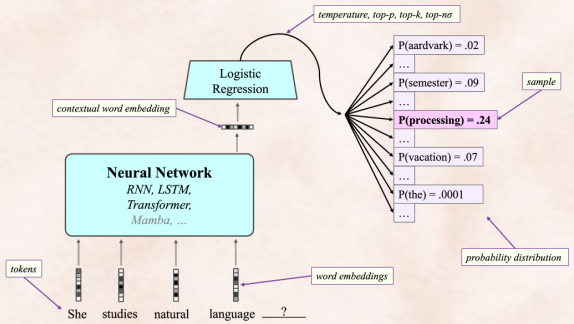
ITCS 6101/8101: Natural Language Processing

Post-Training for Alignment with User Intent



Razvan C. Bunescu

Department of Computer Science @ CCI

rbunescu@charlotte.edu

Language Modeling ≠ Assisting Users

PROMPT *Explain the moon landing to a 6 year old in a few sentences.*

COMPLETION GPT-3

```
Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.
```

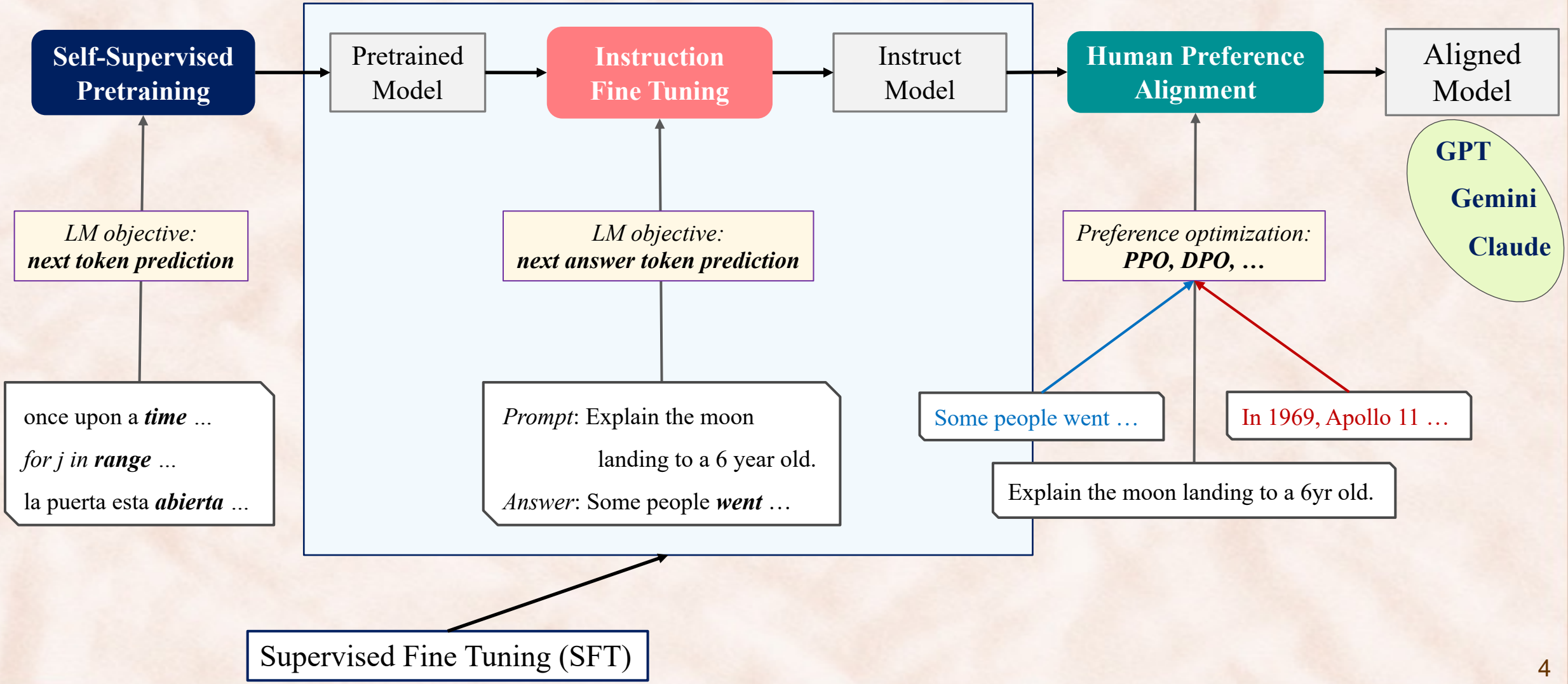
Language models are not *aligned* with user intent [Ouyang et al., 2022].

