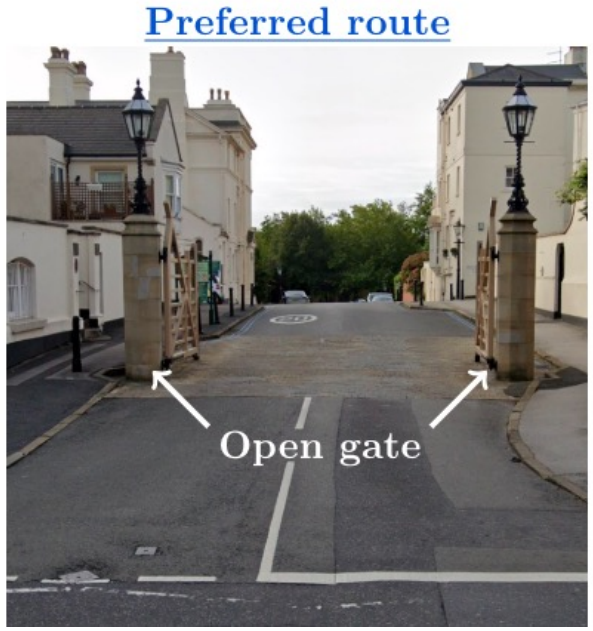
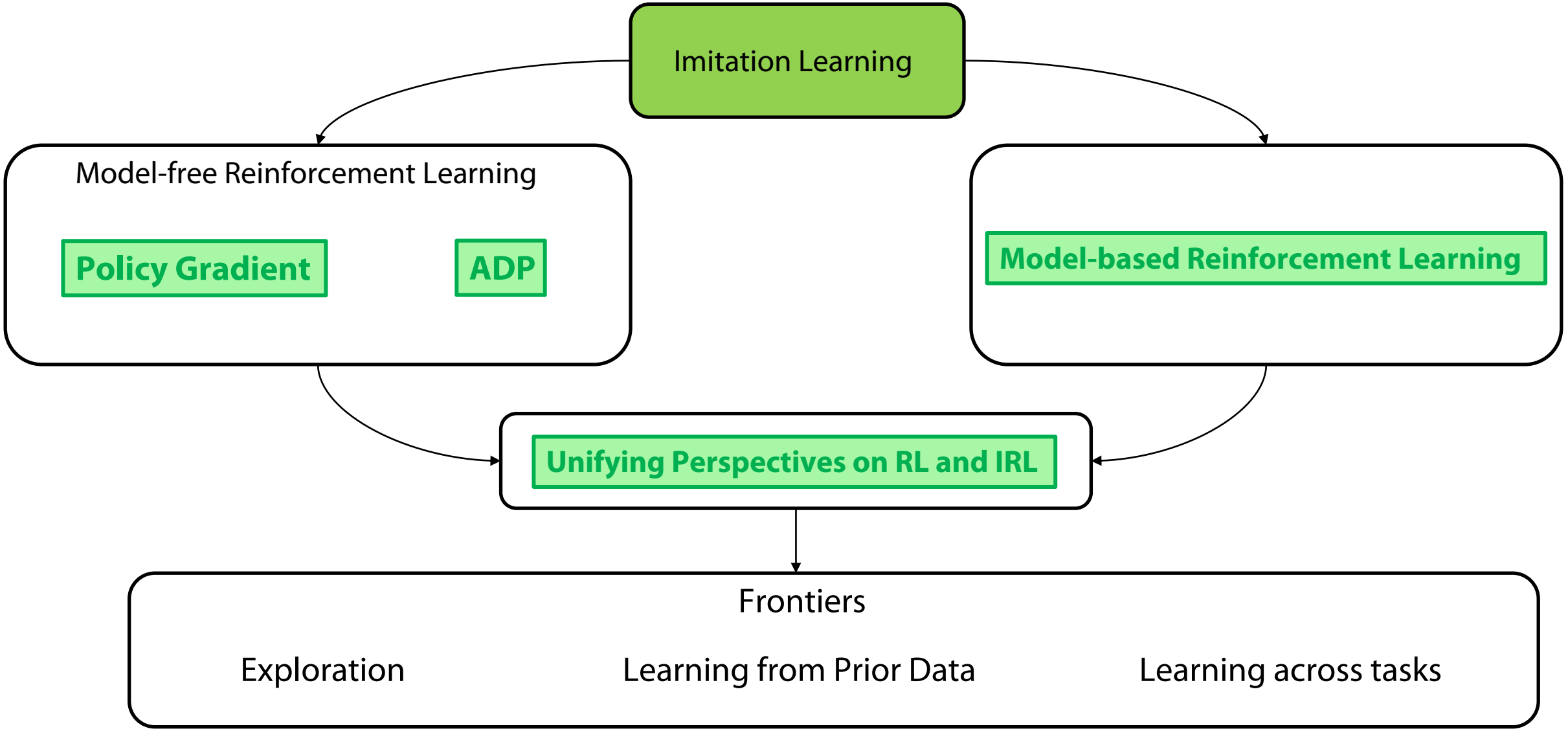
Reinforcement Learning
Autumn 2024

Abhishek Gupta

TA: Jacob Berg



Class Structure



Lecture Outline

Recap – IRL formulation

↓

IRLv1 – max margin planning

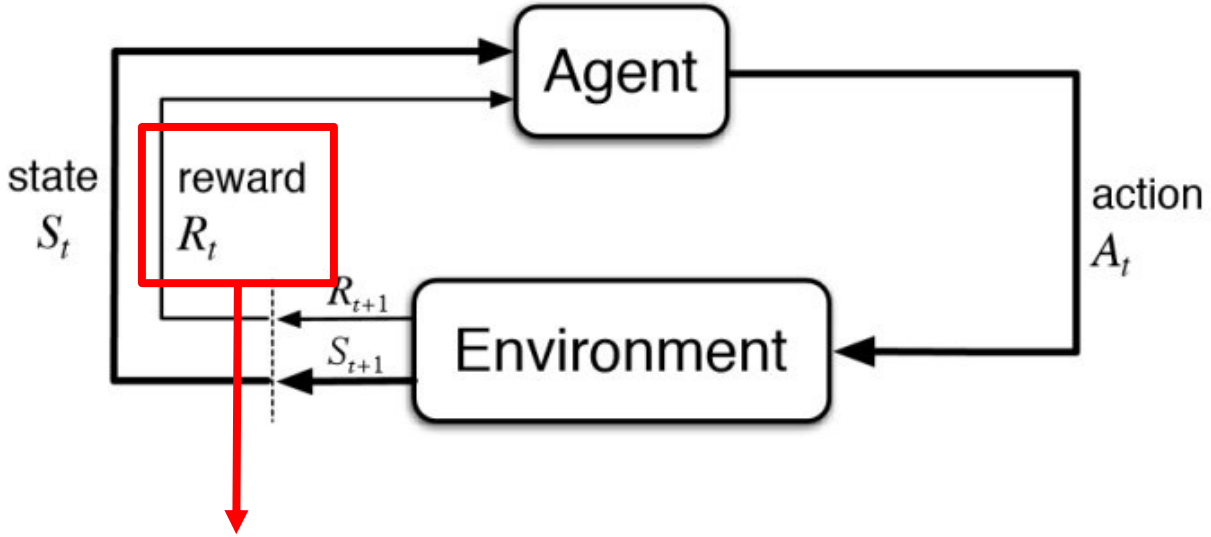
↓

IRLv2 – max entropy IRL

↓

IRL as a GAN

Reinforcement Learning requires Task Specification

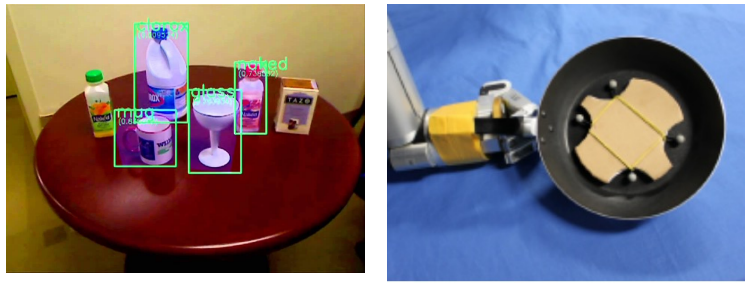


Does not magically appear in most settings

Has to be manually specified

→ can we do better?

Manual state estimation/perception



Complex reward specification

Name	Reward	Heroes	Description
Win	5	Team	
Hero Death	-1	Solo	
Courier Death	-2	Team	
XP Gained	0.002	Solo	
Gold Gained	0.006	Solo	For each unit of gold gained. Reward is not lost when the gold is spent or lost.
Gold Spent	0.0006	Solo	Per unit of gold spent on items without using courier.
Health Changed	2	Solo	Measured as a fraction of hero's max health.‡
Mana Changed	0.75	Solo	Measured as a fraction of hero's max mana.
Killed Hero	-0.6	Solo	For killing an enemy hero. The gold and experience reward is very high, so this reduces the total reward for killing enemies.
Last Hit	-0.16	Solo	The gold and experience reward is very high, so this reduces the total reward for last hit to ~ 0.4.
Deny	0.15	Solo	
Gained Aegis	5	Team	
Ancient HP Change	5	Team	Measured as a fraction of ancient's max health.
Megas Unlocked	4	Team	
T1 Tower*	2.25	Team	
T2 Tower*	3	Team	
T3 Tower*	4.5	Team	
T4 Tower*	2.25	Team	
Shrine*	2.25	Team	
Barracks*	6	Team	
Lane Assign†	-0.15	Solo	Per second in wrong lane.

