535514: Reinforcement Learning
Lecture 25 — SAC and Imitation Learning

Ping-Chun Hsieh

May 27, 2024

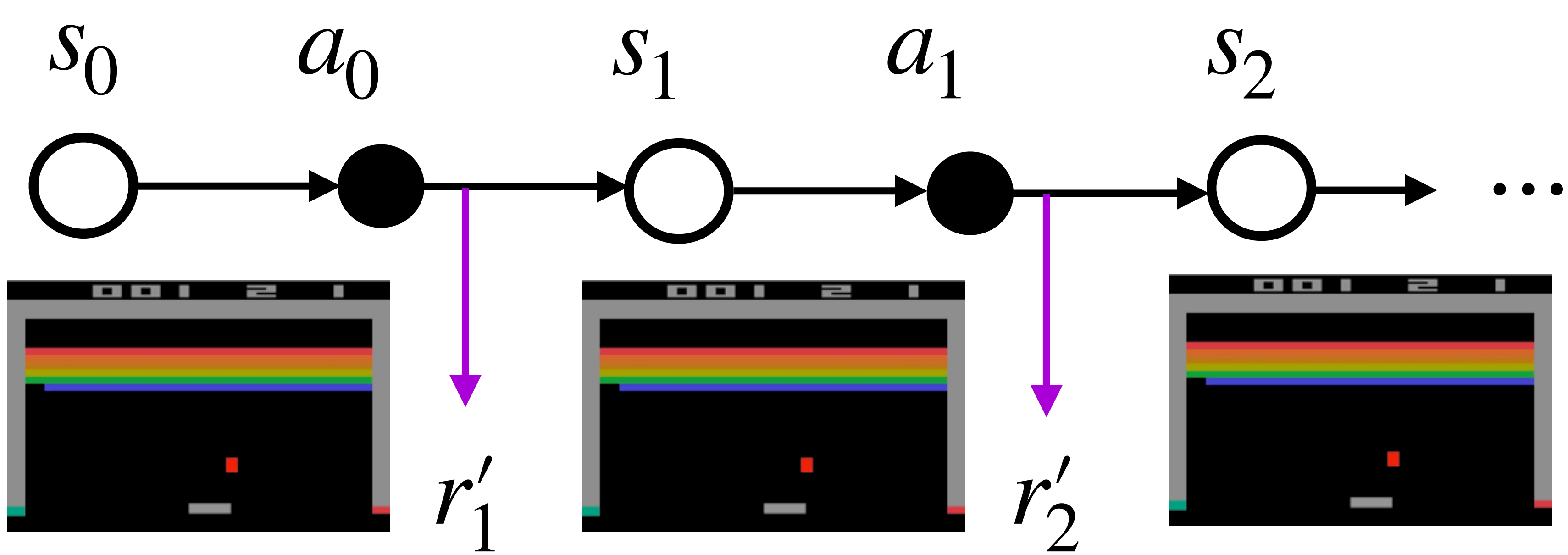
On-Policy vs Off-Policy Methods

<table>
<tr><th></th><th>Policy Optimization</th><th>Value-Based</th><th>Model-Based</th><th>Imitation-Based</th></tr>
<tr><td>On-Policy</td><td>Exact PG
REINFORCE (w/i baseline)
A2C
On-policy DAC
TRPO
Natural PG (NPG)
PPO-KL & PPO-Clip
RLHF by PPO-KL</td><td>Epsilon-Greedy MC
Sarsa
Expected Sarsa</td><td rowspan="2">Model-Predictive Control (MPC)
PETS</td><td rowspan="2">IRL
GAIL
IQ-Learn</td></tr>
<tr><td>Off-Policy</td><td>Off-policy DPG & DDPG
Twin Delayed DDPG (TD3)</td><td>Q-learning
Double Q-learning
DQN & DDQN
Rainbow
C51 / QR-DQN / IQN
Soft Actor-Critic (SAC)</td></tr>
</table>

Soft Policy Iteration

Review: *Regularized MDPs*

Regularized MDP = Standard MDP + Regularized rewards!



$= r_1 + \Omega(\pi(\cdot \mid s_0))$ $\quad$ $= r_2 + \Omega(\pi(\cdot \mid s_1))$

- A regularized MDP can be specified by $(\mathcal{S}, \mathcal{A}, P, R, \Omega, \gamma)$
  - $\Omega(\cdot)$: A function that maps an *action distribution* to a *real number*

Question: How to define $Q^\pi(s, a)$ and $V^\pi(s)$ with a regularizer?

Review: Value Functions of *Regularized MDPs*

	Unregularized MDP	Entropy-Regularized MDP
Return	$G_t := r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots$	$G_t := r_{t+1} + \gamma\big(r_{t+2} + \Omega(\pi(\cdot\,\vert s_{t+1})\big)$ $+\gamma^2\big(r_{t+3} + \Omega(\pi\cdot\vert s_{t+2})\big) + \cdots$
Value function	$V^\pi(s) := \mathbb{E}[G_t \,\vert\, s_t = s; \pi]$	$V^\pi_\Omega(s) := \mathbb{E}\big[G_t \vert s_t = s; \pi\big] + \Omega(\pi(\cdot\,\vert s))$
Q function	$Q^\pi(s,a) := \mathbb{E}[G_t \,\vert\, s_t = s, a_t = a; \pi]$	$Q^\pi_\Omega(s,a) := \mathbb{E}\big[G_t \vert s_t = a, a_t = a; \pi\big]$
Bellman expectation equations	$V^\pi(s) = \sum_{a\in\mathcal{A}} \pi(a\,\vert\, s) Q^\pi(s,a)$ $Q^\pi(s,a) = R_{s,a} + \gamma \sum_{s'\in\mathcal{S}} P^a_{ss'} V^\pi(s')$	$V^\pi_\Omega(s) = \sum_{a\in\mathcal{A}} \pi(a\vert s) Q^\pi_\Omega(s,a) + \Omega(\pi(\cdot\,\vert s))$ $Q^\pi_\Omega(s,a) = R_{s,a} + \gamma \sum_{s'\in\mathcal{S}} P^a_{s,s'} V^\pi_\Omega(s')$

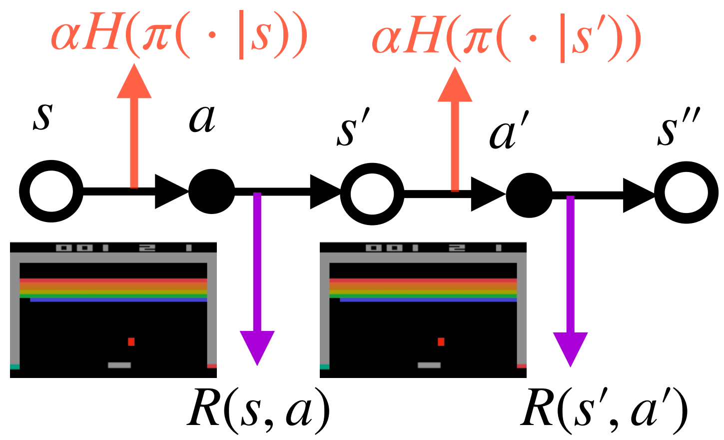
If $\Omega(\pi(\cdot\,\vert s)) \equiv \alpha \cdot H(\pi(\cdot\,\vert s))$, the value functions are called “**soft functions**”

Soft Policy Evaluation for Soft Q-Function

Let's extend **policy evaluation** to entropy-regularized case

$$Q^{\pi}_{soft}(s,a) = R^a_s + \gamma E_{s' \sim P(\cdot|s,a)}[V^{\pi}_{soft}(s')]$$

$$(= R^a_s + \gamma E_{s' \sim P(\cdot|s,a), a' \sim \pi(\cdot|s')}[Q^{\pi}_{soft}(s',a') - \alpha \log(\pi(a'|s'))])$$

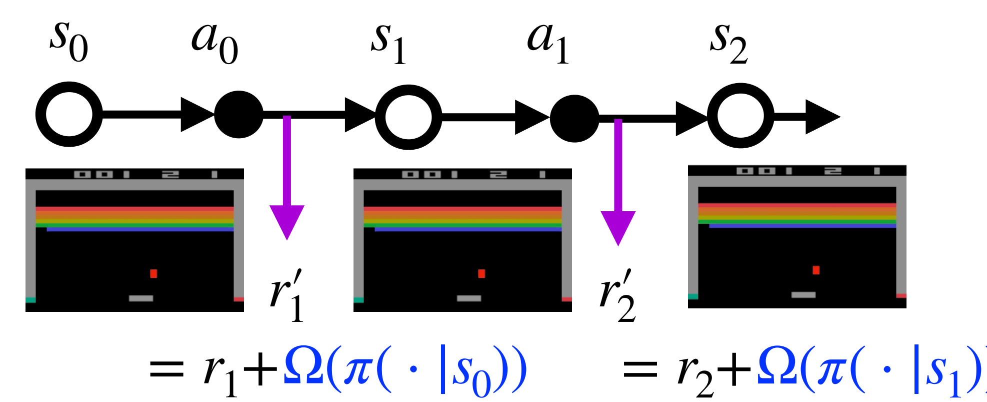


Soft policy evaluation: Find $Q^{\pi}_{soft}(s,a)$ for a policy π

1. (Optimal control) Given R, P, and a policy π:

2. (Learning) Given a policy π (with unknown R, P):

Review: Optimal Value Functions and Bellman Optimality Equations of *Regularized MDPs*



	Unregularized MDP	Regularized MDP
Bellman optimality equations	$V^*(s) := \max_{\pi\in\Pi} V^\pi(s)$ $Q^*(s,a) := \max_{\pi\in\Pi} Q^\pi(s,a)$	$V^*_\Omega(s) := \max_{\pi\in\Pi} V^\pi_\Omega(s)$ $Q^*_\Omega(s,a) := \max_{\pi\in\Pi} Q^\pi_\Omega(s,a)$
Bellman optimality equations	$V^*(s) = \max_{a\in\mathscr{A}} R^a_s + \gamma P^a_s V^*$ $= \max_{\pi\in\Pi} R^\pi_s + \gamma P^\pi_s V^*$ $Q^*(s,a) = R^a_s + \gamma E_{s'\sim P(\cdot\|s,a)}[V^*(s')]$	$V^*_\Omega(s) = \max_{\pi\in\Pi} R^\pi_s + \gamma P^\pi_s V^*_\Omega$ $Q^*_\Omega(s,a) = R^a_s + \gamma E_{s'\sim P(\cdot\|s,a)}[V^*_\Omega(s')]$ $=$

Soft Policy Improvement

Bellman optimality equation

$$Q^*_{soft}(s,a) = R_s^a + \gamma E_{s'\sim P(\cdot|s,a)}[V^*_{soft}(s')]$$

$$(= R_s^a + \gamma E_{s'\sim P(\cdot|s,a)}[\max_\pi \left\{ \langle \pi(\cdot|s'), Q^\pi_{soft}(s',\cdot)\rangle + \alpha H(\pi(\cdot|s'))\right\}])$$

