

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

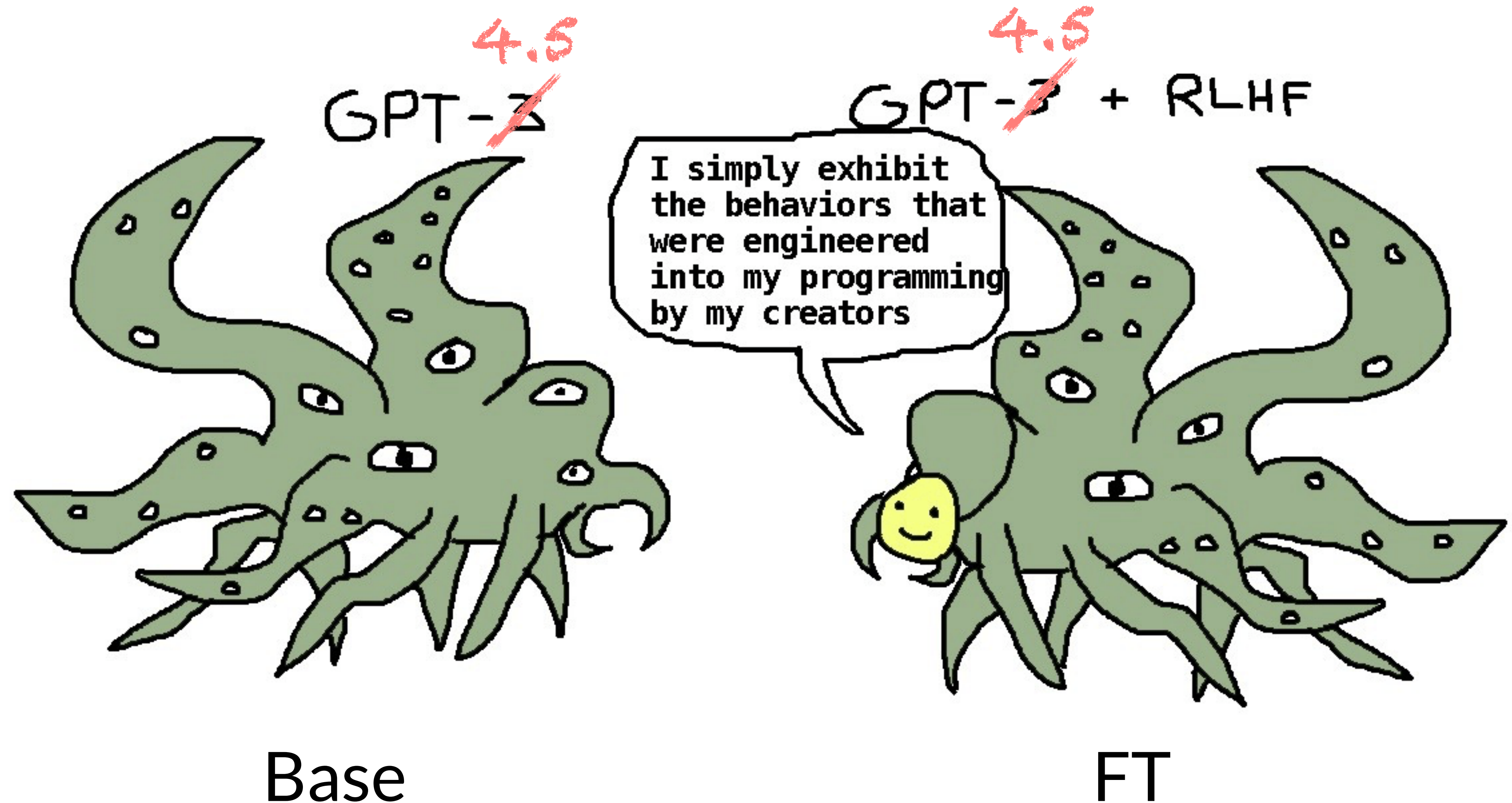
1. What is the fine-tuning problem?

