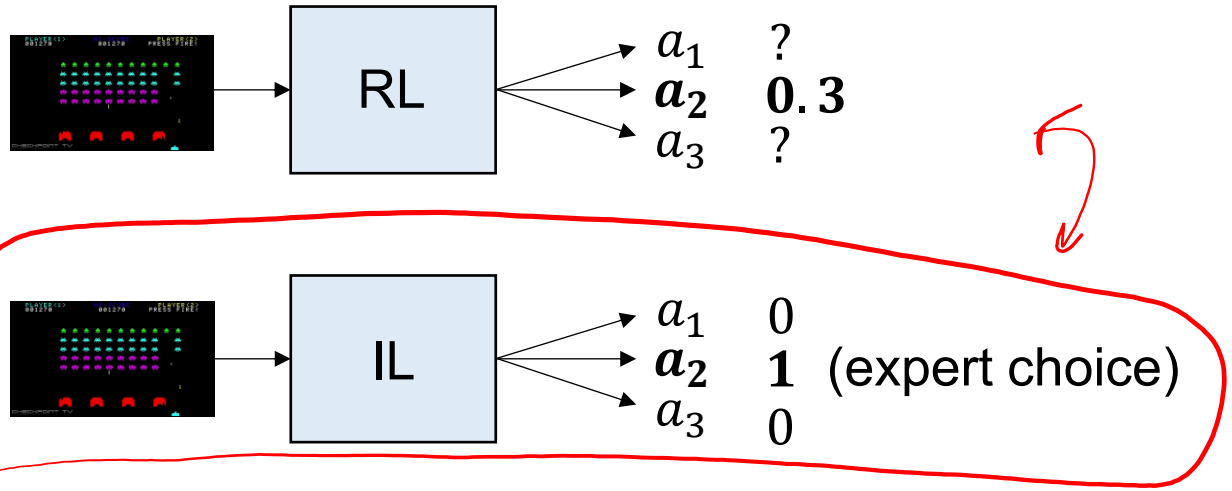
Imitation Learning

Chen-Yu Wei

Imitation Learning



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

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

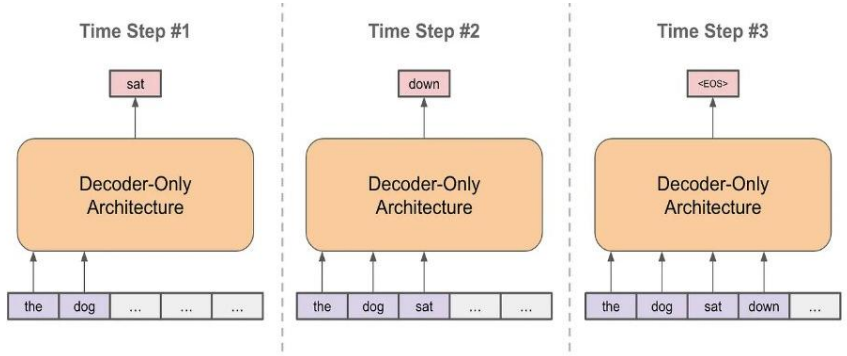
action chosen by expert

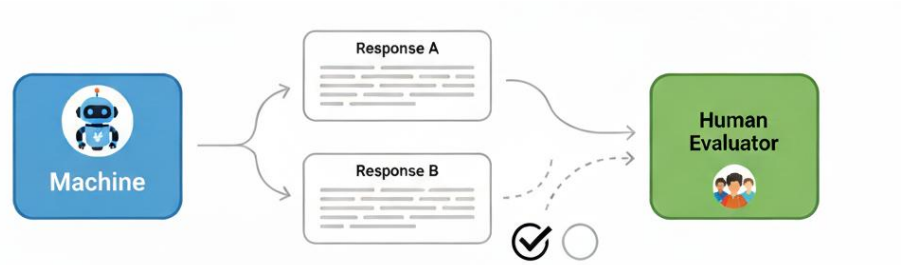
generated by expert

generated by the learner

Examples

- Language models





- 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 Supervised Learning

Behavior Cloning (Offline IL)

