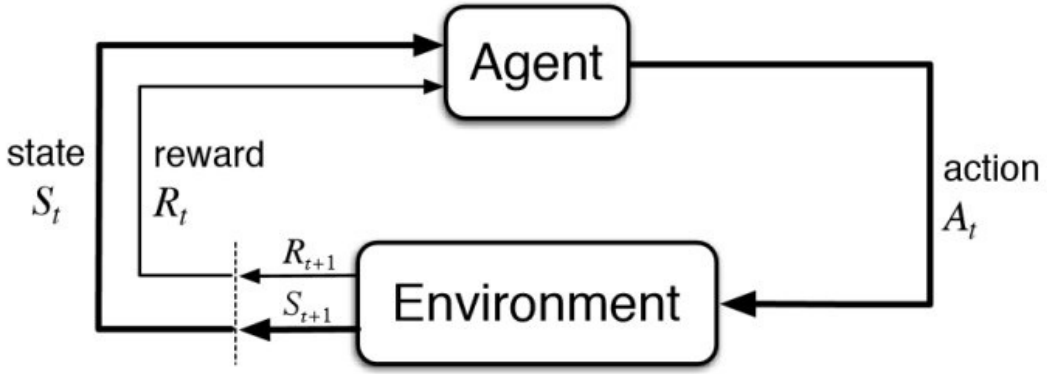


Inverse Reinforcement Learning

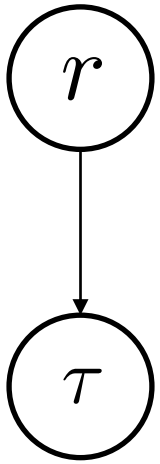
Infer rewards from demonstrations of optimal behavior

→ Optimal behavior optimizes some reward via RL, so the RL process must be "inverted" to find the reward

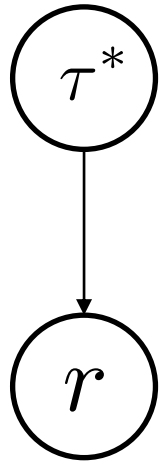
Why is this "inverse" reinforcement learning?



RL: Rewards generate trajectories



IRL: Expert trajectories generate rewards

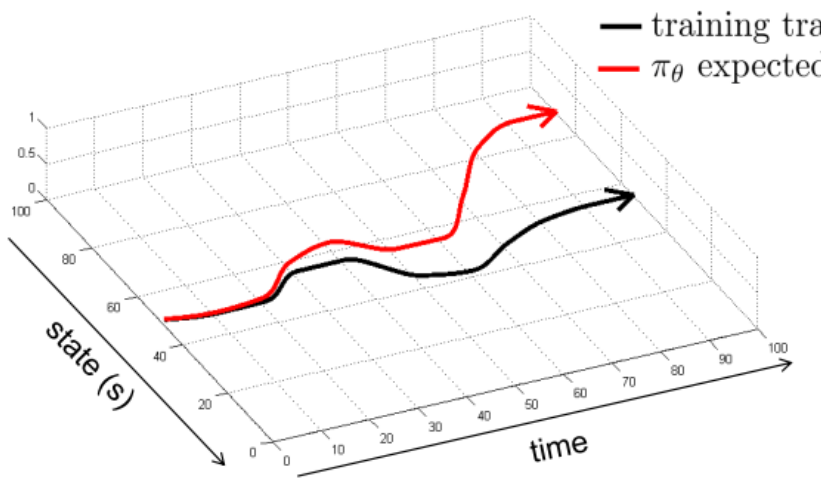


Is this well defined?

But haven't we already learned from demonstrations?

Imitation learning via Behavior Cloning (L2)

$$\arg\max_{\theta} \mathbb{E}_{(s^*,a^*)\sim\mathcal{D}} \left[\log \pi_\theta(a^*|s^*)\right]$$



Main difference between BC and IRL:

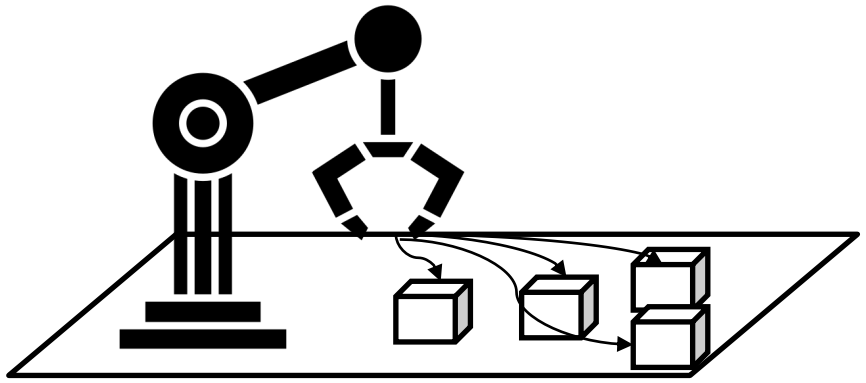
1. BC learns policies, IRL learns rewards
2. BC assumes no environment access, IRL typically assumes either known model or sampling access

Why does this matter?

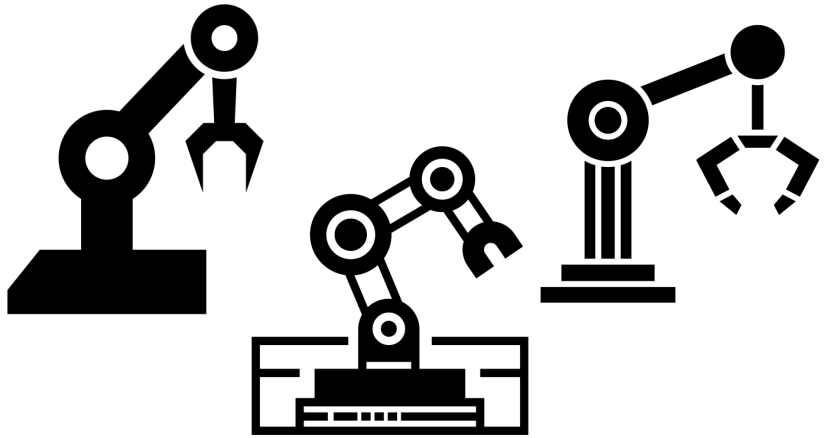
Zooming out – why do we care about imitation?

Imitation learning is all about generalization

Generalization across states



Generalization across dynamics



Covariate shift is just a manifestation of generalization

What if learning something else generalized better than policies?

IRL problem statement + assumptions

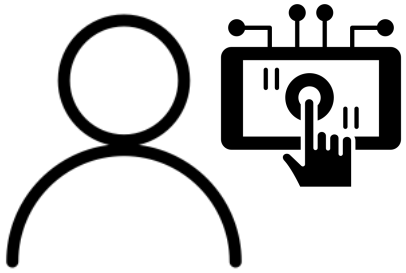
Reinforcement Learning

State: Known
Action: Known
Transition Dynamics: Unknown but can sample
Reward: **Known**
Expert policy: Unknown
Expert traces: **Unknown**

Inverse Reinforcement Learning

State: Known
Action: Known
Transition Dynamics: Unknown but can sample
Reward: **Unknown**
Expert policy: Unknown
Expert traces: **Known**