Language Modeling ≠ Assisting Users

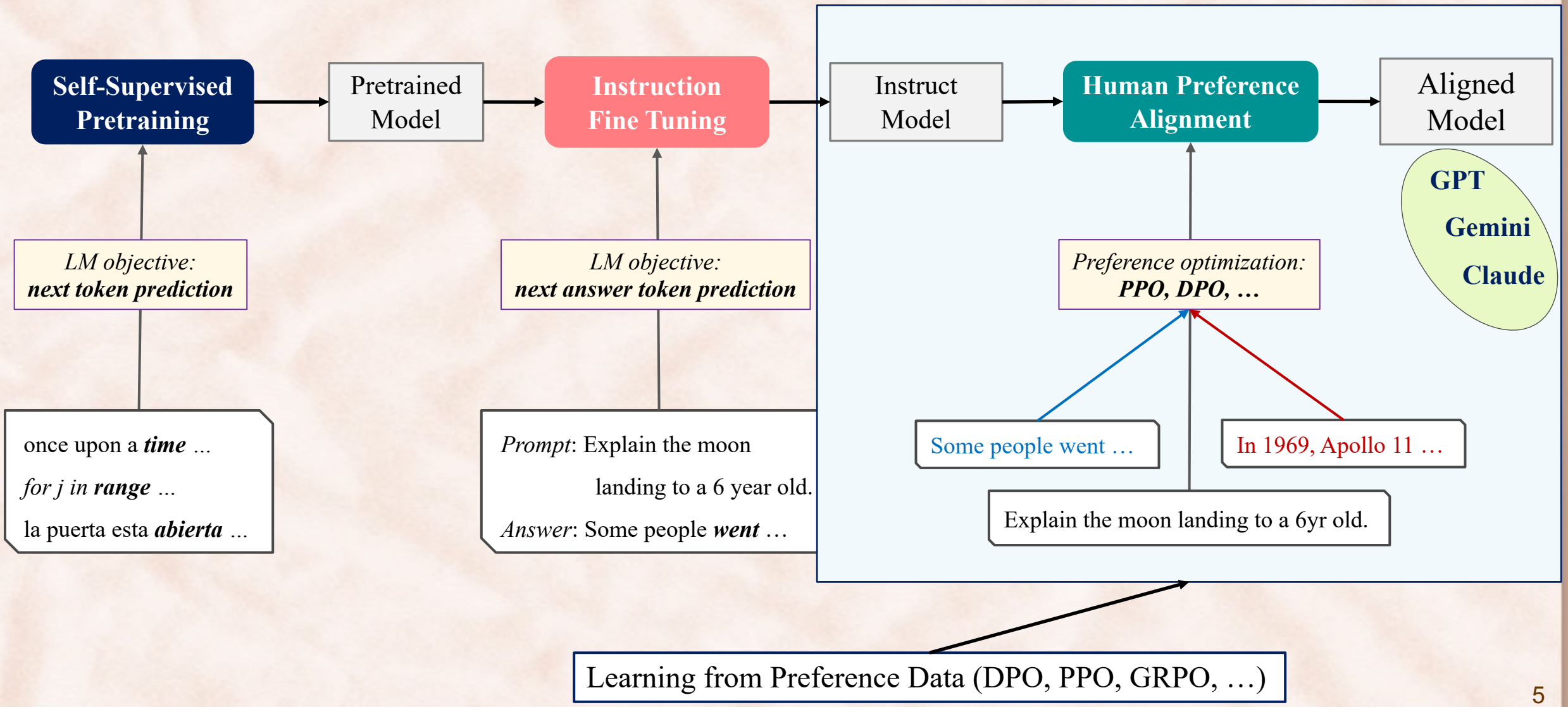
PROMPT	*Explain the moon landing to a 6 year old in a few sentences.*
COMPLETION	**Human** A giant rocket ship blasted off from Earth carrying astronauts to the moon. The astronauts landed their spaceship on the moon and walked around exploring the lunar surface. Then they returned safely back to Earth, bringing home moon rocks to show everyone.