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



We Live in the Era of Fine-Tuning

Base



FT



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

We Live in the Era of Fine-Tuning

Step 1

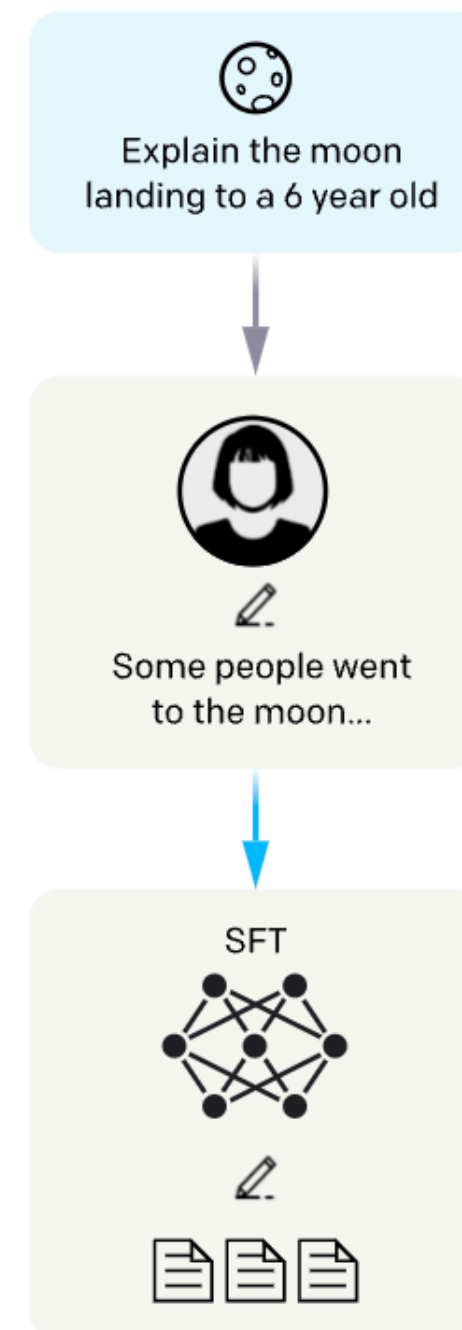
**Collect demonstration data,
and train a supervised policy.**

A prompt is
sampled from our
prompt dataset.

A labeler
demonstrates the
desired output
behavior.

This data is used
to fine-tune GPT-3
with supervised
learning.

SFT / DL



π_{ref}

BC

Step 2

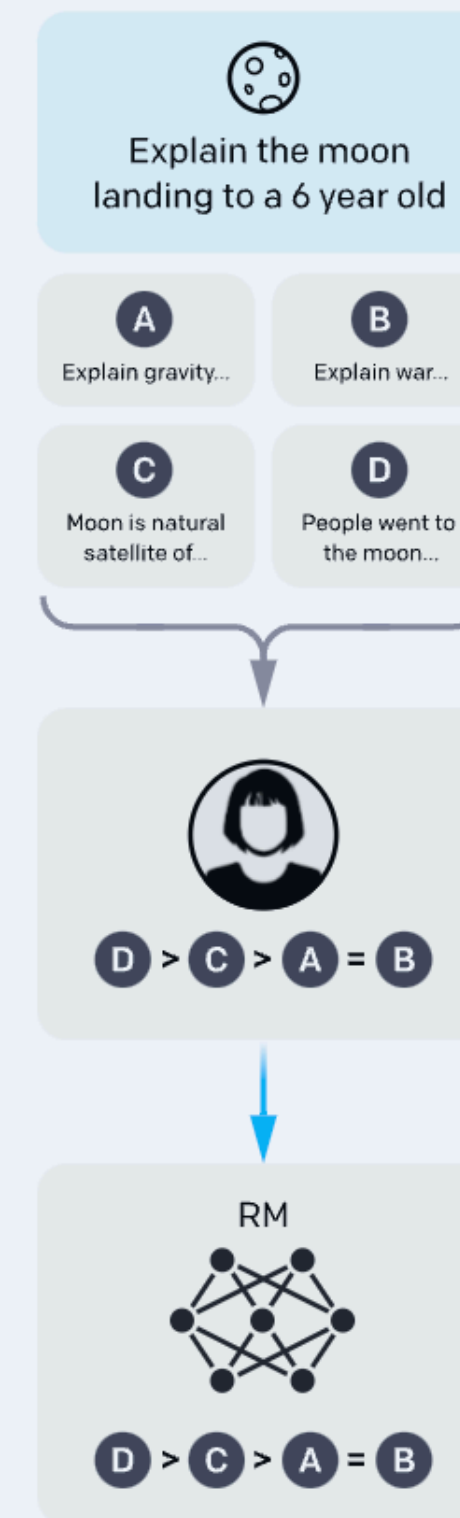
**Collect comparison data,
and train a reward model.**

A prompt and
several model
outputs are
sampled.