Find r that **explains** the demonstrator behavior as noisily optimal



Inverse RL

Reward
$r_\theta(s,a)$

Reinforcement Learning

Policy
$\pi(a|s)$

New dynamics/state

Lecture Outline

Why Imitation? + Problem formulation

↓

IRLv1 – max margin planning

↓

IRLv2 – max entropy IRL

↓

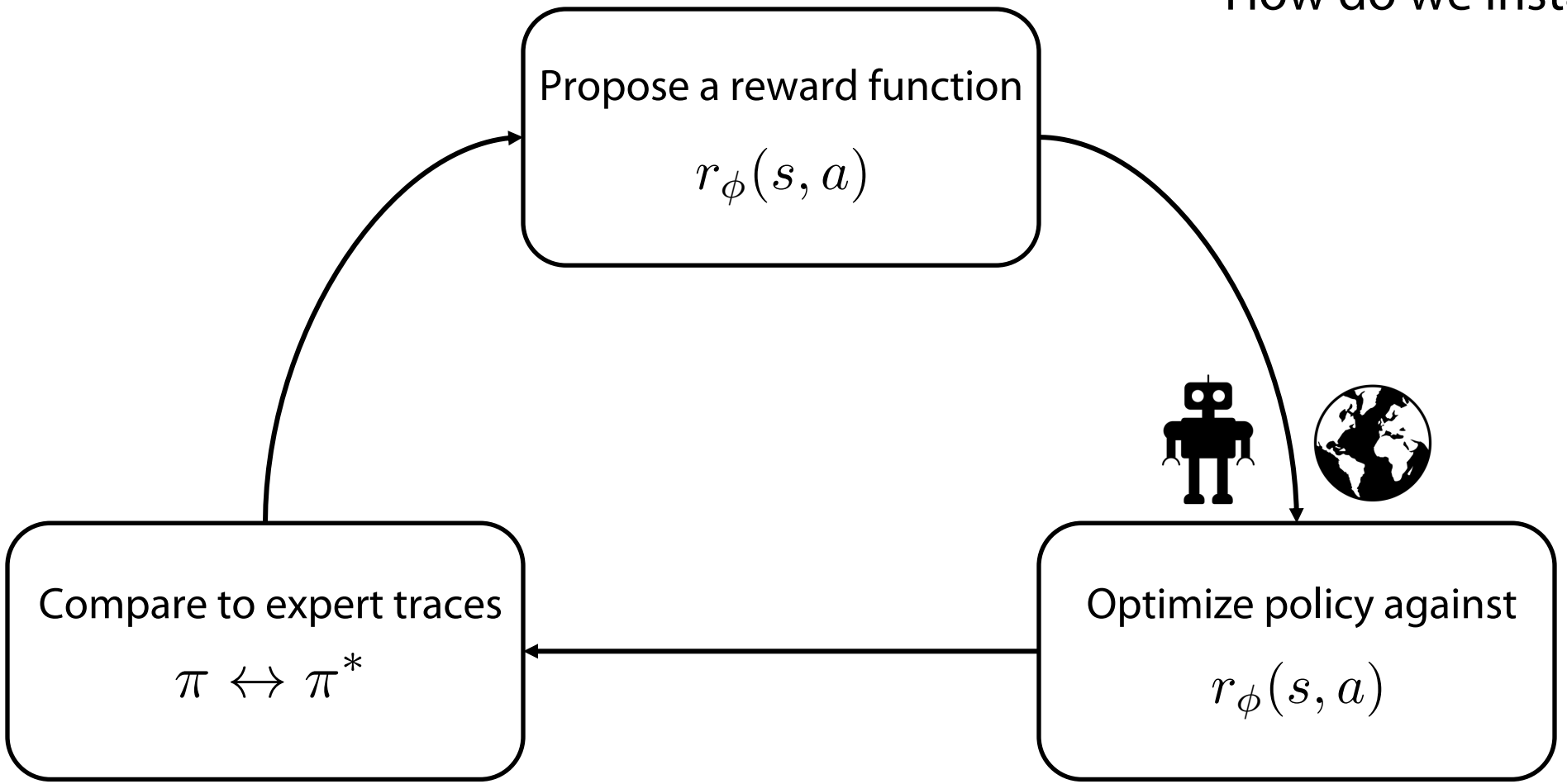
IRLv3 – partial policy optimization

↓

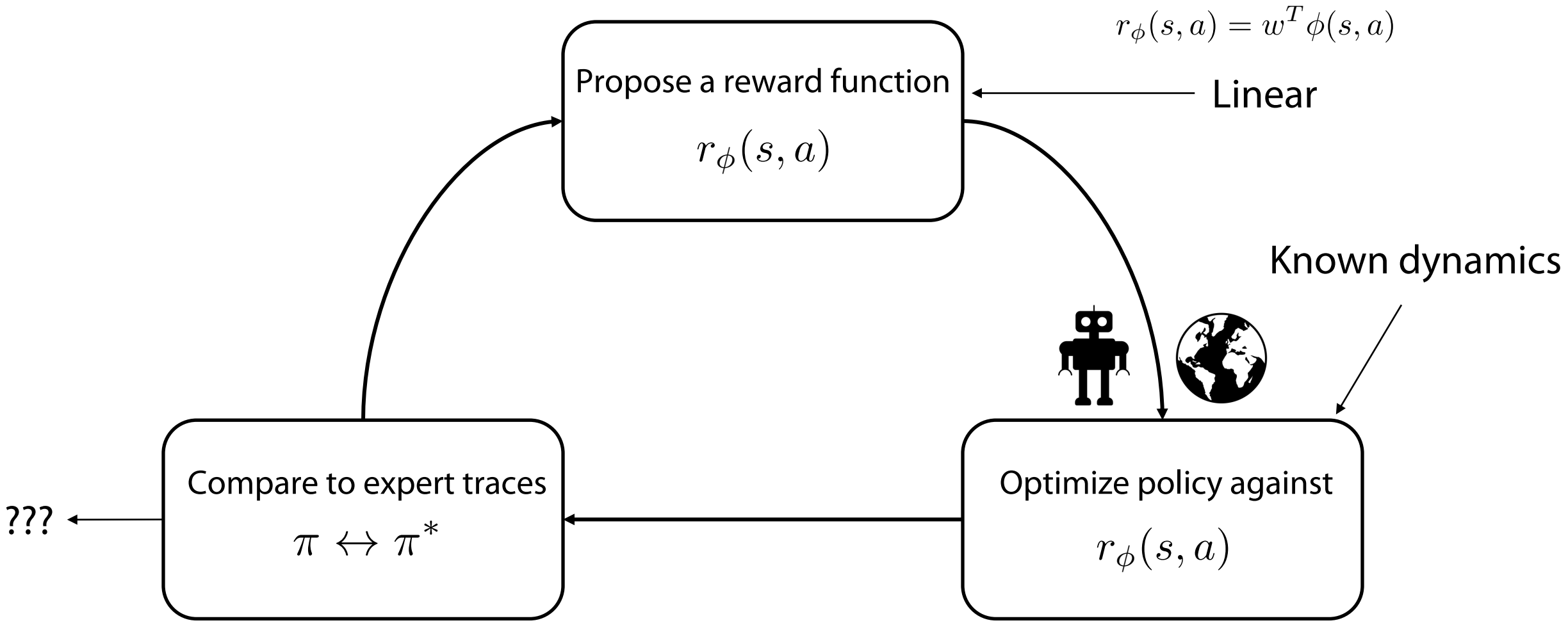
IRLv4 – adversarial IRL

A Formula for Inverse Reinforcement Learning

How do we instantiate?



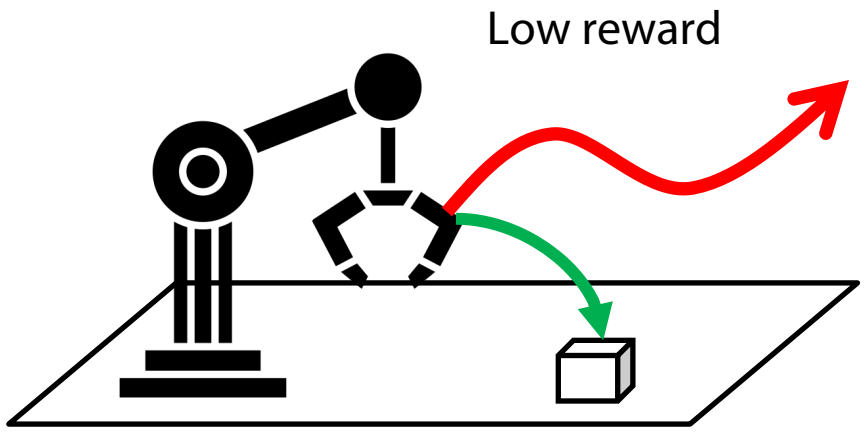
IRL v0 – Assumptions



IRL v0 – What is a good reward function?

A good reward would evaluate optimal data higher than all other data

$$V_r^{\pi^*}(s) \geq V_r^{\pi}(s) \quad \forall \pi, \forall s$$



Find w* such that $r(s,a) = w^{*T}\phi(s,a)$

$$\mathbb{E}_{\pi^*}\left[\sum_t \gamma^t r(s_t,a_t)\right] \geq \mathbb{E}_{\pi}\left[\sum_t \gamma^t r(s_t,a_t)\right], \quad \forall \pi$$

$$\mathbb{E}_{\pi^*}\left[\sum_t \gamma^t w^{*T}\phi(s_t,a_t)\right] \geq \mathbb{E}_{\pi}\left[\sum_t \gamma^t w^{*T}\phi(s_t,a_t)\right], \quad \forall \pi$$

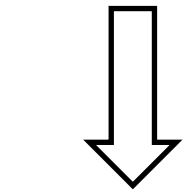
$$w^{*T}\underbrace{\mathbb{E}_{\pi^*}\left[\sum_t \gamma^t \phi(s_t,a_t)\right]}_{\mu(\pi^*,\phi)} \geq w^{*T}\underbrace{\mathbb{E}_{\pi}\left[\sum_t \gamma^t \phi(s_t,a_t)\right]}_{\mu(\pi,\phi)}, \quad \forall \pi$$

Underdefined, w* = 0 trivially satisfies!

IRL v0 – What is a good reward function?

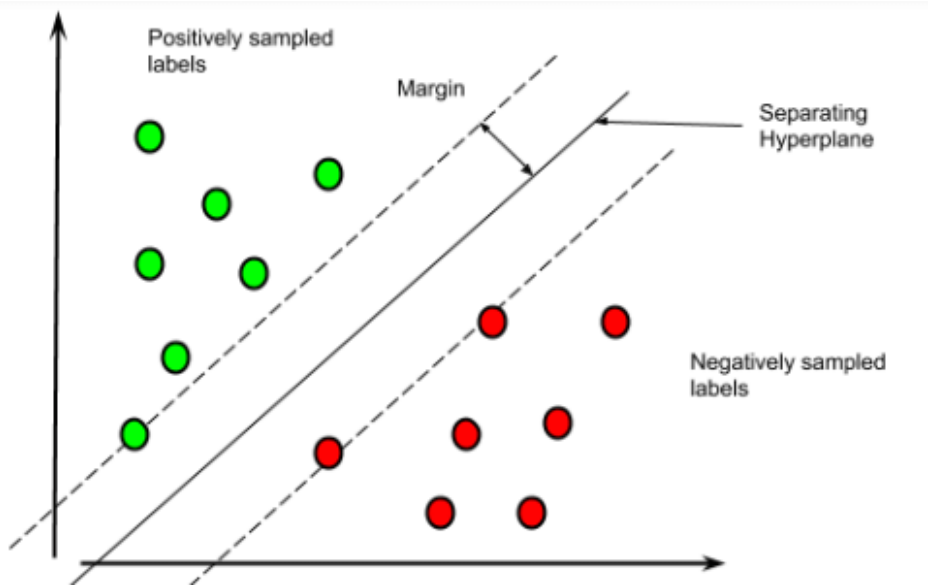
How do we tackle ambiguity?

$$w^{*T}\mathbb{E}_{\pi^*}\left[\phi(s,a)\right] \geq w^{*T}\mathbb{E}_{\pi^*}\left[\phi(s,a)\right] \quad \forall \pi, \forall s$$



$$\max_{w,m} \quad m$$

$$\text{s.t} \quad w^T\mu^{\pi^*} \geq w^T\mu^{\pi} + m, \forall \pi \in \Pi$$



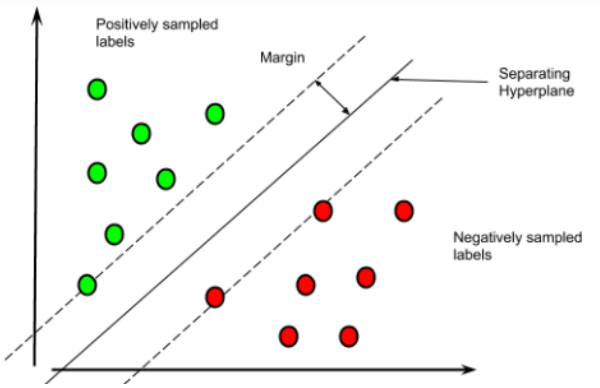
Find rewards which maximize the gap between the expert and all other policies

IRL v1 – Max Margin Feature Matching

Choose w such that "margin" is maximized

$$\max m$$
$$\text{s.t} \;\; w^T \mu^{\pi^*} \geq w^T \mu^{\pi} + m, \forall \pi \in \Pi$$

Looks a lot like an SVM!



$$\min \|w\|_2$$
$$\text{s.t} \;\; w^T \mu^{\pi^*} \geq w^T \mu^{\pi} + 1, \forall \pi \in \Pi$$