Language models are not *aligned* with user intent [Ouyang et al., 2022].
Finetuning to the rescue!

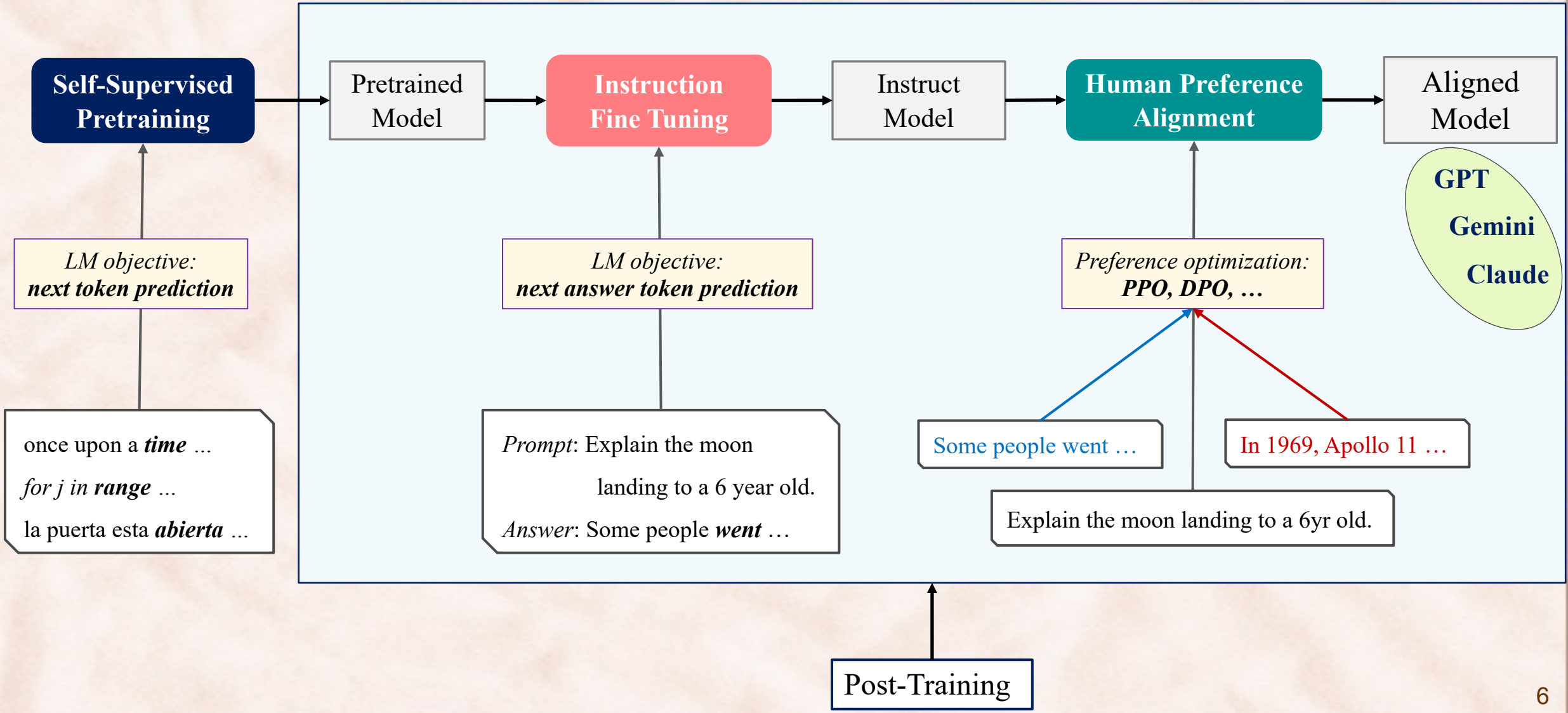
Training LLMs: Pretraining, Instruction Tuning, Alignment



Training LLMs: Pretraining, Instruction Tuning, Alignment



Training LLMs: Pretraining, Instruction Tuning, Alignment



Methods Used During LLM Training

Pre-Training

Self-supervised Learning (SSL)