Soft policy improvement: Given π_k, improve the policy by

$$\pi_{k+1}(\cdot \mid s) = \arg\max_\pi \left\{ \langle \pi(\cdot \mid s), Q^{\pi_k}_{soft}(s,\cdot)\rangle + \alpha H(\pi(\cdot \mid s))\right\}$$

Solution to Soft Policy Improvement

- **Theorem**: Under soft policy iteration, we have

$$\pi_{k+1}(\,\cdot\,|\,s) = \arg\max_{\pi} \left\{ \langle \pi(\,\cdot\,|\,s), Q^{\pi_k}_{soft}(s,\,\cdot\,) \rangle + \alpha H(\pi(\,\cdot\,|\,s)) \right\}$$

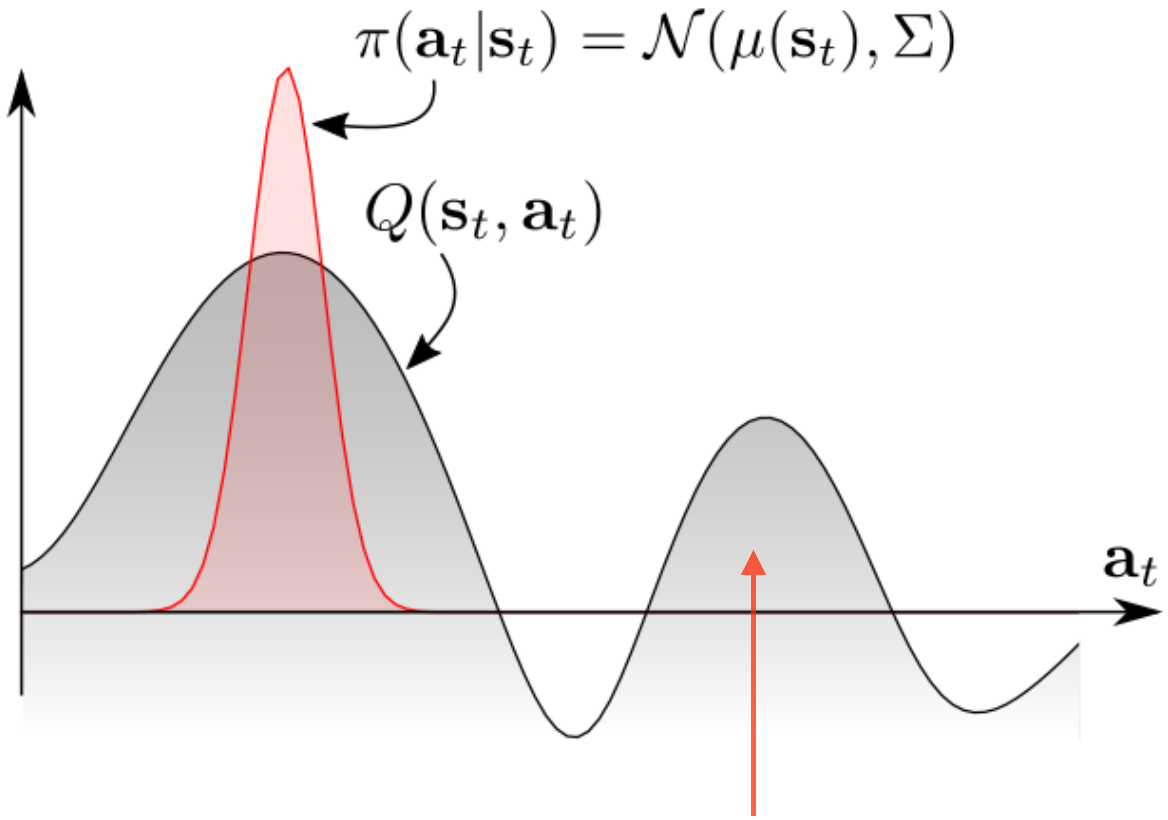
$$= \frac{\exp\left(\frac{1}{\alpha} Q^{\pi_k}_{soft}(s,\,\cdot\,)\right)}{\sum_{a \in \mathscr{A}} \exp\left(\frac{1}{\alpha} Q^{\pi_k}_{soft}(s, a)\right)}$$

- Question: Could you explain why this is called "soft" policy improvement?

Why is Soft Policy Improvement a Good Idea?

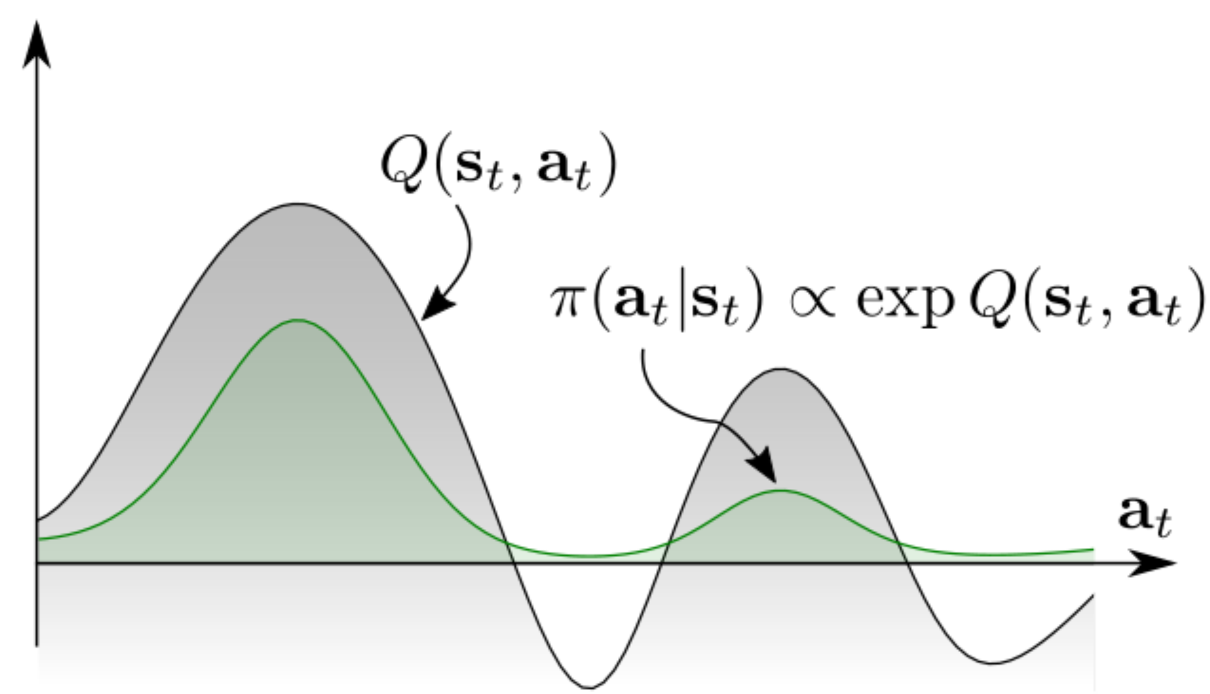
Suppose we'd like to use stochastic policies

Standard Gaussian policies



Unimodal policies completely ignore this part

Energy-based policies



https://bair.berkeley.edu/blog/2017/10/06/soft-q-learning/

Soft Policy Iteration (Soft PI)

Soft Policy Iteration

1. Initialize $k = 0$ and set $\pi_0(\cdot \mid s)$ arbitrarily for all states
2. While $\underline{k \text{ is zero}}$ or $\underline{\pi_k \neq \pi_{k-1}}$:
   - Derive $V_{soft}^{\pi_k}$ and $Q_{soft}^{\pi_k}$ via soft policy evaluation
   - Derive π_{k+1} by greedy soft policy improvement:

$$\pi_{k+1}(\cdot \mid s) = \arg\max_{\pi} \left\{ \langle \pi(\cdot \mid s), Q_{soft}^{\pi_k}(s, \cdot) \rangle + H(\pi(\cdot \mid s)) \right\}$$

Soft Actor-Critic (SAC)

Soft Actor-Critic: The “Learning” Version of Soft-PI

State $\longrightarrow$ $\pi_\theta(\cdot|s)$ $\longrightarrow$

Policy Network (Actor)

State, Action $\longrightarrow$ $Q_\phi(s,a)$ $\longrightarrow$

Q Network (Critic)

State $\longrightarrow$ $V_\psi(s)$ $\longrightarrow$

Value Network

Actor Loss

$$L_\pi(\theta) = \mathbb{E}_{s\sim D}\Big[D_{KL}\Big(\pi_\theta(\cdot|s)\Big\|\underbrace{\frac{\exp(Q_{\bar\phi}(s,\cdot))}{Z_{\bar\phi}(s)}}_{\rho_Q(\cdot|s)}\Big)\Big]$$

$$= \mathbb{E}_{s\sim D, a\sim\pi_\theta}\Big[\log \pi_\theta(a|s) - Q_{\bar\phi}(s,a) + \log Z_{\bar\phi}(s)\Big]$$

Critic Loss

$$L_Q(\phi) = \mathbb{E}_{(s,a)\in D}\Big[\frac{1}{2}(Q_\phi(s,a) - (\mathbb{E}_{r,s'}[r + \gamma V_{\bar\psi}(s')|s,a]))^2\Big]$$

$$\nabla_\phi L(\phi) =$$

Value Loss

$$L(\psi) = \mathbb{E}_{s\sim D}\Big[\frac{1}{2}\Big(V_\psi(s) - \mathbb{E}_{a\sim\pi_{\bar\theta}(\cdot|s)}[Q_{\bar\phi}(s,a) - \log \pi_{\bar\theta}(a|s)]\Big)^2\Big]$$

$$\nabla_\psi L_V(\psi) =$$

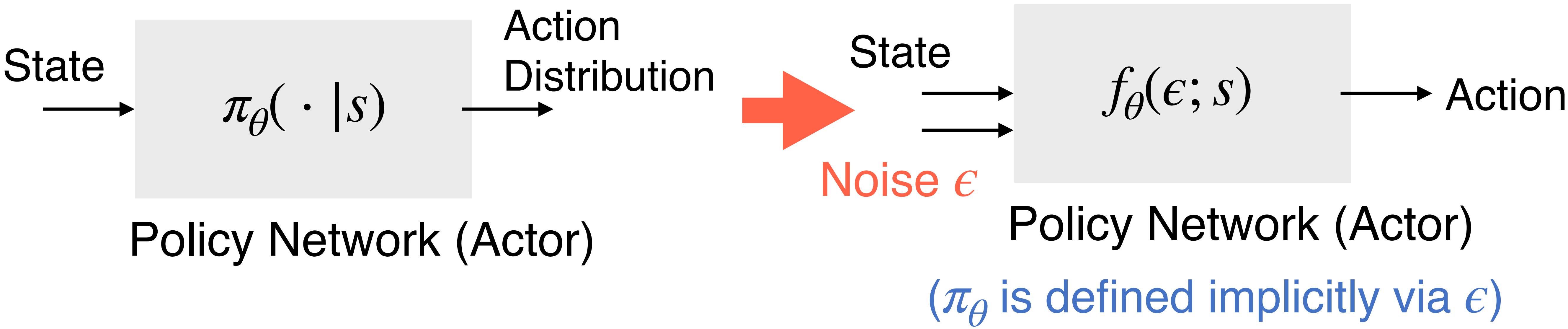
Inherent Difficulty in Finding $\nabla L_\pi(\theta)$

$$L_\pi(\theta) = \mathbb{E}_{s\sim D, a\sim\pi_\theta}\Big[\log \pi_\theta(a|s) - Q_{\bar{\phi}}(s,a) + \log Z_{\bar{\phi}}(s)\Big]$$

