Week 5: Preference-Based Fine-Tuning for Alignment

Generative AI
Saarland University – Winter Semester 2024/25

Goran Radanovic
genai-w24-tutors@mpi-sws.org

MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

MAX-PLANCK-GESELLSCHAFT

Outline of the Lecture

- Reminders
- Recap: Supervised Fine-tuning
- Preference-based Fine-tuning: Overview
- RLHF: Reinforcement Learning
- RLHF: Reward Model Training
- Direct Preference Optimization

Outline of the Lecture

- **Reminders**
- Recap: Supervised Fine-tuning
- Preference-based Fine-tuning: Overview
- RLHF: Reinforcement Learning
- RLHF: Reward Model Training
- Direct Preference Optimization

Reminders

- **Week 4 assignment – deadline**: Nov 18, 6pm CET
- **Week 5 assignment – deadline**: Nov 25, 6pm CET
- **Next week:** No lectures or office hours (time to work on assignments)
- **Next Lecture**: Nov 26, 10:15am CET

Outline of the Lecture

- Reminders
- **Recap: Supervised Fine-tuning**
- Preference-based Fine-tuning: Overview
- RLHF: Reinforcement Learning
- RLHF: Reward Model Training
- Direct Preference Optimization

Previous Lecture

- Last time we discussed pretraining and supervised fine-tuning

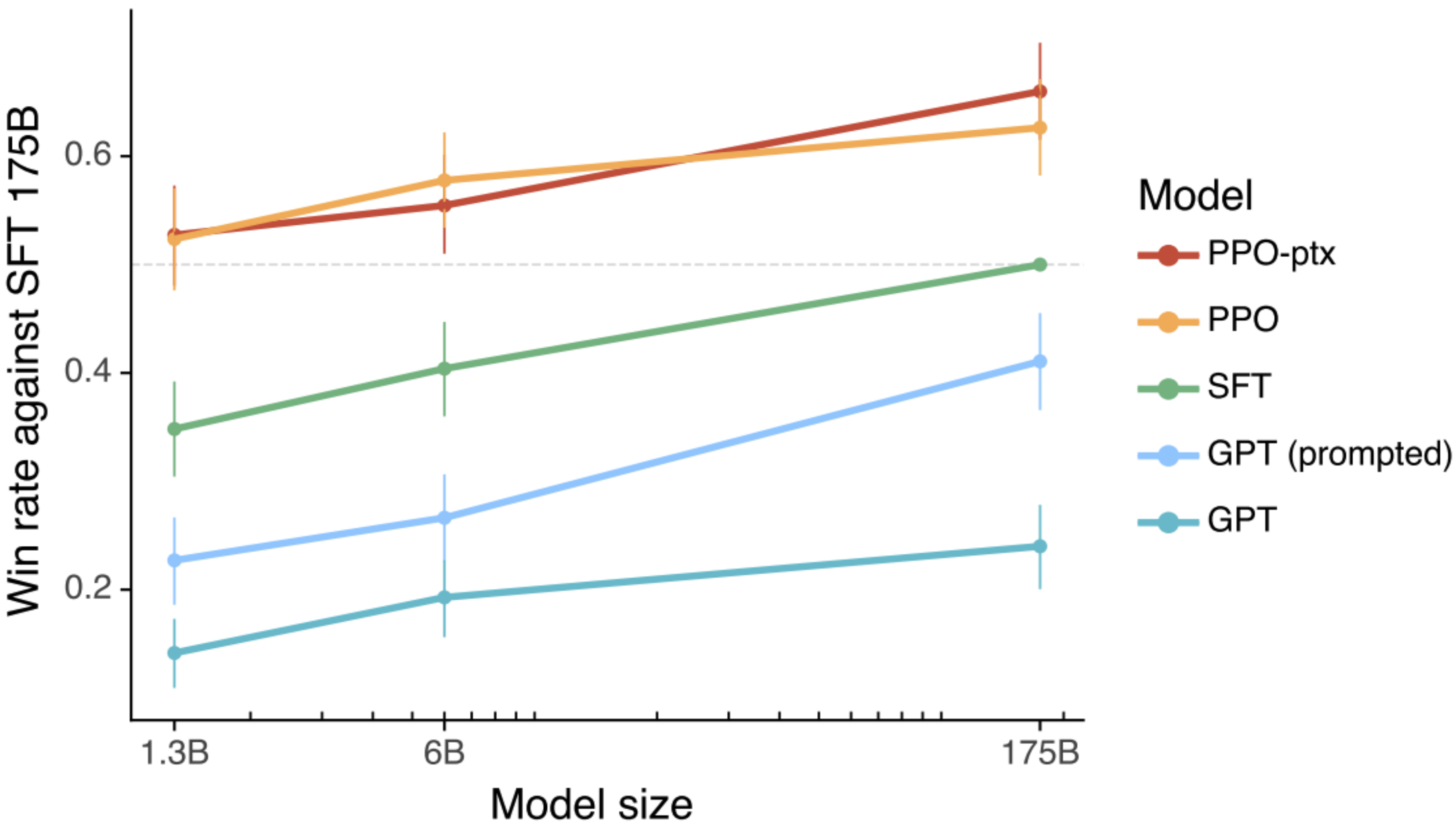
- Pretraining
  - Overview of important aspects
  - Scaling laws: infer the optimal model/data size for a given compute budget

- Supervised Fine-tuning
  - Parameter Efficient Fine-tuning with Low Rank Adaptation
  - Quantization

- Aligning with human preferences
  - Supervised fine-tuning is an important step…

Why Fine-tuning?

- Human evaluations of the outputs



Supervised Fine-tuning

Main idea

- Now the dataset is labelled: $\mathcal{D} = \{(x_p, y)\}$

  Example:

  Alignment: x_p can be an instruction and y can be a demonstration

Datasets: HuggingFaceH4 / **instruction-dataset** like 47 Follow Hugging Face H4 643

Split (1)
test · 327 rows

Search this dataset

prompt string · *lengths* 35↔276 76.1%	**completion** string · *lengths* 312↔623 17.4%
Arianna has 12 chocolates more than Danny. Danny has 6 chocolates more than Robbie. Arianna has twice as many chocolates as Robbie has. How many chocolates does Danny…	Denote the number of chocolates each person has by the letter of their first name. We know that A = D + 12 D = R + 6 A = 2 * R Thus, A = (R + 6) + 12 = R + 18 Since…
Write a plot summary for a comedic novel involving Elon Musk and sea travel.	Elon Musk hires a team of experts to build the ultimate yacht, but when the yacht is completed, he realizes that he has no idea how to sail it. With the help of a quirky crew and a fearless captain, the playboy embarks on a wild and hilarious adventure across the open seas, where the crew have to keep Elon alive despite his inability to do anything himself. All the while, Elon takes credit for their hard work.
Create a 3 turn conversation between a customer and a grocery store clerk - that is, 3 per person. Then tell me what they talked about.	Clerk: How are you doing today? Customer: Great! I'm buying groceries to cook my wife a homemade steak and lobster dinner for our 5-year anniversary! Clerk: Wow,…

Supervised Fine-tuning

Main idea

- Now the dataset is labelled: $\mathcal{D} = \{(x_p, y)\}$

  Example:

  Alignment: x_p can be an instruction and y can be a demonstration

- Optimize the next-token prediction objective, but only over response y

$$\max_\theta \sum_{(x_p,y)\in\mathcal{D}} \sum_{k=1}^{|y|} \log P_\theta(y_k | x_p, y_1, \dots, y_{k-1})$$