* For buildings, two-thirds of the reward is earned linearly as the building loses health, and one-third is earned as a lump sum when it dies.
† See item O.2.
‡ Hero's health is quartically interpolated between 0 (dead) and 1 (full health); health at fraction x of full health is worth $(x + 1 - (1 - x)^4)/2$. This function was not tuned; it was set once and then untouched for the duration of the project.

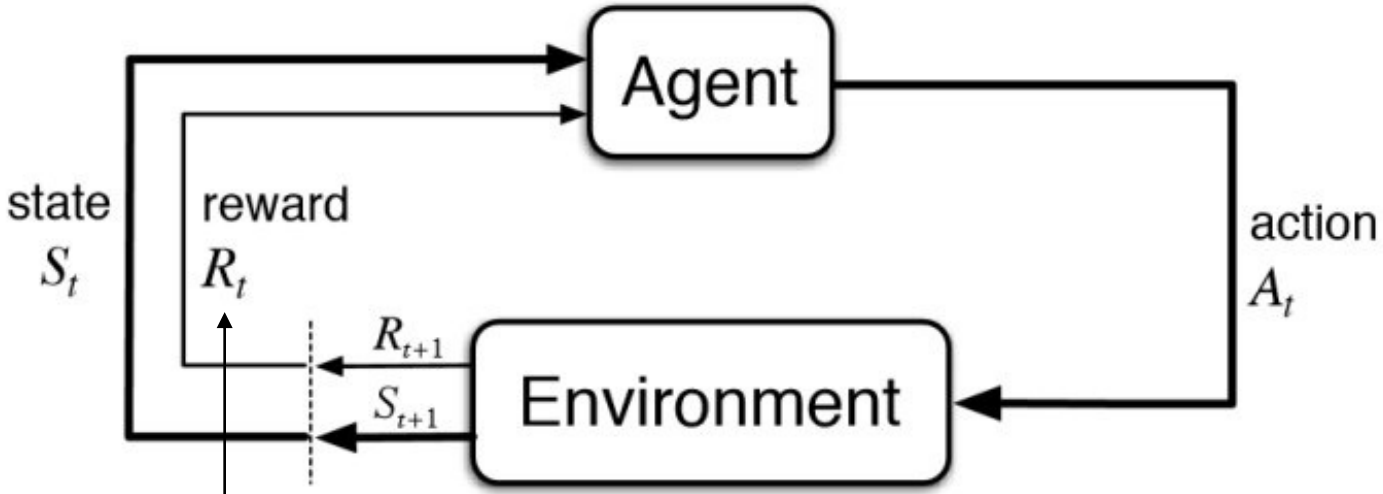
Table 6: Shaped Reward Weights

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

IRL problem statement + assumptions

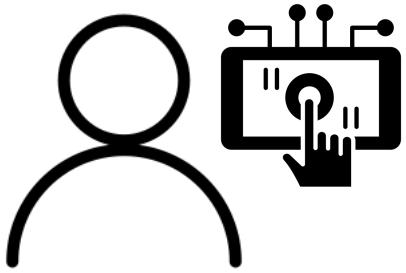
Reinforcement Learning

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
$r_\theta(s,a)$

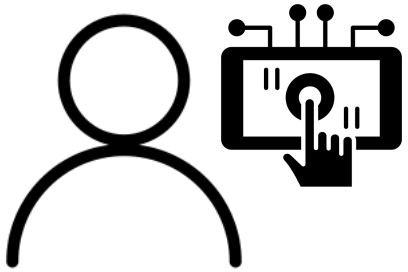
Reinforcement Learning

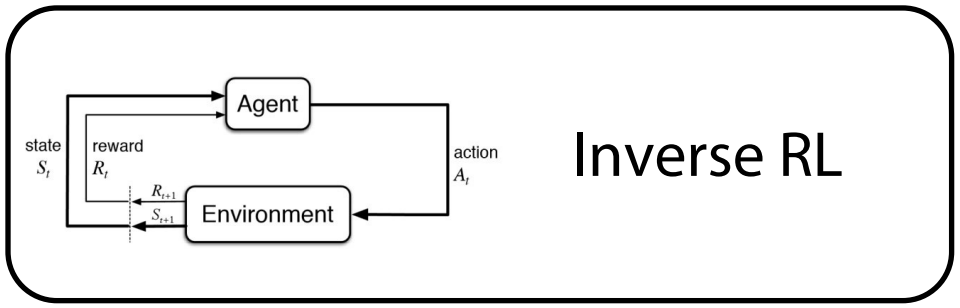
Policy
$\pi(a|s)$

New dynamics/state

Why is this hard?

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





Inverse RL

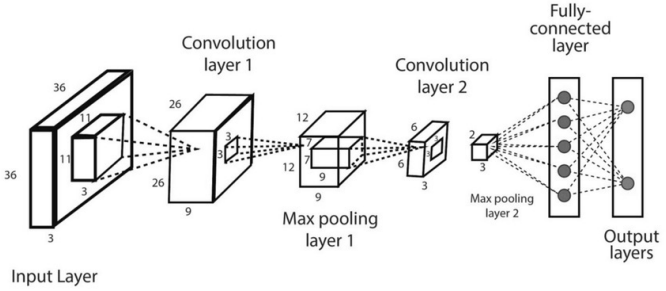
Challenging for a variety of reasons:

1. Inherently underspecified
2. R and π both unknown
3. Difficult optimization with T unknown.
4. Distributions/comparison metrics unknown