A labeler ranks
the outputs from
best to worst.

This data is used
to train our
reward model.

RM / classifier



Step 3

**Optimize a policy against
the reward model using
reinforcement learning.**

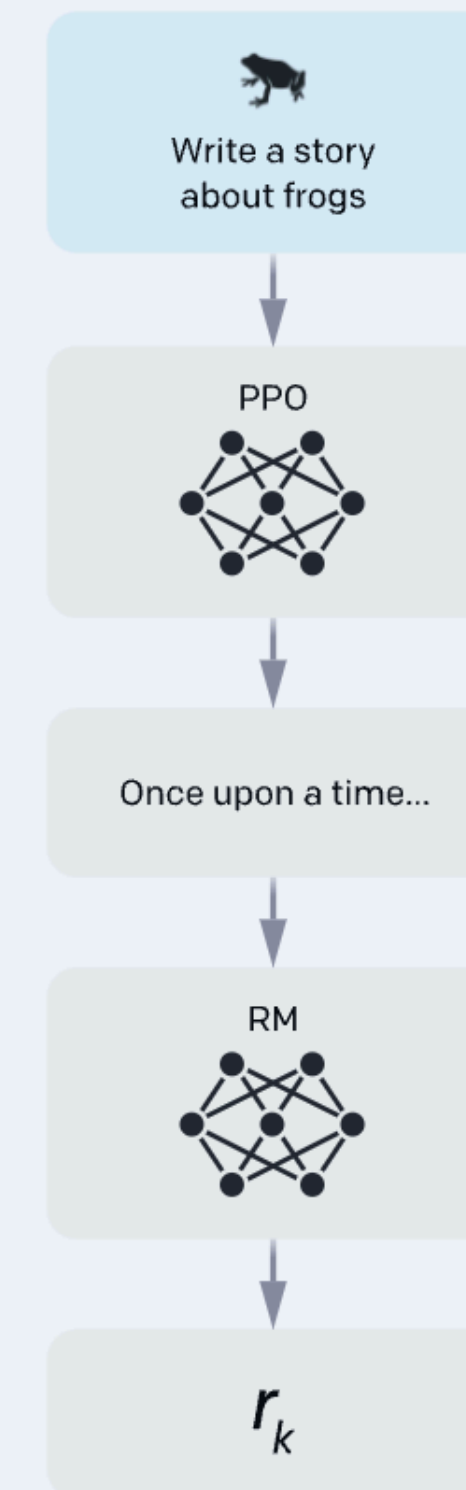
A new prompt
is sampled from
the dataset.

The policy
generates an output.

The reward model
calculates a
reward for the
output.

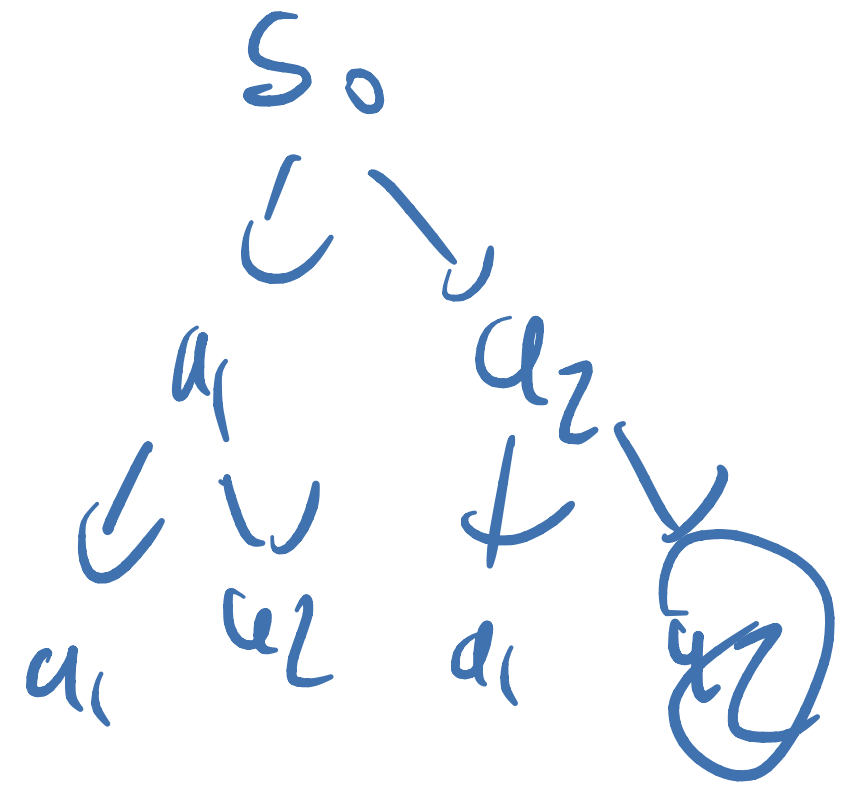
The reward is
used to update
the policy
using PPO.