Unlabeled Text Corpus

Common Crawl

GitHub

>>2T tokens

"I like cats"

$$\min_{\pi} -\log \pi(\mathrm{I}) - \log \pi(\mathrm{like} \mid \mathrm{I}) - \log \pi(\mathrm{cats} \mid \mathrm{I\ like})$$

Methods Used During LLM Training

Pre-Training

Self-supervised Learning (SSL)

Unlabeled Text Corpus

>>2T tokens

"*I like cats*"

$$\min_{\pi} -\log \pi\,(\mathrm{I}) - \log \pi\,(\mathrm{like} \mid \mathrm{I}) - \log \pi\,(\mathrm{cats} \mid \mathrm{I\ like})$$

Post-training Method 1: Supervised Fine-tuning (SFT)

(Supervised / Imitation Learning)

Labeled Prompt-Response Pairs

Prompt: Explain LLM to me
Response: LLM is …

~1K-1B tokens

$$\min_{\pi} -\log \pi\,(\mathrm{Response} \mid \mathrm{Prompt})$$

Methods Used During LLM Training

Post-training Method 2: Direct Preference Optimization (DPO)

Prompt + Good and Bad Responses

Prompt: Explain LLM to me
Good Response: LLM is …
Bad Response: Sorry …

~1K-1B tokens

$$\min_{\pi} -\log \sigma\left(\beta\left(\log \frac{\pi(\text{Good R} \mid \text{Prompt})}{\pi_{\text{ref}}(\text{Good R} \mid \text{Prompt})} - \log \frac{\pi(\text{Bad R} \mid \text{Prompt})}{\pi_{\text{ref}}(\text{Bad R} \mid \text{Prompt})}\right)\right)$$

Methods Used During LLM Training

Post-training Method 2: Direct Preference Optimization (DPO)

Prompt + Good and Bad Responses

Prompt: Explain LLM to me
Good Response: LLM is …
Bad Response: Sorry …

~1K-1B tokens

$$\min_{\pi} -\log \sigma\left(\beta\left(\log \frac{\pi(\text{Good R} \mid \text{Prompt})}{\pi_{\text{ref}}(\text{Good R} \mid \text{Prompt})} - \log \frac{\pi(\text{Bad R} \mid \text{Prompt})}{\pi_{\text{ref}}(\text{Bad R} \mid \text{Prompt})}\right)\right)$$

Post-training Method 3: Online Reinforcement Learning

Prompt + Reward Function

Prompt: Explain LLM to me
Response: LLM is …
Reward: 1.9

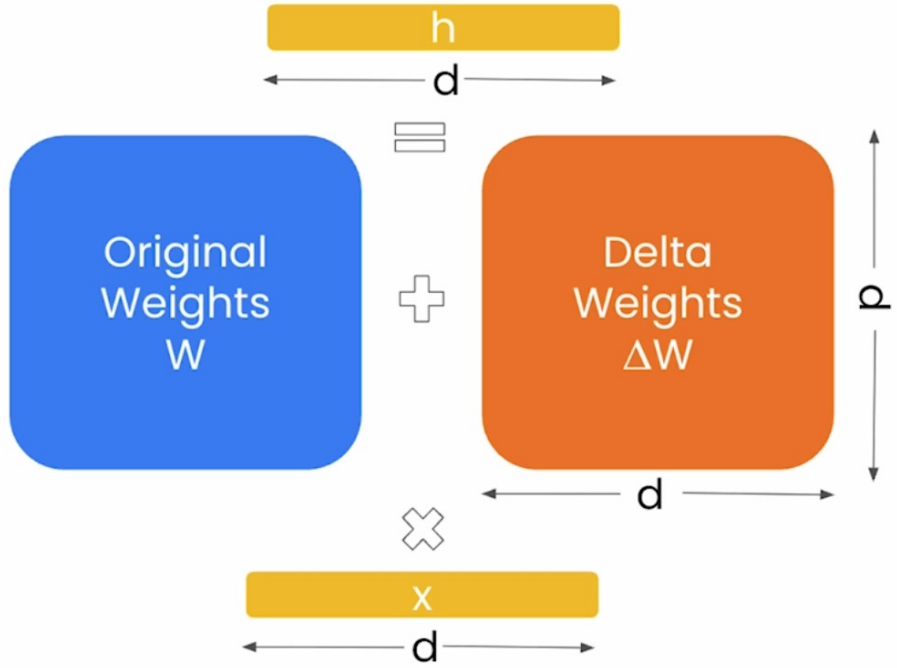
~1K-10M prompts

$$\max_{\pi} \text{Reward}(\text{Prompt}, \text{Response}(\pi))$$

Full Fine Tuning vs. Parameter Efficient Fine Tuning (PEFT)