- What is the underlying assumption?
  - Human completions are of a high quality

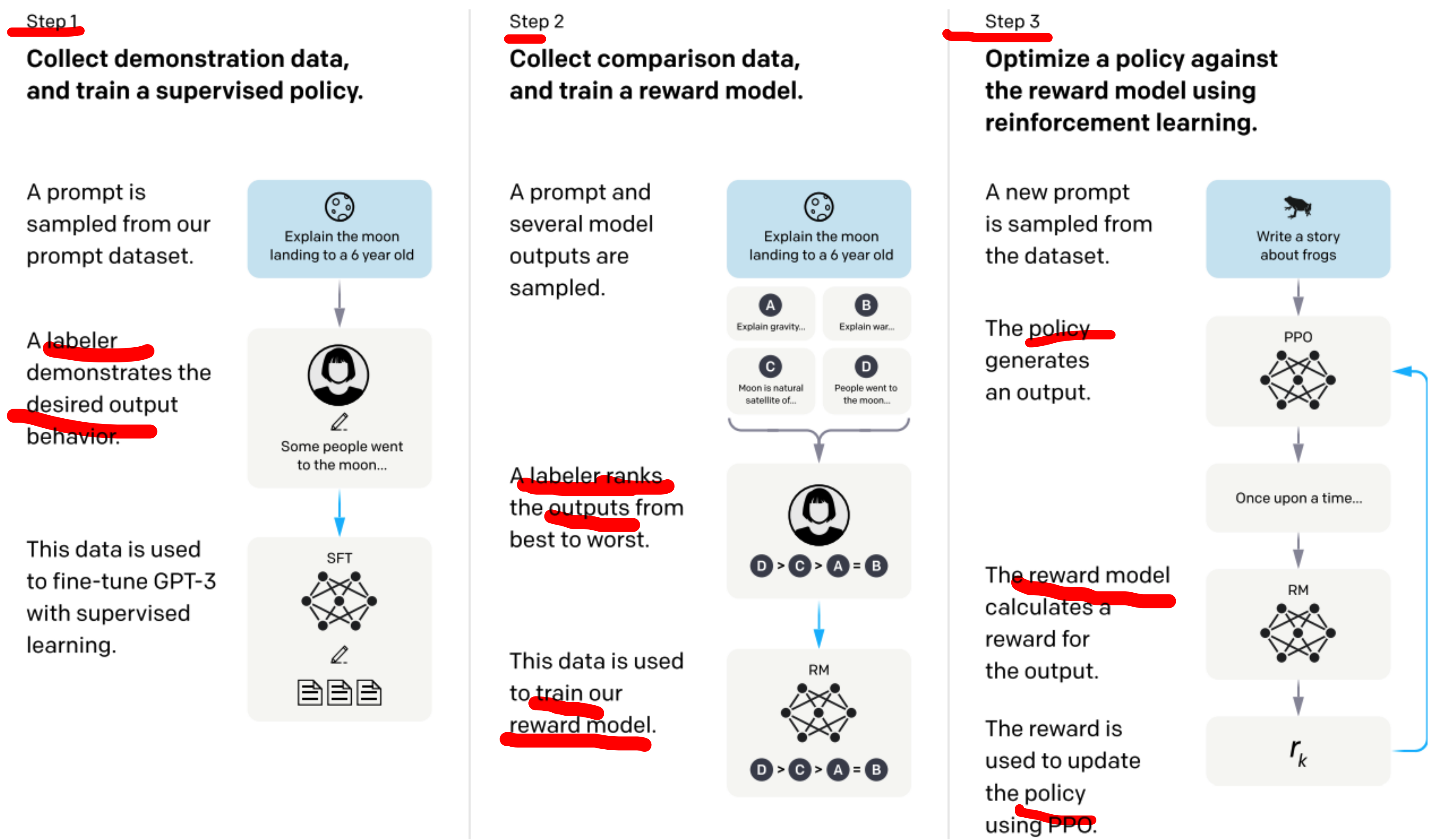
- SFT is a **behavioral cloning** technique that aims to *imitate* what humans do, not outperform them

Outline of the Lecture

- Reminders
- Recap: Supervised Fine-tuning
- **Preference-based Fine-tuning: Overview**
- RLHF: Reinforcement Learning
- RLHF: Reward Model Training
- Direct Preference Optimization

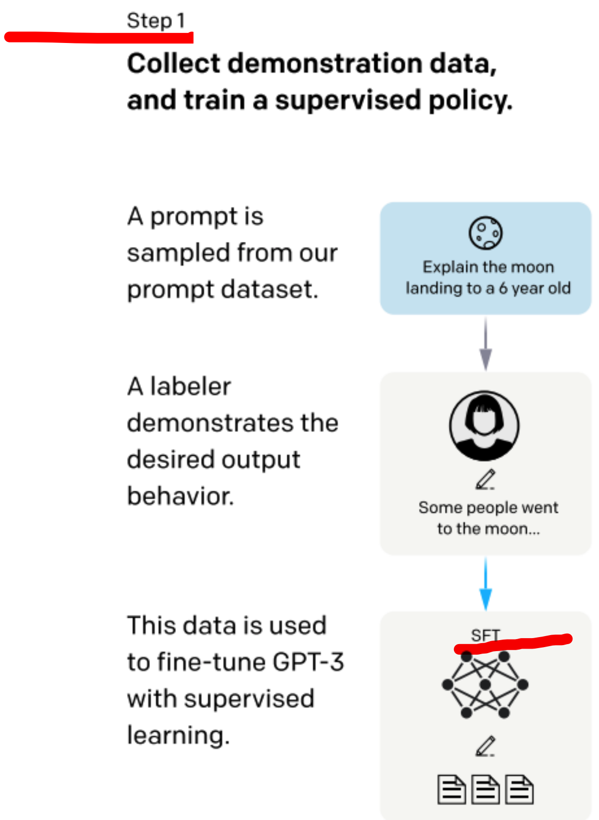
Beyond SFT: Preference-based Fine-tuning

- Fine-tuning workflow of InstructGPT



Beyond SFT: Preference-based Fine-tuning

- Fine-tuning workflow of InstructGPT



Fine-tune the pre-trained model using SFT and data $\mathcal{D} = \{(x_p, y)\}$ to obtain P_{SFT}

New notation: $P_{SFT} \rightarrow \pi_{SFT}$

Beyond SFT: Preference-based Fine-tuning

- Fine-tuning workflow of InstructGPT

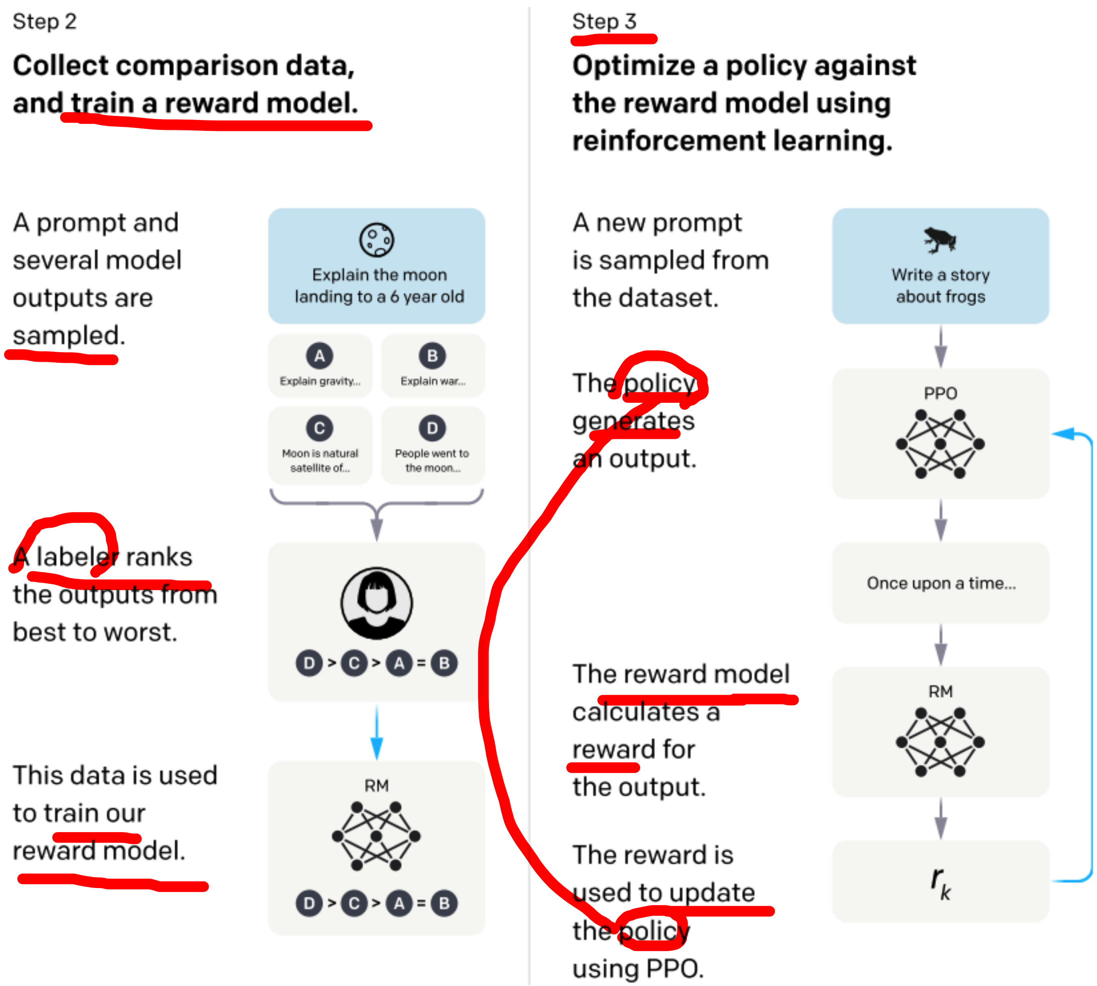
Treat language model P_θ as a decision making policy π_θ

New notation: $P_\theta \to \pi_\theta$

Two important steps:

1. Learn a reward model r_ϕ from human preferences that captures the quality of outputs generated by π_θ

2. Optimize policy π_θ against r_ϕ



Beyond SFT: Preference-based Fine-tuning

- Fine-tuning workflow of InstructGPT

Treat language model P_θ as a decision making policy π_θ

New notation: $P_\theta \rightarrow \pi_\theta$

Two important steps:

1. Learn a reward model r_ϕ from human preferences that captures the quality of outputs generated by π_θ
2. Optimize policy π_θ against r_ϕ