Reward Function

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

Can be parameterized by arbitrary function approximator



Lecture Outline

Recap – IRL formulation

↓

IRLv1 – max margin planning

↓

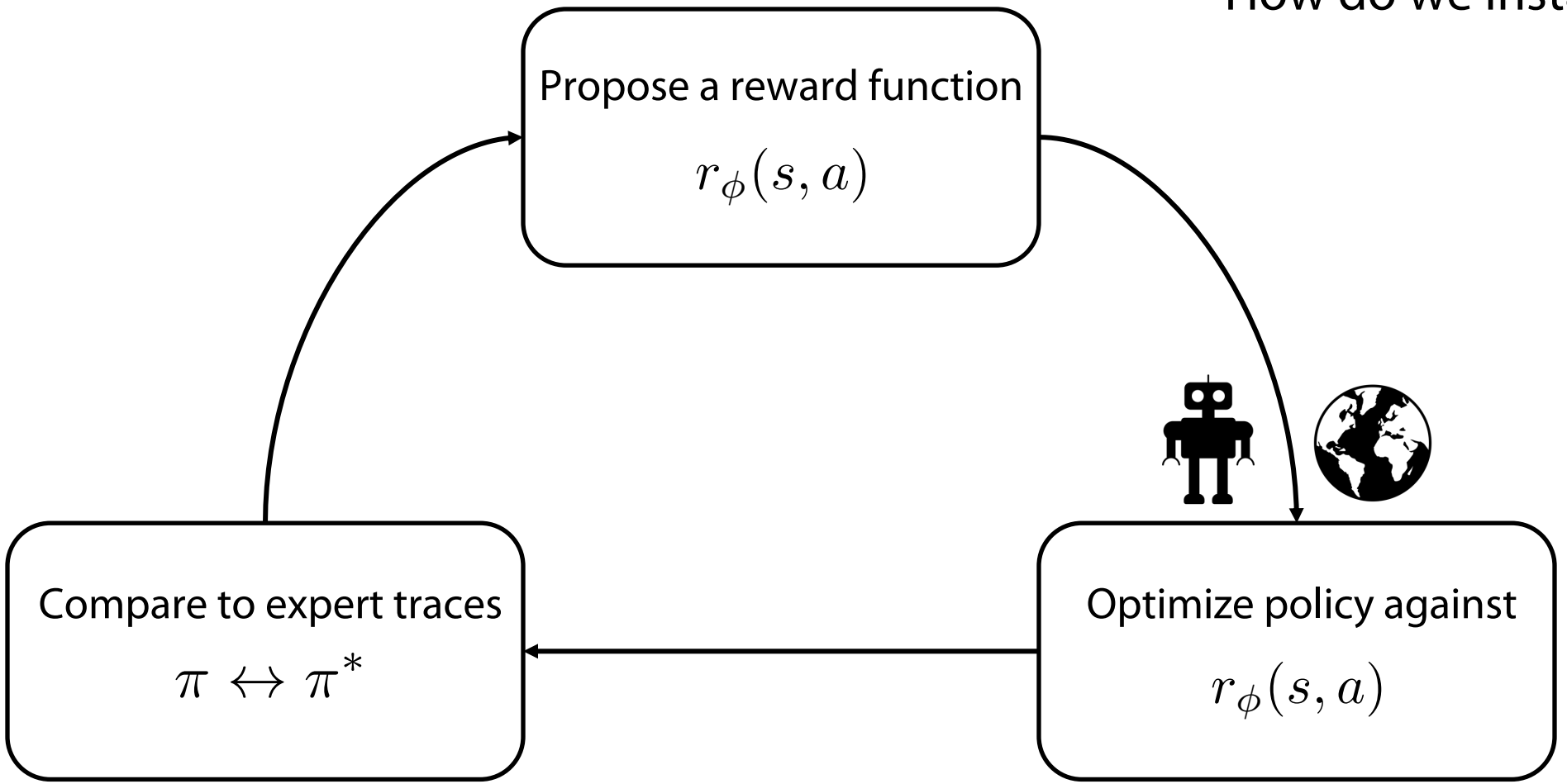
IRLv2 – max entropy IRL

↓

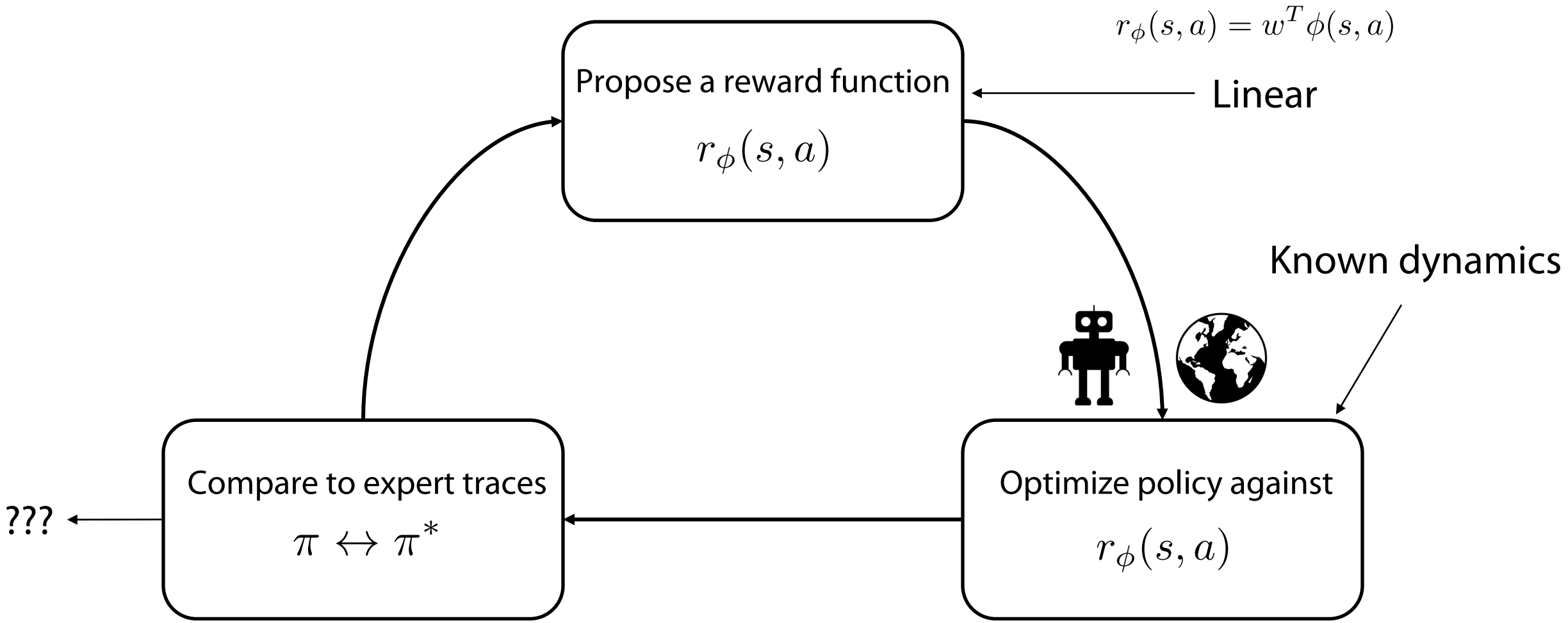
IRL as a GAN

A Formula for Inverse Reinforcement Learning

How do we instantiate?



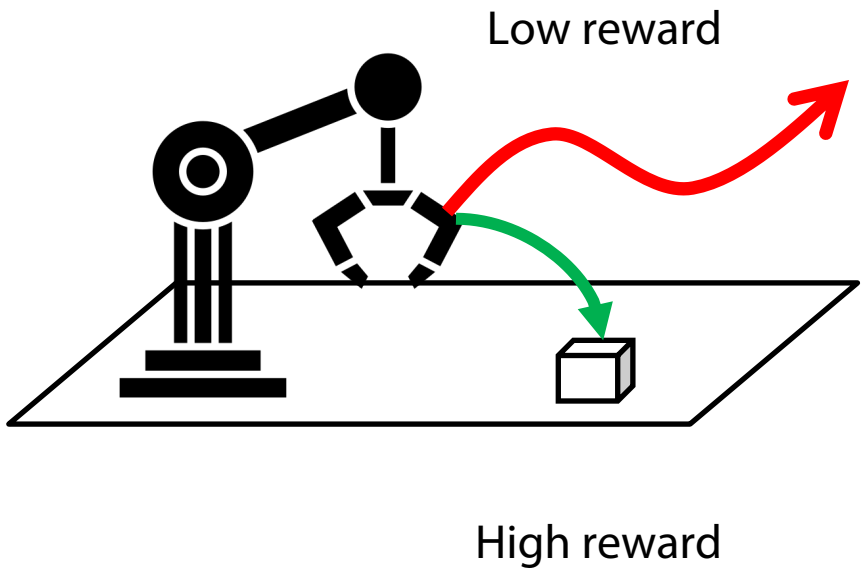
IRL v0 – Assumptions



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

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

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

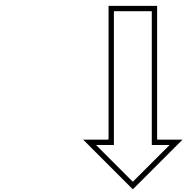
$$w^{*T}\underbrace{\mathbb{E}_{\pi^*}\left[\sum_t \gamma^t \phi(s_t,a_t)\right]}_{\mu(\pi^*,\phi)} \geq w^{*T}\underbrace{\mathbb{E}_{\pi}\left[\sum_t \gamma^t \phi(s_t,a_t)\right]}_{\mu(\pi,\phi)}, \quad \forall \pi$$

Underdefined, w* = 0 trivially satisfies!

IRL v0 – What is a good reward function?

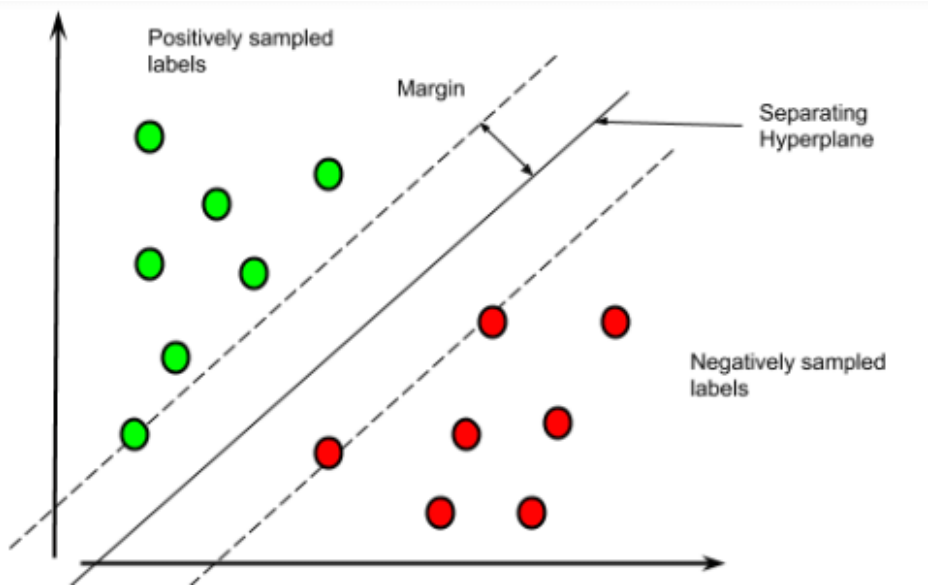
How do we tackle ambiguity?