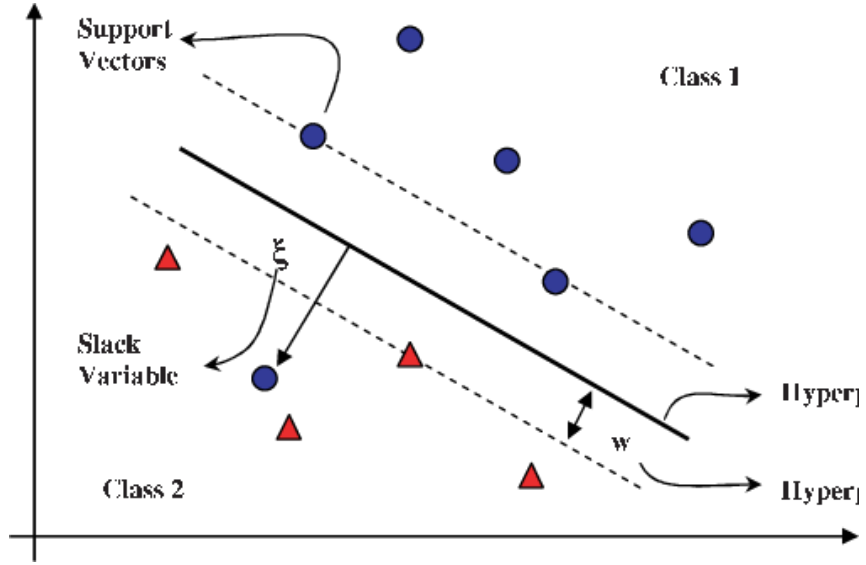
What might the issues be →

1. Uniform gap across all π, π^*
2. Noisily optimal may compromise the optimization

IRL v1 – (Fancy) Max Margin Feature Matching

Maximum margin → Structured Max-Margin + Slack

$$\min \|w\|_2$$
$$\text{s.t} \;\; w^T \mu^{\pi^*} \geq w^T \mu^{\pi} + 1, \forall \pi \in \Pi$$



Bigger for more different policies

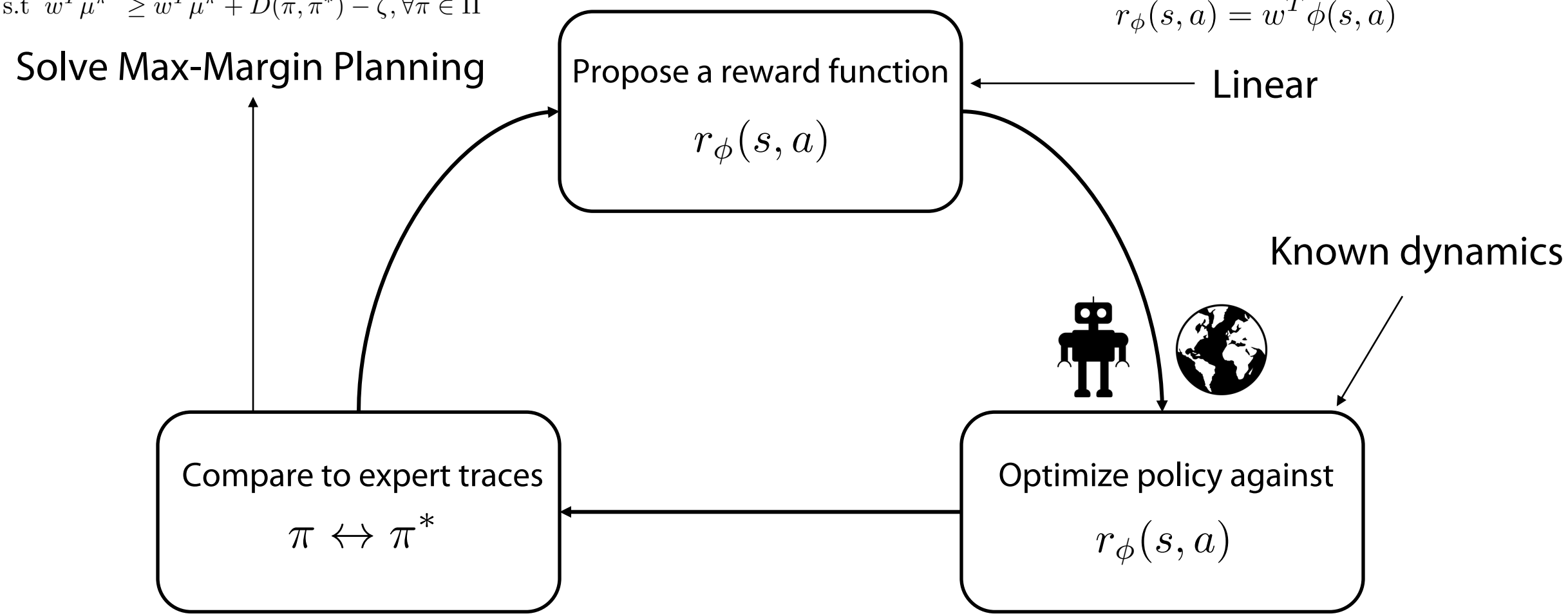
$$\min \|w\|_2 + C\zeta$$
$$\text{s.t} \;\; w^T \mu^{\pi^*} \geq w^T \mu^{\pi} + D(\pi, \pi^*) - \zeta, \forall \pi \in \Pi$$

Slack allows for noisy optimality

IRL v1 – Max Margin Feature Matching

$$\min \|w\|_2 + C\zeta$$
$$\text{s.t } w^T\mu^{\pi^*} \geq w^T\mu^{\pi} + D(\pi, \pi^*) - \zeta, \forall \pi \in \Pi$$

$$r_\phi(s,a) = w^T\phi(s,a)$$



IRL v1 – Max Margin Feature Matching

1. Start with a random policy π_0
2. Find the w that optimizes

$$\min_{w,\zeta} \|w\|_2 + C\zeta$$

$$\text{s.t} \;\; w^T \mu^{\pi^*} \geq w^T \mu^{\pi} + D(\pi, \pi^*) - \zeta, \forall \pi \in \{\pi_0, \pi_1, \ldots, \pi_i\}$$

3. Solve for the optimal policy against $r_\phi(s,a) = {w^{(i)}}^T \phi(s,a)$

$$\pi_{i+1} \rightarrow \text{Opt}(r_\phi(s,a), T)$$

4. Add to constraint set and repeat

Output the optimal reward function w*

Max Margin Feature Matching in Action

Lecture Outline

Why Imitation? + Problem formulation

↓

IRLv1 – max margin planning

↓

IRLv2 – max entropy IRL

↓

IRLv3 – partial policy optimization

↓

IRLv4 – adversarial IRL

IRL v1 – Why this may not be enough?

$$\min \|w\|_2 + C\zeta$$

$$\text{s.t} \;\; w^T \mu^{\pi^*} \geq w^T \mu^{\pi} + D(\pi, \pi^*) - \zeta, \forall \pi \in \Pi$$

May not be able to deal with scenario where true margin is quite small for some policies

Not clear if this is a good way to deal with suboptimality

Constrained optimization is tough to optimize for non-linear functions

What if we had a ”softer” notion of margin?

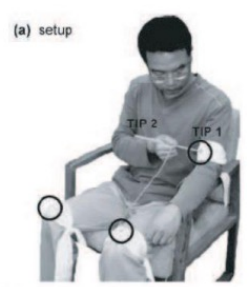
We have talked about "soft" optimality before!

We derived max-ent RL as maximum likelihood on optimality (lower bound) wrt policy

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

Control as inference

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

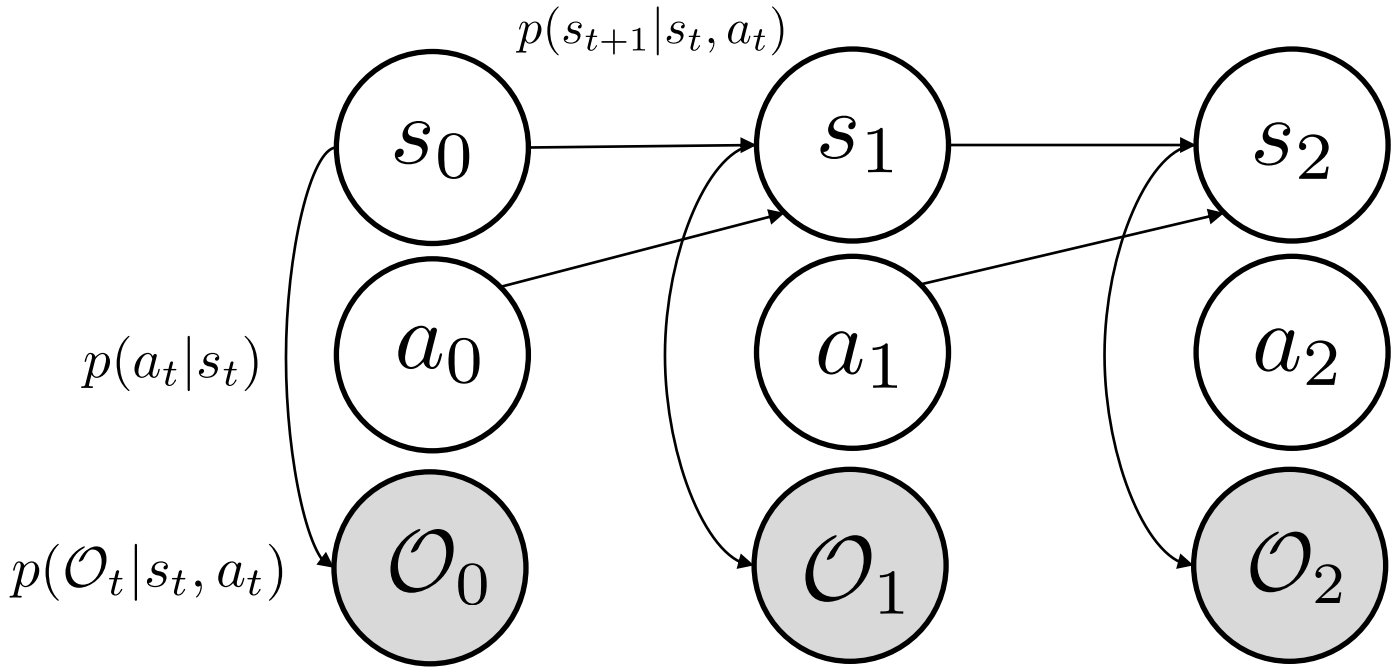


Li & Todorov '06


Ziebart '08

Can we invert this to do inverse RL with a softer notion of margin?