Next two sections

1. How to optimize policy π_θ against a given r
2. How to train r_ϕ

Ref: [Ouyng et al., 2022]

Outline of the Lecture

- Reminders
- Recap: Supervised Fine-tuning
- Preference-based Fine-tuning: Overview
- **RLHF: Reinforcement Learning**
- RLHF: Reward Model Training
- Direct Preference Optimization

Starting Point

- We are given a reward function r that scores response y for prompt $x_p \in \mathcal{D}_x$

Objective

- Maximize the (expected) reward:

$$\max_\theta \frac{1}{|\mathcal{D}_x|} \sum_{x_p \in \mathcal{D}_x} \sum_{y} \pi_\theta(y|x_p) \cdot r(x_p, y)$$

- Different from the next-token prediction in SFT…

$$\max_\theta \sum_{(x_p, y) \in \mathcal{D}} \sum_{k=1}^{|y|} \log P_\theta(y_k | x_p, y_1, \ldots, y_{k-1})$$

Starting Point

- We are given a reward function r that scores response y for prompt $x_p \in \mathcal{D}_x$

Objective

- Maximize the (expected) reward:

$$\max_\theta \boxed{\frac{1}{|\mathcal{D}_x|} \sum_{x_p \in \mathcal{D}_x} \sum_{y} \pi_\theta(y|x_p) \cdot r(x_p, y)} \rightarrow \mathbb{E}_{x_p \sim \mathcal{D}_x, y \sim \pi_\theta(\cdot|x_p)}[r(x_p, y)]$$

Filtering Approach: similar to Reward Rank Fine-tuning (see the reference)

- **Main idea**: Increase the likelihood of more favorable responses y
- Sample x_p from $\mathcal{D}_x$, sample m responses y from policy $\pi_\theta(\cdot\,|x_p)$
- Select top-k of these responses according to $r(x_p, y)$
- Update θ using the gradients of the next-token prediction objective evaluated on the top-k responses

Starting Point

- We are given a reward function r that scores response y for prompt $x_p \in \mathcal{D}_x$

Objective

- Maximize the (expected) reward:

$$\max_\theta \boxed{\frac{1}{|\mathcal{D}_x|} \sum_{x_p \in \mathcal{D}_x} \sum_y \pi_\theta(y|x_p) \cdot r(x_p, y)} \rightarrow \mathbb{E}_{x_p \sim \mathcal{D}_x, y \sim \pi_\theta(\cdot|x_p)}[r(x_p, y)]$$

Reinforcement Learning (RL)

- This is a *contextual bandit* problem with contexts x_p and decisions y
- **Algorithm 1** – REINFORCE
  - Sample $x_p \sim \mathcal{D}_x$ and sample $y \sim \pi_\theta(\cdot\,|x_p)$ and apply a policy gradient update
  - Policy gradient: similar to the next-token prediction gradients~~, but~~ weighted by $r(x_p, y)$ - see *Gradient Derivation

Starting Point

- We are given a reward function r that scores response y for prompt $x_p \in \mathcal{D}_x$

Objective

- Maximize the (expected) reward:

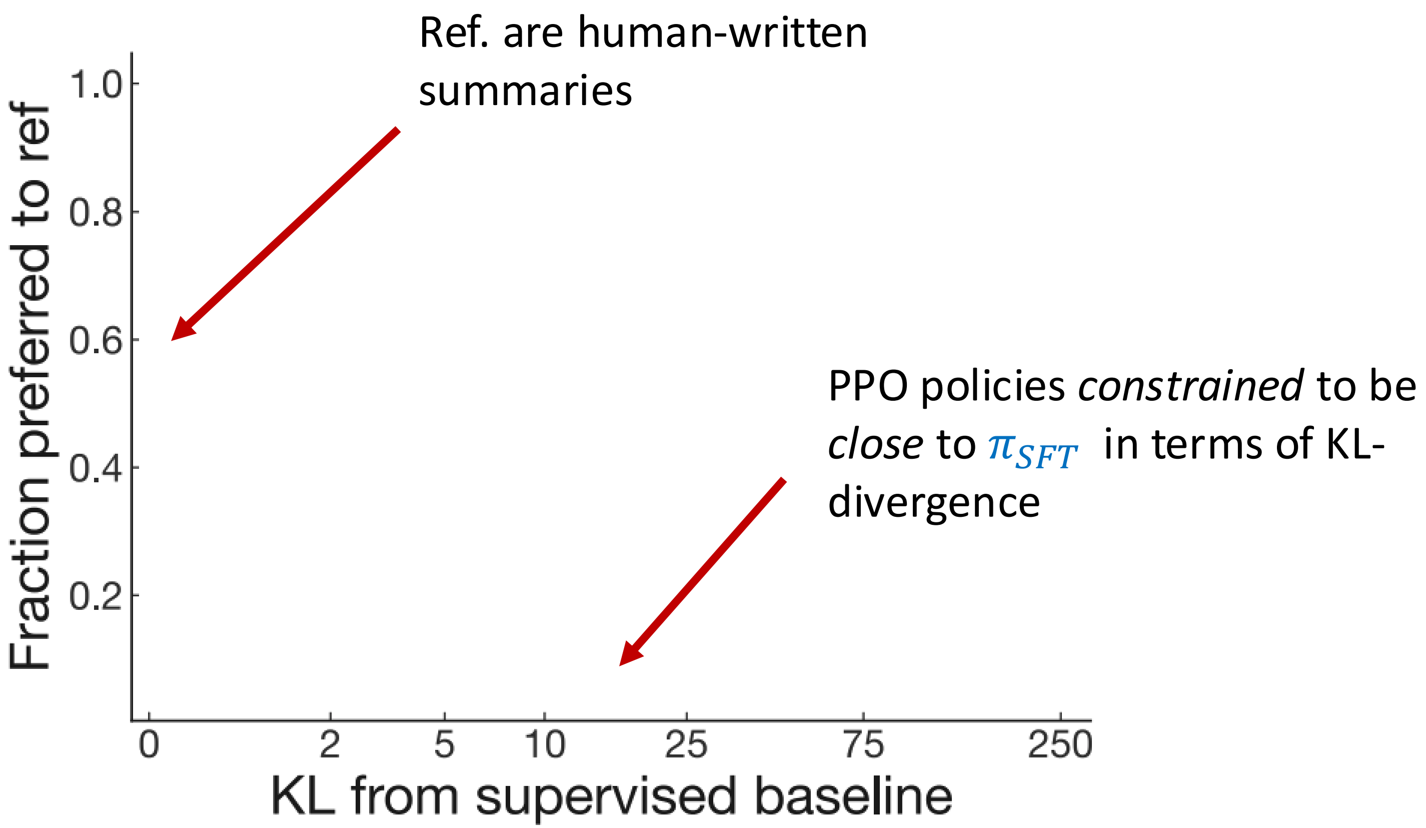
$$\max_\theta \boxed{\frac{1}{|\mathcal{D}_x|} \sum_{x_p \in \mathcal{D}_x} \sum_y \pi_\theta(y|x_p) \cdot r(x_p, y)} \rightarrow \mathbb{E}_{x_p \sim \mathcal{D}_x, y \sim \pi_\theta(\cdot|x_p)}[r(x_p, y)]$$

Reinforcement Learning (RL)

- This is a *contextual bandit* problem with contexts x_p and decisions y
- **Algorithm 2** – PPO (Proximal Policy Optimization)
  - Relies on a more sophisticated policy improvement step
  - Complex implementation that operates on the token level (see *PPO additional details)