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



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

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



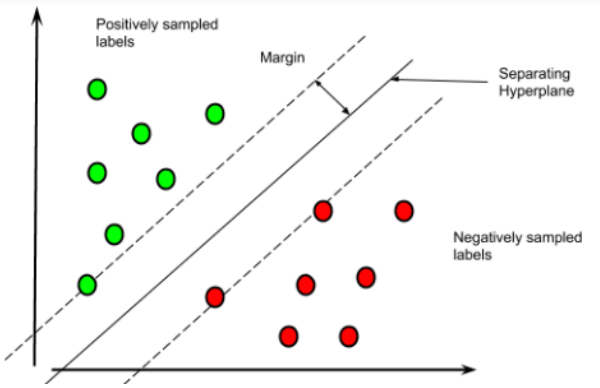
Find rewards which maximize the gap between the expert and all other policies

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 →

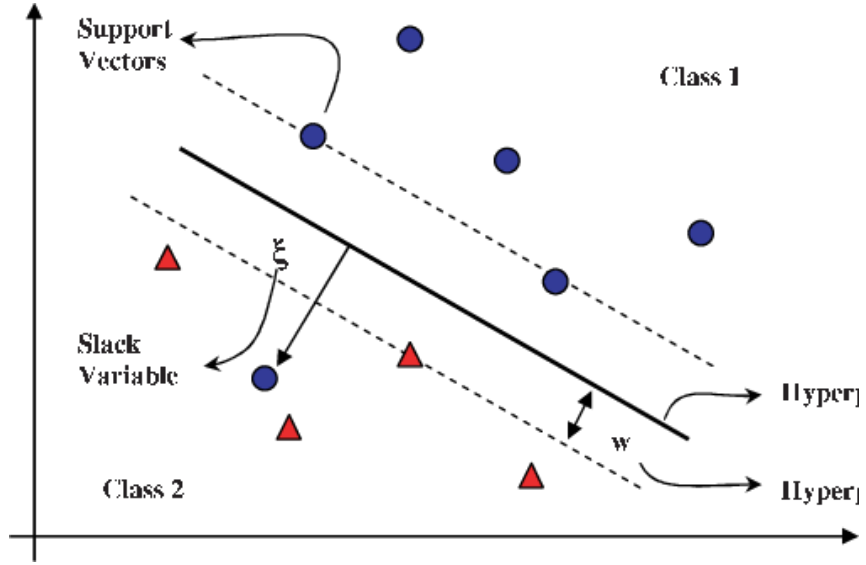
1. Uniform gap across all π, π^*
2. Noisily optimal may compromise the optimization

https://cs229.stanford.edu/lectures-spring2022/main_notes.pdf

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

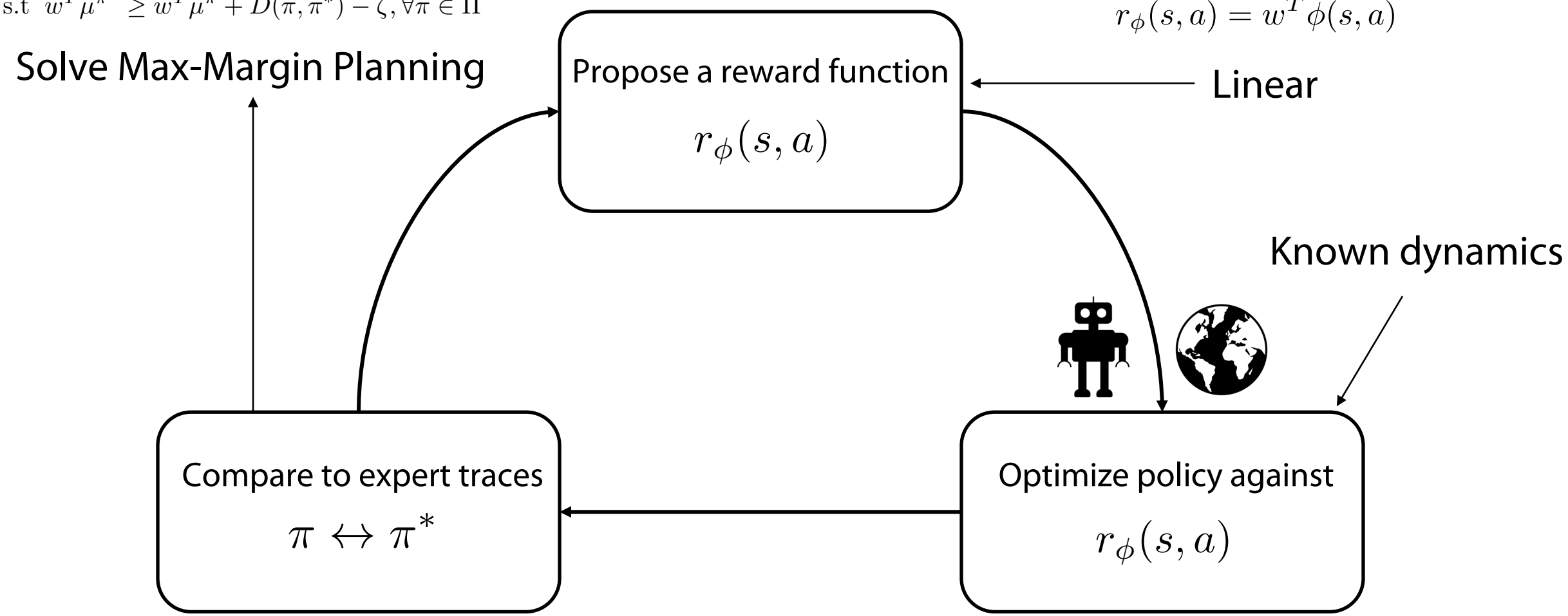
$$\min \|w\|_2 + C\zeta$$
$$\text{s.t} \;\; w^T \mu^{\pi^*} \geq w^T \mu^{\pi} + D(\pi, \pi^*) - \zeta, \forall \pi \in \Pi$$

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

Recap – IRL formulation

↓

IRLv1 – max margin planning

↓

IRLv2 – max entropy IRL

↓

IRL as a GAN

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

Can we do better?

Aside: Feature Matching

Instead of focusing on the reward function, focus on the feature expectations

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

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

$$= \left| w^T \mu(\pi^*) - w^T \mu(\pi) \right|$$

$$\leq \|w\|_2 \|\mu(\pi^*) - \mu(\pi)\|_2 \qquad \|w\|_2 < 1 \qquad \|\mu(\pi^*) - \mu(\pi)\|_2 < \epsilon$$

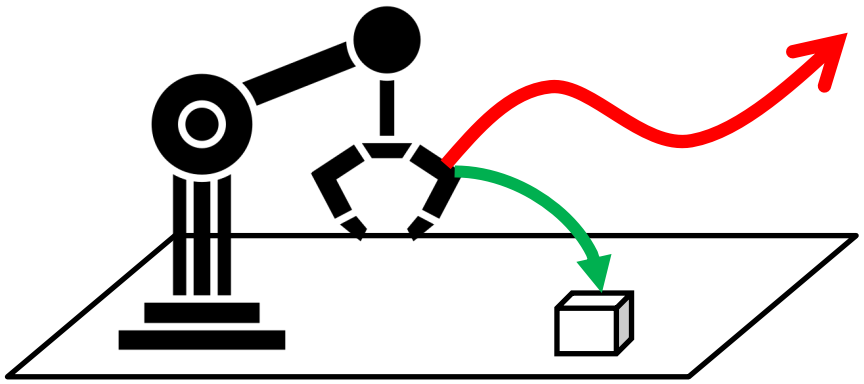
$$\leq \epsilon$$

=> If average feature expectations are close, then values are close

Abbeel, Ng '04

Intuition on Feature Matching

Let's provide some intuition



Features - distance to object
end effector position
object orientation
….

Matching features probably means that behavior is roughly similar

From max margin to max-ent IRL

Two key ideas in maximum-entropy IRL:

1. Prefer good trajectories → Feature matching
2. Weight other trajectories equally to deal with ambiguity → Maximum entropy

Notation:

Trajectory distribution – $p(\tau)$

Feature expectations:

Policy $\mu(p) = \mathbb{E}_{p(\tau)}\left[\sum_t \gamma^t \phi(s_t, a_t)\right]$

Expert $\mu(\pi^*) = \mathbb{E}_{\mathcal{D}^e}\left[\sum_t \gamma^t \phi(s_t, a_t)\right]$

$\max_p \ \mathcal{H}(p(\tau)) = -\int p(\tau)\log p(\tau) d\tau$ Max-entropy

$\mu(p) = \mu(\pi^*)$ Match features

$\int p(\tau) = 1$ Be a probability

Let's simplify

$$\max_p \ \mathcal{H}(p(\tau)) = -\int p(\tau)\log p(\tau)d\tau$$ Max-entropy

$$\mu(p) = \mu(\pi^*)$$ Match features

$$\int p(\tau) = 1$$ Be a probability

Set up the Lagrangian

$$\max_p \min_{w,\lambda} \mathcal{H}(p(\tau)) + w^T(\mu(p) - \mu(\pi^*)) - \lambda(\int p(\tau)d\tau - 1)$$

$$\min_{w,\lambda} \max_p \mathcal{H}(p(\tau)) + w^T(\mu(p) - \mu(\pi^*)) - \lambda(\int p(\tau)d\tau - 1)$$

Solve wrt p

Solve wrt w, λ

Connect the dots!