Issue: Is it easy to directly compute the gradient of this KL divergence?

$$\nabla_\theta L_\pi(\theta) = \nabla_\theta \mathbb{E}_{s\sim D, a\sim\pi_\theta}\Big[\log \pi_\theta(a|s) - Q_{\bar{\phi}}(s,a) + \log Z_{\bar{\phi}}(s)\Big]$$

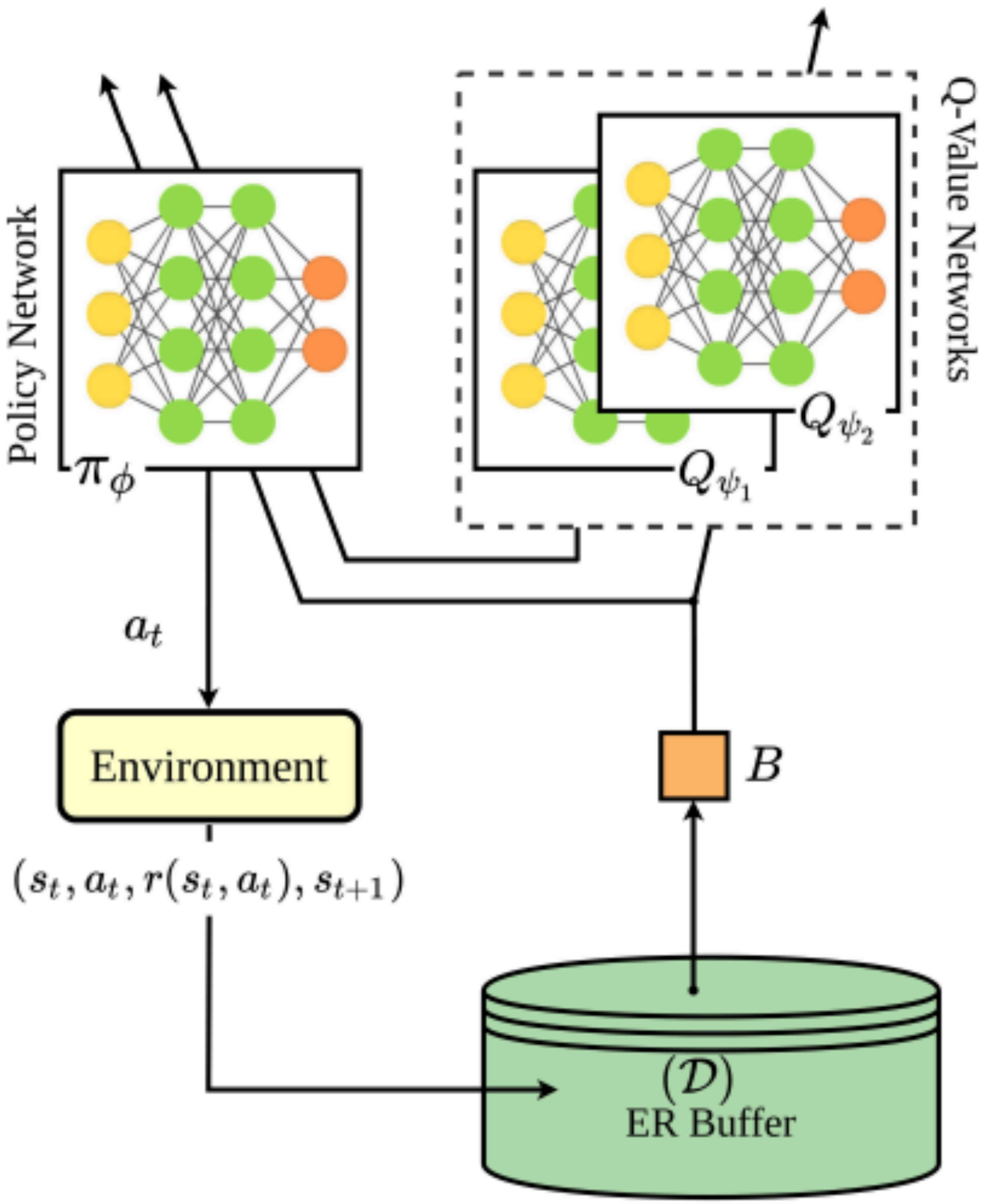
Actor Loss Under Reparameterization Trick



Reparamterization Trick:

$$L_\pi(\theta) = \mathbb{E}_{s\sim D, \epsilon\sim G}\Big[\log \pi_\theta(f_\theta(\epsilon; s)|s) - Q_{\bar{\phi}}(s, f_\theta(\epsilon; s))\Big]$$

Architecture of SAC



1. Clipped double Q networks as TD3
2. Gaussian policies
3. Experience replay buffer for off-policy learning

 (Figure Source: https://arxiv.org/pdf/2109.11767.pdf)

Imitation Learning

Imitation Learning: 2 Major Paradigms

- Suppose we are given *expert demonstrations*.
  How to learn from them?

1. Direct imitation learning
   - Copy the actions of the expert
   - No reasoning about the outcomes of actions

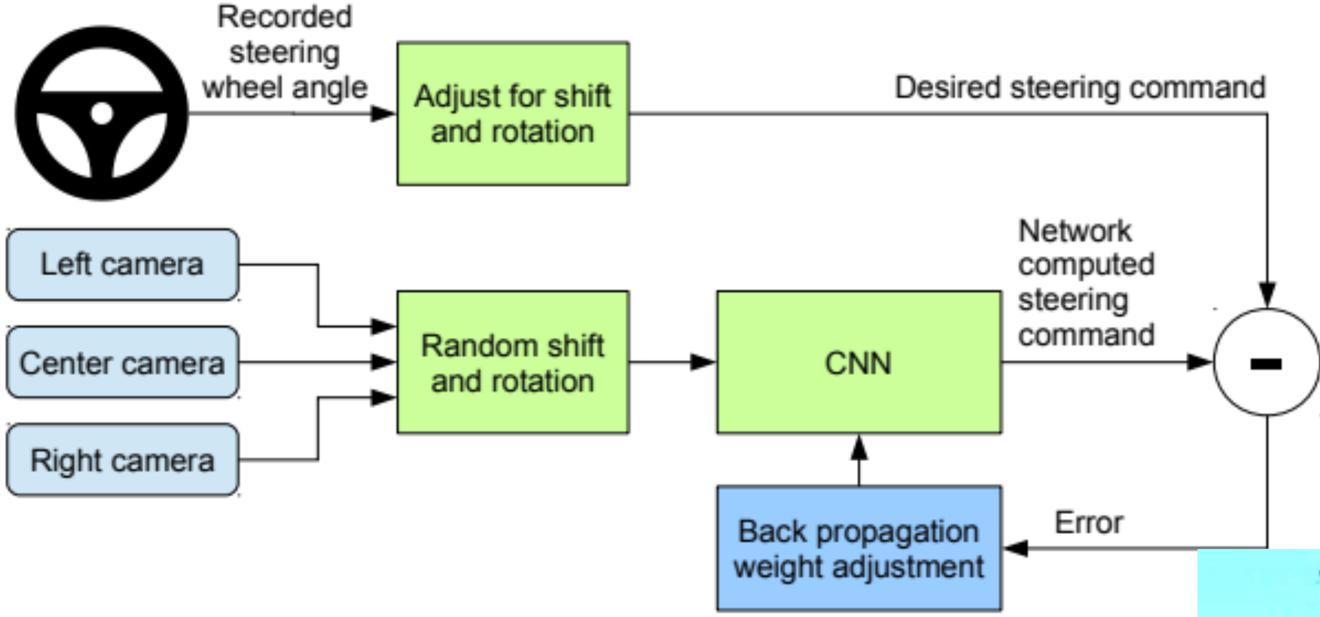
2. Human imitation learning
   - Copy the intent of the expert
   - May take very different actions from the expert

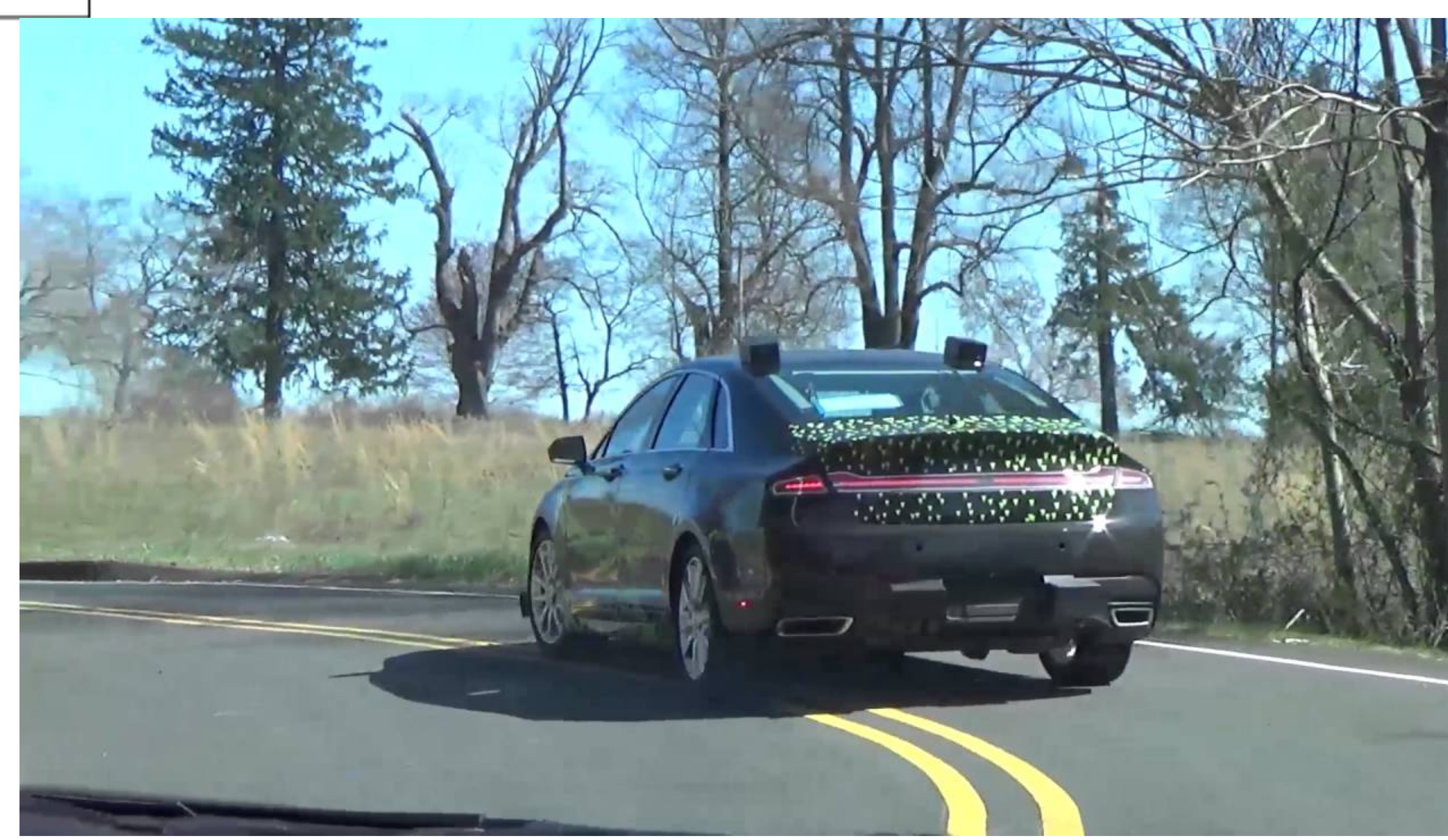
Inverse RL!