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

Train a policy π_θ such that

$$\pi_\theta(s_i) \approx a_i$$

(classification if actions are discrete) $\mathbb{1}\{\pi_\theta(s_i) \neq a_i\}$

(regression if actions are continuous) $\|\pi_\theta(s_i) - a_i\|^2$

Behavior Cloning: Reduction to Supervised Learning

Behavior Cloning (Offline IL)

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

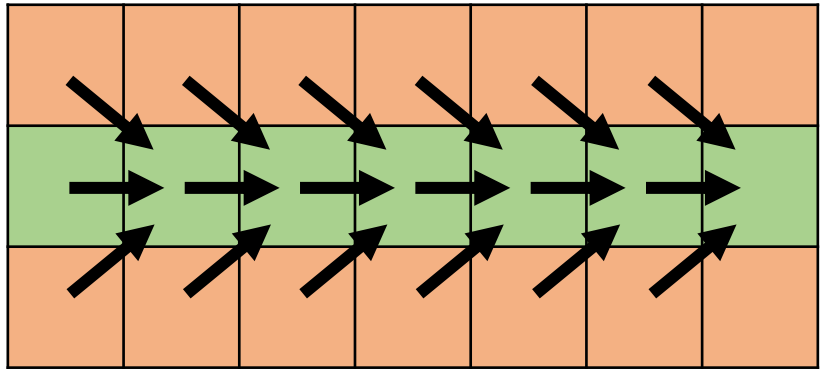
Obtain policy π_θ by finding

$$\theta = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \log \frac{1}{\pi_\theta(a_i|s_i)} \qquad \text{(discrete action)}$$

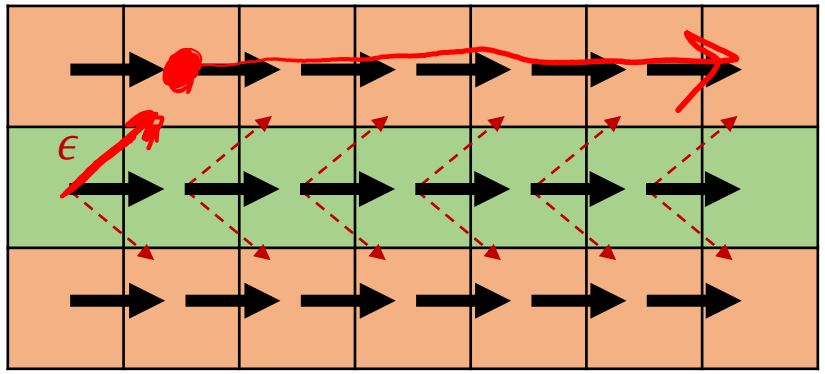
$$\theta = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \|a_i - \pi_\theta(s_i)\|^2 \qquad \text{(continuous action)}$$