Let's revisit the graphical model

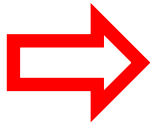


$$p(\mathcal{O}_t|s_t, a_t) = \exp(r_\phi(s_t, a_t))$$

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

$p(\tau)$

Uninformed behavior according to prior/dynamics

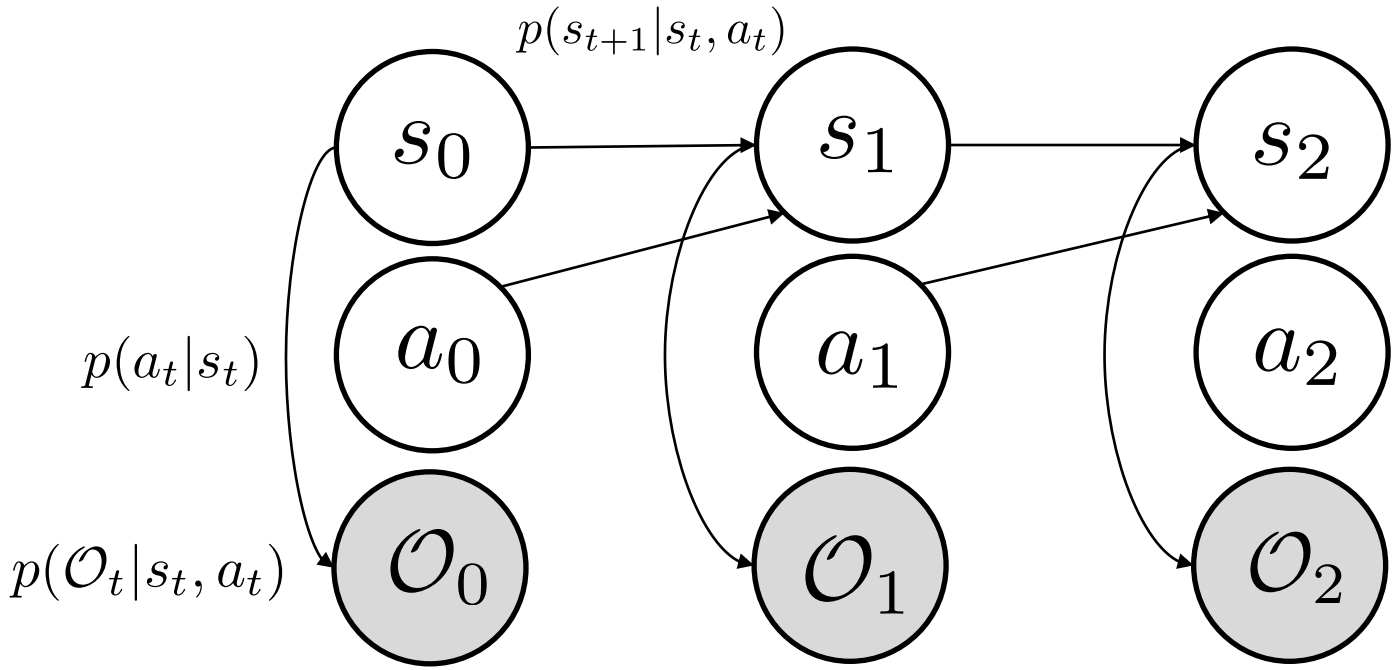


$p(\tau|\mathcal{O}_{0:T} = 1)$

Soft optimal behavior conditioned on optimality

We were trying to find $p(a_t|s_t, \mathcal{O}_{t:T} = 1)$ given reward

IRLv2 – Maximum Entropy Inverse RL



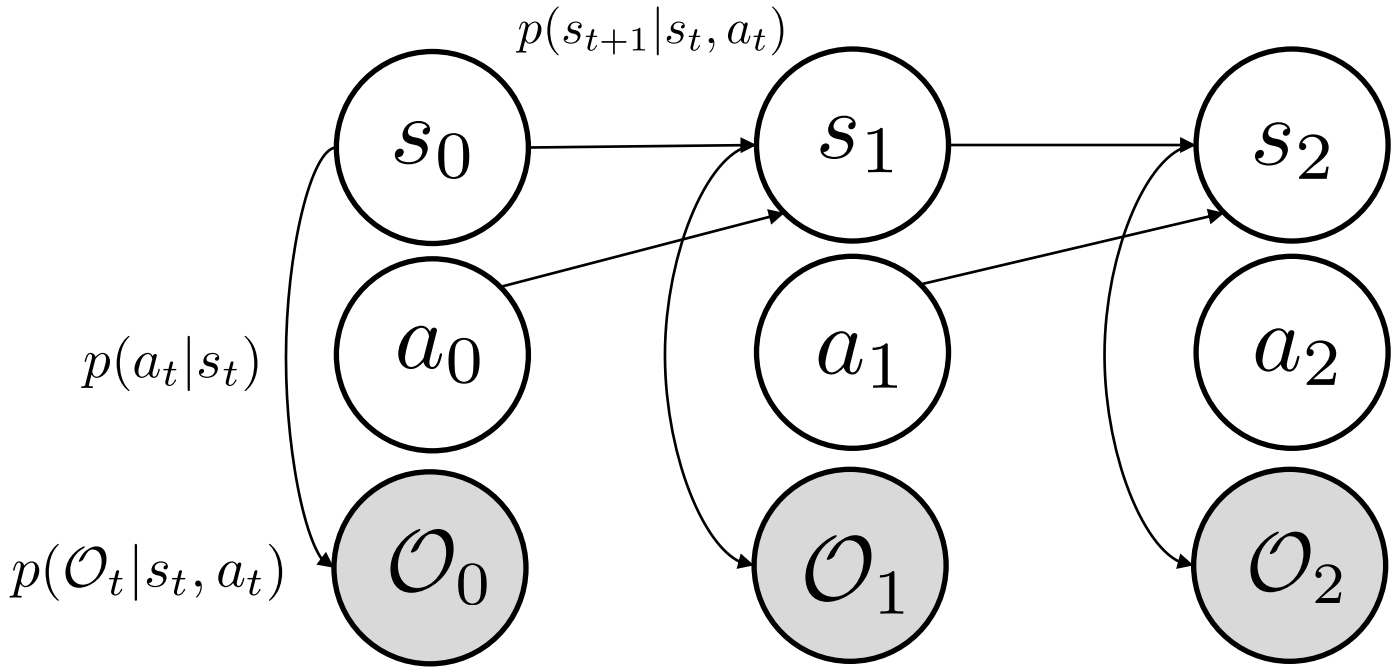
$$p(\mathcal{O}_t|s_t, a_t) = \exp(r_\phi(s_t, a_t))$$

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

Now we are given (s, a) from optimal, we need to find the reward function that best explains the data

→ Maximum likelihood estimation!
(Find r, that maximizes the likelihood of (s, a) being produced on observed optimality

Inverse RL in CAI graphical model



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

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

→ Maximum likelihood estimation!

(Find r, that maximizes the likelihood of (s, a) being produced on observed optimality

$$\max_\phi \mathbb{E}_{\tau \sim \mathcal{D}^*} [\log p(\tau|\mathcal{O}_{0:T} = 1)]$$

(Find optimality CPD that best explains observed data)

Maximum likelihood optimality estimation

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

Independent of reward

$$= \frac{\exp\left(\sum_{t=0}^{T} r(s_t, a_t)\right)}{\int\int p(\tau) \exp\left(\sum_{t=0}^{T} r(s_t, a_t)\right) ds_{0:T} da_{0:T}}$$

Hard to estimate – partition function (Z)

$$\max_{\phi} \mathbb{E}_{\tau \sim \mathcal{D}^*} \left[\log p(\tau|\mathcal{O}_{0:T}=1)\right]$$

Difficult to compute analytically, but it's gradient has a nice form!

Maximum likelihood optimality estimation

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

$$\max_{\phi} \mathbb{E}_{\tau\sim\mathcal{D}^*}\left[\log p(\tau|\mathcal{O}_{0:T}=1)\right]$$

$$= \mathbb{E}_{\tau\sim\mathcal{D}^*}\left[\log\left(\exp(\sum_{t=0}^{T} r_\phi(s_t,a_t))\right) - \log Z\right]$$

$$= \mathbb{E}_{\tau\sim\mathcal{D}^*}\left[\sum_{t=0}^{T} r_\phi(s_t,a_t)\right] - \log Z$$

Easy to compute

Hard to compute

Let's take the gradient

$$\max_{\phi} \mathbb{E}_{\tau \sim \mathcal{D}^*} \left[\log p(\tau | \mathcal{O}_{0:T} = 1) \right]$$

$$\mathcal{L}(\phi) = \mathbb{E}_{\tau \sim \mathcal{D}^*} \left[\sum_{t=0}^{T} r_\phi(s_t, a_t) \right] - \log Z$$

$$\nabla_\phi \mathcal{L}(\phi) = \mathbb{E}_{\tau \sim \mathcal{D}^*} \left[\sum_{t=0}^{T} \nabla_\phi r_\phi(s_t, a_t) \right] - \nabla_\phi \log Z$$

$$\nabla_\phi \log Z = \frac{1}{Z} \nabla_\phi Z$$

$$Z = \int p(\tau) \exp(r(\tau)) d\tau$$

$$\nabla_\phi \mathcal{L}(\phi) = \mathbb{E}_{\tau \sim \mathcal{D}^*} \left[\sum_{t=0}^{T} \nabla_\phi r_\phi(s_t, a_t) \right] - \frac{1}{Z} \int p(\tau) \exp(r_\phi(\tau)) \nabla_\phi r_\phi(\tau) d\tau$$

Notice this is exactly the soft optimality posterior

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

Let's take the gradient

$$\mathcal{L}(\phi) = \mathbb{E}_{\tau \sim \mathcal{D}^*} \left[\sum_{t=0}^{T} r_\phi(s_t, a_t) \right] - \log Z$$

$$\nabla_\phi \mathcal{L}(\phi) = \mathbb{E}_{\tau \sim \mathcal{D}^*} \left[\sum_{t=0}^{T} \nabla_\phi r_\phi(s_t, a_t) \right] - \frac{1}{Z} \int p(\tau) \exp(r_\phi(\tau)) \nabla_\phi r_\phi(\tau) d\tau$$

Notice this is exactly the soft optimality posterior

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

$$\nabla_\phi \mathcal{L}(\phi) = \mathbb{E}_{\tau \sim \mathcal{D}^*} \left[\sum_{t=0}^{T} \nabla_\phi r_\phi(s_t, a_t) \right] - \mathbb{E}_{\tau \sim p(\tau | \mathcal{O}_{0:T}=1)} \left[\sum_{t=0}^{T} \nabla_\phi r_\phi(s_t, a_t) \right]$$

