The Information Geometry of RL from Human Feedback

Gokul Swamy

Outline for Today

1. What is the fine-tuning problem?

2. End-to-end, what is the two-stage RLHF process doing?

3. What are direct alignment algorithms?

Outline for Today

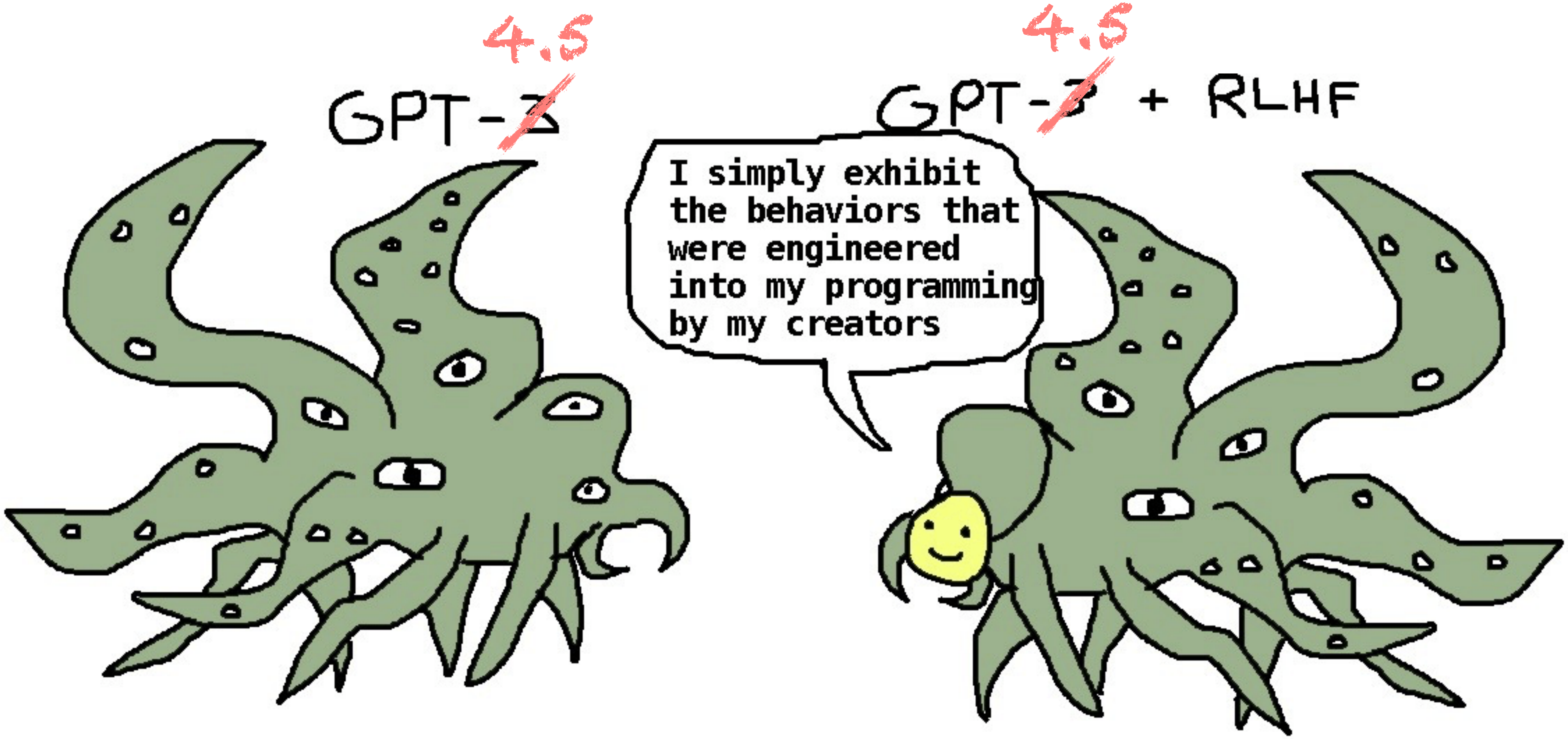
1. What is the fine-tuning problem?

***A**: Regularized maximum likelihood estimation.*

2. End-to-end, what is the two-stage RLHF process doing?

3. What are direct alignment algorithms?

We Live in the Era of Fine-Tuning



Base

FT

We Live in the Era of Fine-Tuning

Base

FT

[Oertell et al.]