$\pi_\theta(s_i) \approx a_i$

Behavior Cloning: Reduction to Supervised Learning



Optimal policy

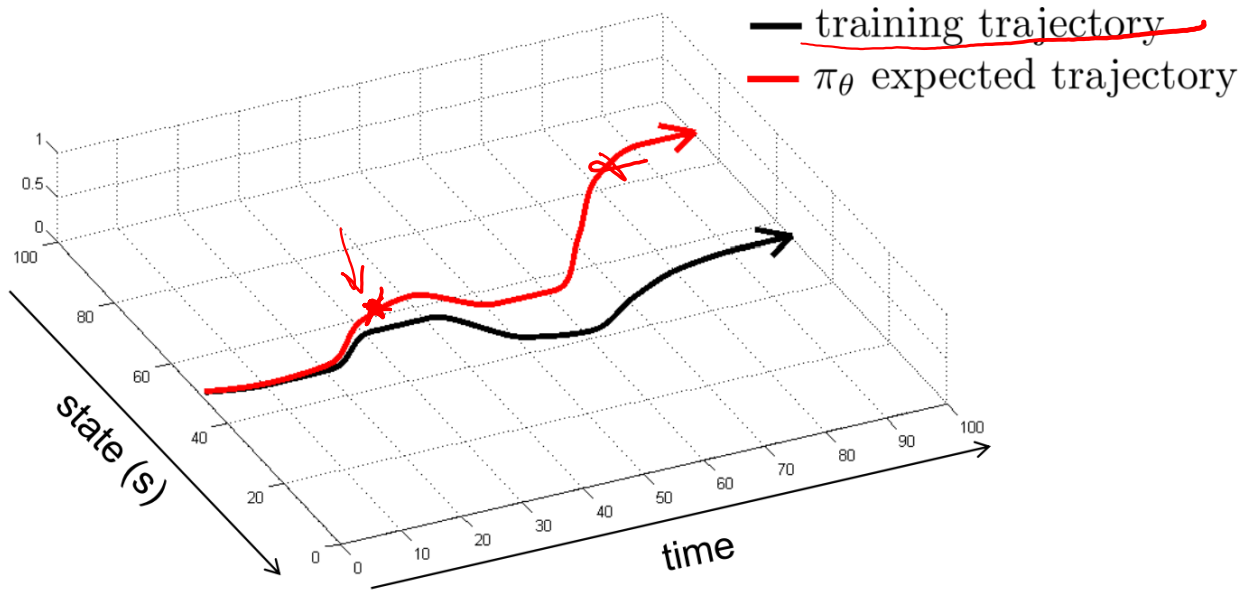


A policy with low training error (ϵ)

but high testing error ($\approx \epsilon \times$ Horizon_length)

Key issue: the training data does not cover the states visited by the learner's policy

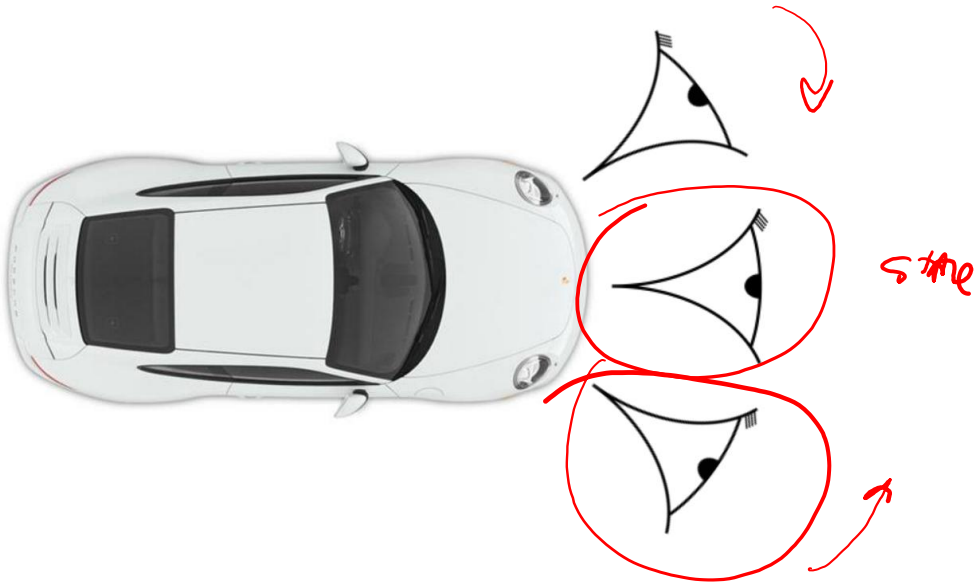
Behavior Cloning: Reduction to Supervised Learning



Distribution shift

Solution 1

- Data augmentation



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

https://www.youtube.com/watch?v=NJU9ULQUwng

Solution 2: 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 $\mathcal{B}$

Stephane Ross, Geoffrey J. Gordon, J. Andrew Bagnell. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning

Solution 2: Interact with Expert (Online IL)