Push up gradients along experts

Push down gradients along soft optimal policy under current reward

Computable, with RL in the inner loop

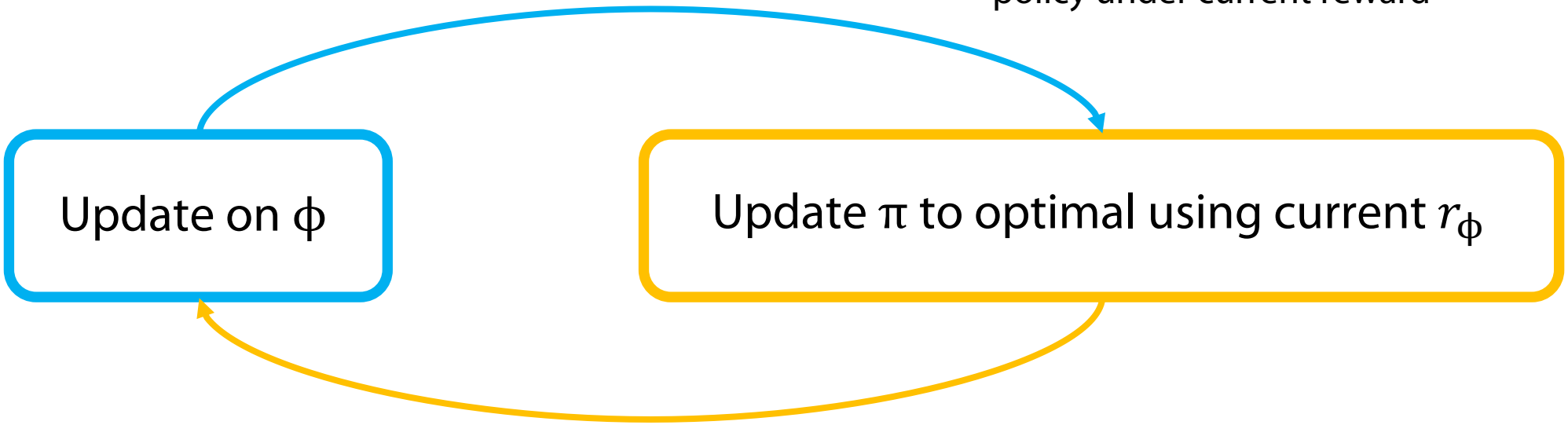
IRLv2 – Maximum Entropy Inverse RL

$$\nabla_\phi \mathcal{L}(\phi) = \mathbb{E}_{\tau \sim \mathcal{D}^*} \left[\sum_{t=0}^{T} \nabla_\phi r_\phi(s_t, a_t) \right] - \frac{1}{Z} \int p(\tau) \exp(r_\phi(\tau)) \nabla_\phi r_\phi(\tau) d\tau$$

$$\nabla_\phi \mathcal{L}(\phi) = \mathbb{E}_{\tau \sim \mathcal{D}^*} \left[\sum_{t=0}^{T} \nabla_\phi r_\phi(s_t, a_t) \right] - \mathbb{E}_{\tau \sim p(\tau | \mathcal{O}_{0:T}=1)} \left[\sum_{t=0}^{T} \nabla_\phi r_\phi(s_t, a_t) \right]$$

Push up gradients along experts

Push down gradients along soft optimal policy under current reward



IRL v2 – Max-Ent IRL – Put it together

Maximum Entropy

$$\nabla_w \mathcal{L} = \mathbb{E}_{\pi^*}\left[\sum_t \gamma^t \phi(s_t, a_t)\right] - \mathbb{E}_{p_w(\tau)}\left[\sum_t \gamma^t \phi(s_t^\tau, a_t^\tau)\right]$$

$$r_\phi(s,a) = w^T \phi(s,a)$$

Linear

Propose a reward function

$r_\phi(s,a)$

Known dynamics

Optimize policy against

$r_\phi(s,a)$

Compare to expert traces

$\pi \leftrightarrow \pi^*$

IRL v2 –Max-Entropy Inverse RL (Pseudocode)

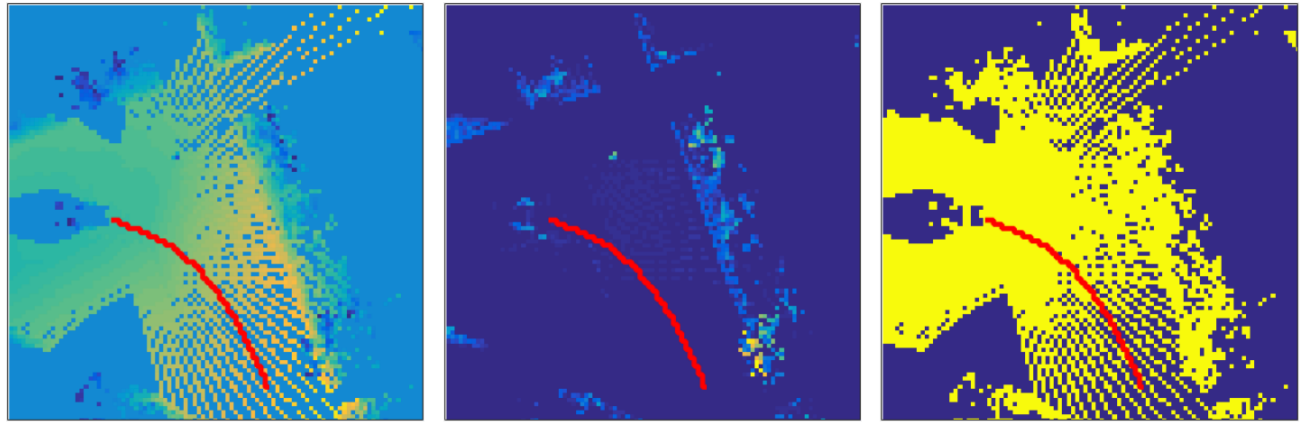
1. Start with a random policy π_0 and weight vector w
2. Find the "soft" optimal policy under w – $p_w(\tau)$
3. Take a gradient step on w

$$\nabla_w \mathcal{L} = \mathbb{E}_{\pi^*}\left[\sum_t \gamma^t \phi(s_t, a_t)\right] - \mathbb{E}_{p_w(\tau)}\left[\sum_t \gamma^t \phi(s_t^\tau, a_t^\tau)\right]$$

4. Repeat

Output the optimal reward function w*

Max-Ent IRL in Action



Lecture Outline

Why Imitation? + Problem formulation

↓

IRLv1 – max margin planning

↓

IRLv2 – max entropy IRL

↓

IRLv3 – partial policy optimization

↓

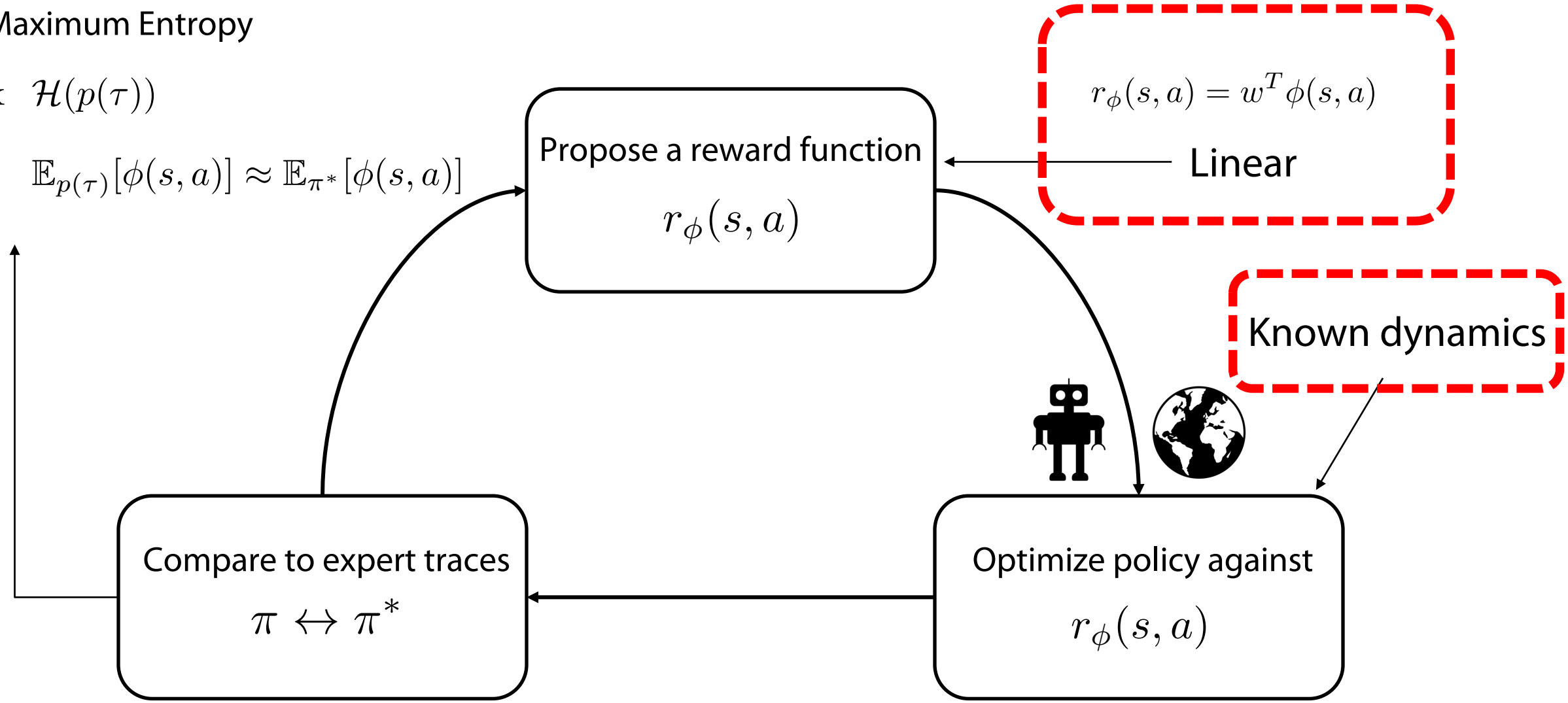
IRLv4 – adversarial IRL

Ok but no way this could work?