Let's simplify – solve for p

Set up the Lagrangian

$$\max_{p} \min_{w,\lambda} \mathcal{H}(p(\tau)) + w^T(\mu(p) - \mu(\pi^*)) - \lambda(\int p(\tau)d\tau - 1)$$

$$\min_{w,\lambda} \max_{p} \mathcal{H}(p(\tau)) + w^T(\mu(p) - \mu(\pi^*)) - \lambda(\int p(\tau)d\tau - 1)$$

Solve wrt p

$$\nabla_p \left[\mathcal{H}(p(\tau)) + w^T(\mu(p) - \mu(\pi^*)) - \lambda(\int p(\tau)d\tau - 1) \right] = 0$$

$$\nabla_p \left[-\int p(\tau)\log p(\tau)d\tau + w^T(\int p(\tau)\mu(\tau)d\tau - \mu(\pi^*)) - \lambda(\int p(\tau)d\tau - 1) \right] = 0$$

$$-\log p(\tau) - 1 + w^T\mu(\tau) - \lambda = 0$$

$$p(\tau) = \exp(-1 + w^T\mu(\tau) - \lambda)$$

Intuition: p(τ) is proportional to the exponential reward of a trajectory $w^T \mu(\tau)$

Let's simplify – solve for λ

$$\min_{w,\lambda} \max_{p} \mathcal{H}(p(\tau)) + w^T(\mu(p) - \mu(\pi^*)) - \lambda(\int p(\tau)d\tau - 1)$$

$$p(\tau) = \exp(-1 + w^T\mu(\tau) - \lambda)$$

$$\min_{w,\lambda} - \int p(\tau)\log p(\tau)d\tau + w^T(\int p(\tau)\mu(\tau)d\tau - \mu(\pi^*)) - \lambda(\int p(\tau)d\tau - 1)$$

$$\min_{w,\lambda} - \int p(\tau)(-1 + w^T\mu(\tau) - \lambda)d\tau + w^T(\int p(\tau)\mu(\tau)d\tau - \mu(\pi^*)) - \lambda(\int p(\tau)d\tau - 1)$$

$$\min_{w,\lambda} \int p(\tau)d\tau - w^T\mu(\pi^*) + \lambda$$

$$\min_{w,\lambda} \int \exp(-1 + w^T\mu(\tau) - \lambda)d\tau - w^T\mu(\pi^*) + \lambda \quad = \min_{w,\lambda} \exp(-1-\lambda) \int \exp(w^T\mu(\tau))d\tau - w^T\mu(\pi^*) + \lambda$$

Z

$$\nabla_\lambda \left[\exp(-1-\lambda)Z - w^T\mu(\pi^*) + \lambda \right] = 0 \implies \exp(-1-\lambda) = \frac{1}{Z}$$

$$\min_w 1 - w^T\mu(\pi^*) + \lambda$$

$$= \min_w \log Z - w^T\mu(\pi^*)$$

Ok – let's unpack what we have so far

$$\max_p \ \mathcal{H}(p(\tau)) = -\int p(\tau)\log p(\tau)d\tau$$ Max-entropy

$$\mu(p) = \mu(\pi^*)$$ Match features

$$\int p(\tau) = 1$$ Be a probability

Solve wrt p

$$p(\tau) = \exp(-1 + w^T\mu(\tau) - \lambda)$$

Solve wrt λ

$$Z = \int \exp(w^T\mu(\tau))d\tau \qquad \exp(-1-\lambda) = \frac{1}{Z}$$

Objective reduces to $\min_w \log Z - w^T\mu(\pi^*)$

⇩

Solve wrt *w*

Find reward function!

Turns out this has nice intuitive properties

$$\max_p \ \mathcal{H}(p(\tau)) = -\int p(\tau)\log p(\tau)d\tau$$ Max-entropy

$$\mu(p) = \mu(\pi^*)$$ Match features

$$\int p(\tau) = 1$$ Be a probability

⇩

Objective reduces to $\min_w \log Z - w^T\mu(\pi^*)$

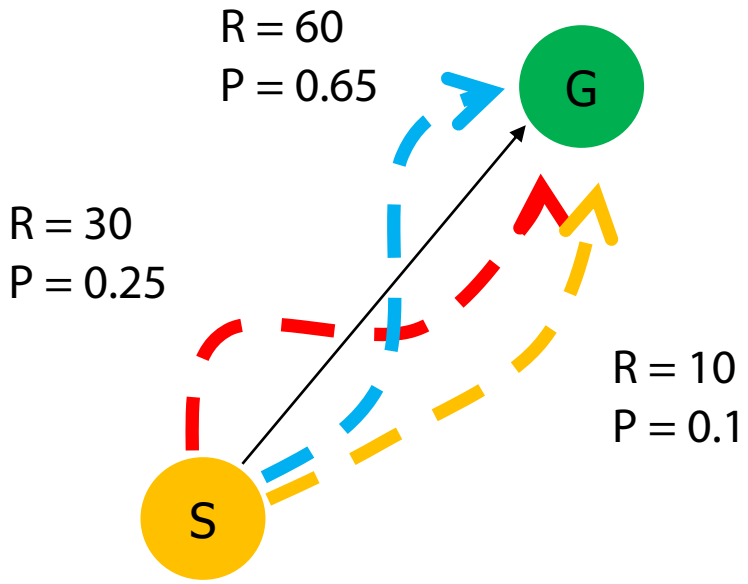
$$Z = \int \exp(w^T\mu(\tau))d\tau$$

⇩

$$\max_w \log \frac{\exp(w^T\mu(\pi^*))}{\int \exp(w^T\mu(\tau))d\tau}$$

Maximum likelihood with exponential family

$$= \max_w \mathbb{E}_{\tau^*\sim\mathcal{D}^e}\left[\log \frac{\exp(w^T\mu(\tau^*))}{\int \exp(w^T\mu(\tau))d\tau}\right]$$



Intuition: trajectories are chosen **<u>proportional</u>** to their reward

Turns out this has nice intuitive properties

$$\max_p \ \mathcal{H}(p(\tau)) = -\int p(\tau)\log p(\tau)d\tau$$ Max-entropy

$$\mu(p) = \mu(\pi^*)$$ Match features

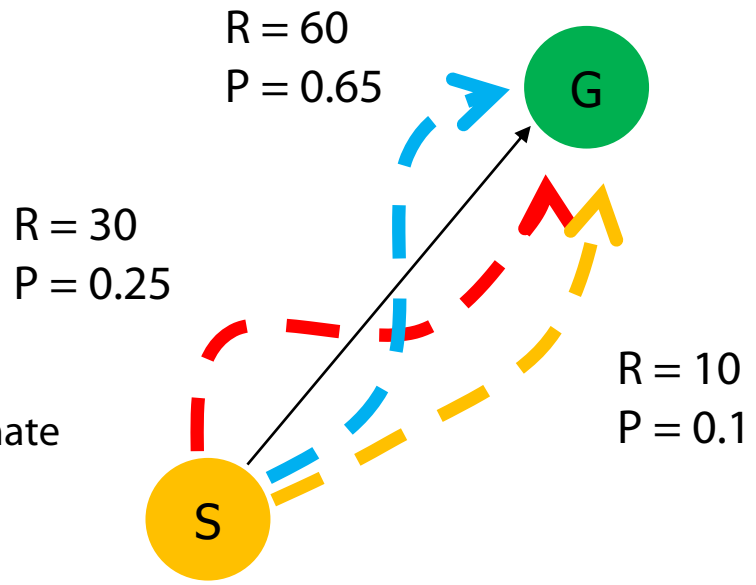
$$\int p(\tau) = 1$$ Be a probability

Maximum likelihood with exponential family

$$\max_w \mathbb{E}_{\tau^* \sim \mathcal{D}^e}\left[\log \frac{\exp(w^T\mu(\tau^*))}{\int \exp(w^T\mu(\tau))d\tau}\right]$$ → Hard to estimate

Intuition: trajectories are chosen **proportional** to their reward

Let's solve with gradient descent! Has a nice tractable form



Maximum likelihood estimation of w

$$\max_w \mathbb{E}_{\tau^* \sim \mathcal{D}^e} \left[\log \frac{\exp(w^T \mu(\tau^*))}{\int \exp(w^T \mu(\tau)) d\tau} \right]$$

$$J(w) = \mathbb{E}_{\tau^* \sim \mathcal{D}^e} \left[w^T \mu(\tau^*) \right] - \log \int \exp(w^T \mu(\tau)) d\tau$$

Gradient has a much nicer form

Painful to estimate log integral

$$\nabla J(w) = \nabla_w \mathbb{E}_{\tau^* \sim \mathcal{D}^e} \left[w^T \mu(\tau^*) \right] - \nabla_w \log \int \exp(w^T \mu(\tau)) d\tau$$

$$\nabla J(w) = \mathbb{E}_{\tau^* \sim \mathcal{D}^e} \left[\nabla_w w^T \mu(\tau^*) \right] - \frac{\int \exp(w^T \mu(\tau)) \nabla_w w^T \mu(\tau) d\tau}{\int \exp(w^T \mu(\tau)) d\tau}$$

$$\nabla J(w) = \mathbb{E}_{\tau^* \sim \mathcal{D}^e} \left[\nabla_w w^T \mu(\tau^*) \right] - \int p_w^*(\tau) \nabla_w w^T \mu(\tau) d\tau$$

$$\nabla J(w) = \mathbb{E}_{\tau^* \sim \mathcal{D}^e} \left[\nabla_w w^T \mu(\tau^*) \right] - \mathbb{E}_{\tau \sim p_w^*(\tau)} \left[\nabla_w w^T \mu(\tau) \right]$$