A Variant of DAgger

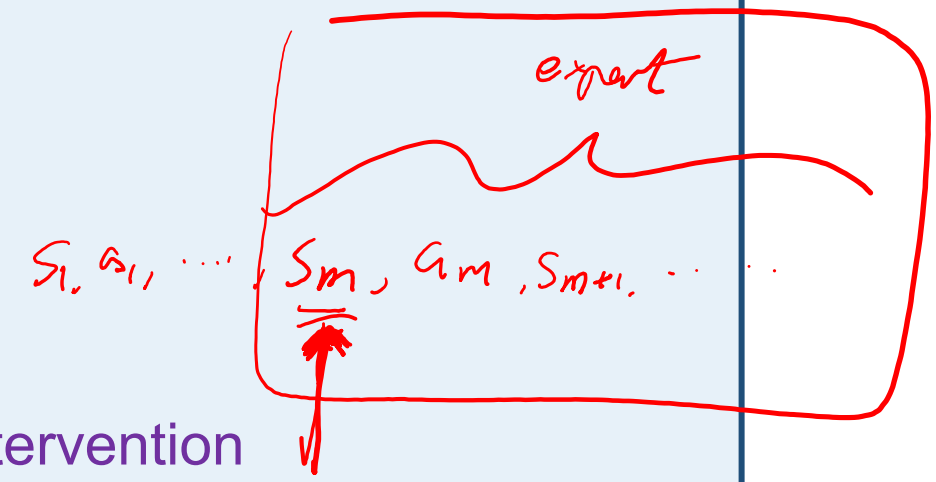
For $k = 1,\ 2, \ldots$

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

Run π_θ

Ask the expert to take over at some time step

Store in $\mathcal{B}$ the samples generated by expert intervention



Stephane Ross, Geoffrey J. Gordon, J. Andrew Bagnell. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning

Key: collecting (s, a) samples with s generated from the learner (+ expert), and a from the expert

input output

Imitation + RL

Imitate then Reinforce

- A common paradigm of learning decision making:
  - Learn a policy π_{BC} through behavior cloning from experts
  - Run RL algorithm with the policy network initialized as $\pi_\theta = \pi_{\mathrm{BC}}$

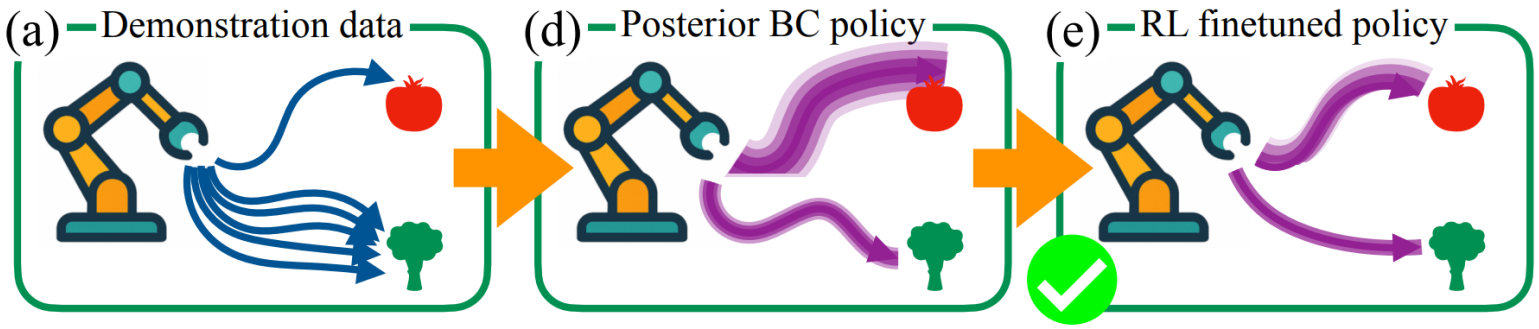
- This could largely save the time for RL to find the optimal policy
  - The amount / cost of exploration can be largely reduced

- Questions
  - What exploration strategy should the RL use?
  - How should we perform behavior cloning to facilitate downstream RL algorithm?