$$h = (W + \Delta W)x$$

$$W, \Delta W \in \mathbb{R}^{a \times d}, h, x \in \mathbb{R}^{d \times 1}$$

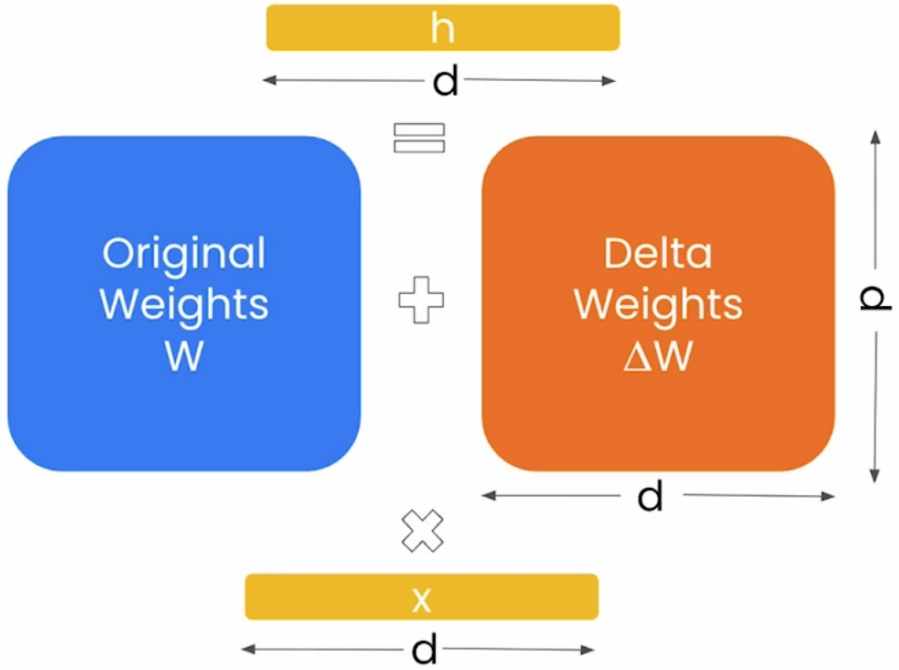


[1] Biderman, Dan, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King et al. "Lora learns less and forgets less." *arXiv preprint arXiv:2405.09673* (2024).

Full Fine Tuning vs. Parameter Efficient Fine Tuning (PEFT)