We Live in the Era of Fine-Tuning

Prompt:
What is the purpose of the list C in the code below?

```
def binomial_coefficient(n, r):
     C = [0 for i in range(r + 1)];
     C[0] = 1;
     for i in range(1, n + 1):
          j = min(i, r);
          while j > 0:
               C[j] += C[j - 1];
               j -= 1;
     return C[r]
```

GPT-3 175B completion:
A. to store the value of C[0]
B. to store the value of C[1]
C. to store the value of C[i]
D. to store the value of C[i - 1]

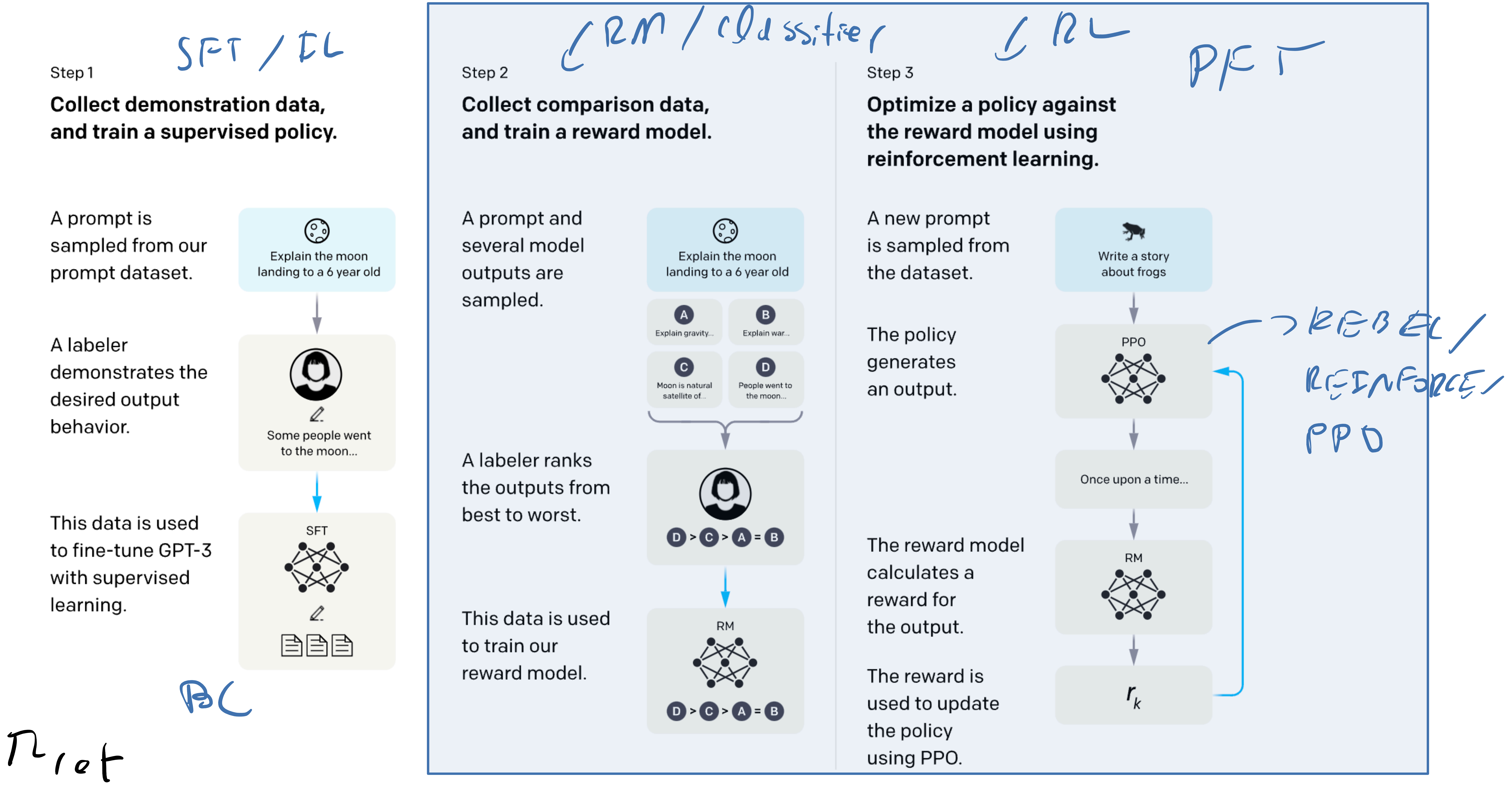
InstructGPT 175B completion:
The list C in this code is used to store the values of the binomial coefficient as the function iterates through the values of n and r. It is used to calculate the value of the binomial coefficient for a given value of n and r, and stores the result in the final return value of the function.