(Slide Credit: Sergey Levine)

Direct Imitation Learning

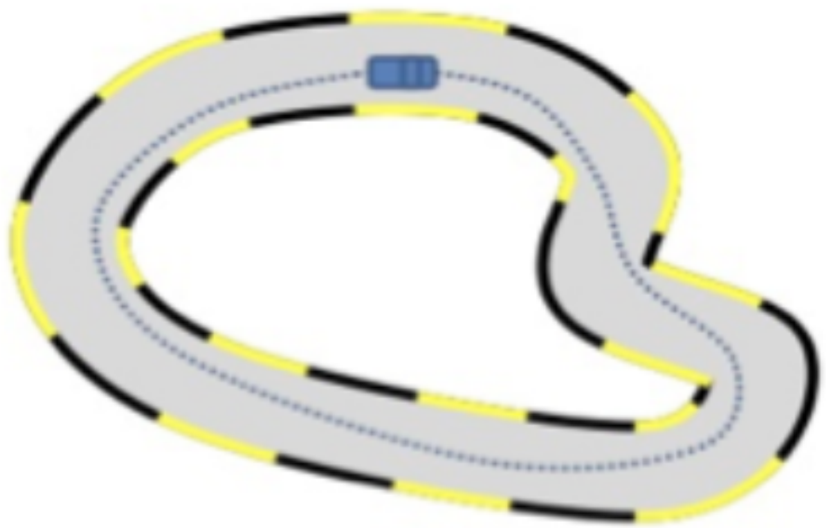
- Example: Self-driving cars



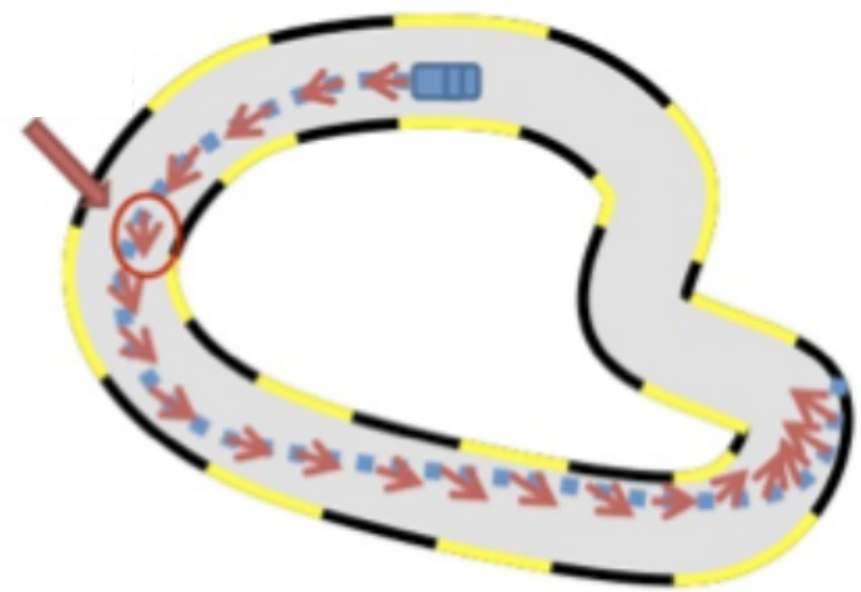


Mariusz Bojarski et al., End to End Learning for Self-Driving Cars, 2016

Direct Imitation Learning: Out-of-Distribution Issue



Expert Trajectories
(Training Distributions)



Direct Imitation Learning
(Testing Distributions)

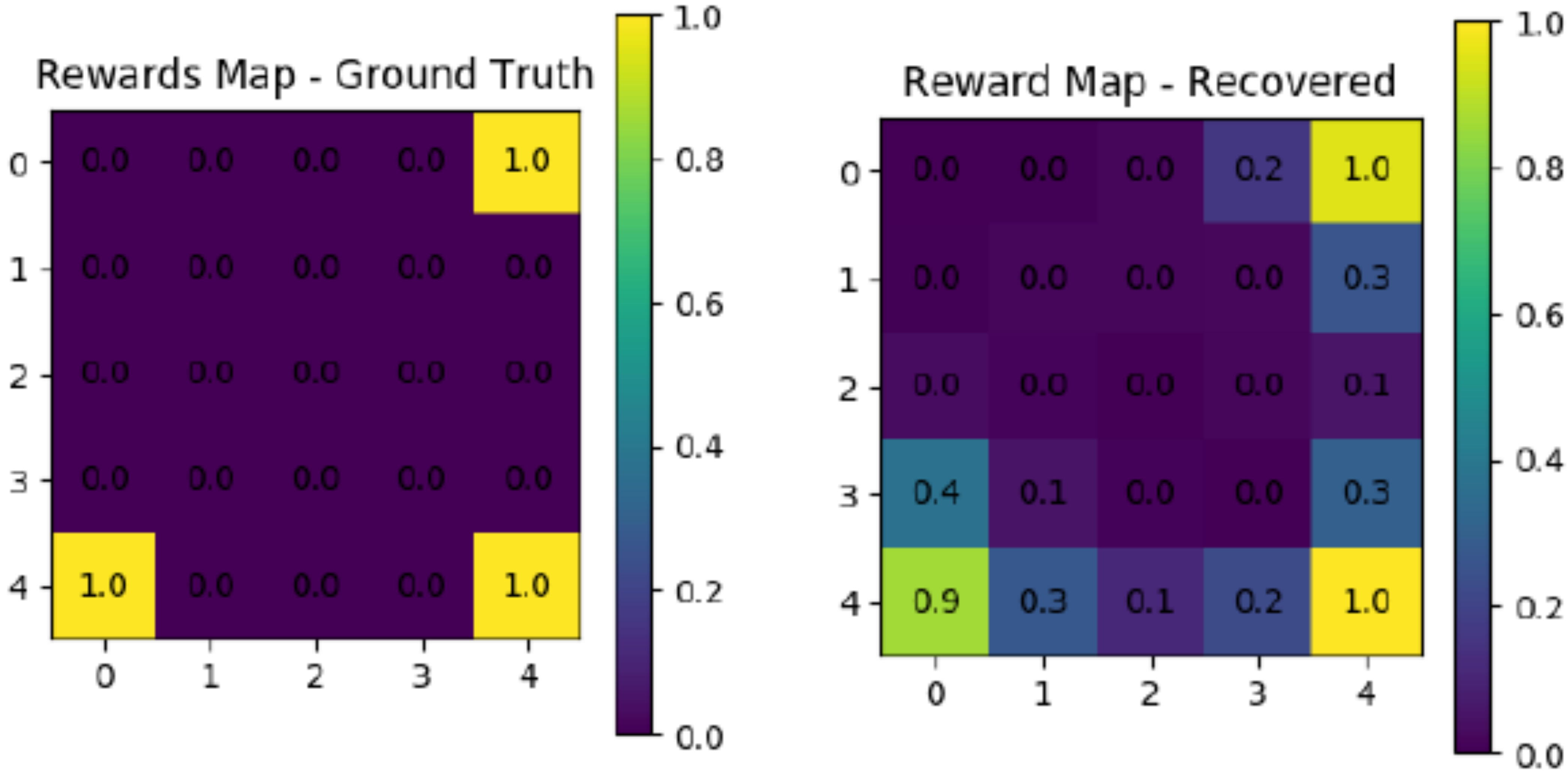
Makes mistakes, enter new states

Cannot recover from new states

(Image Source: Stephane Ross)

Inverse RL

- Example: Reward recovery in Gridworld

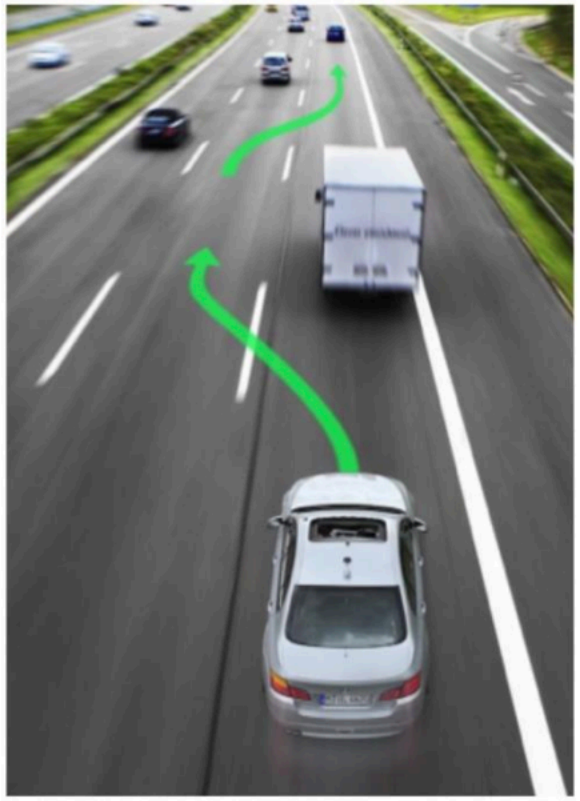


Inverse RL (Informal)