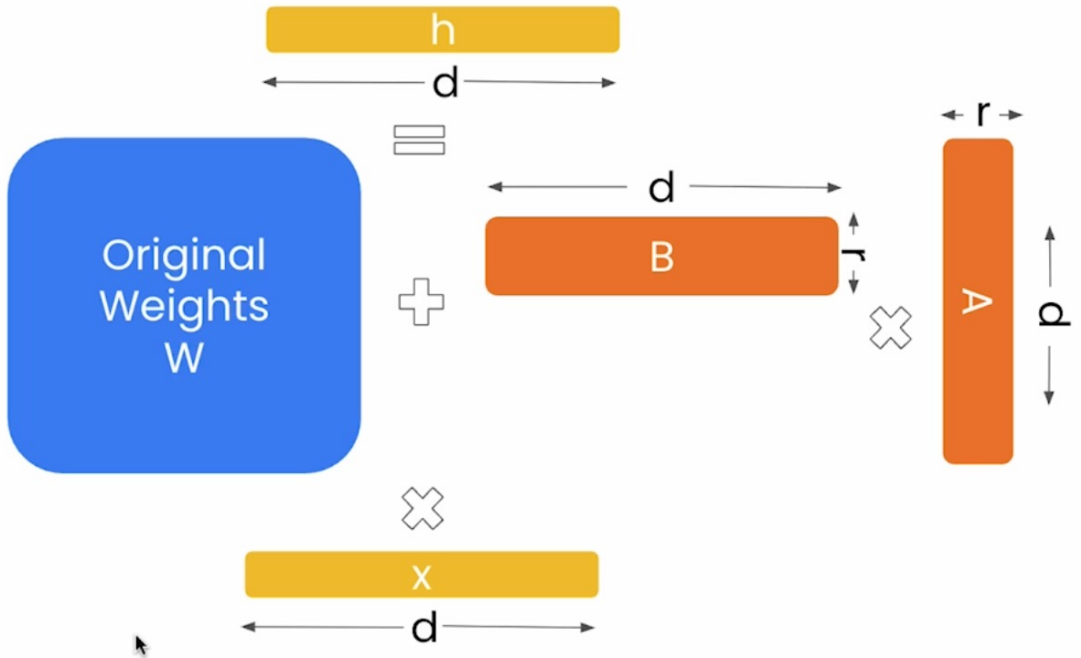
$$h = (W + \Delta W)x$$

$$W, \Delta W \in \mathbb{R}^{d \times d}, h, x \in \mathbb{R}^{d \times 1}$$



$$h = (W + BA)x$$

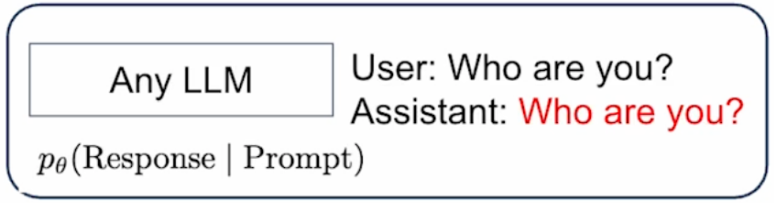
$$B \in \mathbb{R}^{d \times r}, A \in \mathbb{R}^{r \times d}$$



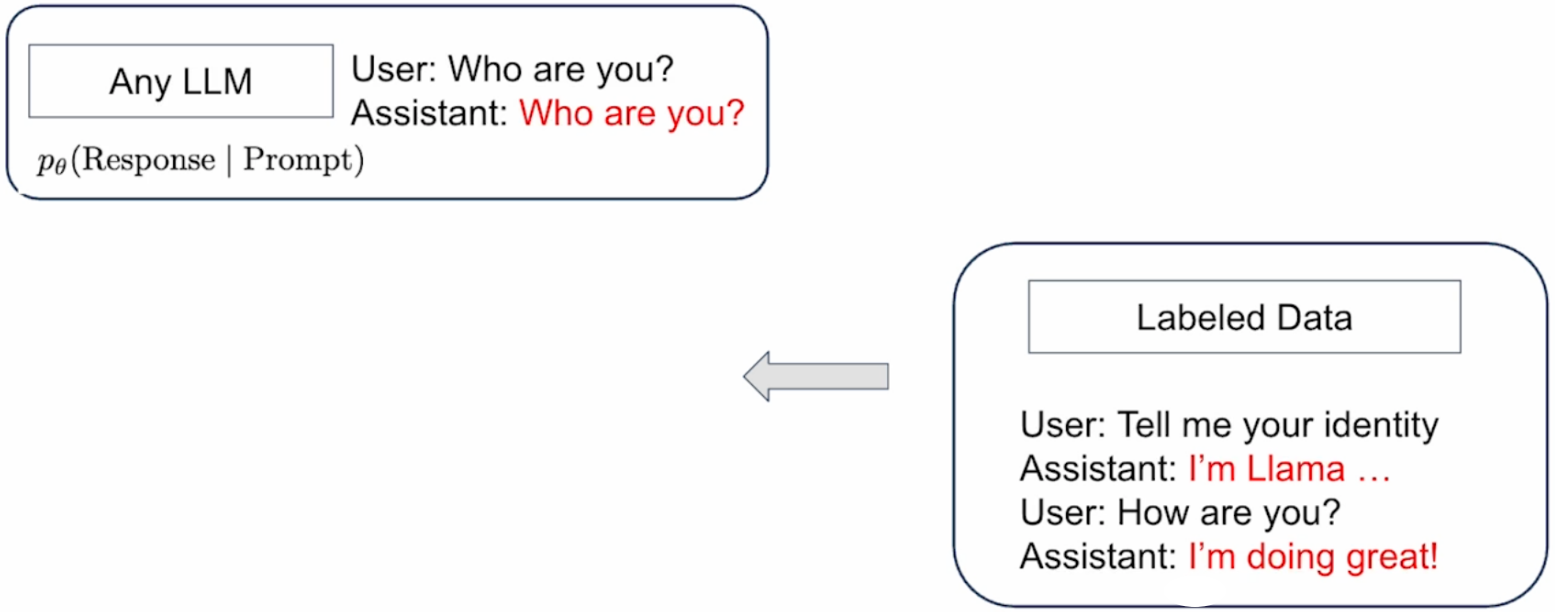
Both full-finetuning and PEFT can be used in any of the post-training methods.
PEFT like Lora saves memory, learns less while forgets less [1]

