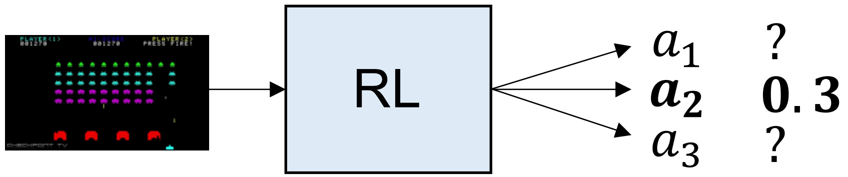
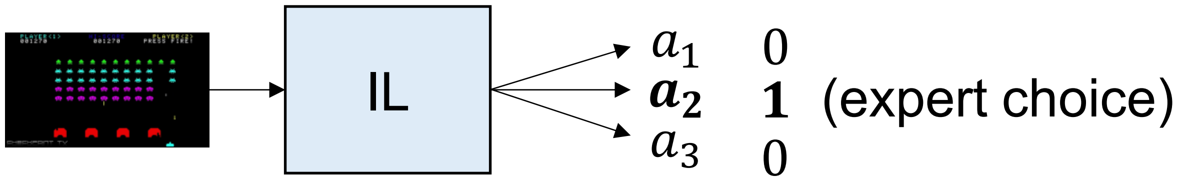
Imitation Learning

Chen-Yu Wei

Imitation Learning ∈ Supervised Learning





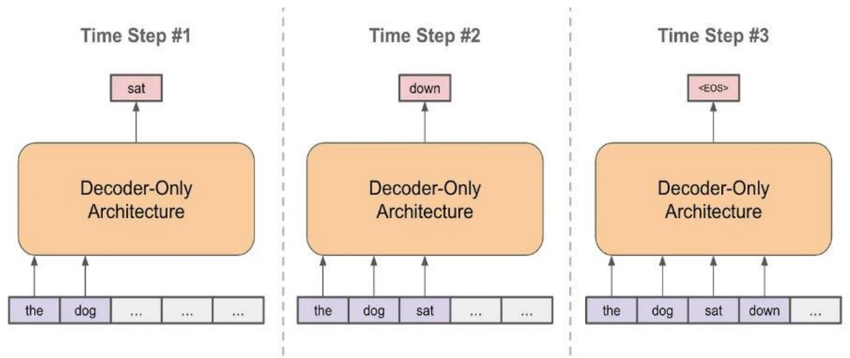
Offline IL: learn from static data generated by the expert $\{(s_1, a_1^\star, s_2, a_2^\star, \dots, s_H, a_H^\star)\}$

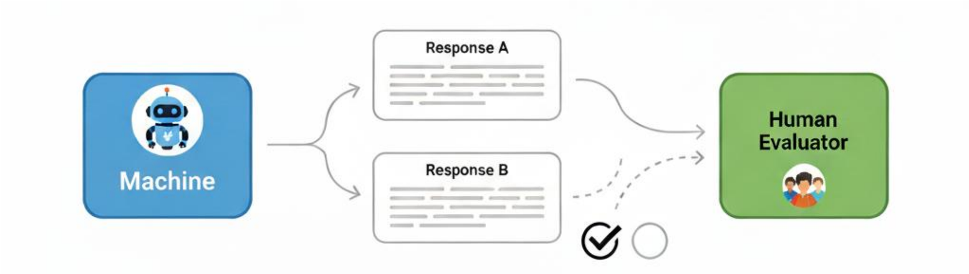
Online IL: may interact with MDP and query the expert $\{(s_1, a_1, a_1^\star, s_2, a_2, a_2^\star, \dots, s_H, a_H, a_H^\star)\}$

Goal: output a policy $\hat{\pi}$ such that $V^{\pi^\star}(\rho) - V^{\hat{\pi}}(\rho)$ is small

Examples

- Language models





RLHF

- Robotics

Types

- Direct Imitation: directly learn policy to imitate the expert
  - Behavior cloning
  - DAgger
  - Direct preference optimization (preference feedback)
- Inverse RL: learn reward function from expert, and perform RL on it
  - Adversarial IRL (paper)
  - MaxEnt IRL (paper)
  - RLHF (preference feedback)

Direct Imitation

Behavior Cloning: Reduction to Classification

Relate $V^{\pi^\star}(\rho) - V^{\widehat{\pi}}(\rho)$ to $\mathbb{E}^{\pi^\star}\left[\frac{1}{H}\sum_{h=1}^{H}\mathbb{I}\{\widehat{\pi}_h(s_h) \neq \pi_h^\star(s_h)\}\right]$