Push up on data

Push down on policy

Soft optimal policy for

$$r_w(s_t, a_t) = w^T \phi(s_t, a_t)$$

$$p_w^*(\tau) = \frac{\exp(w^T \mu(\tau))}{\int \exp(w^T \mu(\tau')) d\tau'}$$

IRLv2 – Maximum Entropy Inverse RL

$$\nabla J(w) = \mathbb{E}_{\tau^* \sim \mathcal{D}^e} \left[\nabla_w w^T \mu(\tau^*) \right] - \mathbb{E}_{\tau \sim p_w^*(\tau)} \left[\nabla_w w^T \mu(\tau) \right]$$

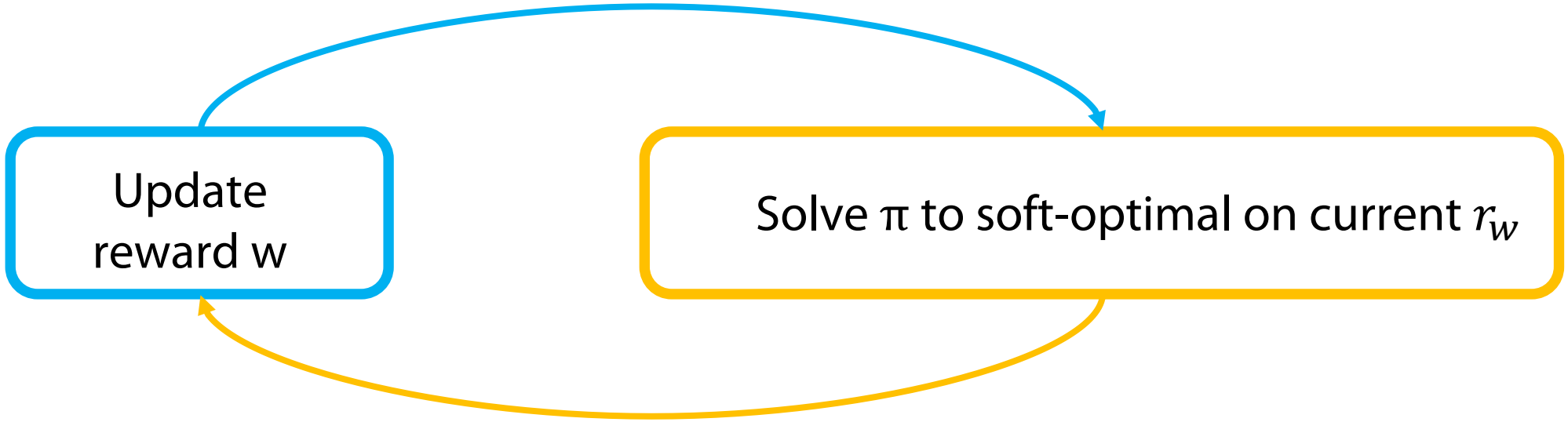
Push up on data

Push down on policy

Soft optimal policy for

$$r_w(s_t, a_t) = w^T \phi(s_t, a_t)$$

$$p_w^*(\tau) = \frac{\exp(w^T \mu(\tau))}{\int \exp(w^T \mu(\tau')) d\tau'}$$



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)

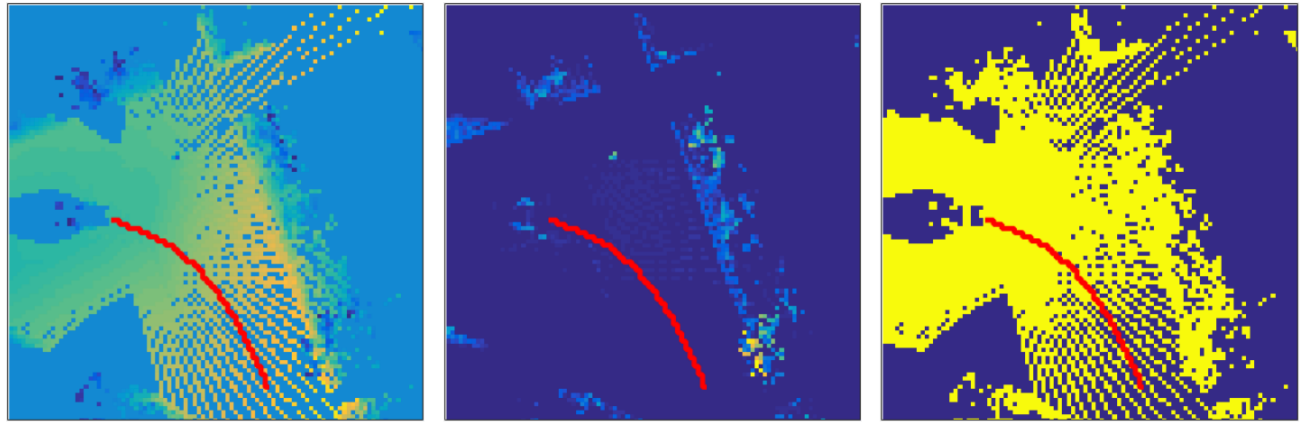
1. Start with a random policy π_0 and weight vector w
2. Find the "soft" optimal policy under w – $p_w(\tau)$
3. Take a gradient step on w

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

4. Repeat

Output the optimal reward function w*

Max-Ent IRL in Action

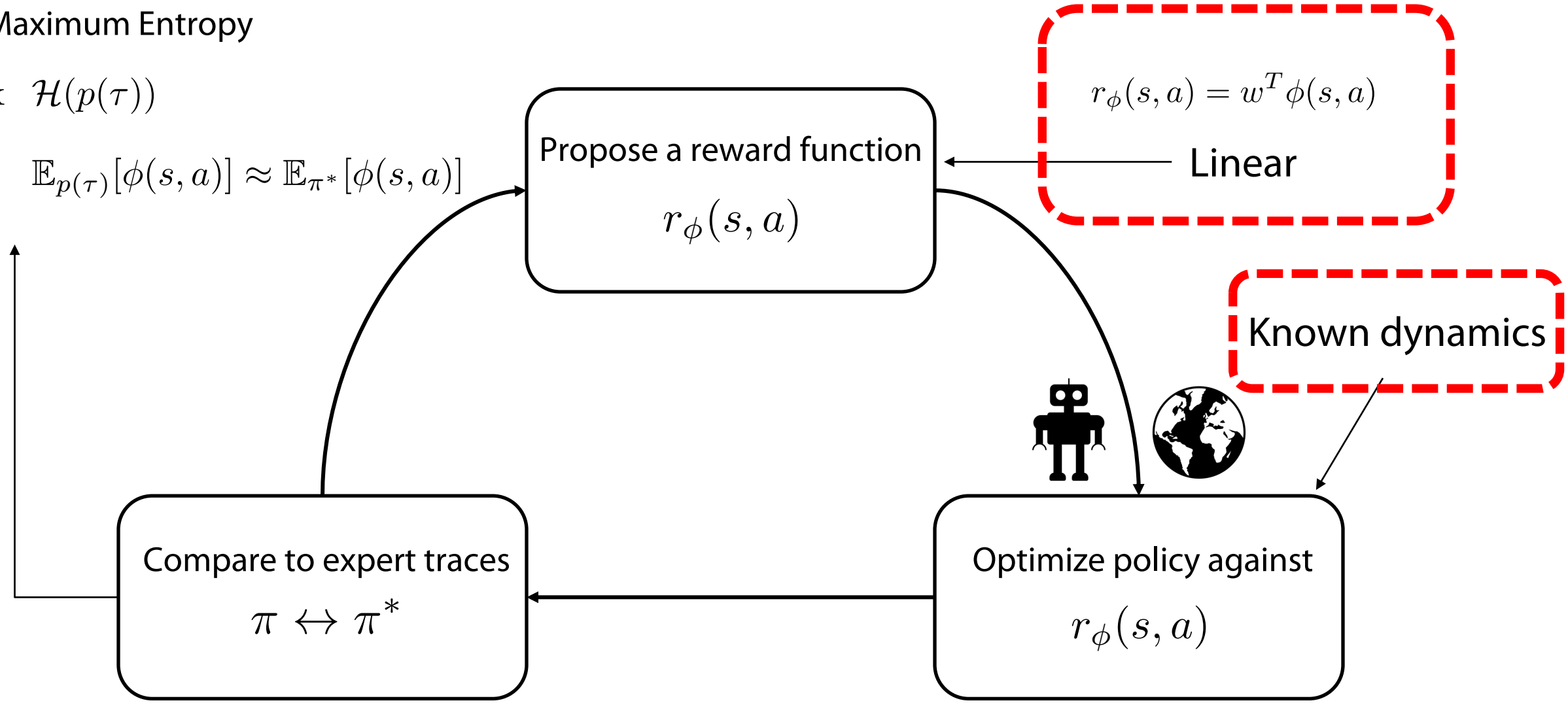


Ok but no way this could work?