Base

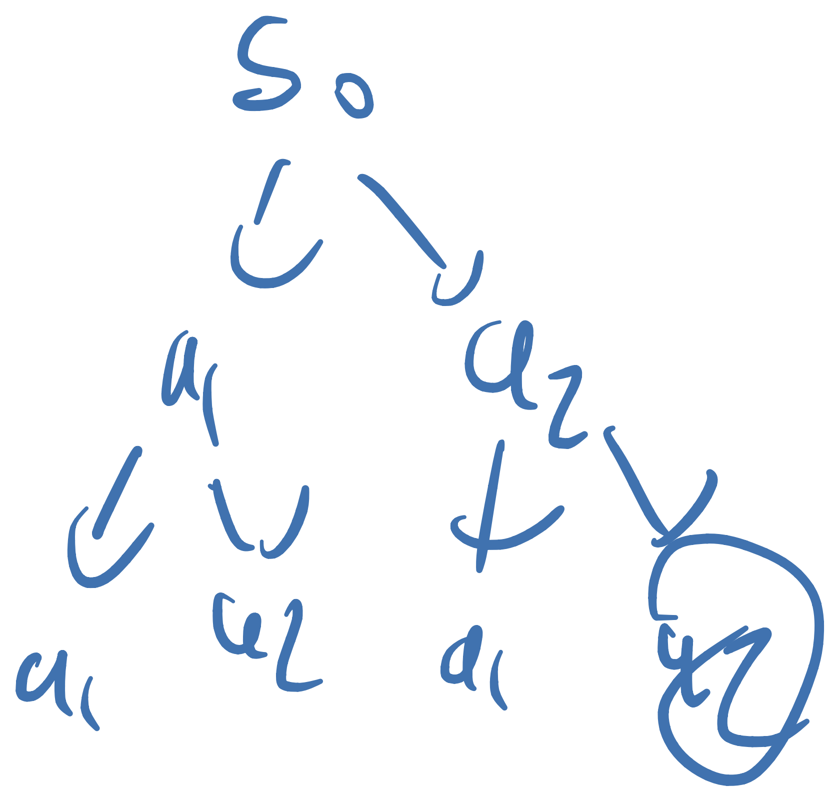
FT

[Ouyang et al.]

We Live in the Era of Fine-Tuning



Language Modeling as an MDP



$s_1 = [s_0, a_1]$, $s_2 = [s_0, a_1, a_2]$, ..., $s_H = [s_0, a_1 \dots a_H]$

↳ reset = generating from a prefix

Prompt	Token 1	Token 2	...	Token H	Reward
$s_0 \sim \rho_0$	$a_1 \in A$	$a_2 \in A$	...	$a_H \in A$	$r(s_H)$

Completion ($s \sim \pi \mid s_0$)