$\text{Range}(Q) = \max_{s,a,b} |Q(s,a) - Q(s,b)|$

$$V^{\pi^\star}(\rho) - V^{\widehat{\pi}}(\rho) = \sum_{h=1}^{H}\sum_{s} d_h^{\pi^\star}(s)\left(Q_h^{\widehat{\pi}}(s,\pi_h^\star(s)) - Q_h^{\widehat{\pi}}(s,\widehat{\pi}_h(s))\right)$$

$$\leq \text{Range}\left(Q^{\widehat{\pi}}\right)\sum_{h=1}^{H}\sum_{s} d_h^{\pi^\star}(s)\,\mathbb{I}\{\pi_h^\star(s) \neq \widehat{\pi}_h(s)\}$$

$$= H\text{Range}\left(Q^{\widehat{\pi}}\right) \times \frac{1}{H}\sum_{h=1}^{H}\sum_{s} d_h^{\pi^\star}(s)\,\mathbb{I}\{\pi_h^\star(s) \neq \widehat{\pi}_h(s)\}$$

Behavior Cloning with Logistic Loss

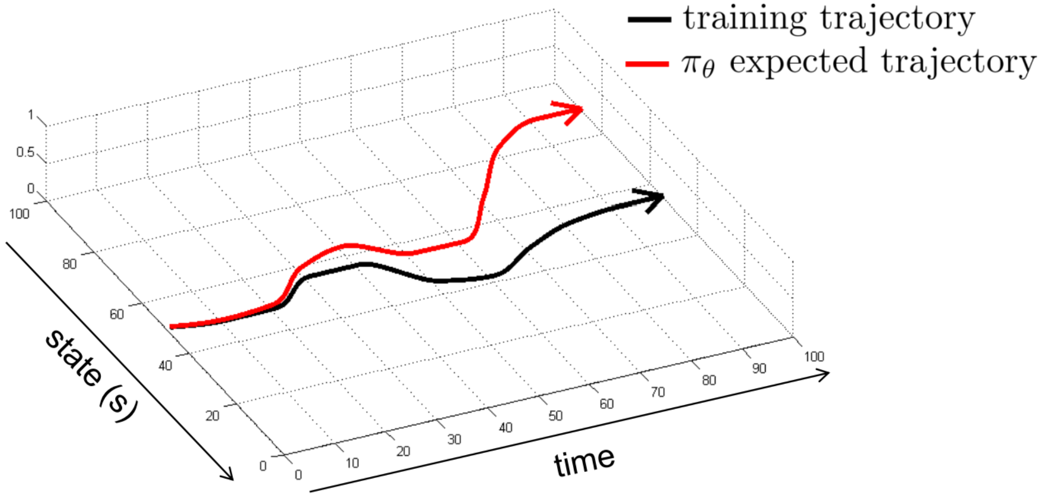
Behavior Cloning (Offline IL)

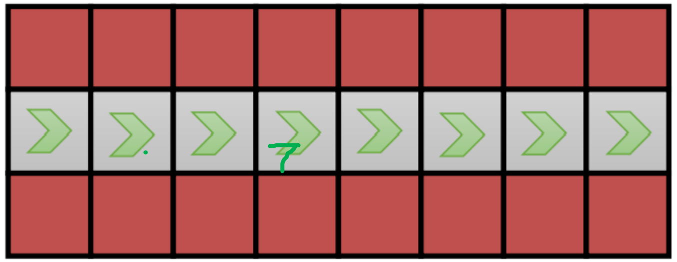
Run expert policy $\pi^\star$ and obtain $(s_1, a_1), \ldots, (s_N, a_N)$

Obtain policy π_θ by finding

$$\theta = \underset{\theta}{\operatorname{argmin}} \ \frac{1}{N} \sum_{i=1}^{N} \log \frac{1}{\pi_\theta(a_i|s_i)}$$

Behavior Cloning: Reduction to Classification





total loss $= \sum_h \mathbb{1}(\text{off-track at step } h)$

$\mathbb{E}$

$= \epsilon \times \mathbb{1}\{\text{go off-track at 1st step}\} \times H \simeq \epsilon H$

$+ (1-\epsilon)\epsilon\, \mathbb{1}\{\text{—— 2nd step}\} \times (H-1) \simeq \epsilon H$