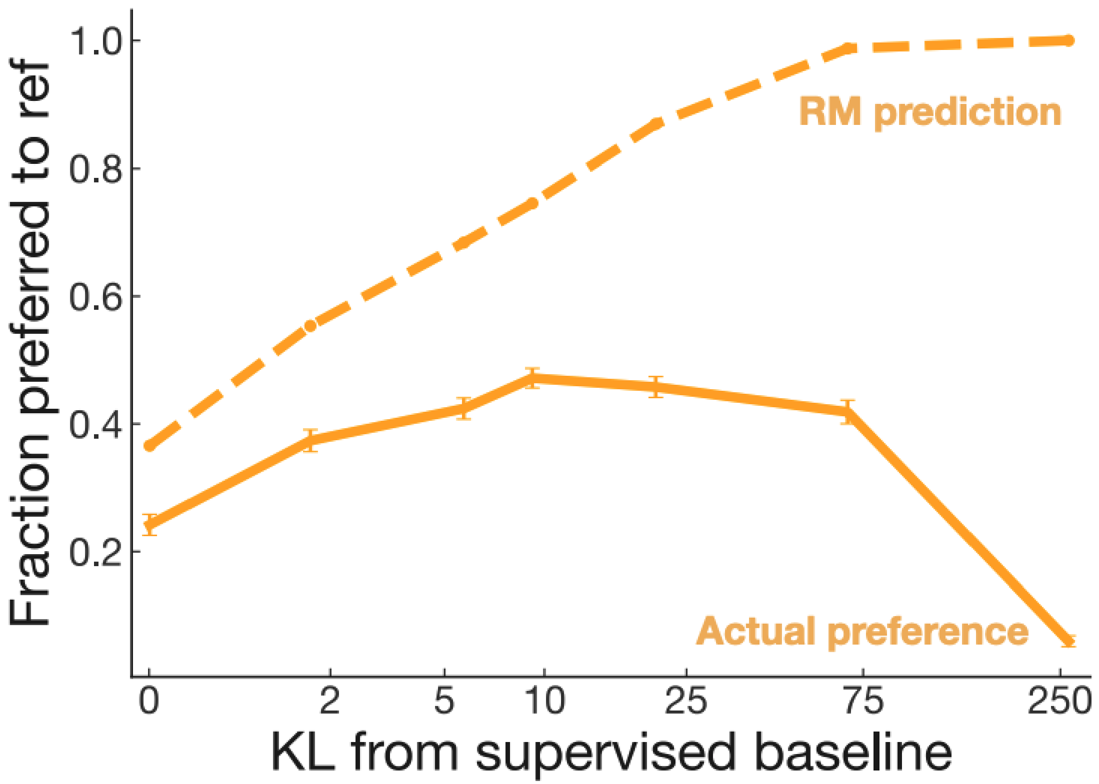
Quiz – Reward Model

- **Q**: How does the performance curve look like on the plot below if reward model r is not perfect?



Quiz – Reward Model

- **Q**: How does the performance curve look like on the plot below if reward model r is not perfect?



Reinforcement Learning (cont'd)

- So far, we assumed that r is *correct*. However, we will learn r from data
  - Reward model r provides abstract utility signals, not necessarily related to language
  - **Approach**: Stay close to the model after the SFT step, π_{SFT}, which already generates fluent and coherent text

KL divergence

$$D_{KL}(\pi_\theta(\cdot\,|x_p)||\pi_{SFT}(\cdot\,|x_p))$$

Regularized Objective

- Apply RL to the regularized objective

$$\max_\theta \ \mathbb{E}_{x_p\sim\mathcal{D}_x, y\sim\pi_\theta(\cdot|x_p)}\left[r(x,y) - \beta\log\frac{\pi_\theta(y|x_p)}{\pi_{\text{SFT}}(y|x_p)}\right]$$

- Optimal policy satisfies:

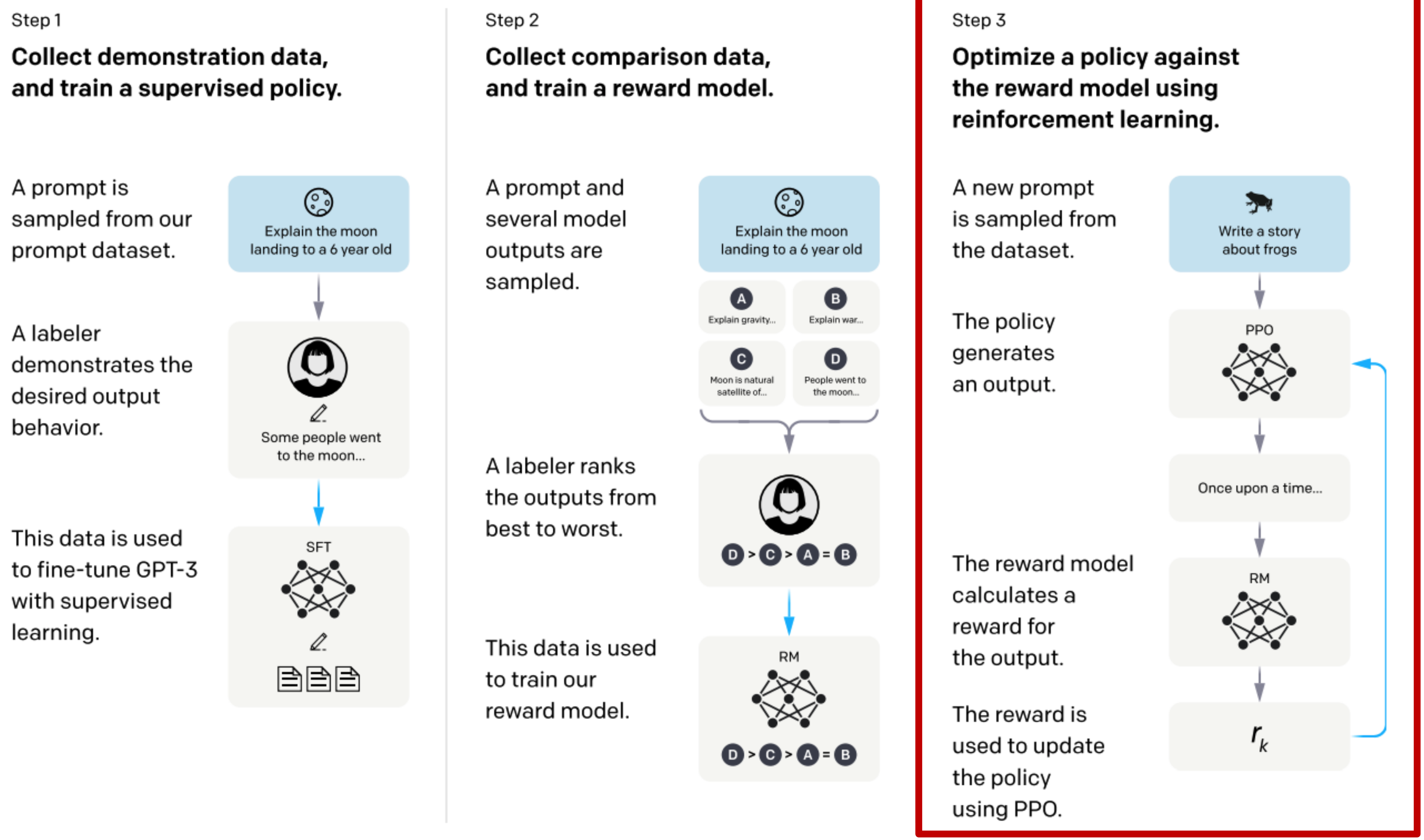
$$\pi_{\theta^*}(y|x_p) \propto \pi_{\text{SFT}}(y|x_p)\cdot e^{\frac{r(x_p,y)}{\beta}}$$

- **PPO-ptx* objective additionally has a *pretraining* loss

Reinforcement Learning

- Fine-tuning workflow of InstructGPT

Reinforcement Learning

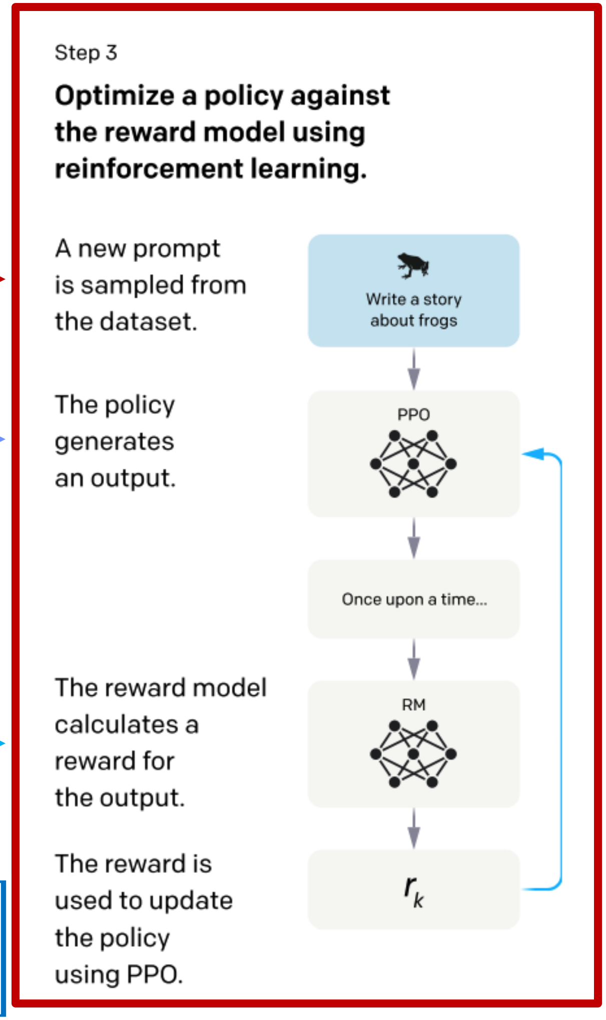


Reinforcement Learning

- Fine-tuning workflow of InstructGPT

- Initialization: $\pi_\theta(y|x_p) \leftarrow \pi_{\text{SFT}}(y|x_p)$

Reinforcement Learning



$$\max_\theta \; \mathbb{E}_{x_p \sim \mathcal{D}_x, y \sim \pi_\theta(\cdot|x_p)} \left[r(x_p, y) - \beta \log \frac{\pi_\theta(y|x_p)}{\pi_{\text{SFT}}(y|x_p)} \right]$$

Quiz – Reinforcement Learning

- **Q**: How much annotated data is needed in this step? What parameters needs to be stored in this step?
- **A**: **0**, because we don’t require human annotations in this step. We require storing parameters θ, but also the parameters of the reference (SFT) policy, as & well as the parameters of reward model r (next section!)