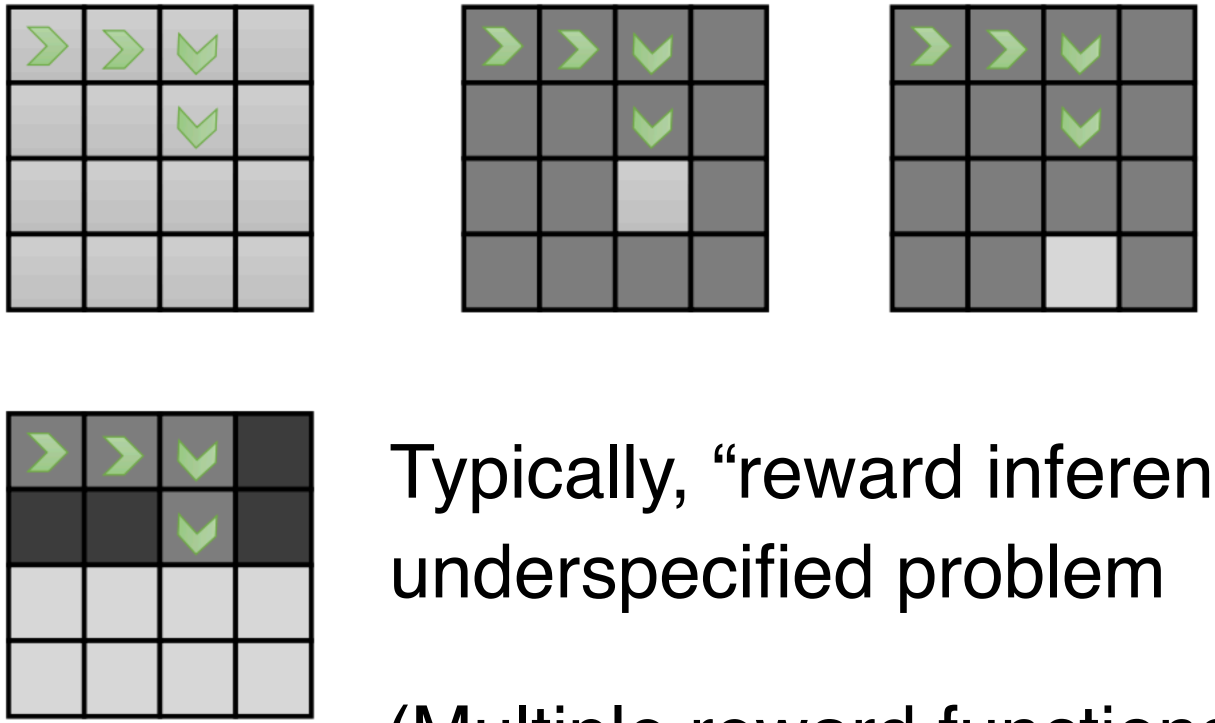
- Suppose the agent is in an MDP (S, A, P, γ)
- Suppose we are given expert demonstrations (under some unknown policy π_e)
- Goal: Infer the reward function R behind the expert actions solely from expert demonstrations (and thereafter learn a good policy)

First Attempt: Infer Rewards from Demonstrations

- Example: Human driving



What's reward $R(s, a)$?



Typically, "reward inference" is an underspecified problem

(Multiple reward functions can explain the same behavior)

Reward identifiability issue!

Rethinking Imitation?

Recall: *Occupancy measure* (or *discounted state visitation*)

$$d^{\pi}_{\mu}(s) := (1-\gamma)\mathbb{E}_{s_0\sim\mu}\Big[\sum_{t=0}^{\infty}\gamma^t P(s_t = s \,|\, s_0, \pi)\Big]$$

$$d^{\pi}_{\mu}(s,a) := (1-\gamma)\mathbb{E}_{s_0\sim\mu}\Big[\sum_{t=0}^{\infty}\gamma^t P(s_t = s, a_t = a \,|\, s_0, \pi)\Big]$$

(Q1) If $\pi_\theta = \pi_e$, then do we have $d^{\pi_\theta}_{\mu}(s,a) = d^{\pi_e}_{\mu}(s,a)$?

(Q2) If $d^{\pi_\theta}_{\mu}(s,a) = d^{\pi_e}_{\mu}(s,a)$, then do we have $V^{\pi_\theta}(\mu) = V^{\pi_e}(\mu)$?

(Q3) If $d^{\pi_\theta}_{\mu}(s,a) = d^{\pi_e}_{\mu}(s,a)$, then do we have $\pi_\theta = \pi_e$?

About (Q3): A Bijection Theorem

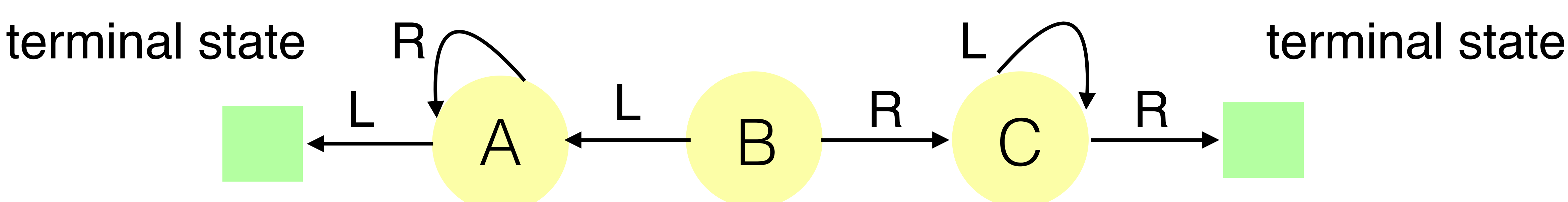
Theorem [Syed et al., 2008]: For any valid discounted state visitation distribution $d^\pi(s,a)$, define a policy $\pi'(a \mid s) := d^\pi(s,a) / \sum_{a' \in A} d^\pi(s,a')$.

Then, we have $d^{\pi'}(s,a) = d^\pi(s,a)$, for all (s,a).

(In other words, the mapping from $d^\pi \to \pi$ is a bijection)

- However, the above Bijection Theorem does NOT implies that (Q3).
- Regarding (Q3), Bijection Theorem only implies that $\pi_\theta(\cdot \mid s) = \pi_e(\cdot \mid s)$ at those states with $d^{\pi_e}(s) > 0$

Example:



Syed, Bowling, and Schapire, "Apprenticeship Learning Using Linear Programming," ICML 2008

Inverse RL: Occupancy Measure Matching

Brian Ziebart et al., Maximum entropy inverse reinforcement learning, AAAI 2008

Jonathan Ho and S. Ermon, Generative adversarial imitation learning, NIPS 2016

Xiao et al., Wasserstein Adversarial Imitation Learning, NeurIPS 2019

Garg et al., IQ-Learn: Inverse soft-Q Learning for Imitation, NeurIPS 2021

Occupancy Measure Matching: Formulation

Recall: *Occupancy measure* (or *discounted state visitation*)

$$d_\mu^\pi(s) := (1-\gamma)\mathbb{E}_{s_0\sim\mu}\Big[\sum_{t=0}^{\infty}\gamma^t P(s_t = s \mid s_0, \pi)\Big]$$

$$d_\mu^\pi(s,a) := (1-\gamma)\mathbb{E}_{s_0\sim\mu}\Big[\sum_{t=0}^{\infty}\gamma^t P(s_t = s, a_t = a \mid s_0, \pi)\Big]$$

Claim: $V^\pi(\mu) = \sum_{(s,a)} d_\mu^\pi(s,a) R(s,a)$ (Why?)

Occupancy measure matching:

Find a policy π such that $d_\mu^\pi(s,a) = d^{\pi_e}(s,a), \quad \forall (s,a)$

Occupancy measure matching implies $V^\pi(\mu) = V^{\pi_e}(\mu)$

- Question: Is $d_\mu^\pi(s,a)$ easy to parameterize?

(Direct) Occupancy Measure Matching (OMM)

$$\min_{\pi \in \Pi} \quad L(\pi) := D(d_\mu^\pi, d_\mu^{\pi_e})$$

($D(\cdot, \cdot)$ is some distance)

(d_μ^π could be hard to express!)

Dual of each other!

$$\max_{R \in \mathscr{R}} \min_{\pi \in \Pi} \underbrace{\left[\left(E_{d_\mu^{\pi_e}}[R(s,a)] - E_{d_\mu^\pi}[R(s,a)] \right) \right]}_{:=L(\pi,R)}$$

OR

$$\min_{\pi \in \Pi} \max_{R \in \mathscr{R}} \underbrace{\left[\left(E_{d_\mu^{\pi_e}}[R(s,a)] - E_{d_\mu^\pi}[R(s,a)] \right) \right]}_{:=L(\pi,R)}$$

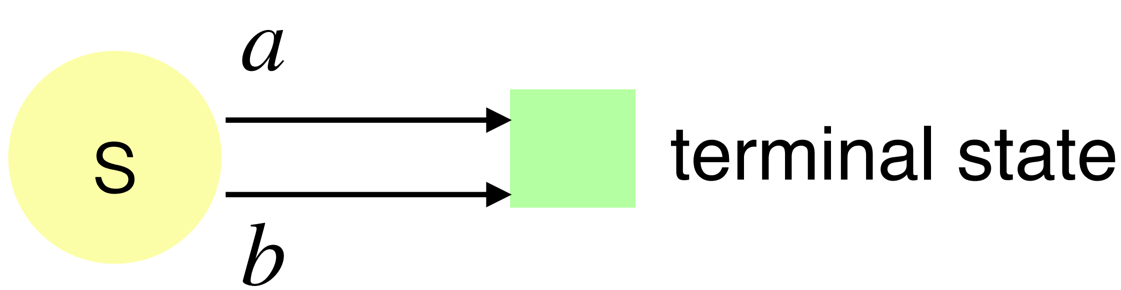
(Easier for training!)