RL P/E T



REBEL /
REINFORCE /
PPO

Language Modeling as an MDP



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

\hookrightarrow next = generating from a prefix



Prompt

$$s_0 \sim p_0$$

Token 1

$$a_1 \in A$$

Token 2

$$a_2 \in A$$

...

Token H

$$a_H \in A$$

$$r(s_H)$$

Reward

Completion ($\xi \sim \pi | s_0$)

$$\tau(s' | s, a) = 1 \text{ if } s' = s \cdot a$$

$$0 \text{ o/w}$$

\hookrightarrow 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

$$D = \{s_0, \xi^+, \xi^-\}$$



Prompt
($s_0 \sim \rho_0$)

Completion 1 ($\xi_1 \sim \pi_{ref} | s_0$)

Token 1
($a_1 \in \mathcal{A}$)

Token 2
($a_2 \in \mathcal{A}$)

Token 3
($a_3 \in \mathcal{A}$)

Token 1
($a'_1 \in \mathcal{A}$)

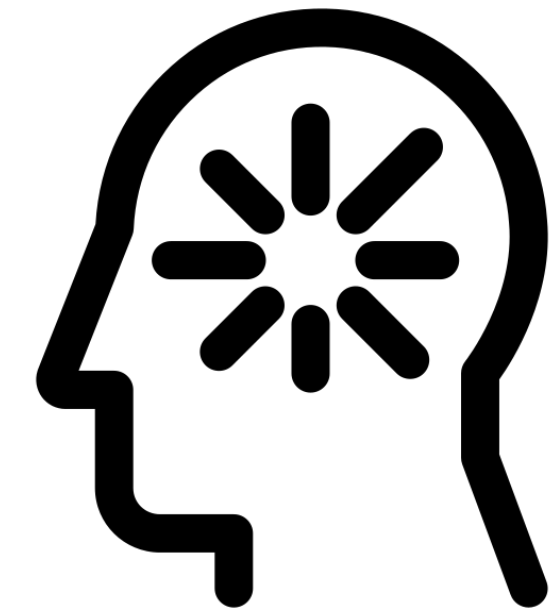
Token 2
($a'_2 \in \mathcal{A}$)

Token 3
($a'_3 \in \mathcal{A}$)

Completion 2 ($\xi_2 \sim \pi_{ref} | s_0$)

(+) see star 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.

$$\pi^{\star} = \arg \min_{\pi \in \Pi} \mathbb{D}_{KL}(\mathcal{D} || \pi) + \mathbb{D}_{KL}(\pi || \pi_{\text{ref}})$$

(Data Likelihood)

(Prior Reg.)

FKL

RKL

limited coverage

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

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

*assuming transitivity ✓
all raters mostly agree*

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

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 .