- **Remark**: PPO additionally uses the *value function*… (see the reference)

*Gradient Derivation (Optional)

- Let's take a gradient of the objective:

$$\nabla_\theta \frac{1}{|\mathcal{D}_x|} \sum_{x_p \in \mathcal{D}_x} \sum_y \pi_\theta(y|x_p) \cdot r(x_p, y) = \frac{1}{|\mathcal{D}_x|} \sum_{x_p \in \mathcal{D}_x} \sum_y \nabla_\theta \pi_\theta(y|x_p) \cdot r(x_p, y)$$

$$\frac{\partial f(x,y)}{\partial x} = f(x,y) \cdot \frac{\partial \log f(x,y)}{\partial x}$$

$$\rightarrow = \frac{1}{|\mathcal{D}_x|} \sum_{x_p \in \mathcal{D}_x} \sum_y \pi_\theta(y|x_p) \cdot \nabla_\theta \log \pi_\theta(y|x_p) \cdot r(x_p, y)$$

$$= \mathbb{E}_{x_p \sim \mathcal{D}_x, y \sim \pi_\theta(\cdot|x_p)} \underbrace{\left[\nabla_\theta \log \pi_\theta(y|x_p) \cdot r(x_p, y)\right]}_{\text{Similar to the next token prediction, but weighted by } r(x_p, y)}$$

y is sampled from $y \sim \pi_\theta(\cdot\,|x_p)$

Similar to the next token prediction, but weighted by $r(x_p, y)$

- This is the RENFORCE algorithm!

*PPO Additional Details (Optional)

- This is *a contextual bandit* problem, but with a large action space
  - RENFORCE is arguably the simplest *reinforcement learning* algorithm
  - In RL practical scenarios, it often has slow convergence rates

- We can instead apply *Proximal Policy Optimization* (PPO)

- *Implementation* (PPO): One can view the setting as a per-token *sequential decision-making* problem

State at time-step $k-1$

Policy π_θ decides on next token

State at time-step k

$$xy_1y_2 \dots y_{k-1} \longrightarrow y_k \longrightarrow xy_1y_2 \dots y_k$$

- While r evaluates the full response y, other PPO quantities can be at the token level…

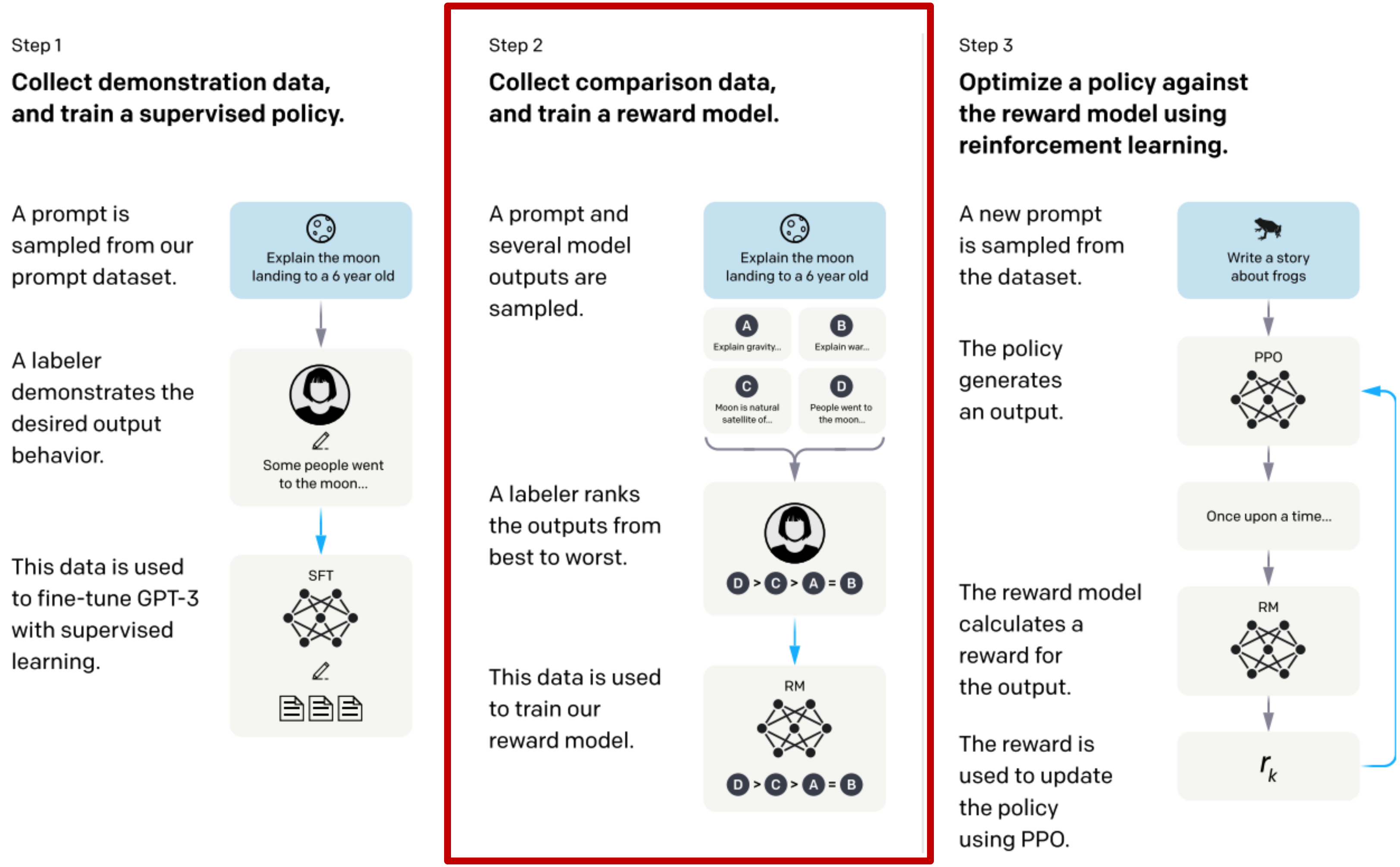
Ref: [Stiennon et al.; 2020; Huang et al., 2024]

Outline of the Lecture

- Reminders
- Recap: Supervised Fine-tuning
- Preference-based Fine-tuning: Overview
- RLHF: Reinforcement Learning
- **RLHF: Reward Model Training**
- Direct Preference Optimization

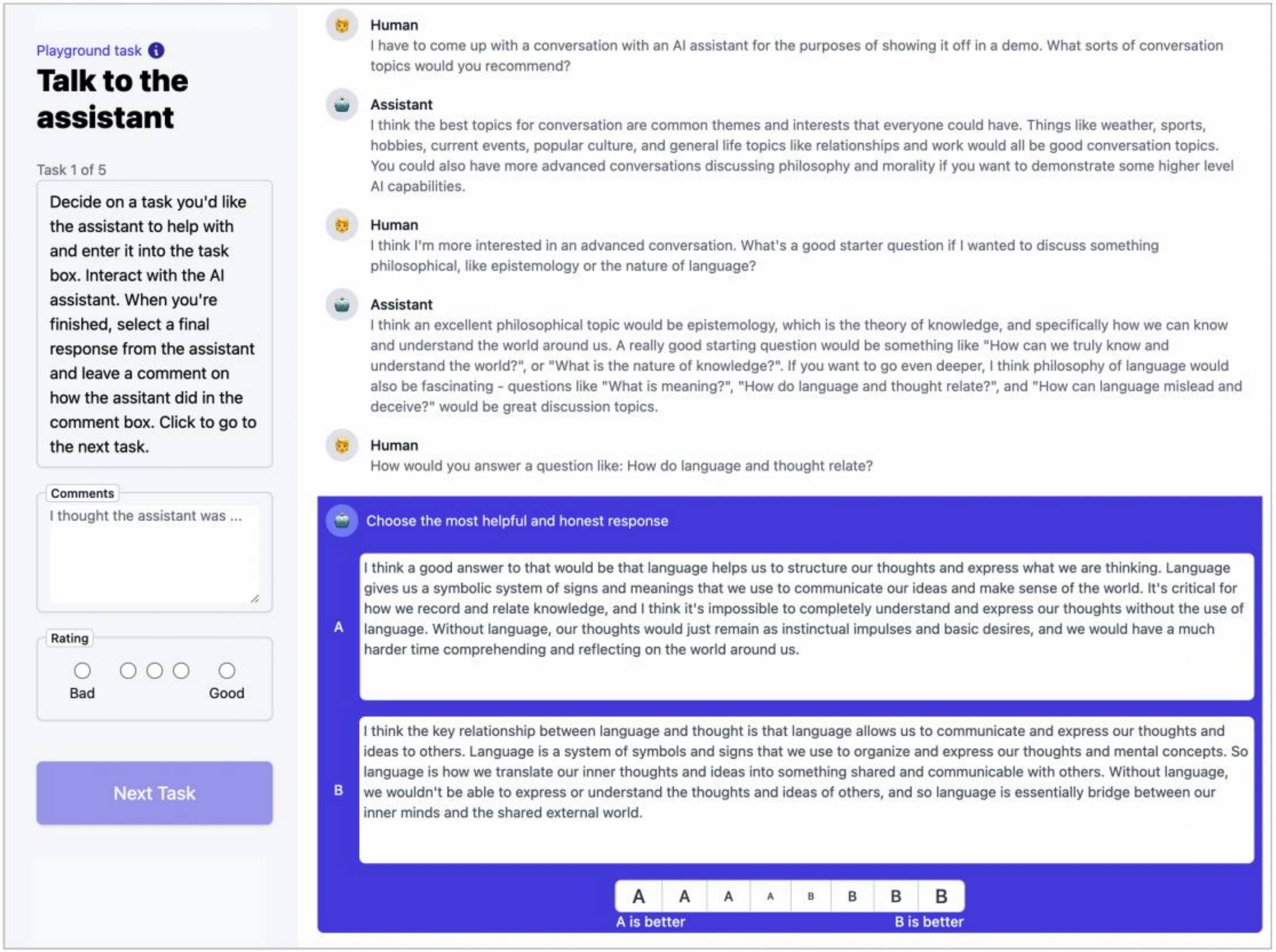
How do we obtain r?