Apprenticeship Learning (APPLE)

A Motivating Example: Connecting OMM & APPLE

$$\min_{\pi \in \Pi} L(\pi) := D(d_\mu^\pi, d_\mu^{\pi_e}) \longleftrightarrow \min_{\pi \in \Pi} \max_{R \in \mathscr{R}} \underbrace{\left[\left(E_{d_\mu^{\pi_e}}[R(s,a)] - E_{d_\mu^\pi}[R(s,a)]\right)\right]}_{:=L(\pi,R)}$$

Consider a simple 1-state, 2-action MDP



Suppose $\mathscr{R} = \mathbb{R}^2$

$$\pi_e(a|s) = \pi_e(b|s) = 0.5$$

Let's write down $R \in \mathscr{R}$ that maximizes $L(\pi, R)$ under a fixed π

For (s, a) with $d_\mu^\pi(s,a) > d_\mu^{\pi_e}(s,a)$:

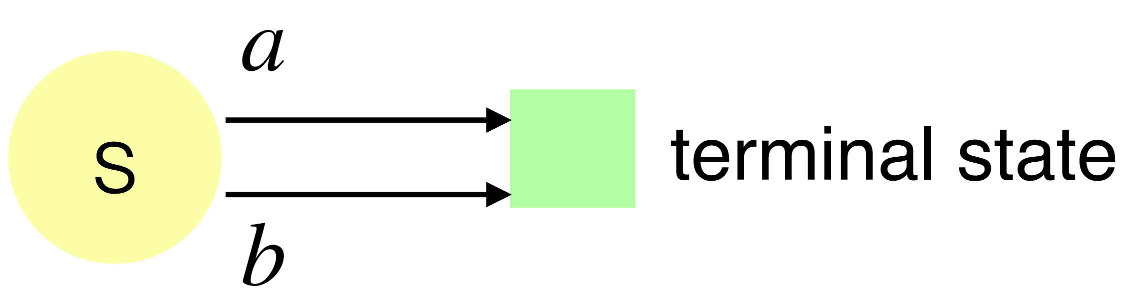
For (s, a) with $d_\mu^\pi(s,a) < d_\mu^{\pi_e}(s,a)$:

For (s, a) with $d_\mu^\pi(s,a) = d_\mu^{\pi_e}(s,a)$:

A Motivating Example: Connecting OMM & APPLE

$$\min_{\pi \in \Pi} L(\pi) := D(d_\mu^\pi, d_\mu^{\pi_e}) \longleftrightarrow \min_{\pi \in \Pi} \max_{R \in \mathscr{R}} \underbrace{\left[\left(E_{d_\mu^{\pi_e}}[R(s,a)] - E_{d_\mu^\pi}[R(s,a)] \right) \right]}_{:= L(\pi, R)}$$

Consider a simple 1-state, 2-action MDP



Suppose $\mathscr{R} = \mathbb{R}^2$

$$\pi_e(a|s) = \pi_e(b|s) = 0.5$$

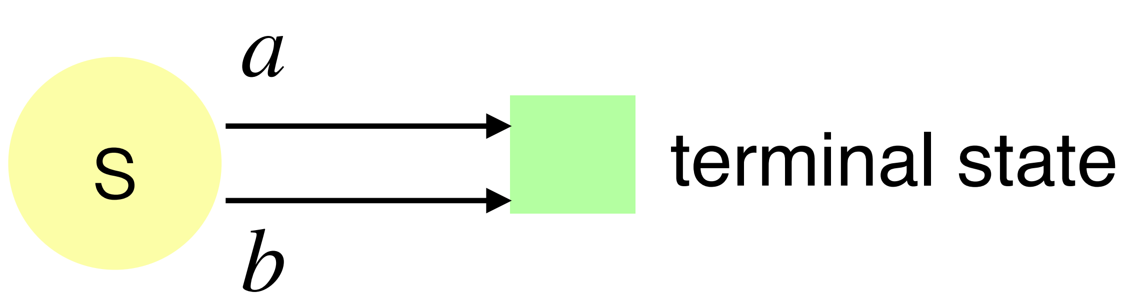
Nice Property: Under $\mathscr{R} = \mathbb{R}^2$, the corresponding metric D is

$$D(d_\mu^\pi, d_\mu^{\pi_e}) = \begin{cases} 0, & \text{if } d_\mu^\pi(s,a) = d_\mu^{\pi_e}(s,a), \forall (s,a) \\ \infty, & \text{otherwise} \end{cases}$$

A Motivating Example: Connecting OMM & APPLE (Cont.)

$$\min_{\pi \in \Pi} L(\pi) := D(d_\mu^\pi, d_\mu^{\pi_e}) \longleftrightarrow \min_{\pi \in \Pi} \max_{R \in \mathscr{R}} \underbrace{\left[\left(E_{d_\mu^{\pi_e}}[R(s,a)] - E_{d_\mu^\pi}[R(s,a)] \right) \right]}_{:=L(\pi,R)}$$

Consider a simple 1-state, 2-action MDP



Suppose $\mathscr{R} = \left\{ R \in \mathbb{R}^2 \,\middle|\, \|R\|_\infty \leq 1 \right\}$

$$\pi_e(a|s) = \pi_e(b|s) = 0.5$$

Let's write down $R \in \mathscr{R}$ that maximizes $L(\pi, R)$ under a fixed π

For (s, a) with $d_\mu^\pi(s,a) > d_\mu^{\pi_e}(s,a)$:

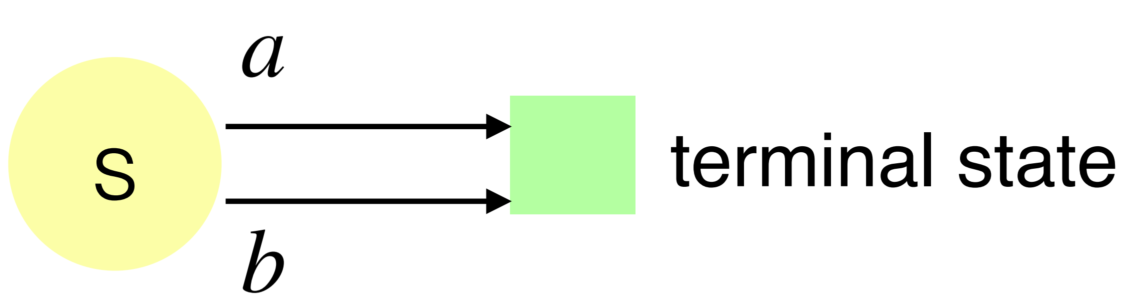
For (s, a) with $d_\mu^\pi(s,a) < d_\mu^{\pi_e}(s,a)$:

For (s, a) with $d_\mu^\pi(s,a) = d_\mu^{\pi_e}(s,a)$:

A Motivating Example: Connecting OMM & APPLE (Cont.)

$$\min_{\pi\in\Pi} \; L(\pi) := D(d_\mu^\pi, d_\mu^{\pi_e}) \quad \longleftrightarrow \quad \min_{\pi\in\Pi}\max_{R\in\mathscr{R}} \underbrace{\left[\left(E_{d_\mu^{\pi_e}}[R(s,a)] - E_{d_\mu^{\pi}}[R(s,a)]\right)\right]}_{:=L(\pi,R)}$$

Consider a simple 1-state, 2-action MDP



Suppose $\mathscr{R} = \left\{ R \in \mathbb{R}^2 \,\middle|\, \|R\|_\infty \leq 1 \right\}$

$$\pi_e(a|s) = \pi_e(b|s) = 0.5$$

Nice Property: Under $\mathscr{R} = \left\{ R \in \mathbb{R}^2 \,\middle|\, \|R\|_\infty \leq 1 \right\}$, the metric D is

$$D(d_\mu^\pi, d_\mu^{\pi_e}) = \sum_{(s,a)} \left| d_\mu^\pi(s,a) - d_\mu^{\pi_e}(s,a) \right|$$

(usually called *"total variation distance"*)

How to choose $\mathscr{R}$ to get some widely-used D?

Example #1: Wasserstein Metric and APPLE

$$\min_{\pi\in\Pi} L(\pi) := W(d_\mu^\pi, d_\mu^{\pi_e})$$

(Wasserstein)

$$\longleftrightarrow$$

$$\min_{\pi\in\Pi} \max_{R\in\mathscr{R}} \underbrace{\left[\left(E_{d_\mu^{\pi_e}}[R(s,a)] - E_{d_\mu^\pi}[R(s,a)]\right)\right]}_{:=L(\pi,R)}$$

where $\mathscr{R} = \left\{R \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}} \middle| \text{Lip}(R) \leq 1\right\}$

This is also known as the *Kantorovich-Rubenstein duality*

Xiao et al., Wasserstein Adversarial Imitation Learning, NeurIPS 2019

Wasserstein Metric

Metric for random vectors

- $U : \Omega \to \mathbb{R}^d$: a random vector from the sample space Ω to $\mathbb{R}^d$
- For $1 \leq p < \infty$: $||U||_p := \left(\mathbb{E}\left[||U(\omega)||_p^p\right]\right)^{\frac{1}{p}}$

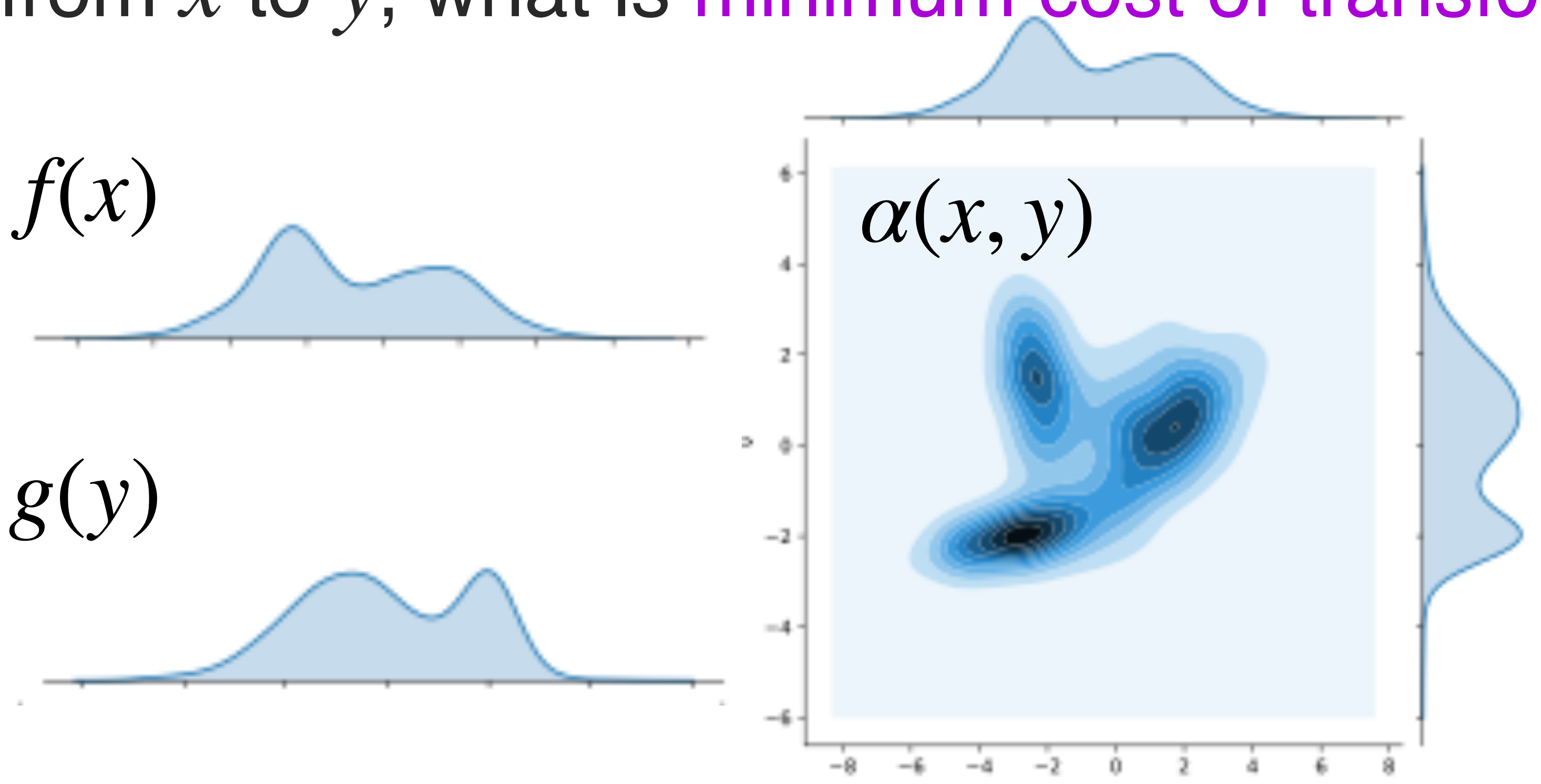
- **Wasserstein Metric**: For two CDFs F, G over the reals, the Wasserstein metric is defined as