$T(s' \mid s, a) = 1$ if $s' = s \circ a$

0 o/w

↳ deterministic, tree-structured

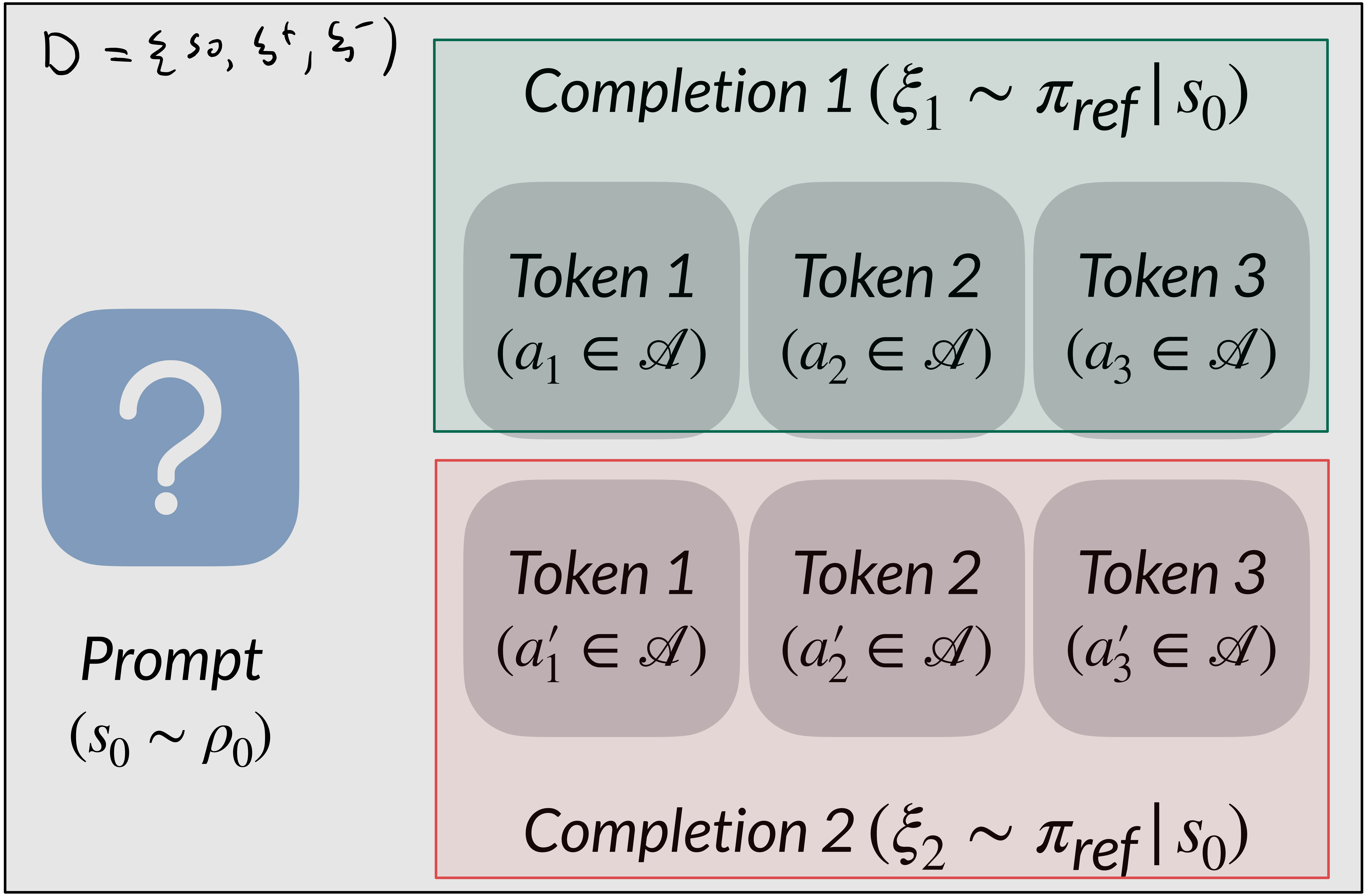
What makes the Language MDP Special

1. Dynamics are deterministic, known, and tree-structured.

2. Resets are just generating from a prefix — easy to do.

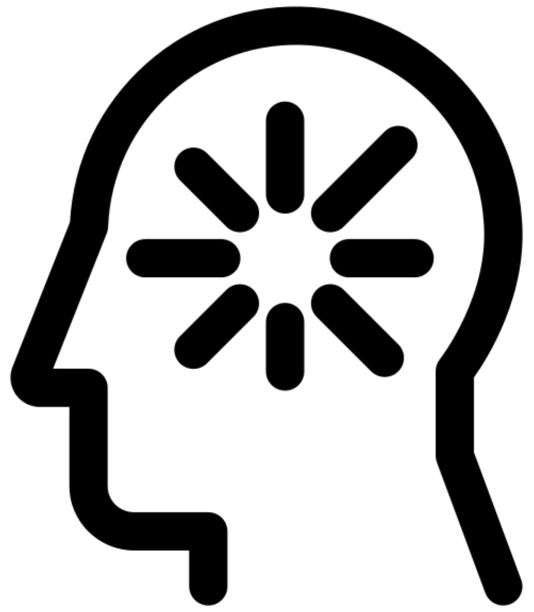
3. The reward function is non-Markovian and doesn't decompose into token-wise rewards.

Preference Fine-Tuning



(+) easier data collection

(−) one bit of information



Preference

$\xi_1 \succ \xi_2$

Preference Fine-Tuning

***Goal**: Maximize the relative likelihood of preferred to dis-preferred completions.*

FKL RKL

$$\pi^\star = \arg\min_{\pi \in \Pi} \mathbb{D}_{KL}\left(\mathscr{D} \,||\, \pi\right) \quad + \quad \mathbb{D}_{KL}\left(\pi \,||\, \pi_{\text{ref}}\right)$$

(Data Likelihood) (Prior Reg.)