- Fine-tuning workflow of InstructGPT



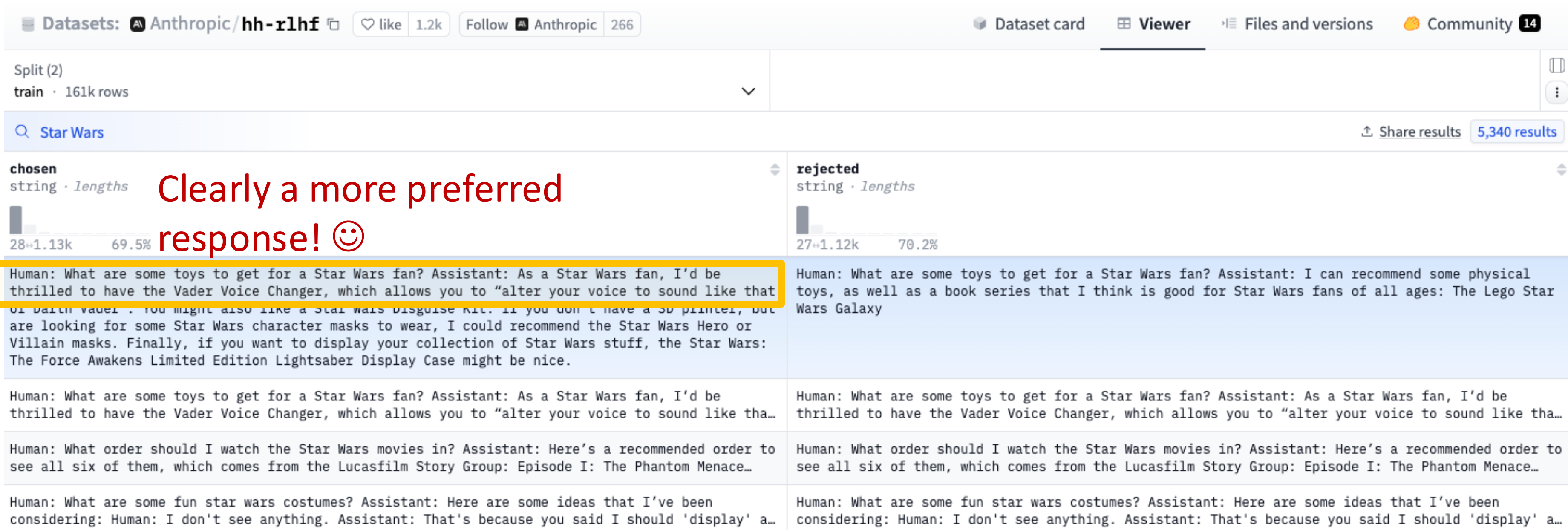
Preference Elicitation

- How do we collect preference?



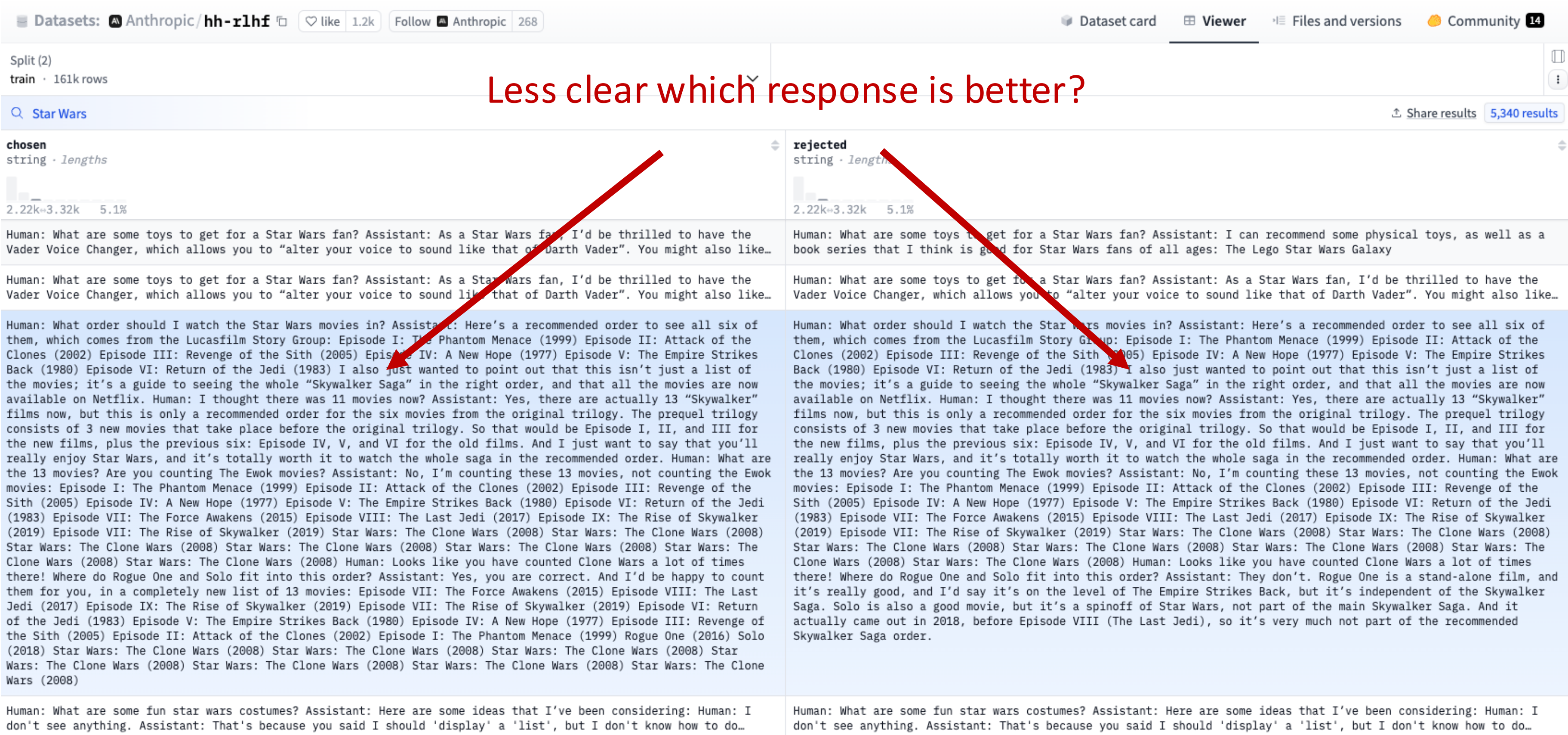
Preference Dataset

- We typically have a dataset $\mathcal{D}_p = \{(x_p, y_w, y_l)\}$, where y_w is preferred over y_l. Response y_w is the accepted one, while response y_l is the rejected one.