[1] Biderman, Dan, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King et al. "Lora learns less and forgets less." *arXiv preprint arXiv:2405.09673* (2024).

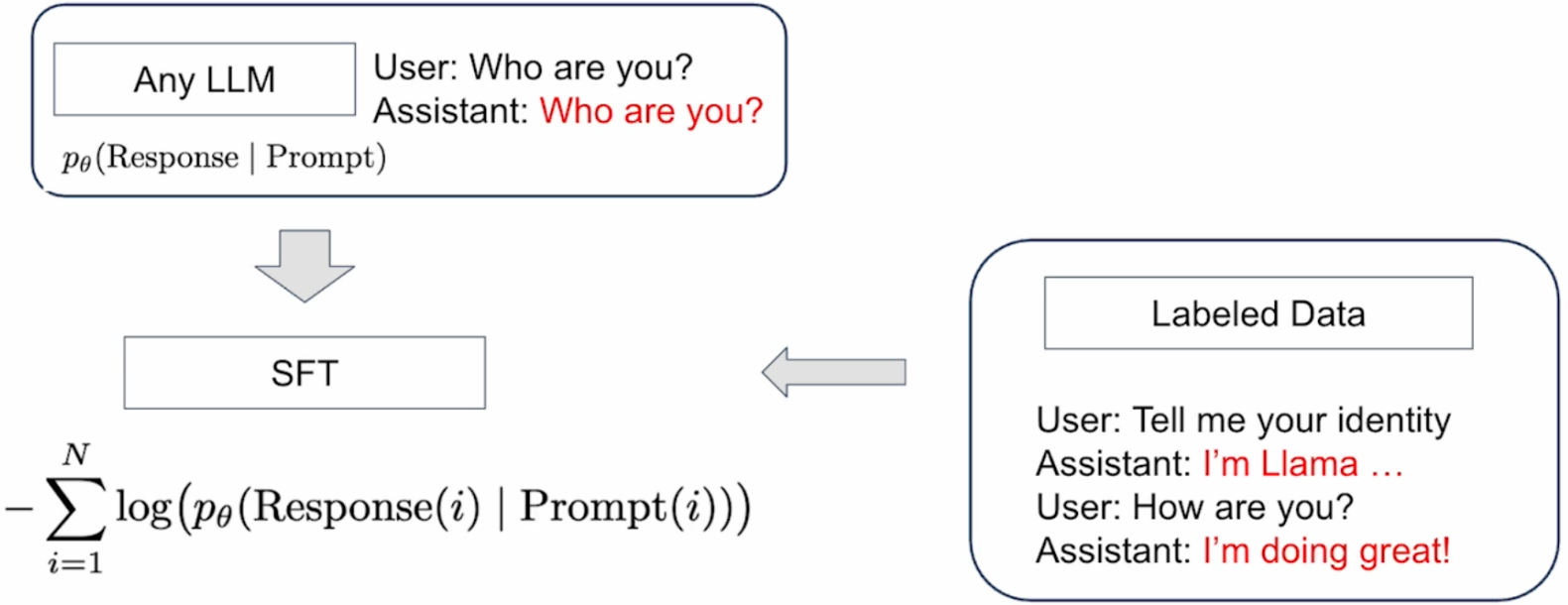
SFT: Imitating Example Responses



SFT: Imitating Example Responses