& imitate
contrast

Outline for Today

1. What is the fine-tuning problem?

A: *Regularized maximum likelihood estimation.*

2. End-to-end, what is the two-stage RLHF process doing?

A: *MLE over reward models followed by MaxEnt over policies.*

3. What are direct alignment algorithms?

Notation

assuming transitivity ✓
all raters mostly agree

For simplicity, we're going to assume the "Bradley-Terry" model of preferences:

$$\mathbb{P}_r(\xi_1 \succ \xi_2 \,|\, s_0) = \sigma(r(\xi_1) - r(\xi_2))$$

Also, let's denote the empirical preference distribution as:

$$\mathbb{P}_{\mathcal{D}}(\xi_1 \succ \xi_2 \,|\, s_0)$$

i.e. how often raters preferred ξ_1 to ξ_2 given prompt s_0.

Reward Modeling is MLE

Then,

$$\hat{r}_{\text{mle}} = \arg\min_{r \in \mathscr{R}} \mathbb{E}_{s_0 \sim \mathscr{D}}[\mathbb{D}_{KL}(\mathbb{P}_{\mathscr{D}} \,||\, \mathbb{P}_r)]$$

Forward KL

$D_{KL}(P \| Q) = \mathbb{E}_p[\log P] - \mathbb{E}_p[\log Q]$

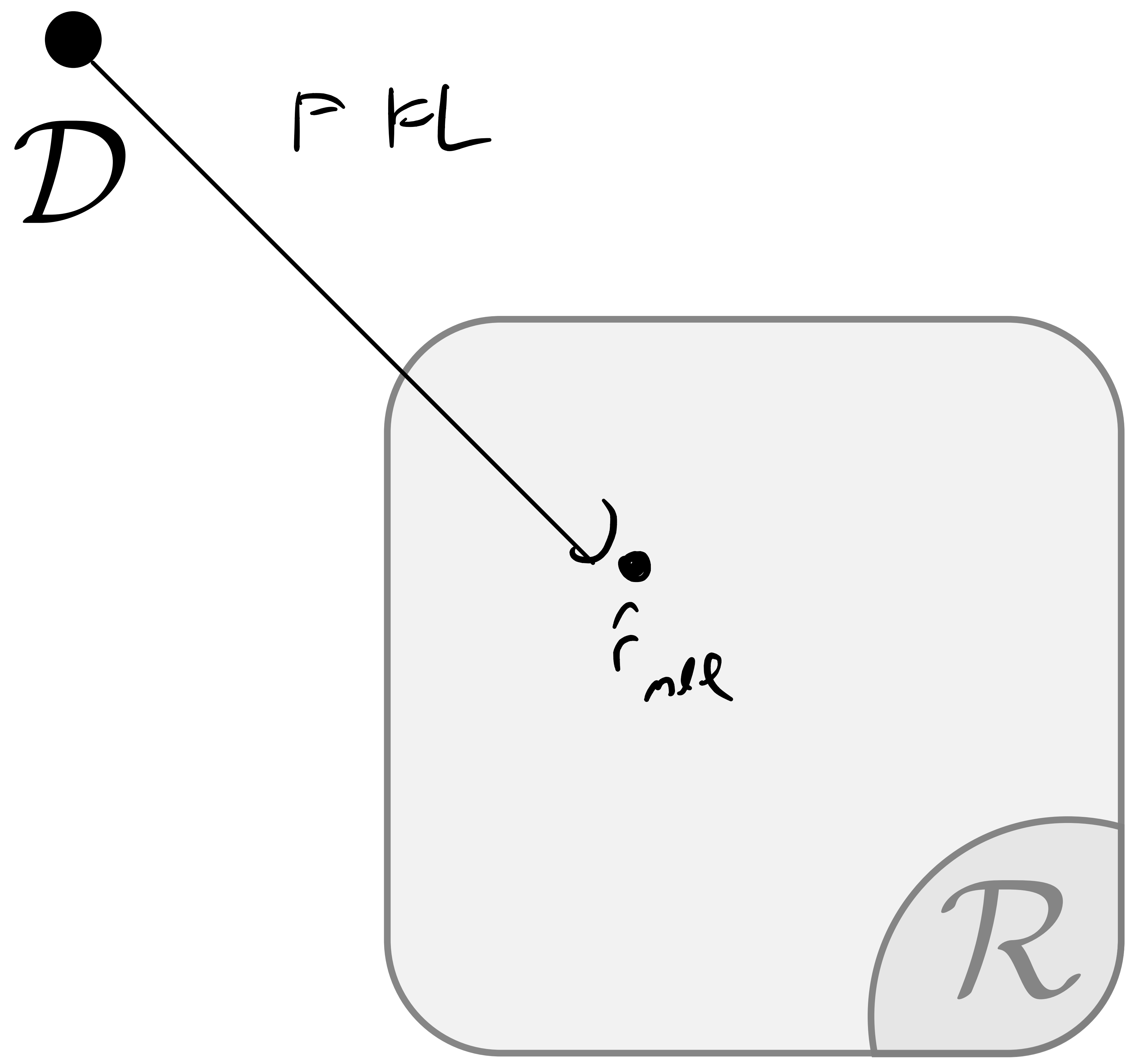
$$= \arg\max_{r \in \mathscr{R}} \mathbb{E}_{(s_0, \xi^+, \xi^-) \sim \mathscr{D}}[\log \mathbb{P}_r(\xi^+ \succ \xi^- \mid s_0)]$$