Maximum Entropy

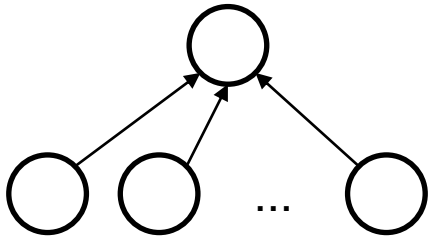
$$\max_{p(\tau)} \quad \mathcal{H}(p(\tau))$$

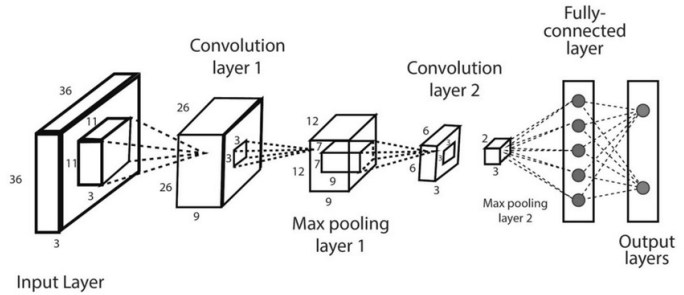
$$\text{s.t} \quad \mathbb{E}_{p(\tau)}[\phi(s,a)] \approx \mathbb{E}_{\pi^*}[\phi(s,a)]$$



Linear Rewards → Neural Net Rewards

Max-ent IRL allows us to go from linear rewards to arbitrary neural network rewards





Linear Max-Ent IRL

$$\max_w \mathbb{E}_{\pi^*}\left[\sum_t w^T \gamma^t \phi(s_t, a_t)\right] - \log \int_\tau \left[\exp\left(\sum_t w^T \gamma^t \phi(s_t, a_t)\right)\right] d\tau$$

Non-Linear Max-Ent IRL

$$\max_\theta \mathbb{E}_{\pi^*}\left[\sum_t \gamma^t r_\theta(s_t, a_t)\right] - \log \int_\tau \left[\exp\left(\sum_t \gamma^t r_\theta(s_t, a_t)\right)\right] d\tau$$

Can simply replace, w with arbitrary θ and use autodiff!

Avoiding Complete Policy Optimization

Optimize policy against

$$r_\phi(s, a)$$

← Assumes dynamics are known so we can just do (fast) planning

What happens when dynamics are unknown!

$$\mathbb{E}_{\pi^*}\left[\sum_t \gamma^t \nabla_\theta r_\theta(s_t, a_t)\right]$$

← What if we only **improved** the policy a little bit

$$-\mathbb{E}_{p_w(\tau)}\left[\sum_t \gamma^t \nabla_\theta r_\theta(s_t, a_t)\right]$$

← Biased!

Requires complete "soft" policy optimization

Avoiding Complete Policy Optimization

Importance sampling to the rescue!

$$\mathbb{E}_{p(x)}\left[f(x)\right] = \mathbb{E}_{q(x)}\left[\frac{p(x)}{q(x)} f(x)\right]$$

$$\mathbb{E}_{\pi^*}\left[\sum_t \gamma^t \nabla_\theta r_\theta(s_t, a_t)\right]$$

$$-\mathbb{E}_{p_w(\tau)}\left[\sum_t \gamma^t \nabla_\theta r_\theta(s_t, a_t)\right]$$

Importance Sampling

$$\mathbb{E}_{\pi^*}\left[\sum_t \gamma^t \nabla_\theta r_\theta(s_t, a_t)\right]$$

$$-\mathbb{E}_{q}\left[\frac{p_w(\tau)}{q(\tau)} \sum_t \gamma^t \nabla_\theta r_\theta(s_t, a_t)\right]$$

$$\frac{\exp(\sum_t r_\theta(s_t, a_t))}{\Pi_t \pi_\theta(a_t|s_t)}$$

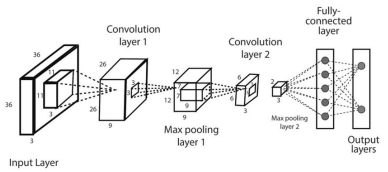
Can transfer significantly more from iteration to iteration rather than doing full nested optimization

IRLv4 – Guided Cost Learning

Gradient Step on Reward

$$\mathbb{E}_{\pi^*}\left[\sum_t \gamma^t \nabla_\theta r_\theta(s_t, a_t)\right]$$

$$-\mathbb{E}\left[\frac{p_w(\tau)}{q(\tau)} \sum_t \gamma^t \nabla_\theta r_\theta(s_t, a_t)\right]$$



Propose a reward function

$r_\phi(s, a)$

$r_\phi(s, a)$

Neural network

Gradient step on policy

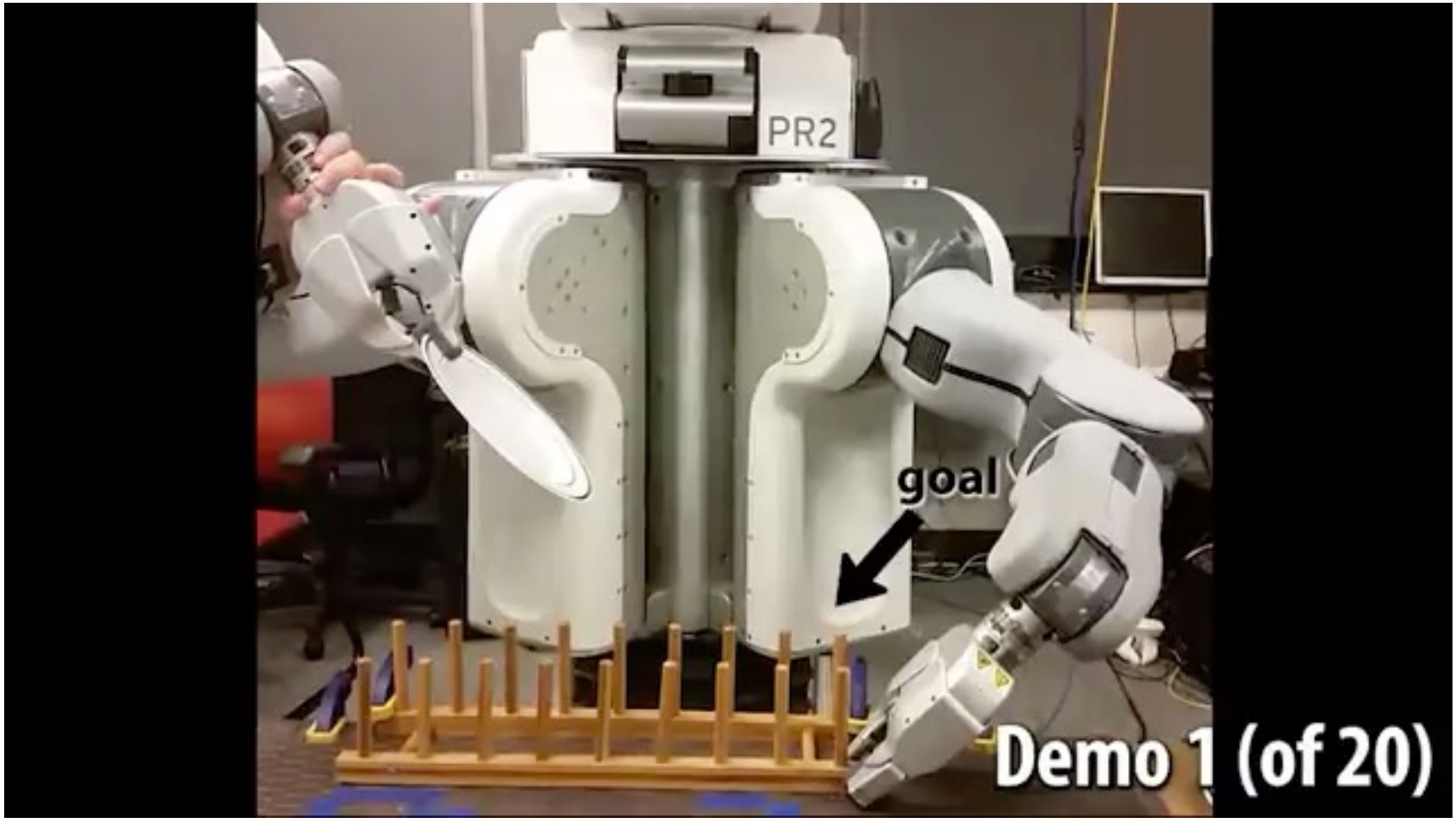
Optimize policy against

$r_\phi(s, a)$

Compare to expert traces

$\pi \leftrightarrow \pi^*$

IRLv4 – Guided Cost Learning



Lecture Outline

Why Imitation? + Problem formulation

↓

IRLv1 – max margin planning

↓

IRLv2 – max entropy IRL

↓

IRLv3 – partial policy optimization

↓

IRLv4 – adversarial IRL

Connecting Maximum-Entropy RL to GANs

Looks like a game

Generator

Discriminator

1. Start with a random policy π_0 and weight vector w
2. Take a step on "soft" optimal policy under w – $p_w(\tau)$
3. Take a gradient step on w

$$\nabla_\theta \mathcal{L} = \mathbb{E}_{\pi^*}\left[\sum_t \gamma^t \nabla_\theta r_\theta(s_t, a_t)\right] - \mathbb{E}_q\left[\frac{p_w(\tau)}{q(\tau)} \sum_t \gamma^t \nabla_\theta r_\theta(s_t, a_t)\right]$$

4. Repeat

Output the optimal reward function w*

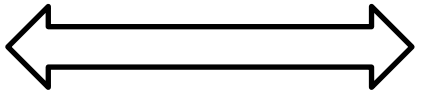
Recasting GAIL as an IRL method

For a particular parameterization of the discriminator, GAIL recovers a reward

Max-Ent Inverse RL

$$\mathbb{E}_{\pi^*}\left[\sum_t \gamma^t \nabla_\theta r_\theta(s_t, a_t)\right] - \mathbb{E}_q\left[\frac{p_w(\tau)}{q(\tau)} \sum_t \gamma^t \nabla_\theta r_\theta(s_t, a_t)\right]$$

Push up demos, push down policy



With some massaging

GAIL

$$\mathbb{E}_{\pi^*}\left[D_\psi(\tau)\right] + \mathbb{E}_{\pi_\theta}\left[(1 - D_\psi(\tau))\right]$$

Push up real data, push down generated

$$D_\theta(\tau) = \frac{\frac{1}{Z}\exp(r_\theta(\tau))}{\frac{1}{Z}\exp(r_\theta(\tau)) + \Pi_t \pi_\theta(a_t|s_t)}$$