$+ (1-\epsilon)^2 \epsilon$ —— $\times (H-2)$

$\vdots$

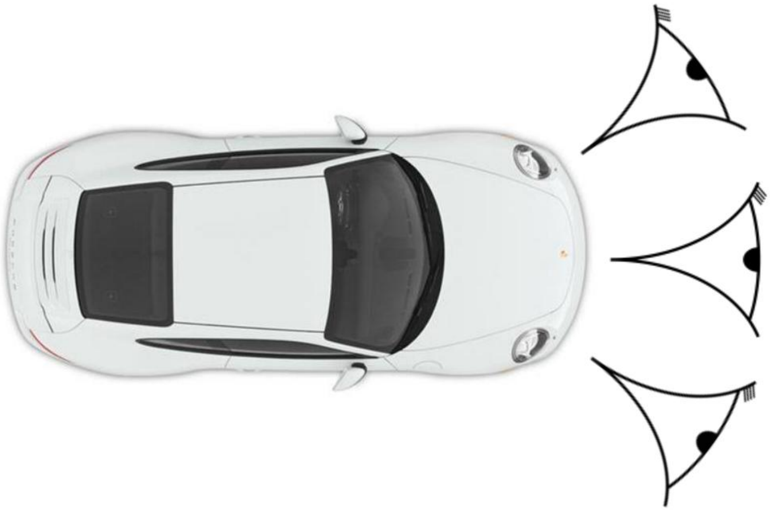
ϵH^2

Issue: **distribution shift**

$$\frac{1}{H} \sum_h \mathbb{1}(\hat{\pi}_h(s_h) \neq \pi(s_h)) \leq \epsilon$$

Solution

- Data augmentation



Bojarsky et al. End to End Learning for Self-Driving Cars. 2016

Solution: Interact with Expert (Online IL)

Dataset Aggregation (DAgger)

For $k = 1,\ 2, \ldots$

Train $\pi_\theta(a|s)$ with dataset $\mathcal{B}$

Run π_θ to collect states $s_1, s_2, \ldots, s_N$

Ask the expert to label actions, giving $(s_1, a_1), \ldots, (s_N, a_N)$

Add $(s_1, a_1), \ldots, (s_N, a_N)$ to $\mathcal{B}$

DAgger with Logistic Loss

DAgger (Online IL)

Run expert policy $\pi^\star$ and obtain (s, a)-pairs and push them to $\mathcal{B}$

For $k = 1,\ 2, \dots$

For $m = 1,\ 2, \dots, M$:

Sample a batch b from $\mathcal{B}$

$$\theta \leftarrow \theta - \alpha \frac{1}{|b|} \sum_{(s,a)\in b} \log \frac{1}{\pi_\theta(a|s)}$$

Use π_θ to generate states $s_1, s_2, \dots, s_N$

Ask expert to provide labels $(s_1, a_1), (s_2, a_2), \dots, (s_N, a_N)$

Push them to $\mathcal{B}$

Analysis

Relate $V^{\pi^\star}(\rho) - V^{\widehat{\pi}}(\rho)$ to $\mathbb{E}^{\widehat{\pi}}\left[\frac{1}{H}\sum_{h=1}^{H} \mathbb{I}\{\widehat{\pi}_h(s_h) \neq \pi_h^\star(s_h)\}\right]$

$$V^{\pi^\star}(\rho) - V^{\widehat{\pi}}(\rho) = \sum_{h=1}^{H}\sum_{s} d_h^{\widehat{\pi}}(s)\left(Q_h^{\pi^\star}(s, \pi_h^\star(s)) - Q_h^{\pi^\star}(s, \widehat{\pi}_h(s))\right)$$

$$\leq \mathrm{Range}\left(Q^{\pi^\star}\right)\sum_{h=1}^{H}\sum_{s} d_h^{\widehat{\pi}}(s)\,\mathbb{I}\{\pi_h^\star(s) \neq \widehat{\pi}_h(s)\}$$

$$= H\mathrm{Range}\left(Q^{\pi^\star}\right) \times \frac{1}{H}\sum_{h=1}^{H}\sum_{s} d_h^{\widehat{\pi}}(s)\,\mathbb{I}\{\pi_h^\star(s) \neq \widehat{\pi}_h(s)\}$$

Imitation with Preference Feedback

Direct Preference Optimization (DPO) (link)

Given context x and two potential actions, the expert chooses the **preferred** one (the preferred one is denoted as y_w and the other is y_l).

$$\mathcal{L}_{\mathrm{DPO}}(\pi_\theta; \pi_{\mathrm{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\mathrm{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\mathrm{ref}}(y_l \mid x)} \right) \right]$$

π_{ref} is a reference policy that π_θ is initialized as.