BT

$$= \arg\max_{r \in \mathscr{R}} \mathbb{E}_{(s_0, \xi^+, \xi^-) \sim \mathscr{D}}[\log \sigma(r(\xi^+) - r(\xi^-))]$$

This is just logistic regression / classification!

Reward Modeling is a FKL Projection onto $\mathscr{R}$



Recap: “Soft” / Entropy Regularized RL

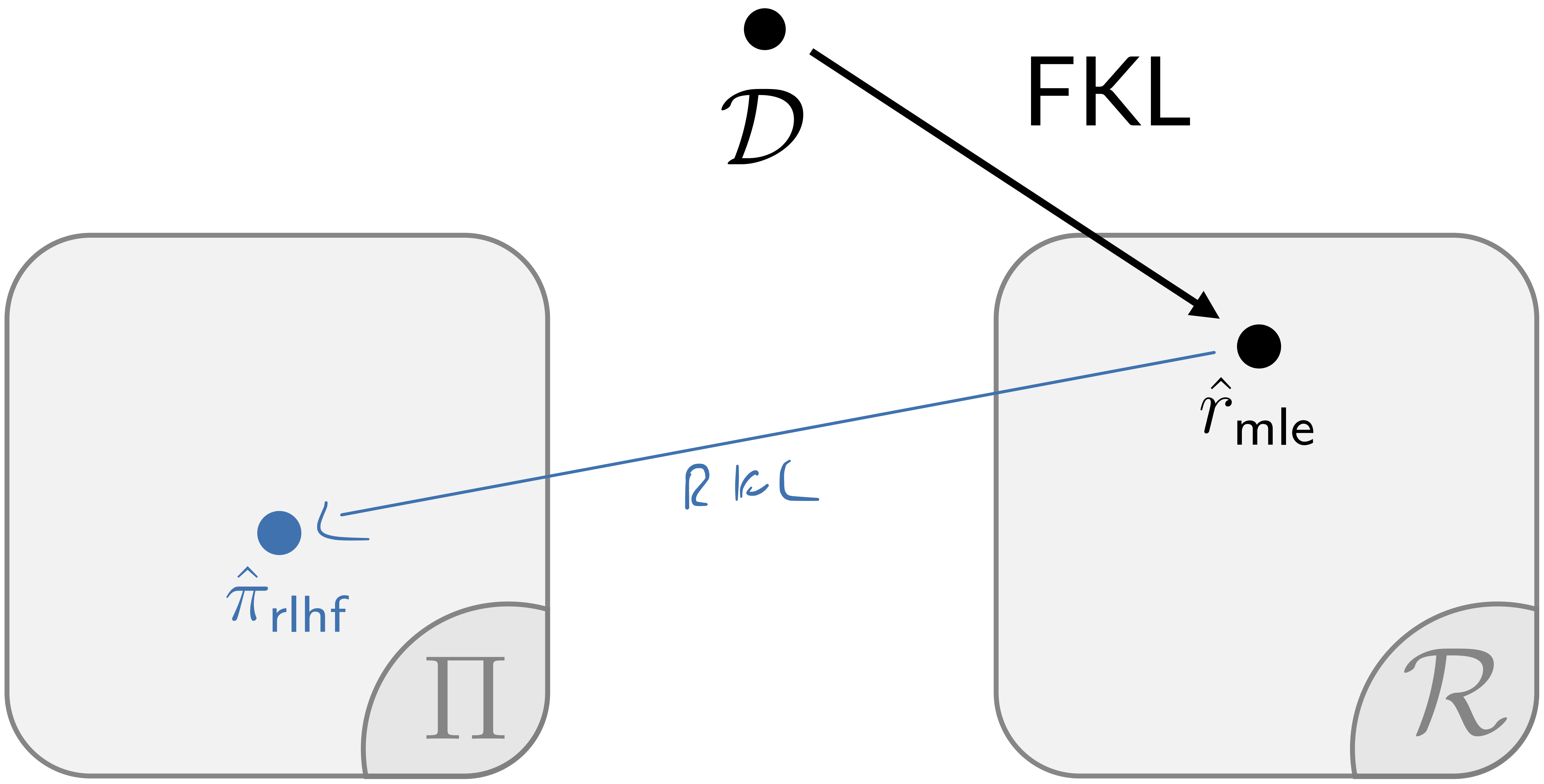
$$\hat{\pi}_{\text{rlhf}} = \arg\max_{\pi \in \Pi} \mathbb{E}_{\xi \sim \pi}[\hat{r}_{\text{mle}}(\xi)] + \mathbb{D}_{KL}(\pi \,||\, \pi_{\text{ref}})$$