$$d_p(F, G) := \inf_{(U,V):U\sim F, V\sim G} ||U - V||_p$$

- Infimum is taken over all joint distributions of random variables (U, V), whose marginal distributions are F, G

Intuition Behind Wasserstein Metric

- Also known as: <u>optimal transport problem</u> or <u>earth mover's distance</u>
- Given two density $f(x)$, $g(x)$ and a cost function $c(x, y)$ of moving mass from x to y, what is minimum cost of transforming from $f(x)$ to $g(y)$?



Minimum cost

$$C^* := \inf_{\alpha} \int c(x, y)\alpha(x, y)dxdy$$

$\alpha(x, y)$:amount of mass to move from x to y

$\alpha(x, y)$describes a feasible transport plan if

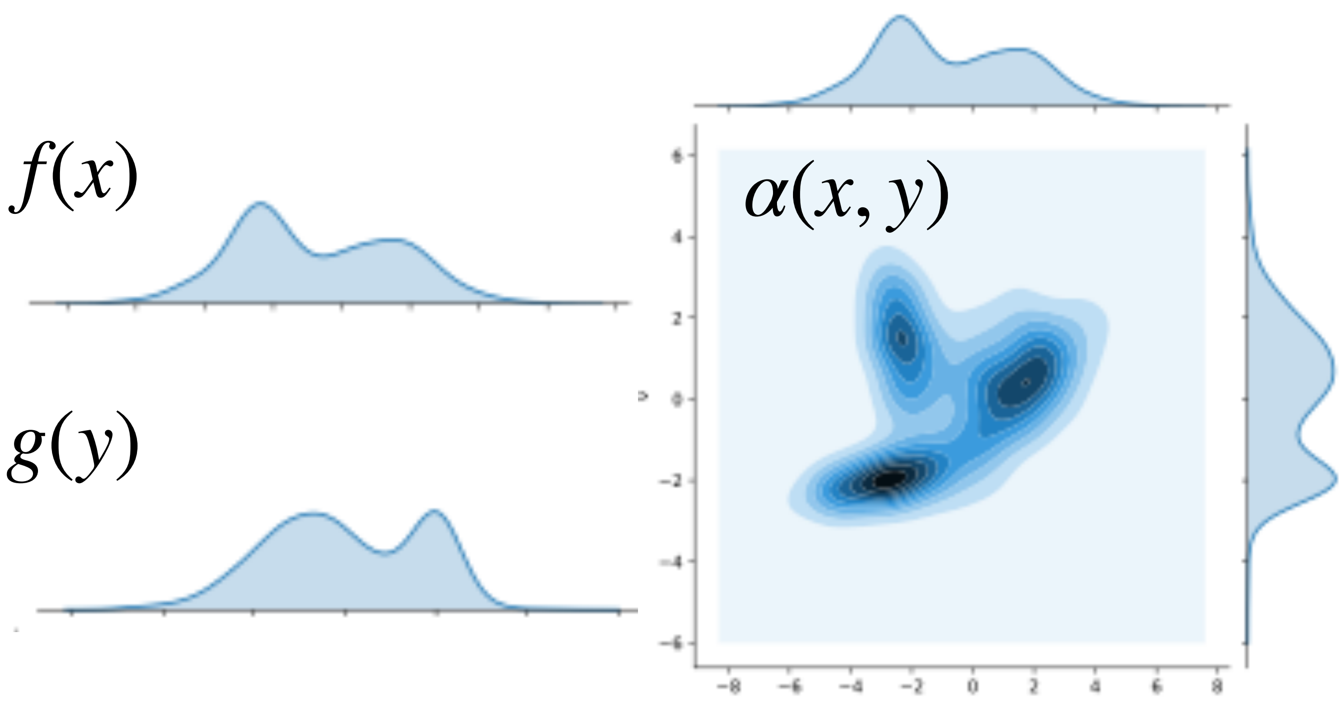
$$\int \alpha(x, y)dy = f(x), \quad \int \alpha(x, y)dx = g(y)$$

(Figure source: Wikipedia)

Summary: Optimal Transport & Wasserstein Metric

Wasserstein $$d_p(F,G) := \inf_{(U,V):U\sim F, V\sim G} ||U-V||_p$$

Optimal Transport (OT)



$c(x,y)$ = cost function of moving one unit of mass from x to y

- OT can be written as an optimization problem:

$$\min_{\alpha} \sum_{x,y} c(x,y)\alpha(x,y)$$

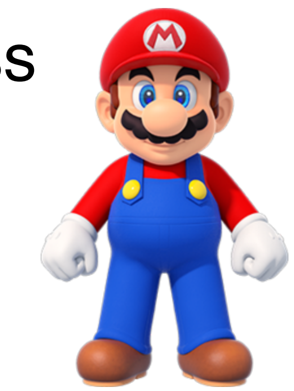
subject to (1) $\sum_{y} \alpha(x,y) = f(x), \forall x$ (2) $\sum_{x} \alpha(x,y) = g(y), \forall y$

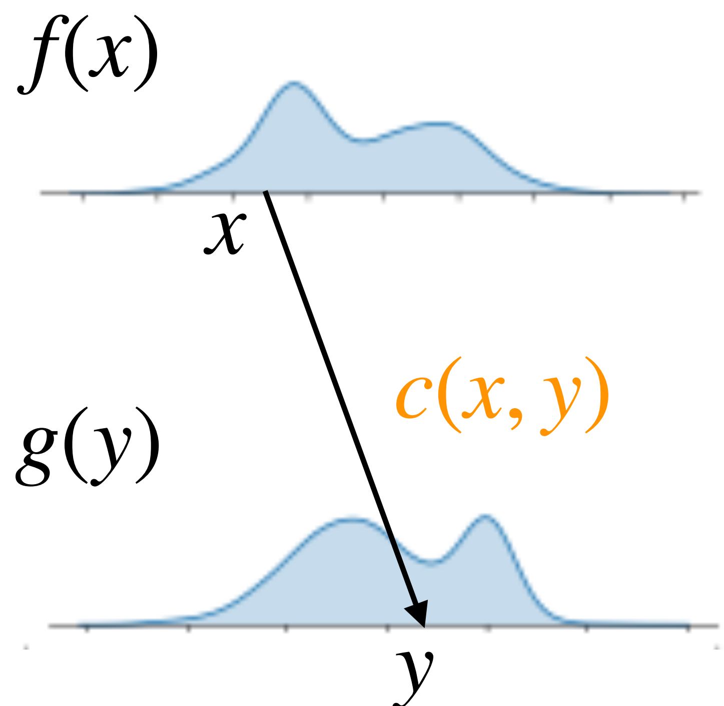
(3) $\alpha(x,y) \geq 0, \forall x, y$

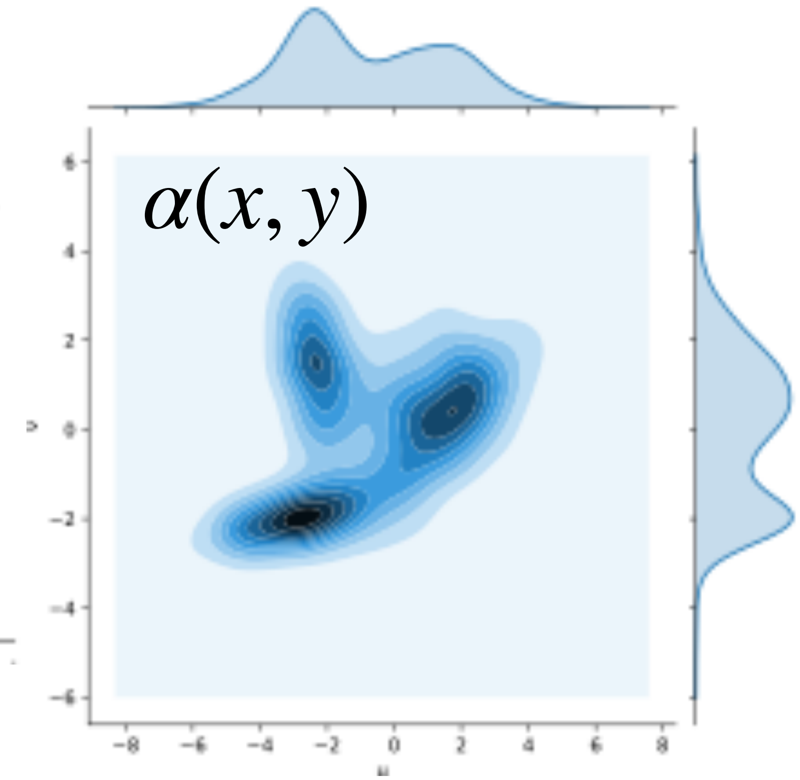
Duality of Optimal Transport: Economic Interpretation

Primal Form of OT (Mario moving the earth by himself)

$c(x, y)$ = Mario's cost function (for moving one unit of mass from x to y)

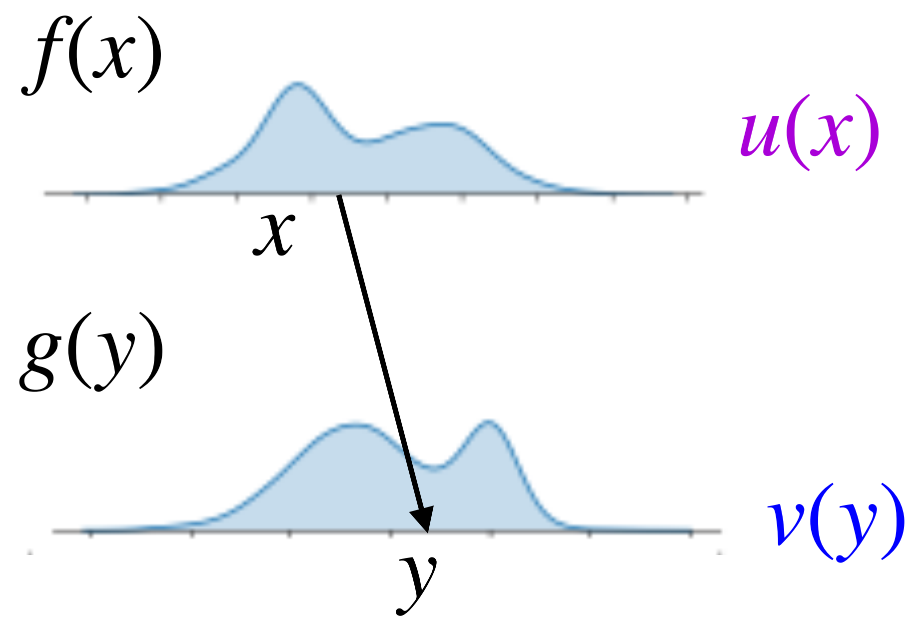






Dual Form of OT (Luigi offers to help Mario)

Mario needs to pay Luigi $u(x)$ and $v(y)$ (for moving one unit of mass from x to y)



Question: Under what condition would Mario ask for Luigi's help?