The previously discussed frameworks of online / offline IL can be directly applied here.

Inverse RL

Inverse Reinforcement Learning (IRL)

Find a reward / value function that explains the behavior of the expert.

- **Behavior assumption:** Assume some connections between expert policy and reward/value. E.g.,

  $\pi^\star(a|s) \propto \exp(Q^\star(s,a))$

  $P^{\pi^\star}(\tau) \propto \exp\left(\sum_h R^\star(s_h, a_h)\right)$

- **Maximum likelihood:** Given expert trajectories, find ϕ that maximizes likelihood
- **Policy training:** Find a policy that approximately maximizes the expected reward / value under ϕ. E.g.,

  $\pi(a|s) \propto \exp\left(Q_\phi(s,a)\right)$

  $\theta = \underset{\theta}{\operatorname{argmax}} \ \mathbb{E}_{s\sim D}\left[\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[R_\phi(s,a)\right] - \beta \, \mathrm{KL}(\pi_\theta(\cdot\,|s), \pi_{\mathrm{ref}}(\cdot\,|s))\right]$

Mathematically, this "learn reward from expert + learn policy from reward" paradigm can always be condensed as "learn policy from expert"

IRL with Preference Feedback

$$p^*(y_1 \succ y_2 \mid x) = \frac{\exp\left(r^*(x, y_1)\right)}{\exp\left(r^*(x, y_1)\right) + \exp\left(r^*(x, y_2)\right)}$$

Behavior assumption

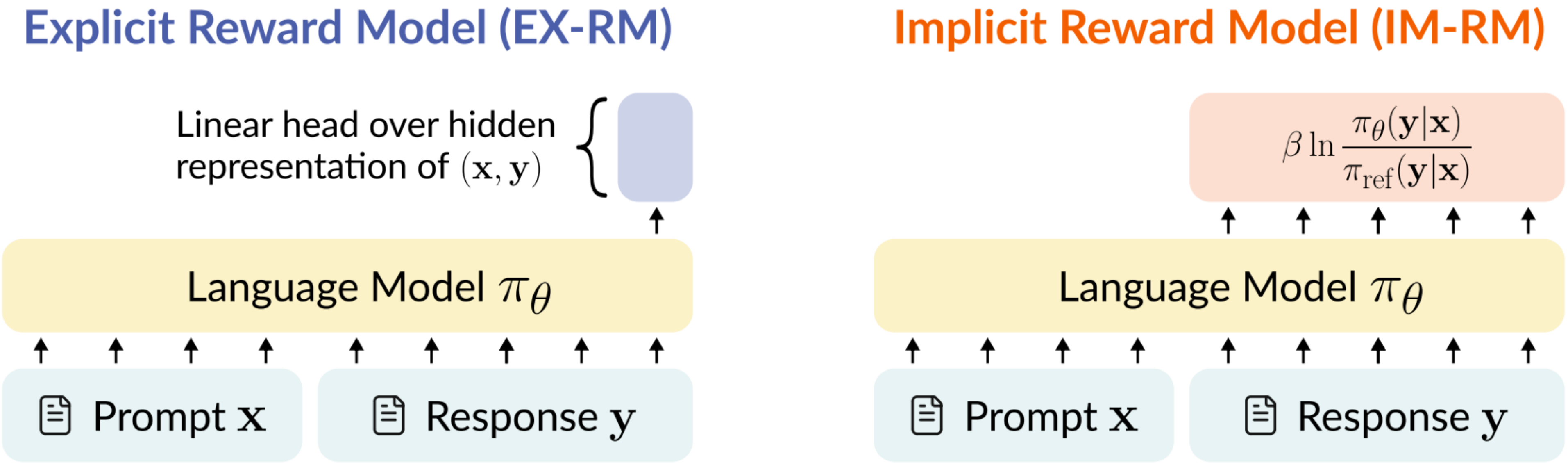
$$\mathcal{L}_R(r_\phi, \mathcal{D}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}}\left[\log \sigma(r_\phi(x, y_w) - r_\phi(x, y_l))\right]$$

Maximum likelihood

$$\max_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(y \mid x)}\left[r_\phi(x, y)\right] - \beta \mathbb{D}_{\mathrm{KL}}\left[\pi_\theta(y \mid x) \mid\mid \pi_{\mathrm{ref}}(y \mid x)\right]$$

Policy training

Benefits of having an explicit reward model



Razin et al., 2025. Why is Your Language Model a Poor Implicit Reward Model?