$$\mathbb{E}_{\xi \sim \pi}\left[\sum_{h}^{H} \log\left(\frac{\pi(a_h \mid s_h)}{\pi_{\text{ref}}(a_h \mid s_h)}\right)\right]$$

deterministic dynamics

$$\prod_{h}^{H} \pi_r^{\star}(a_h \mid s_h) = \mathbb{P}_{\hat{r}}^{\star}(\xi \mid s_0) = \frac{\mathbb{P}_{\text{ref}}(\xi) \cdot \exp(\hat{r}(\xi))}{\sum_{\xi' \in \Xi \mid s_0} \mathbb{P}_{\text{ref}}(\xi') \cdot \exp(\hat{r}(\xi'))}$$

Soft RL is a *Reverse* KL Projection onto Π



E2E, *(1) RLHF is FKL to $\mathscr{R}$ and (2) RKL to Π*

If ⏱️: Soft RL is a *Reverse* KL Projection onto Π

$$\hat{\pi}_{\text{rlhf}} = \arg\min_{\pi \in \Pi} \mathbb{D}_{KL}(\mathbb{P}_{\pi} \,||\, \mathbb{P}^{\star}_{\hat{r}})$$

$$= \arg\min_{\pi \in \Pi} \mathbb{E}_{\xi \sim \mathbb{P}_{\pi}} \left[\log \left(\frac{\mathbb{P}_{\pi}(\xi)}{\mathbb{P}^{\star}_{\hat{r}}(\xi)} \right) \right]$$

$$= \arg\min_{\pi \in \Pi} \sum_{\xi \in \Xi} \mathbb{P}_{\pi}(\xi)(\log \mathbb{P}_{\pi}(\xi) - \log \mathbb{P}^{\star}_{\hat{r}}(\xi))$$

$$= \arg\min_{\pi \in \Pi} \sum_{\xi \in \Xi} \mathbb{P}_{\pi}(\xi)(\log \mathbb{P}_{\pi}(\xi) - \hat{r}(\xi) + \log Z^{\star}_{\hat{r}})$$

$$= \arg\min_{\pi \in \Pi} \sum_{\xi \in \Xi} \mathbb{P}_{\pi}(\xi)(\log \mathbb{P}_{\pi}(\xi) - \hat{r}(\xi))$$

$$= \arg\max_{\pi \in \Pi} \sum_{\xi \in \Xi} \mathbb{P}_{\pi}(\xi)(-\log \mathbb{P}_{\pi}(\xi) + \hat{r}(\xi))$$

$$= \arg\max_{\pi \in \Pi} \mathbb{E}_{\xi \sim \pi}[\hat{r}(\xi)] + \mathbb{H}(\pi)$$

Outline for Today

1. What is the fine-tuning problem?
A: *Regularized maximum likelihood estimation.*

2. End-to-end, what is the two-stage RLHF process doing?
A: *MLE over reward models followed by MaxEnt over policies.*

3. What are direct alignment algorithms?
A: *Algorithms like DPO directly maximize likelihood over Π without passing through $\mathscr{R}$.*

The DPO "Reparameterization Trick"