GAIL (which is just a GAN), recovers Max-Ent IRL

In practice, often use GAIL and just log D as reward

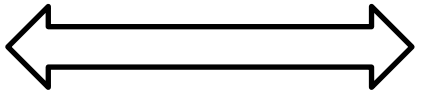
Recasting GAIL as an IRL method

For a particular parameterization of the discriminator, GAIL recovers a reward

Max-Ent Inverse RL

$$\mathbb{E}_{\pi^*}\left[\sum_t \gamma^t \nabla_\theta r_\theta(s_t, a_t)\right] - \mathbb{E}_q\left[\frac{p_w(\tau)}{q(\tau)} \sum_t \gamma^t \nabla_\theta r_\theta(s_t, a_t)\right]$$

Push up demos, push down policy



With some massaging

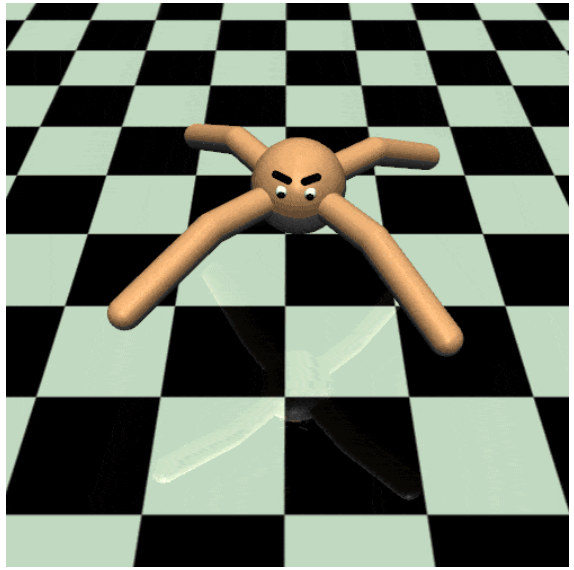
GAIL

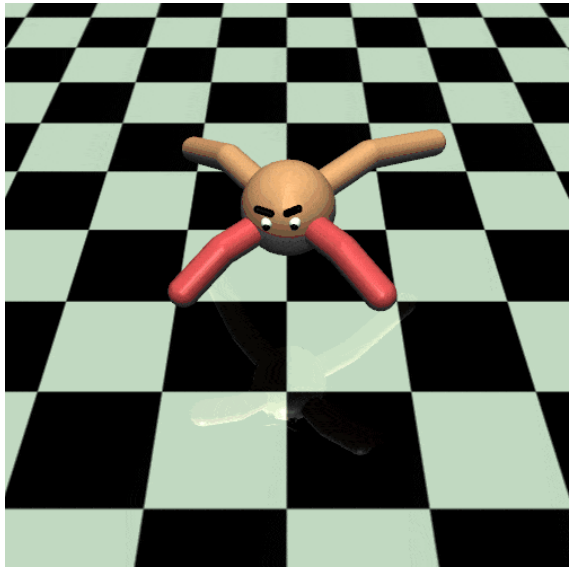
$$\mathbb{E}_{\pi^*}\left[D_\psi(\tau)\right] + \mathbb{E}_{\pi_\theta}\left[(1 - D_\psi(\tau))\right]$$

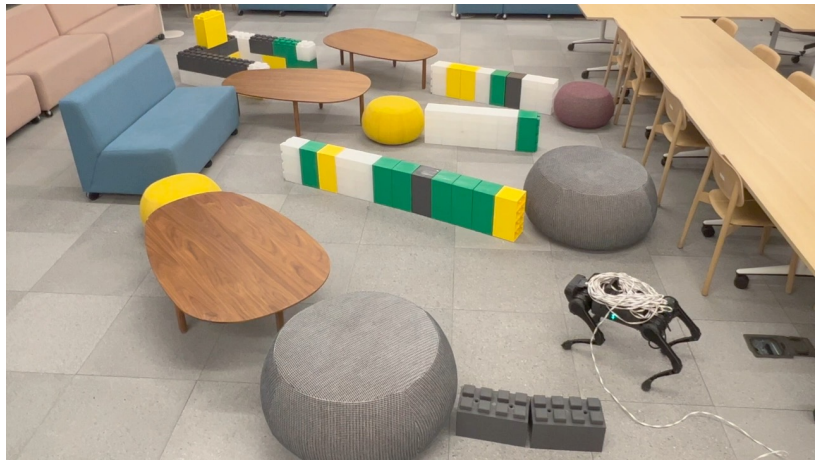
Push up real data, push down generated

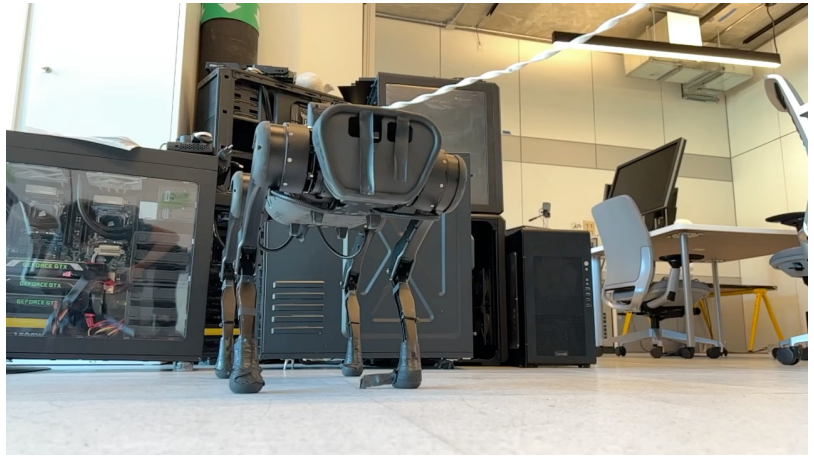
$$\max_r \min_\pi \left[\mathbb{E}_{\tau_E} \sum_{t=0}^{T-1} \gamma^t r_t - \mathbb{E}_{\tau_\pi} \sum_{t=0}^{T-1} \gamma^t \left(r_t - \log \pi(a_t|s_t)\right) - \psi(r)\right].$$

Adversarial IRL in Action









Lecture Outline

Why Imitation? + Problem formulation

↓

IRLv1 – max margin planning

↓

IRLv2 – max entropy IRL

↓

IRLv3 – partial policy optimization

↓

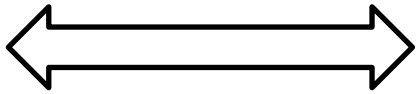
IRLv4 – adversarial IRL

Can we get rid of this adversarial game?

Max-Ent Inverse RL

$$\mathbb{E}_{\pi^*}\left[\sum_t \gamma^t \nabla_\theta r_\theta(s_t, a_t)\right] - \mathbb{E}_q\left[\frac{p_w(\tau)}{q(\tau)} \sum_t \gamma^t \nabla_\theta r_\theta(s_t, a_t)\right]$$

Push up demos, push down policy



GAIL

$$\mathbb{E}_{\pi^*}\left[D_\psi(\tau)\right] + \mathbb{E}_{\pi_\theta}\left[(1 - D_\psi(\tau))\right]$$

Push up real data, push down generated

$$\max_r \min_\pi \left[\mathbb{E}_{\tau_E} \sum_{t=0}^{T-1} \gamma^t r_t - \mathbb{E}_{\tau_\pi} \sum_{t=0}^{T-1} \gamma^t \left(r_t - \log \pi(a_t|s_t)\right) - \psi(r)\right].$$

Yes if we are maximum entropy!

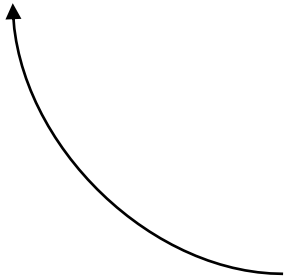
Let's start from the adversarial objective

$$\max_r \min_\pi \left[\mathbb{E}_{\tau_E} \sum_{t=0}^{T-1} \gamma^t r_t - \mathbb{E}_{\tau_\pi} \sum_{t=0}^{T-1} \gamma^t \left(r_t - \log \pi(a_t|s_t) \right) - \psi(r) \right].$$

Expert pushed up

On-policy pushed down

Regularization



How can we get rid of this

r: make expert better than learner

π: optimize the learned reward

Certain properties hold at optimality for maximum entropy RL!

What does max-ent RL tell us?

$$Q_t^r(s,a) = \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[r(s,a,s') + \gamma V_{t+1}^r(s') \right], \qquad \text{(Bellman equation)}$$

$$V_t^r(s) = \log \sum_a \exp Q_t^r(s,a). \qquad \text{(Value expectation)}$$

$$\pi_t^r(a|s) = \exp\left(Q_t^r(s,a) - V_t^r(s)\right). \qquad \text{(Policy extraction)}$$

$$\max_\pi \mathbb{E}_{\tau_\pi} \sum_{t=0}^{T-1} \gamma^t \left(r_t - \log \pi(a_t|s_t) \right). \quad \Longrightarrow \quad \mathbb{E}_{s_0 \sim p_0} \left[V_0^r(s_0) \right].$$

(Max-ent RL objective) (Average value function)

Let's reduce max-ent IRL

$$\max_r \min_\pi \left[\mathbb{E}_{\tau_E} \sum_{t=0}^{T-1} \gamma^t r_t - \mathbb{E}_{\tau_\pi} \sum_{t=0}^{T-1} \gamma^t \left(r_t - \log \pi(a_t|s_t) \right) - \psi(r) \right].$$

$$\mathbb{E}_{s_0 \sim p_0} \left[V_0^r(s_0) \right].$$

Non-adversarial, but V depends on r non-trivially

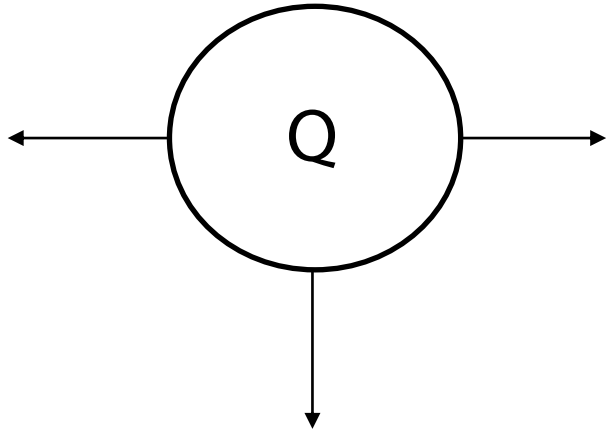
$$\max_r \left[\mathbb{E}_{\tau_E} \sum_{t=0}^{T-1} \gamma^t r_t - \mathbb{E}_{s_0 \sim p_0} \left[V_0^r(s_0) \right] - \psi(r) \right].$$

Can we unify policy and reward via Q?