Duality of Optimal Transport (Formally)

- Primal Form of Optimal Transport

$$\min_{\alpha} \sum_{x,y} c(x,y)\alpha(x,y)$$

subject to (1) $\sum_{y} \alpha(x,y) = f(x), \forall x$ (2) $\sum_{x} \alpha(x,y) = g(y), \forall y$

(3) $\alpha(x,y) \geq 0, \forall x, y$

- Dual Form of Optimal Transport

$$\max_{u,v} \mathbb{E}_{x\sim f(x)}[u(x)] + \mathbb{E}_{y\sim g(y)}[v(y)]$$

subject to $u(x) + v(y) \leq c(x,y), \forall x, y$

The dual form looks exactly like APPLE!

- Both forms lead to the same optimal values (called "strong duality")

Example #2: Generative Adversarial Imitation Learning (GAIL)

- Recall: Dual Form of Optimal Transport

$$\max_{u,v} \mathbb{E}_{x\sim f(x)}[u(x)] + \mathbb{E}_{y\sim g(y)}[v(y)]$$

subject to $u(x) + v(y) \leq c(x, y), \forall x, y$

$D_\phi(s, a)$: A binary classifier that predicts the probability of the event that "the observed (s, a) is drawn from π"

Let's choose the following:

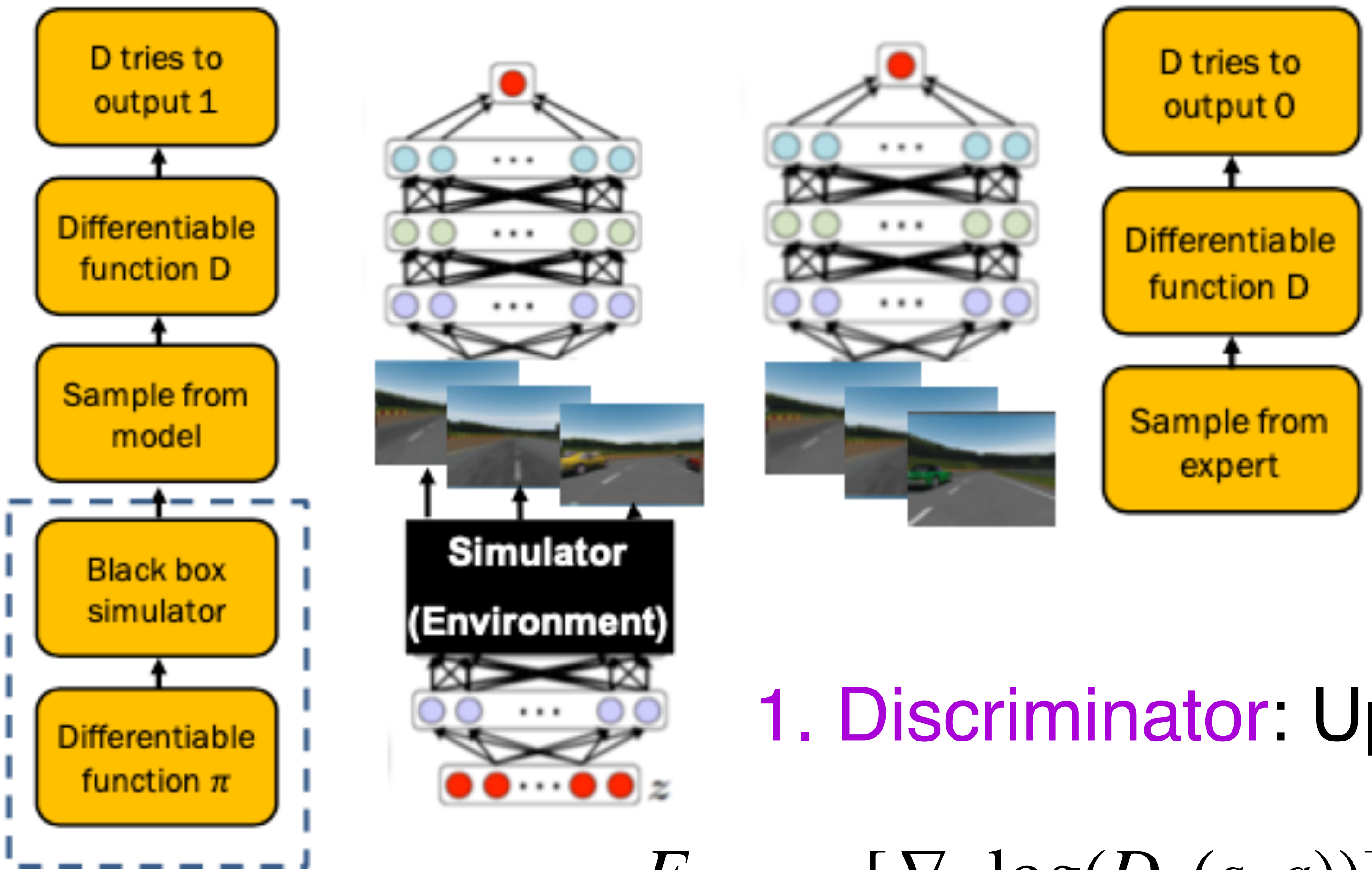
(1) $f(x) \equiv d_\mu^\pi(s, a)$

(2) $g(y) \equiv d_\mu^{\pi_e}(s, a)$

(3) $u(x) \equiv \log(D_\phi(s, a))$

(4) $v(y) \equiv \log(1 - D_\phi(s, a))$

GAIL: Discriminator and Generator



1. Discriminator: Update ϕ by

$$E_{(s,a)\sim d_{\mu}^{\pi}}[\nabla_{\phi}\log(D_{\phi}(s,a))] + E_{(s,a)\sim d_{\mu}^{\pi_e}}[\nabla_{\phi}\log(1 - D_{\phi}(s,a))]$$

2. Generator: Use any RL algorithm with reward function $\log(D_{\phi}(s,a))$

Jonathan Ho and S. Ermon, Generative adversarial imitation learning, NIPS 2016

A Comparison Between Wasserstein AIL and GAIL

Expert state-action pairs (projected onto a 2D board via PCA)

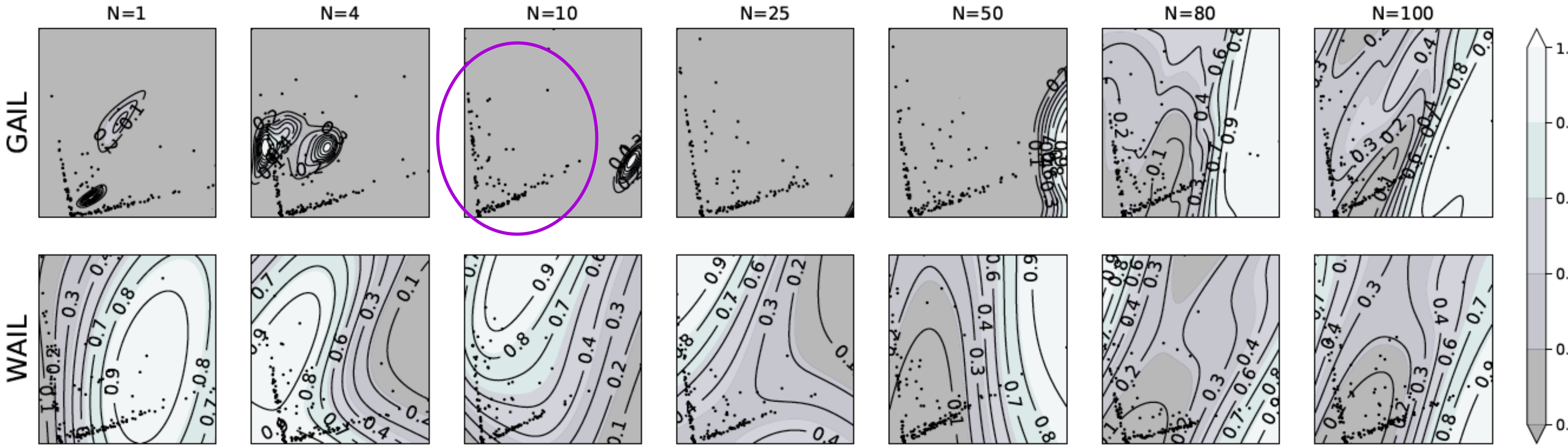


Figure 2: Reward surfaces of WAIL and GAIL on *Humanoid* with respect to different expert data sizes.

Xiao et al., Wasserstein Adversarial Imitation Learning, NeurIPS 2019

Summary: Occupancy Measure Matching via Apprenticeship Learning (With Regularization)

$$\min_{\pi \in \Pi} \max_{R \in \mathscr{R}} \underbrace{\left[\left(E_{(s,a)\sim d_\mu^{\pi_e}}[R(s,a)] - E_{(s,a)\sim d_\mu^{\pi}}[R(s,a)] \right) - H(\pi) + \psi(R) \right]}_{:=L(\pi,R)}$$

where $H(\pi) := E\Big[\sum_t -\gamma^t \log \pi_t(a_t|s_t)\Big]$ is the discounted causal entropy

$\psi(R)$ is a regularizer for the reward function

Key Idea: By choosing different "reward function classes $\mathscr{R}$", we obtain various OMM approaches!