Imitate and Reinforce

- For states where expert demonstration provides **sufficient** guidance
  - The learner does not need to explore that much
- For states where expert demonstration provides **insufficient** guidance
  - The learner may need to explore more

It's tempting to employ uncertainty-aware exploration in the RL phase, where the uncertainty accounts for the **uncertainty of the expert's action**.



Wagenmaker et al. Posterior Behavioral Cloning: Pretraining BC Policies for Efficient RL Finetuning. 2025.

Imitate and Reinforce

Run expert policy $\pi^\star$ and obtain $\mathcal{B} = \{(s_1, a_1), \ldots, (s_N, a_N)\}$

Learn a policy μ_θ through behavior cloning from $\mathcal{B}$

Run RL algorithm with policy initialized as μ_θ. When interacting with the environment, use policy

$$\mu_\theta(s) + \alpha \cdot \mathcal{N}(0, \mathrm{cov}(s))$$

Related to the action uncertainty on state s

How to Estimate Action Uncertainty

- Similar to techniques we discussed in the “Exploration” lecture, though there we mainly focused on the uncertainty of $Q^\star$ estimation
- The following idea is similar to Bootstrapped DQN (and also related to RND):

Train policies $\hat{\mu}_1, \hat{\mu}_2, \ldots, \hat{\mu}_L$ with Behavior Cloning with random initialized weights and randomly sub-sampled data:

For $\ell = 1, \ldots, L$:

Sample $\mathcal{B}_\ell \subset \mathcal{B}$ randomly

$$\hat{\mu}_\ell = \operatorname*{argmin}_{\mu} \sum_{(s,a)\in\mathcal{B}_\ell} \|\mu(s) - a\|^2$$

Define $\operatorname{cov}(s) = \frac{1}{L}\sum_{\ell=1}^{L}(\hat{\mu}_\ell(s) - \overline{\mu}(s))(\hat{\mu}_\ell(s) - \overline{\mu}(s))^\top \in \mathbb{R}^{d\times d}$ where $\overline{\mu}(s) = \frac{1}{L}\sum_{\ell=1}^{L}\hat{\mu}_\ell(s)$

Imitate and Reinforce

Run expert policy $\pi^\star$ and obtain $\mathcal{B} = \{(s_1, a_1), \ldots, (s_N, a_N)\}$

Learn a policy μ_θ through behavior cloning from $\mathcal{B}$, and learn $\hat{\mu}_1, \hat{\mu}_2, \ldots, \hat{\mu}_L$ using the procedure in the previous slide.

Run RL algorithm with policy initialized as μ_θ. When interacting with the environment, use policy