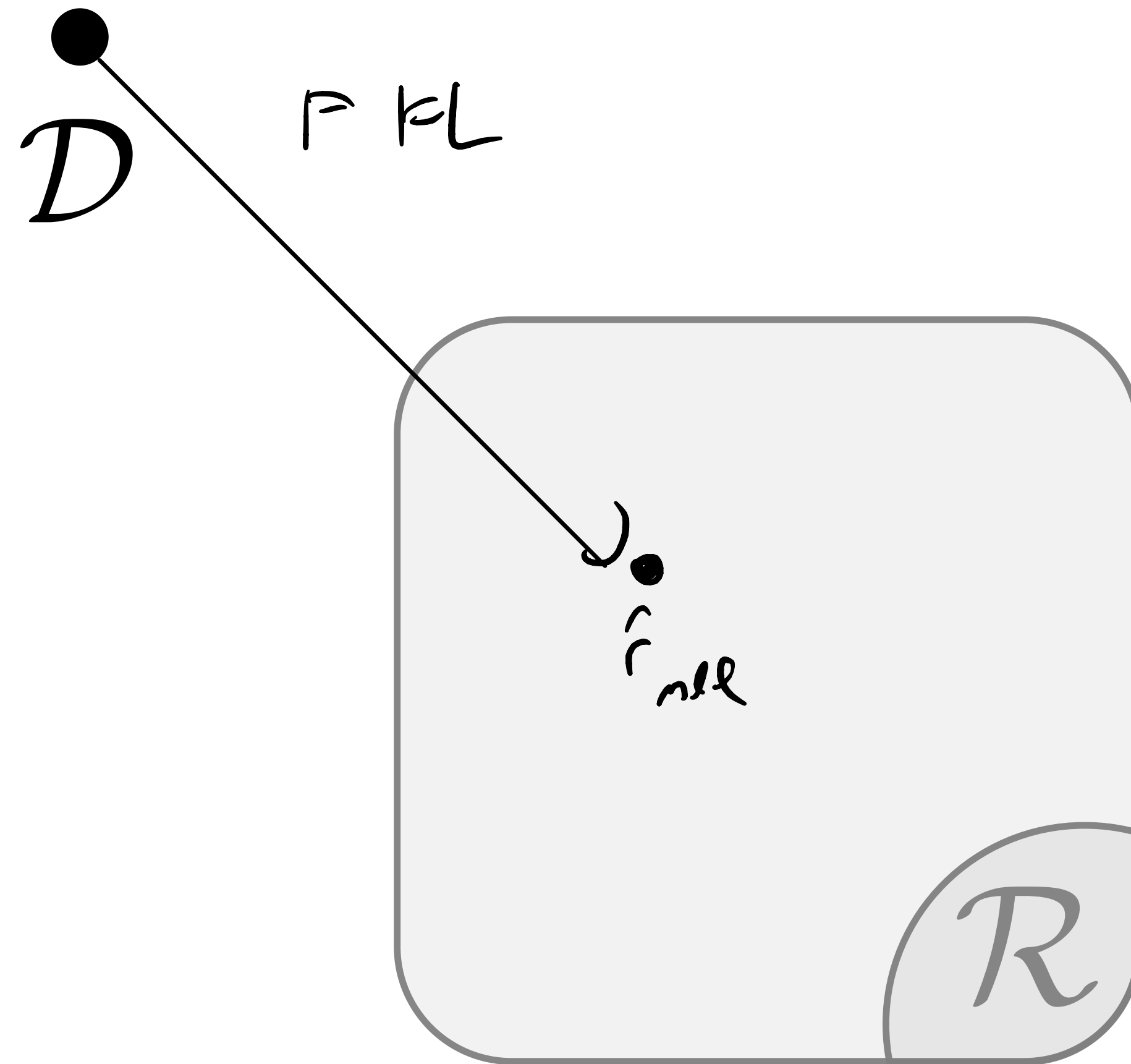
Reward Modeling is MLE

Then,

$$\begin{aligned}\hat{r}_{\text{mle}} &= \arg \min_{r \in \mathcal{R}} \mathbb{E}_{s_0 \sim \mathcal{D}} [\mathbb{D}_{KL}(\mathbb{P}_{\mathcal{D}} \parallel \mathbb{P}_r)] \quad \text{(\textit{forward RL})} \\ &= \arg \max_{r \in \mathcal{R}} \mathbb{E}_{(s_0, \xi^+, \xi^-) \sim \mathcal{D}} [\log \mathbb{P}_r(\xi^+ > \xi^- \mid s_0)] \quad \begin{array}{l} D_{KL}(\mathbb{P} \parallel \mathbb{Q}) \\ = \mathbb{E}_{\mathbb{P}}[\log \mathbb{P}] \\ - \mathbb{E}_{\mathbb{P}}[\log \mathbb{Q}] \end{array} \\ &= \arg \max_{r \in \mathcal{R}} \mathbb{E}_{(s_0, \xi^+, \xi^-) \sim \mathcal{D}} [\log \sigma(r(\xi^+) - r(\xi^-))] \quad \text{(\textit{BT})}\end{aligned}$$

This is just logistic regression / classification!