Maximum Entropy

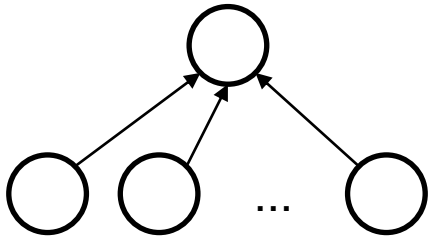
$$\max_{p(\tau)} \quad \mathcal{H}(p(\tau))$$

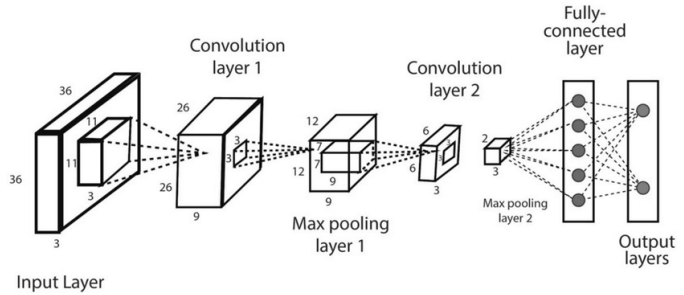
$$\text{s.t} \quad \mathbb{E}_{p(\tau)}[\phi(s,a)] \approx \mathbb{E}_{\pi^*}[\phi(s,a)]$$



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

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

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

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

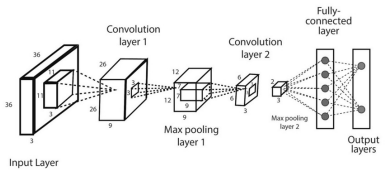
Can transfer significantly more from iteration to iteration rather than doing full nested optimization

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

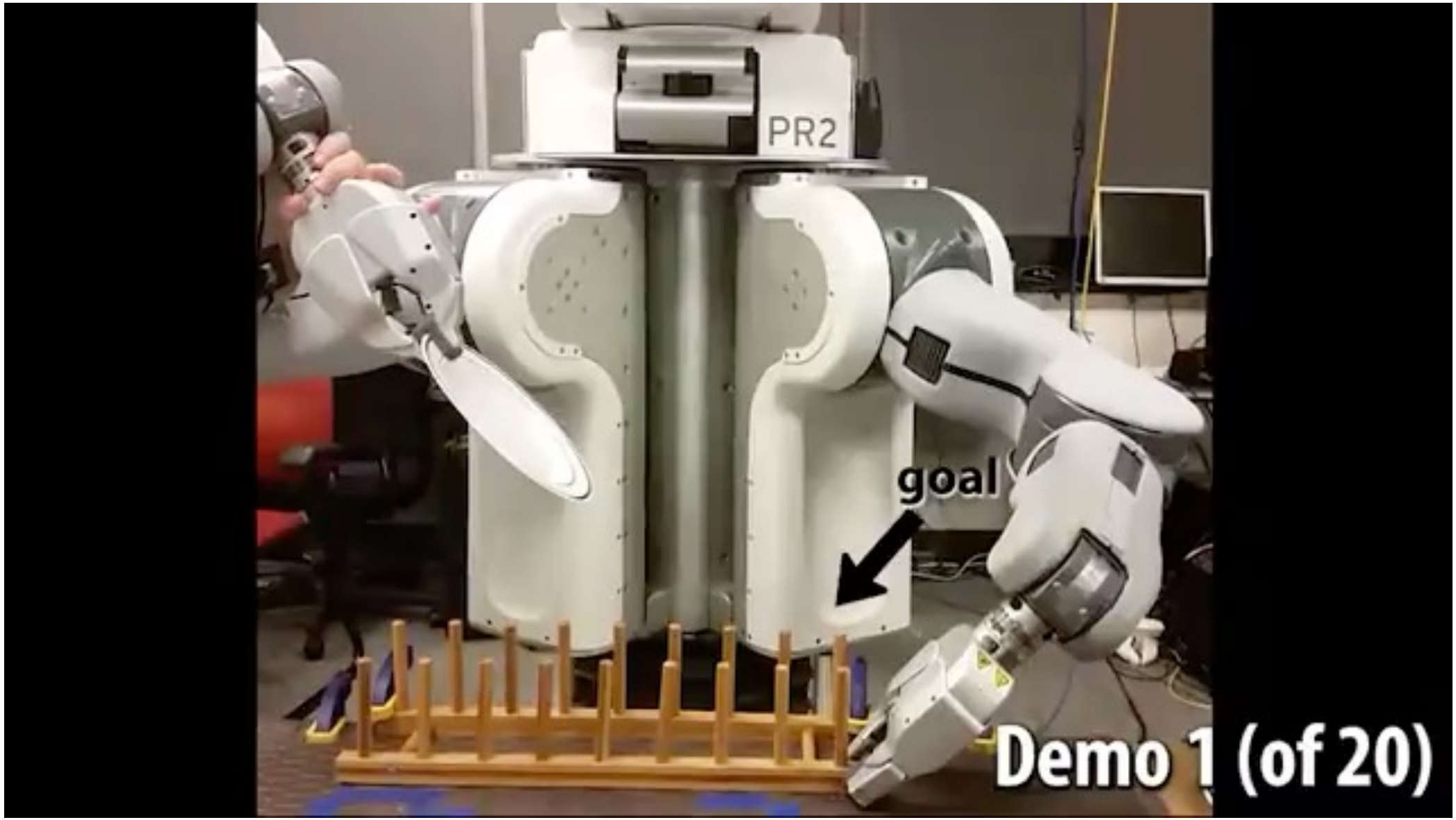
Optimize policy against

$r_\phi(s, a)$

Compare to expert traces

$\pi \leftrightarrow \pi^*$

IRLv4 – Guided Cost Learning



Lecture Outline

Recap – IRL formulation

↓

IRLv1 – max margin planning

↓

IRLv2 – max entropy IRL

↓

IRL as a GAN

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

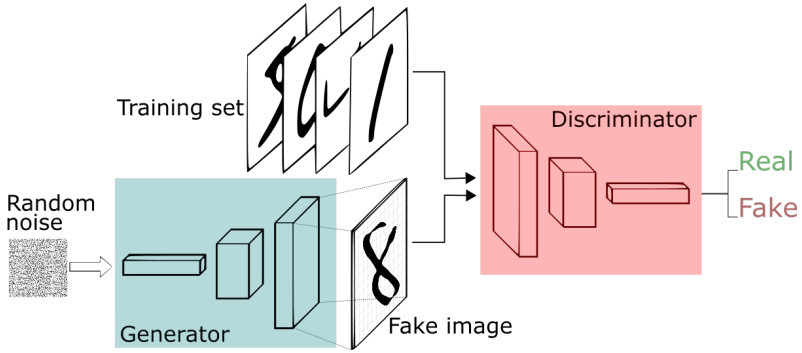
$$\nabla_\theta \mathcal{L} = \mathbb{E}_{\pi^*}\left[\sum_t \gamma^t \nabla_\theta r_\theta(s_t, a_t)\right] - \mathbb{E}_q\left[\frac{p_w(\tau)}{q(\tau)} \sum_t \gamma^t \nabla_\theta r_\theta(s_t, a_t)\right]$$

4. Repeat

Output the optimal reward function w*

Reminder: Generative Adversarial Networks

Technique to learn generative models via a 2 player game



https://sthalles.github.io/intro-to-gans/

Key idea: Generator tries to "confuse" the discriminator.
At convergence generated samples indistinguishable from real samples

$$\min_G \max_D V(D,G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p(z)} [\log(1 - D(G(z)))]$$

Often approximate generator loss as:

$$\min_G \mathbb{E}_{z \sim p(z)} [\log(1 - D(G(z)))] - \mathbb{E}_{z \sim p(z)} [\log(D(G(z)))]$$

Can inverse RL be considered a GAN?