$$\mu_\theta(s) + \alpha \cdot \mathcal{N}(0, \mathrm{cov}(s))$$

Defined as in the previous slide

Imitate and Reinforce

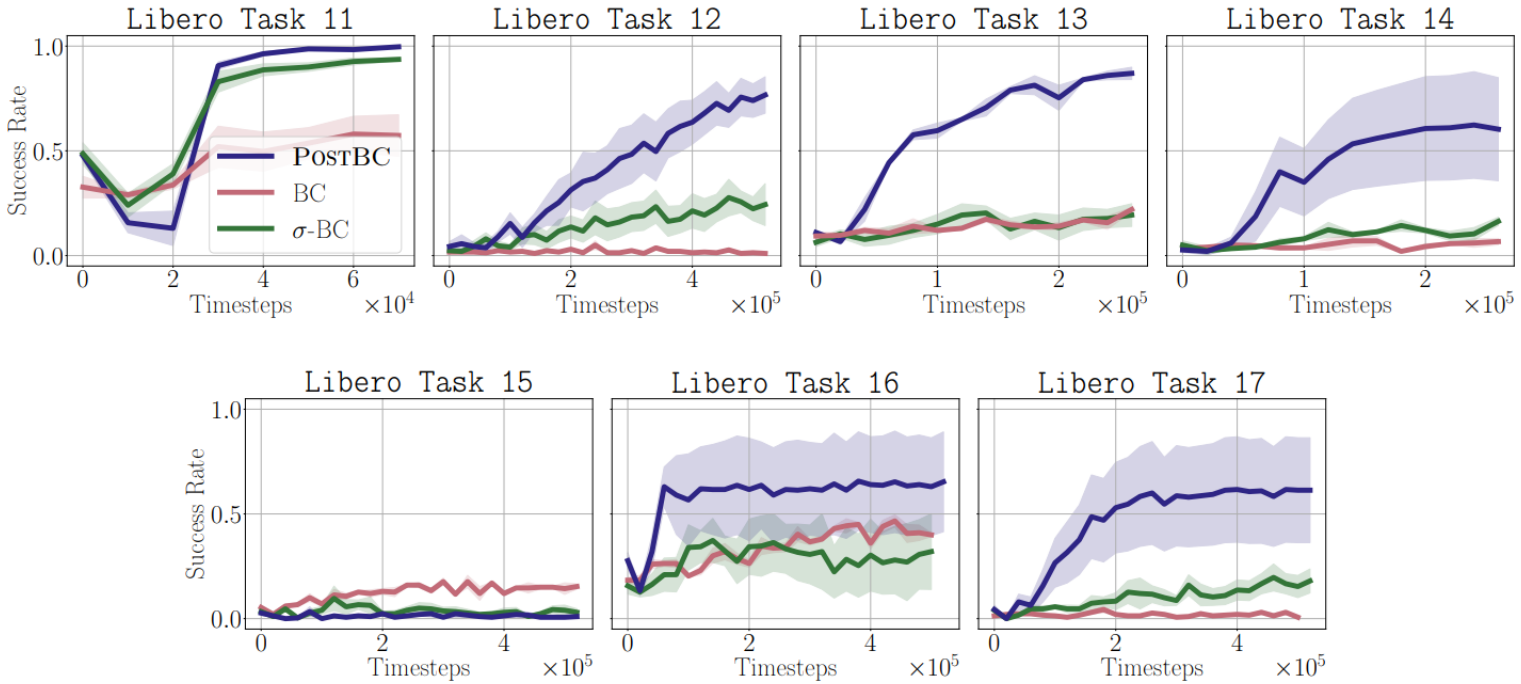


Figure 5: Comparison of DSRL finetuning performance combined with different BC pretraining approaches on all tasks from `Libero 90`, `Kitchen Scene 2`.

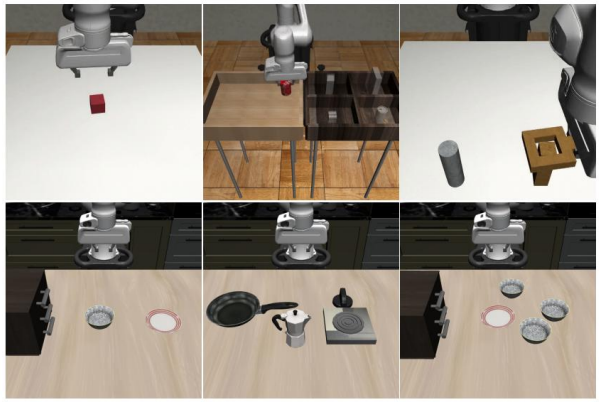


Figure 2: `Robomimic` and `Libero` settings

Wagenmaker et al. Posterior Behavioral Cloning: Pretraining BC Policies for Efficient RL Finetuning. 2025.

Imitation with Preference Feedback

Preference Feedback from “Experts”

Offline

Given dataset $(x_1, a_{1w}, a_{1l}), (x_2, a_{2w}, a_{2l}), \ldots (x_N, a_{Nw}, a_{Nl})$
where a_{iw} is a **preferred action** by the expert than a_{il} when seeing context x_i

Online

For $t = 1, \ldots, T$:

Learner sees context x_i

Learner picks two actions $a_i^{(1)}, a_i^{(2)}$ and query the expert for their preference

With expert’s feedback, the learner forms a tuple (x_i, a_{iw}, a_{il})

Imitation with Preference Feedback

Direct Preference Optimization (DPO) (link) (An important technique for LLM alignment)

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).

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

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

where $\sigma(z) = \frac{1}{1+e^{-z}}$