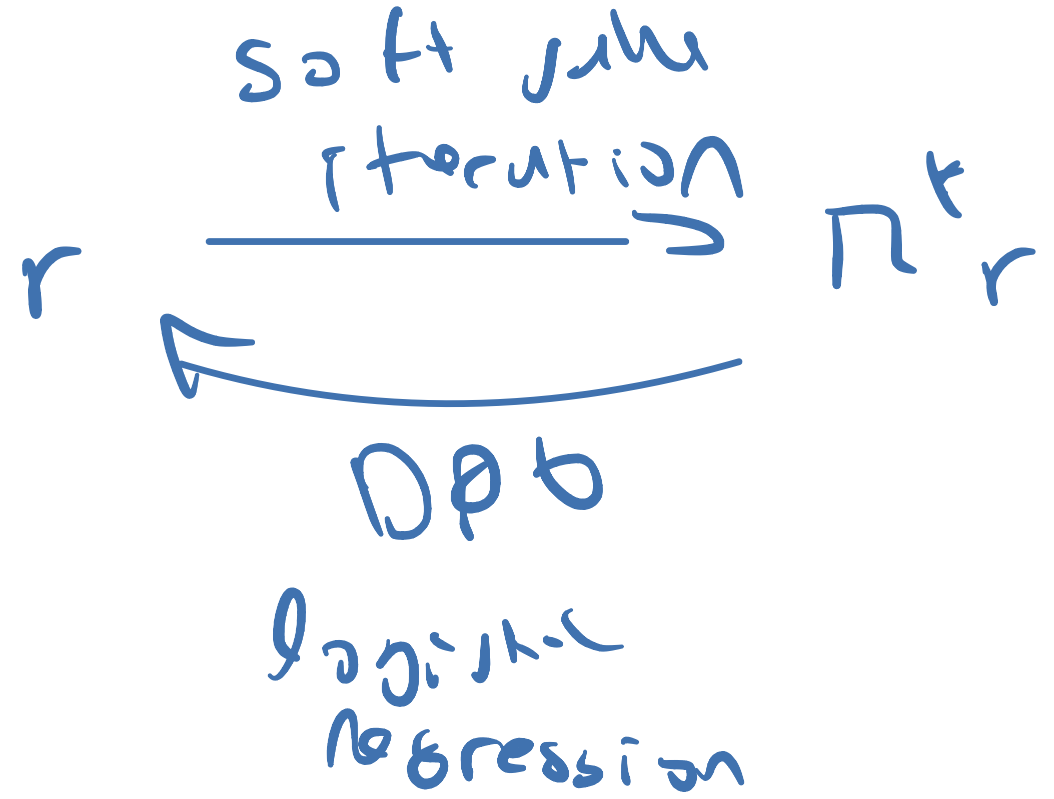
$$\prod_{h}^{H} \pi_r^{\star}(a_h \mid s_h) = \frac{\prod_h^H \pi_{\text{ref}}(a_h \mid s_h) \cdot \exp(r(\xi))}{Z(s_0)}$$

↓ taking log of both sides

$$\sum_{h}^{H} \log \pi_r^{\star}(a_h \mid s_h) = \sum_{h}^{H} \log \pi_{\text{ref}}(a_h \mid s_h) + r(\xi) - \log Z(s_0)$$

$$r(\xi) = \sum_{h}^{H} \log \pi_r^\star(a_h \mid s_h) - \log \pi_{\text{ref}}(a_h \mid s_h) + \log Z(s_0)$$

$$\triangleq r_\pi(\xi)$$



We can express the reward model that makes a policy (soft) optimal in terms of said policy by “inverting” the MaxEnt RL equations!

More explicitly, consider the soft-optimal policy for r_π:

$$\mathbb{P}^\star_{r_\pi}(\xi) \propto \exp(r_\pi(\xi))$$

plug in

$$\propto \exp\left(\sum_h^H \log \pi(a_h \mid s_h) + \log Z(s_0)\right)$$

$$\propto \exp\left(\sum_h^H \log \pi(a_h \mid s_h)\right)$$

$$\propto \prod_h^H \pi(a_h \mid s_h)$$

The soft optimal policy for r_π is π, which means we can optimize over r_π and get the soft optimal policy "for free"!

Now, we proceed by MLE *directly* over policies:

$$\hat{\pi}_{\text{dpo}} = \arg\max_{\pi \in \Pi} \mathbb{E}_{(s_0, \xi^+, \xi^-) \sim \mathcal{D}} [\log \sigma(r_\pi(\xi^+) - r_\pi(\xi^-))]$$

$$= \arg\max_{\pi \in \Pi} \mathbb{E}_{(s_0, \xi^+, \xi^-) \sim \mathcal{D}} \left[\log \sigma \left(\sum_h^H \log \frac{\pi(a_h^+ \mid s_h^+)}{\pi_{\text{ref}}(a_h^+ \mid s_h +)} - \log \frac{\pi(a_h^- \mid s_h^-)}{\pi_{\text{ref}}(a_h^- \mid s_h^-)} \right) \right]$$

So, we end up with a single-step MLE procedure!

DPO is a FKL Projection onto Π