Reward Modeling is a FKL Projection onto \mathcal{R}



Recap: “Soft” / Entropy Regularized RL

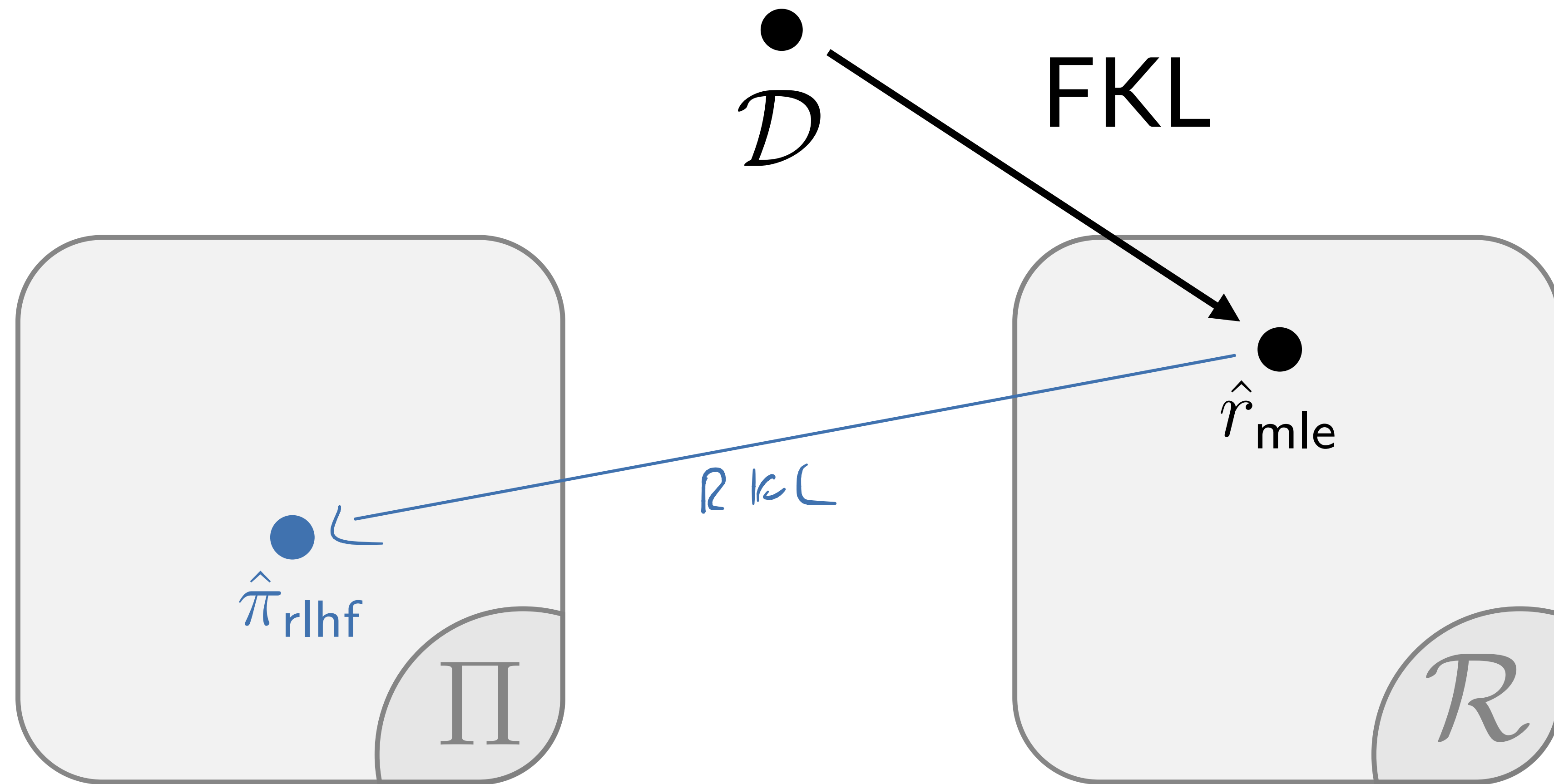
$$\hat{\pi}_{\text{rlhf}} = \arg \max_{\pi \in \Pi} \mathbb{E}_{\xi \sim \pi} [\hat{r}_{\text{mle}}(\xi)] + \underbrace{\mathbb{D}_{KL}(\pi \parallel \pi_{\text{ref}})}_{\text{entropy regularization}}$$