Generator = policy
Discriminator = reward (kinda)

Find a policy which makes a discriminator unable to tell if the samples came from the policy or the demos

$$\min_G \max_D V(D,G) = \mathbb{E}_{\tau \sim p_{\text{demo}}(\tau)} [\log D(\tau)] + \mathbb{E}_{\tau \sim \pi} [\log(1 - D(\tau))]$$

Push up real data

Push down policy data

Discriminator trained with classification between expert/non-expert

Generator trained to max log D with RL

Challenge: only policy, not really a reward

Generative Adversarial Imitation Learning

Jonathan Ho
Stanford University
hoj@cs.stanford.edu

Stefano Ermon
Stanford University
ermon@cs.stanford.edu

Tweaking GAIL to connect with IRL

We can make simple tweaks to GAIL to get back to max-ent IRL

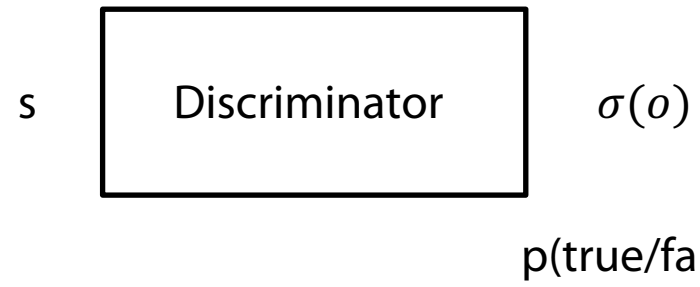
Optimal discriminator

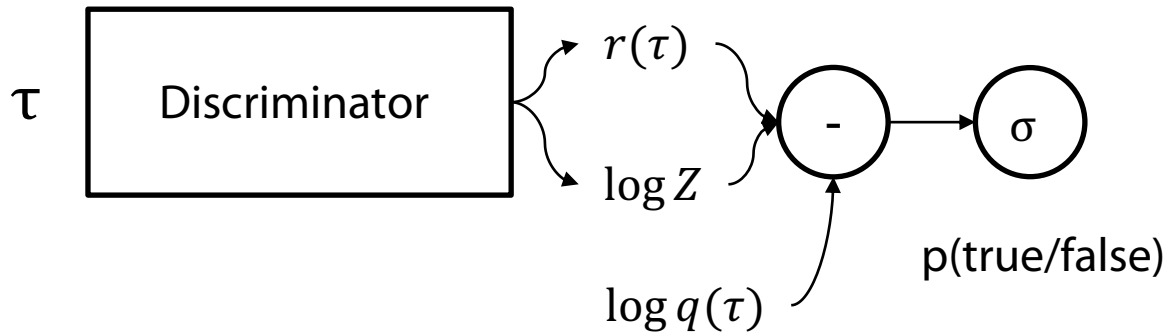
$$D^*(x) = \frac{p(x)}{p(x) + q(x)}$$

Choose a particular form of discriminator

Policy informed discriminator

$$D_\theta(\tau) = \frac{\frac{1}{Z}\exp(r_\theta(\tau))}{\frac{1}{Z}\exp(r_\theta(\tau)) + q(\tau)}$$





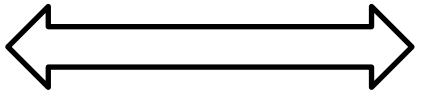
Recasting GAIL as an IRL method

For a particular parameterization of the discriminator, we can show that GAN = max-ent IRL

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

GAN

$$\mathbb{E}_{\tau \sim p_{\text{demo}}(\tau)}\left[\log D(\tau)\right] + \mathbb{E}_{\tau \sim \pi}\left[\log(1 - D(\tau))\right]$$

Push up real data, push down generated

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

Generator Optimization as Max-Ent RL

$$\min_G \mathbb{E}_{z\sim p(z)}\left[\log(1 - D(G(z)))\right] - \mathbb{E}_{z\sim p(z)}\left[\log(D(G(z)))\right]$$

$$D_\theta(\tau) = \frac{\frac{1}{Z}\exp(r_\theta(\tau))}{\frac{1}{Z}\exp(r_\theta(\tau)) + q(\tau)}$$

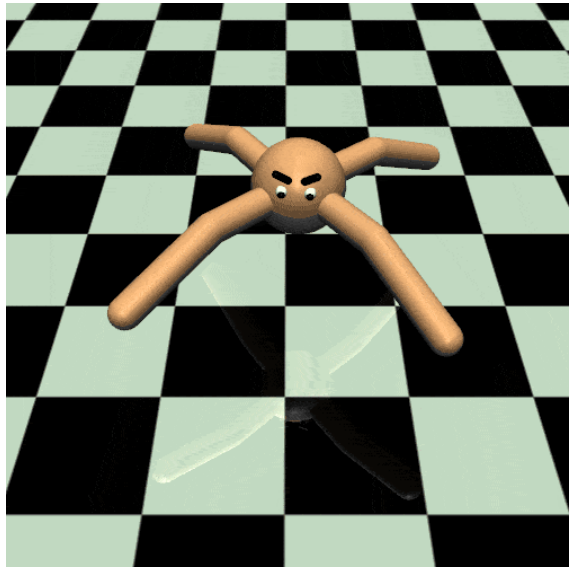
$$\min \mathbb{E}_{\tau\sim q(\tau)}\left[\log \frac{q(\tau)}{\cancel{\frac{1}{Z}\exp(r_\theta(\tau)) + q(\tau)}} - \log \frac{\frac{1}{Z}\exp(r_\theta(\tau))}{\cancel{\frac{1}{Z}\exp(r_\theta(\tau)) + q(\tau)}}\right]$$

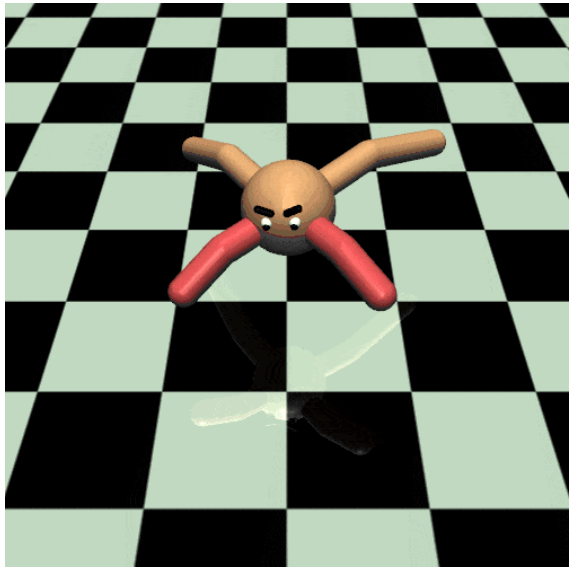
$$\max \mathbb{E}_{\tau\sim q(\tau)}\left[r_\theta(\tau) - \cancel{\log Z} - \log q(\tau)\right]$$

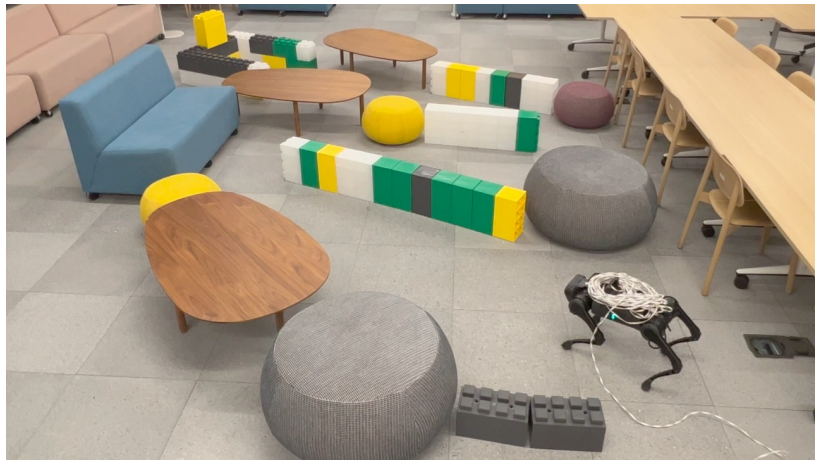
Maximum entropy RL with current reward!

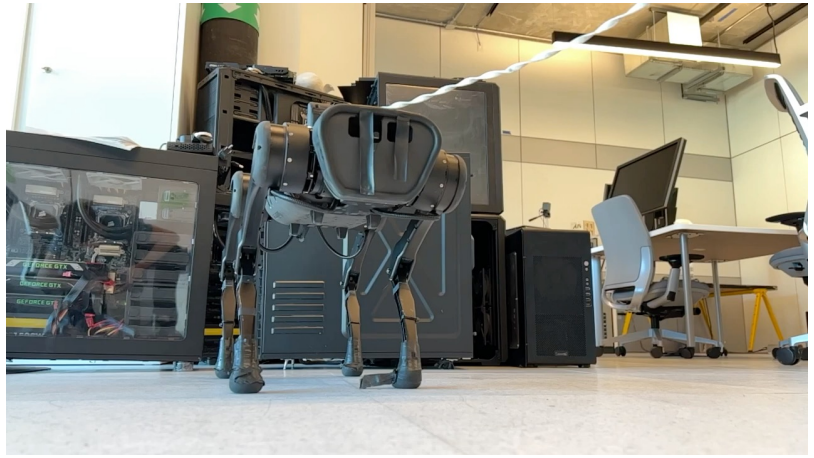
Similar proof holds for the discriminator optimization – refer to https://arxiv.org/pdf/1611.03852

Adversarial IRL in Action









Lecture Outline

Recap – IRL formulation

↓

IRLv1 – max margin planning

↓

IRLv2 – max entropy IRL

↓

IRL as a GAN