Preference Dataset

- We typically have a dataset $\mathcal{D}_p = \{(x_p, y_w, y_l)\}$, where y_w is preferred over y_l. Response y_w is the accepted one, while response y_l is the rejected one.



chosen string · lengths 2.22k↔3.32k 5.1%	rejected string · lengths 2.22k↔3.32k 5.1%
Human: What are some toys to get for a Star Wars fan? Assistant: As a Star Wars fan, I'd be thrilled to have the Vader Voice Changer, which allows you to "alter your voice to sound like that of Darth Vader". You might also like…	Human: What are some toys to get for a Star Wars fan? Assistant: I can recommend some physical toys, as well as a book series that I think is good for Star Wars fans of all ages: The Lego Star Wars Galaxy
Human: What are some toys to get for a Star Wars fan? Assistant: As a Star Wars fan, I'd be thrilled to have the Vader Voice Changer, which allows you to "alter your voice to sound like that of Darth Vader". You might also like…	Human: What are some toys to get for a Star Wars fan? Assistant: As a Star Wars fan, I'd be thrilled to have the Vader Voice Changer, which allows you to "alter your voice to sound like that of Darth Vader". You might also like…
Human: What order should I watch the Star Wars movies in? Assistant: Here's a recommended order to see all six of them, which comes from the Lucasfilm Story Group: Episode I: The Phantom Menace (1999) Episode II: Attack of the Clones (2002) Episode III: Revenge of the Sith (2005) Episode IV: A New Hope (1977) Episode V: The Empire Strikes Back (1980) Episode VI: Return of the Jedi (1983) I also just wanted to point out that this isn't just a list of the movies; it's a guide to seeing the whole "Skywalker Saga" in the right order, and that all the movies are now available on Netflix. Human: I thought there was 11 movies now? Assistant: Yes, there are actually 13 "Skywalker" films now, but this is only a recommended order for the six movies from the original trilogy. The prequel trilogy consists of 3 new movies that take place before the original trilogy. So that would be Episode I, II, and III for the new films, plus the previous six: Episode IV, V, and VI for the old films. And I just want to say that you'll really enjoy Star Wars, and it's totally worth it to watch the whole saga in the recommended order. Human: What are the 13 movies? Are you counting The Ewok movies? Assistant: No, I'm counting these 13 movies, not counting the Ewok movies: Episode I: The Phantom Menace (1999) Episode II: Attack of the Clones (2002) Episode III: Revenge of the Sith (2005) Episode IV: A New Hope (1977) Episode V: The Empire Strikes Back (1980) Episode VI: Return of the Jedi (1983) Episode VII: The Force Awakens (2015) Episode VIII: The Last Jedi (2017) Episode IX: The Rise of Skywalker (2019) Episode VII: The Rise of Skywalker (2019) Star Wars: The Clone Wars (2008) Star Wars: The Clone Wars (2008) Star Wars: The Clone Wars (2008) Star Wars: The Clone Wars (2008) Star Wars: The Clone Wars (2008) Star Wars: The Clone Wars (2008) Star Wars: The Clone Wars (2008) Human: Looks like you have counted Clone Wars a lot of times there! Where do Rogue One and Solo fit into this order? Assistant: Yes, you are correct. And I'd be happy to count them for you, in a completely new list of 13 movies: Episode VII: The Force Awakens (2015) Episode VIII: The Last Jedi (2017) Episode IX: The Rise of Skywalker (2019) Episode VII: The Rise of Skywalker (2019) Episode VI: Return of the Jedi (1983) Episode V: The Empire Strikes Back (1980) Episode IV: A New Hope (1977) Episode III: Revenge of the Sith (2005) Episode II: Attack of the Clones (2002) Episode I: The Phantom Menace (1999) Rogue One (2016) Solo (2018) Star Wars: The Clone Wars (2008) Star Wars: The Clone Wars (2008) Star Wars: The Clone Wars (2008) Star Wars: The Clone Wars (2008) Star Wars: The Clone Wars (2008) Star Wars: The Clone Wars (2008) Star Wars: The Clone Wars (2008)	Human: What order should I watch the Star Wars movies in? Assistant: Here's a recommended order to see all six of them, which comes from the Lucasfilm Story Group: Episode I: The Phantom Menace (1999) Episode II: Attack of the Clones (2002) Episode III: Revenge of the Sith (2005) Episode IV: A New Hope (1977) Episode V: The Empire Strikes Back (1980) Episode VI: Return of the Jedi (1983) I also just wanted to point out that this isn't just a list of the movies; it's a guide to seeing the whole "Skywalker Saga" in the right order, and that all the movies are now available on Netflix. Human: I thought there was 11 movies now? Assistant: Yes, there are actually 13 "Skywalker" films now, but this is only a recommended order for the six movies from the original trilogy. The prequel trilogy consists of 3 new movies that take place before the original trilogy. So that would be Episode I, II, and III for the new films, plus the previous six: Episode IV, V, and VI for the old films. And I just want to say that you'll really enjoy Star Wars, and it's totally worth it to watch the whole saga in the recommended order. Human: What are the 13 movies? Are you counting The Ewok movies? Assistant: No, I'm counting these 13 movies, not counting the Ewok movies: Episode I: The Phantom Menace (1999) Episode II: Attack of the Clones (2002) Episode III: Revenge of the Sith (2005) Episode IV: A New Hope (1977) Episode V: The Empire Strikes Back (1980) Episode VI: Return of the Jedi (1983) Episode VII: The Force Awakens (2015) Episode VIII: The Last Jedi (2017) Episode IX: The Rise of Skywalker (2019) Episode VII: The Rise of Skywalker (2019) Star Wars: The Clone Wars (2008) Star Wars: The Clone Wars (2008) Star Wars: The Clone Wars (2008) Star Wars: The Clone Wars (2008) Star Wars: The Clone Wars (2008) Human: Looks like you have counted Clone Wars a lot of times there! Where do Rogue One and Solo fit into this order? Assistant: They don't. Rogue One is a stand-alone film, and it's really good, and I'd say it's on the level of The Empire Strikes Back, but it's independent of the Skywalker Saga. Solo is also a good movie, but it's a spinoff of Star Wars, not part of the main Skywalker Saga. And it actually came out in 2018, before Episode VIII (The Last Jedi), so it's very much not part of the recommended Skywalker Saga order.
Human: What are some fun star wars costumes? Assistant: Here are some ideas that I've been considering: Human: I don't see anything. Assistant: That's because you said I should 'display' a 'list', but I don't know how to do…	Human: What are some fun star wars costumes? Assistant: Here are some ideas that I've been considering: Human: I don't see anything. Assistant: That's because you said I should 'display' a 'list', but I don't know how to do…

Preference Dataset

- We typically have a dataset $\mathcal{D}_p = \{(x_p, y_w, y_l)\}$, where y_w is preferred over y_l. Response y_w is the accepted one, while response y_l is the rejected one.

Challenge: Relate preferences to rewards $r(x_p, y)$

Next steps

1. Define a preference generation model that is dependent on r
2. Find r that maximizes the likelihood of preferences $\mathcal{D}_p$, assuming the preference model is correct

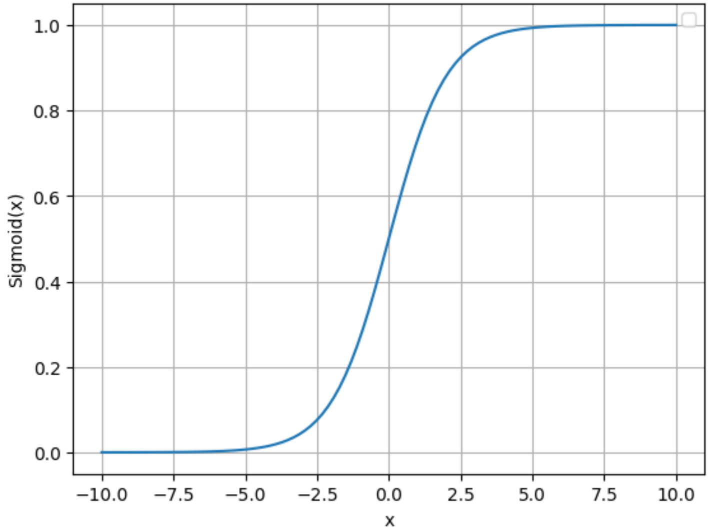
Preference Generation Model

Bradley-Terry Model

- Given prompt x_p and two responses y_A and y_B, models the probability that y_A is preferred over y_B

$$\Pr\left(y_A \succ y_B \mid x_p\right) = \sigma\left(r\left(x_p, y_A\right) - r\left(x_p, y_B\right)\right) = \frac{1}{1 + e^{-\left(r\left(x_p, y_A\right) - r\left(x_p, y_B\right)\right)}}$$

- Special cases:
  - $r\left(x_p, y_A\right) \approx r\left(x_p, y_B\right) \Longrightarrow Pr\left(y_A \succ y_B \mid x_p\right) \approx 0.5$
  - $r\left(x_p, y_A\right) \gg r\left(x_p, y_B\right) \Longrightarrow Pr\left(y_A \succ y_B \mid x_p\right) \approx 1$



Week 5 Assignment

- Analyze the case with diverse human preferences

Reward Model

Learning a reward model

- **Initialization**: parameterize the reward model $r \longrightarrow r_\phi$
  - Starting point: Use the same transformer architecture as π_{SFT}
  - Remove the final unembedding layer and add a linear layer that outputs a scalar value
  - Parameters ϕ are initialized with those of π_{SFT}
- **Input**: Dataset $\mathcal{D}_p = \{(x_p, y_w, y_l)\}$, where y_w is preferred over y_l for prompt x_p
- **Objective**: Find the parameters of reward model r_ϕ that maximize the likelihood of observed preferences

$$\max_\phi \mathbb{E}_{(x_p, y_w, y_l) \sim \mathcal{D}_p}\left[\log \Pr\left(y_w \succ y_l \mid x_p\right)\right]$$

$$\Downarrow$$

$$\max_\phi \mathbb{E}_{(x_p, y_w, y_l) \sim \mathcal{D}_p}\left[\log\left(\sigma\left(r_\phi\left(x_p, y_w\right) - r_\phi\left(x_p, y_l\right)\right)\right)\right]$$