$$\max_r \left[\mathbb{E}_{\tau_E} \sum_{t=0}^{T-1} \gamma^t r_t - \mathbb{E}_{s_0 \sim p_0} \left[V_0^r(s_0) \right] - \psi(r) \right].$$

Turns out all can be written in terms of Q

$$V_t(s) = \log \sum_a \exp Q_t(s,a).$$



$$r_Q(s,a,s') = Q_t(s,a) - \gamma V_{t+1}(s').$$

$$\pi_Q(a|s) = \exp\left(Q_t(s,a) - V_t(s)\right).$$

Let's simplify to a single objective

$$\max_r \left[\mathbb{E}_{\tau_E} \sum_{t=0}^{T-1} \gamma^t r_t - \mathbb{E}_{s_0 \sim p_0} \left[V_0^r(s_0) \right] - \psi(r) \right].$$

$$\max_Q \left[\mathbb{E}_{\tau_E} \sum_{t=0}^{T-1} \gamma^t \left(Q_t(s_t, a_t) - \gamma V_{t+1}(s_{t+1}) \right) - \mathbb{E}_{s_0 \sim p_0} \left[V_0(s_0) \right] - \psi(r_Q) \right],$$

$$V_t(s) = \log \sum_a \exp Q_t(s, a), \qquad r_Q(s, a, s') = Q_t(s, a) - \gamma V_{t+1}(s').$$

$$\max_r \min_\pi \quad \longrightarrow \quad \max_Q .$$

Let's build some intuition on this

$$\boxed{\max_{r}\min_{\pi} \quad \longrightarrow \quad \max_{Q} \quad \longrightarrow \quad \max_{\pi, V}}$$

$$\max_{Q}\left[\mathbb{E}_{\tau_E}\sum_{t=0}^{T-1}\gamma^t\left(Q_t(s_t,a_t)-\gamma V_{t+1}(s_{t+1})\right)-\mathbb{E}_{s_0\sim p_0}\left[V_0(s_0)\right]-\psi(r_Q)\right],$$

$$r_Q(s,a,s') = Q_t(s,a) - \gamma V_{t+1}(s'). \qquad Q_t(s,a) = V_t(s) + \log\pi(a|s).$$

⇩ Re-represent in terms of π, V

$$\min_{\pi,V}\left[-\mathbb{E}_{\tau_E}\sum_{t=0}^{T-1}\gamma^t\log\pi(a_t|s_t)+\psi\left(\log\pi(a_t|s_t)+V_t(s_t)-\gamma V_{t+1}(s_{t+1})\right)\right].$$

Let's build some intuition on this

$$\min_{\pi, V} \left[-\mathbb{E}_{\tau_E} \sum_{t=0}^{T-1} \gamma^t \log \pi(a_t|s_t) + \psi \left(\log \pi(a_t|s_t) + V_t(s_t) - \gamma V_{t+1}(s_{t+1}) \right) \right].$$

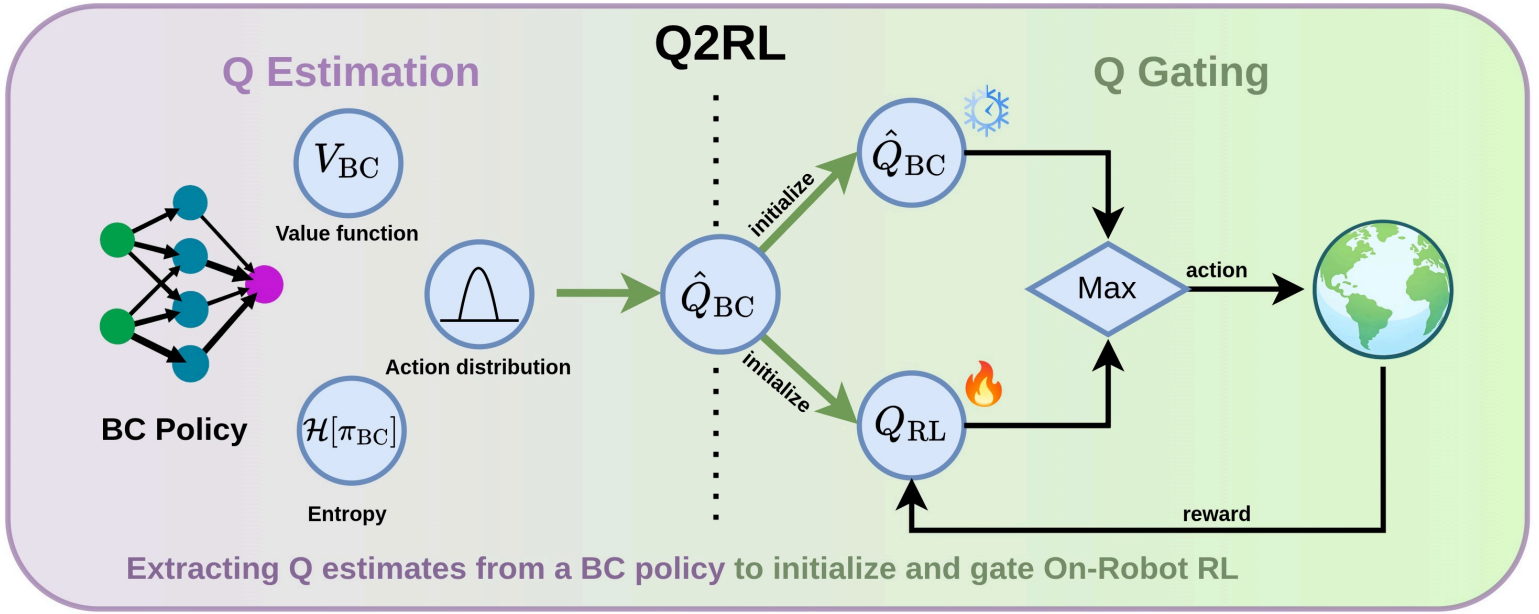
Maximize likelihood (BC)

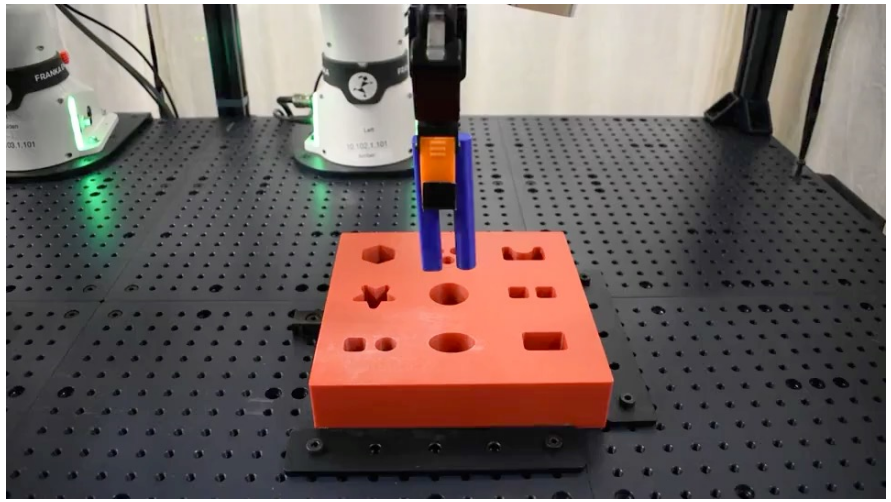
Ensure TD consistency under implied reward

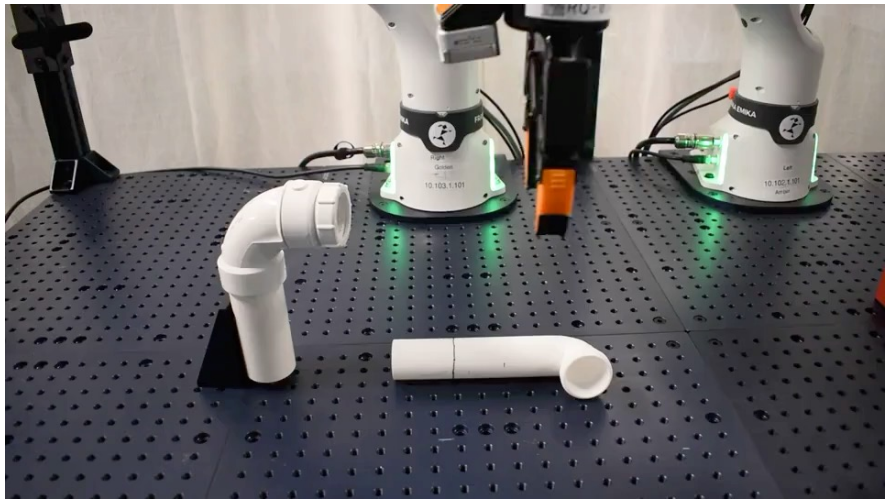
Don't just pick expert actions, but those that also lead to high value

No adversary!

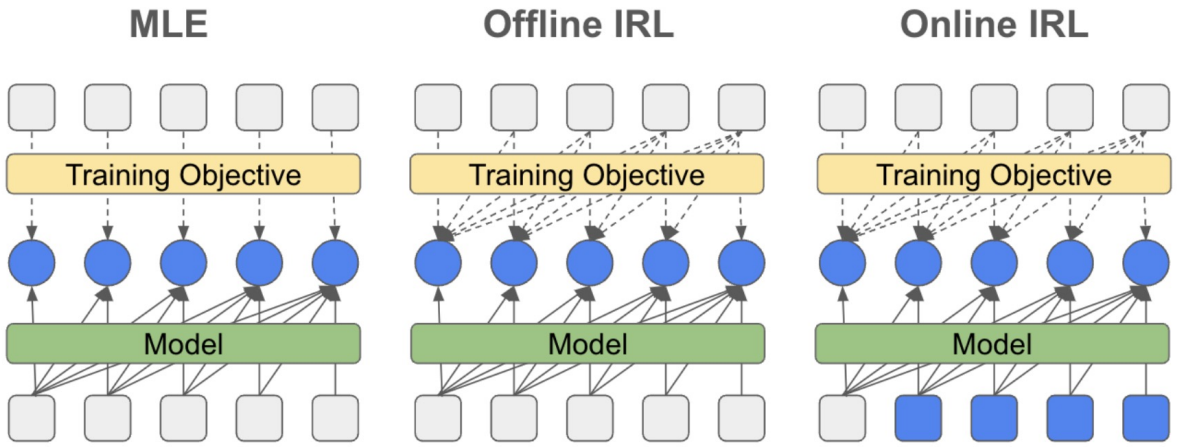
Does this work?







Does this work?



Works at increasing diversity of responses in LLMs