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

deterministic
dynamic

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

Soft RL is a *Reverse* KL Projection onto Π



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

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

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

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

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

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

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

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

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

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 \mathcal{R} .

The DPO “Reparameterization Trick”

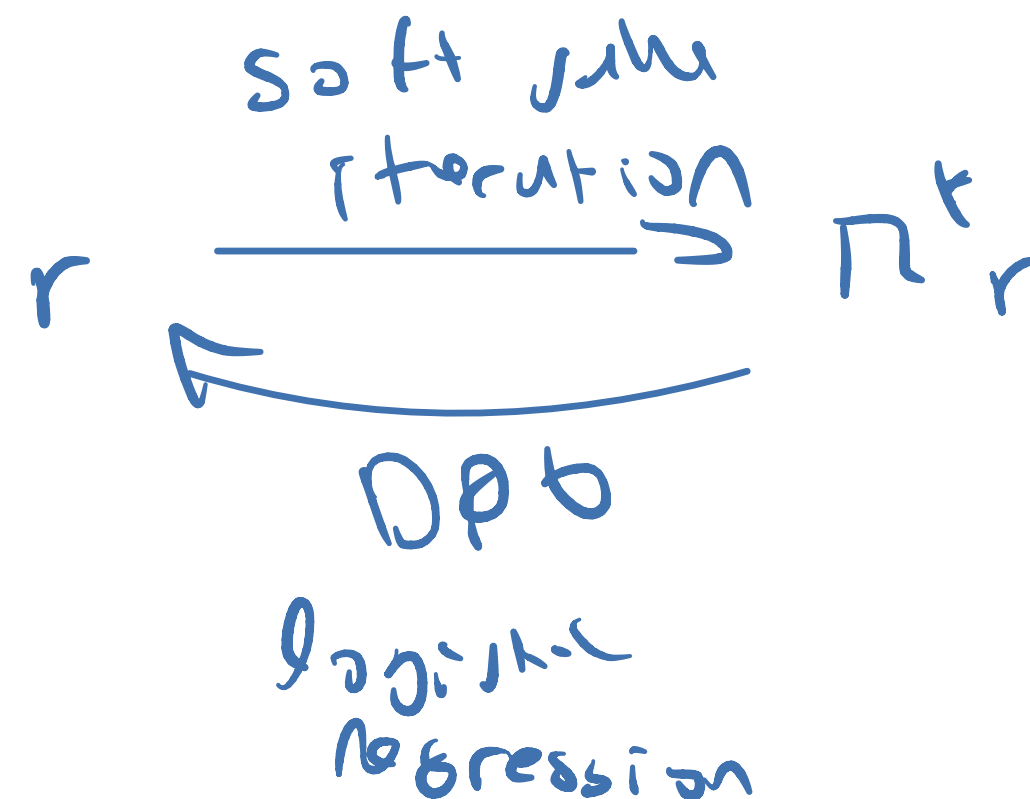
$$\prod_h^H \pi_r^\star(a_h | s_h) = \frac{\prod_h^H \pi_{\text{ref}}(a_h | s_h) \cdot \exp(r(\xi))}{Z(s_0)}$$

↓ taking log of
both sides

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

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

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



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 :

$$\begin{aligned}\mathbb{P}_{r_\pi}^\star(\xi) &\propto \exp(r_\pi(\xi)) \quad \text{planned} \\ &\propto \exp\left(\sum_h^H \log \pi(a_h | s_h) + \log Z(s_0)\right) \\ &\propto \exp\left(\sum_h^H \log \pi(a_h | s_h)\right) \\ &\propto \prod_h^H \pi(a_h | s_h)\end{aligned}$$

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:

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

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

DPO is a FKL Projection onto Π