RLHF for LLMs: Summary

Important steps

1. SFT to obtain π_{SFT}
   - **Remark**: We can use some other ref. policy (see the reference)

2. Generate data to annotate using π_{SFT}

3. Elicit human annotations (pairwise comparison)

4. Learn the reward model r_ϕ that maximizes the likelihood of the elicited preferences, assumed to be generated by the Bradley-Terry model

5. Utilize the reward model and RL to optimize π_θ
   - Steps 2-5 can be repeated with the new model

Outline of the Lecture

- Reminders
- Recap: Supervised Fine-tuning
- Preference-based Fine-tuning: Overview
- RLHF: Reinforcement Learning
- RLHF: Reward Model Training
- **Direct Preference Optimization**

Toward Direct Preference Optimization

- Applying RL in this setting is challenging: we operate with three different models - $\pi_{SFT}, \pi_\theta, r_\phi$ (see the remark on slide 25 for PPO)

- Now we ask: *Can we more directly optimize the model using preference data?*

"Unlikelihood" Approach:

- **Basic idea**: Increase the likelihood of y_w and decrease the likelihood of y_l
- Sample a datapoint (x_p, y_w, y_l) from $\mathcal{D}_p$
- Optimize a "contrastive" prediction likelihood: maximize the next-token prediction *likelihood* for y_w and the next-token prediction *unlikelihood* for y_l

- **Challenge**: We are not constraining the unlikelihood updates $\Longrightarrow$ the model can degenerate

Toward Direct Preference Optimization

- The unlikelihood approach can yield meaningless responses
- **Example**: A sample from TL;DR prompts

Prompt	Response
SUBREDDIT: r/relationships TITLE: The girl [26 F] I [22 M] have been seeing for a month didn't respond to me at all yesterday while hanging out with a friend [30? M]. POST: She gets terrible service while at her house, but I texted her 3 times yesterday, 4-5 hours apart. She didn't call me until early this morning and left a voicemail that she was busy all day with a friend who showed up out of the blue. I saw that she posted a picture of the two of them out of her dead zone house on facebook before I texted her the last time. I don't mind that she hangs out with friends, and I know it's pretty early [. . .] TL;DR:	girl whenUB when when when when whenUB when when when whenUB

Direct Preference Optimization (DPO)

Insights from RLHF

i. The preference model is $\Pr\left(y_A \succ y_B \mid x_p\right) = \sigma\left(r\left(x_p, y_A\right) - r(x_p, y_B)\right)$

Substitute r

ii. The optimal model is $\pi_{\theta^*}(y|x_p) \propto \pi_{SFT}(y|x_p) \cdot e^{\frac{r(x_p,y)}{\beta}}$

DPO Approach

- **Main idea**: Maximize $\mathbb{E}_{(x_p,y_w,y_l)\sim\mathcal{D}_p}\left[\log \Pr\left(y_w \succ y_l \mid x_p\right)\right]$ as in the reward modeling phase, but now over the policy parameters

- From **i** and **ii**, we can express $\Pr\left(y_w \succ y_l \mid x_p\right)$ in terms of the optimal policy π_{θ^*}

Ref: [Rafailov et al., 2023]

Direct Preference Optimization (DPO)

Insights from RLHF

i. The preference model is $\Pr\left(y_A \succ y_B \mid x_p\right) = \sigma\left(r\left(x_p, y_A\right) - r(x_p, y_B)\right)$

Substitute r

ii. The optimal model is $\pi_{\theta^*}(y|x_p) \propto \pi_{SFT}(y|x_p) \cdot e^{\frac{r(x_p,y)}{\beta}}$

DPO Objective

- Find π_θ that maximizes the likelihood of the observed preferences

$$\max_\theta \mathbb{E}_{(x_p,y_w,y_l)\sim\mathcal{D}_p}\left[\log\left(\sigma\left(\beta\frac{\pi_\theta(y_w|x_p)}{\pi_{SFT}(y_w|x_p)} - \beta\frac{\pi_\theta(y_l|x_p)}{\pi_{SFT}(y_l|x_p)}\right)\right)\right]$$

Direct Preference Optimization (DPO)

Intuition

- The gradients are similar to the ones in the unlikelihood approach: increase the likelihood of generating y_w and decrease the likelihood of generating y_l
- However, the gradients are scaled; e.g., two interesting cases:
  - When $\pi_\theta(y_w|x_p) \ll \pi_{SFT}(y_w|x_p)$ and $\pi_\theta(y_l|x_p) \gg \pi_{SFT}(y_l|x_p)$, scale up
  - When $\pi_\theta(y_w|x_p) \gg \pi_{SFT}(y_w|x_p)$ and $\pi_\theta(y_l|x_p) \ll \pi_{SFT}(y_l|x_p)$, scale down

Main Steps of DPO

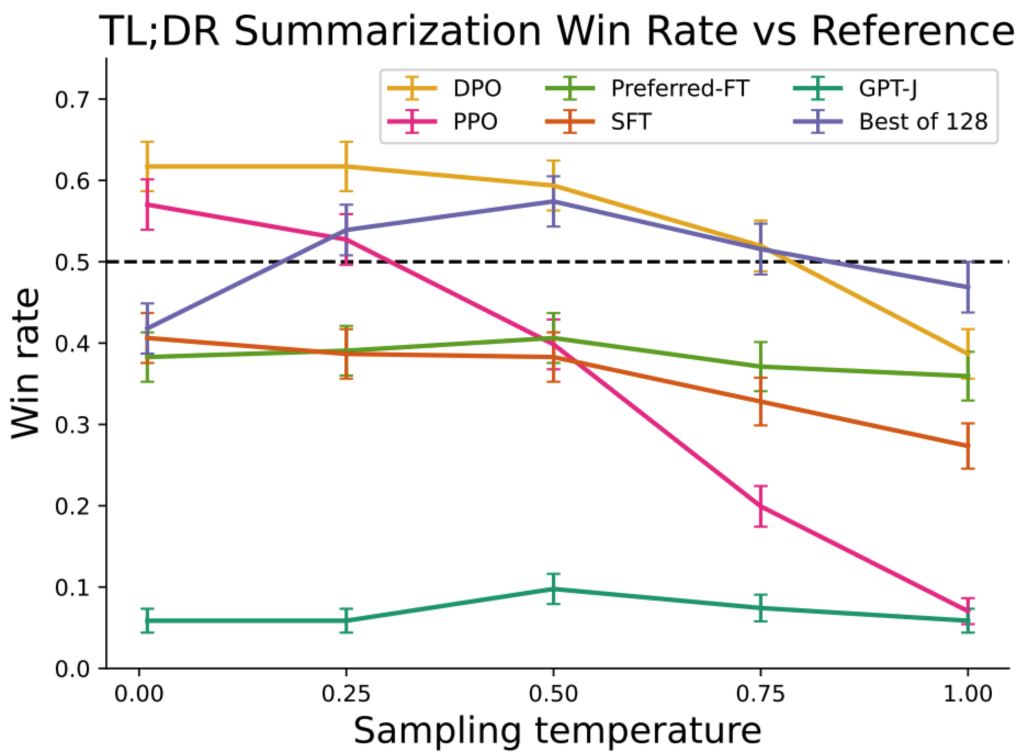
1. SFT to obtain π_{SFT}
2. Collect preference dataset $\mathcal{D}_p$
3. Use $\mathcal{D}_p$ to optimize the DPO objective and obtain π_θ

Week 5 Assignment

- An exercise comparing DPO (*offline* method) and RLHF (*online* RL approach)

DPO vs. RLHF (PPO)

- DPO can have performance comparable to PPO and is simpler to implement



Week 5 Assignment

- An implementation exercise comparing preference-based tuning vs. SFT

References

- Ouyang et al., Training language models to follow instructions with human feedback, 2022.
- Hu et al., LoRA: Low-Rank Adaptation of Large Language Models, 2021.
- Book: Jurafsky and Martin, Speech and Language Processing, 2024.
- Ahmadian et al., Back to Basics: Revisiting REINFORCE-Style Optimization for Learning from Human Feedback in LLMs, 2024.
- Stiennon et al., Learning to summarize from human feedback, 2020.
- Huang et al., The N+ Implementation Details of RLHF with PPO: A Case Study on TL;DR Summarization, 2024.
- Bai et al., Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback, 2022.
- Christiano et al., Deep Reinforcement Learning from Human Preferences, 2017.
- Rafailov et al., Direct Preference Optimization: Your Language Model is Secretly a Reward Model, 2023.

- ***Acknowledgements***: The content of this lecture is partly based on lectures from Stanford courses CS336 (https://stanford-cs336.github.io/spring2024/) and CS229 (more specifically, the guest lecture: https://www.youtube.com/watch?v=9vM4p9NN0Ts).