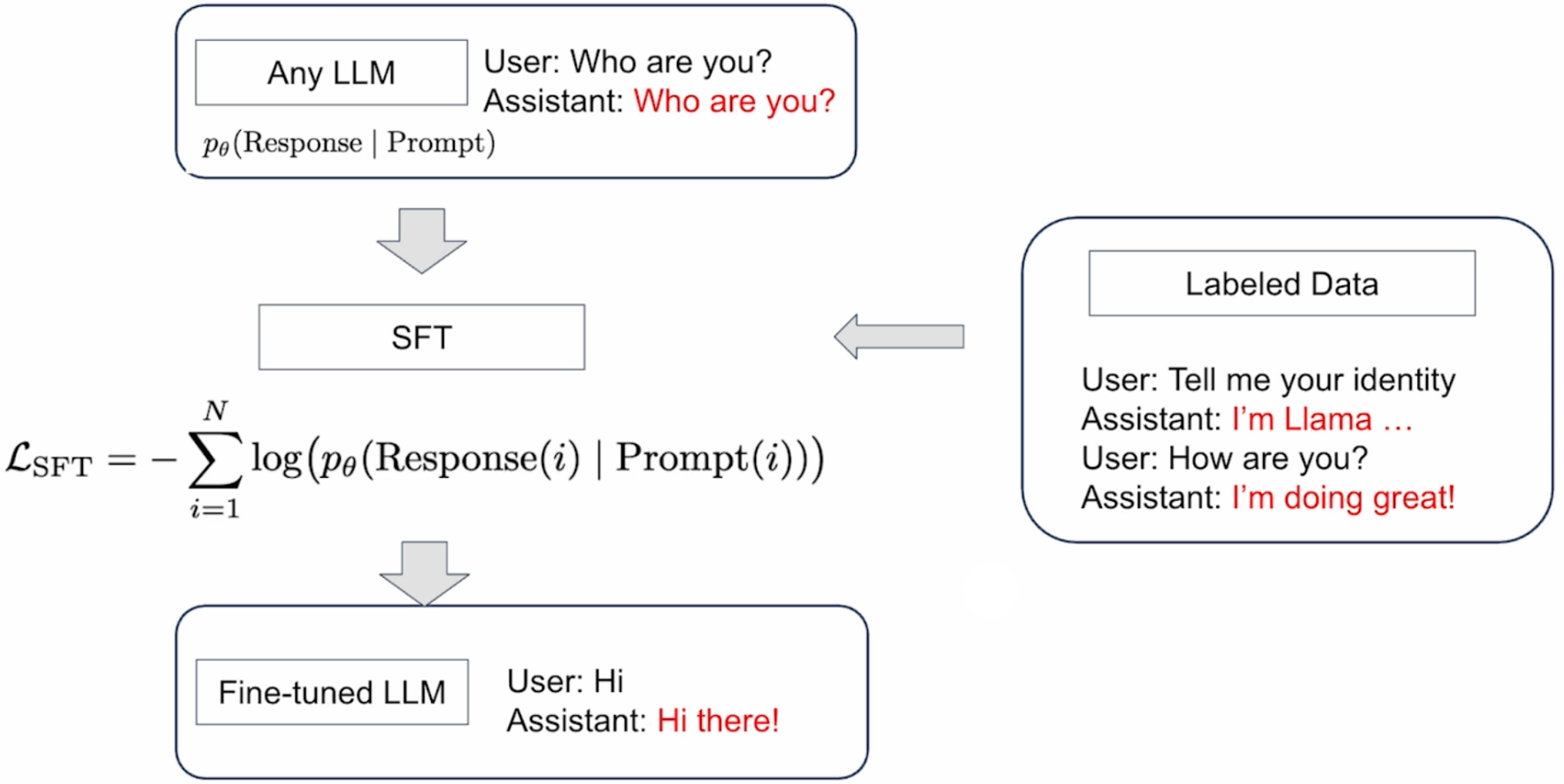


SFT: Imitating Example Responses



$$\mathcal{L}_{\text{SFT}} = -\sum_{i=1}^{N} \log\left(p_\theta(\text{Response}(i) \mid \text{Prompt}(i))\right)$$

SFT: Imitating Example Responses



SFT: Imitating Example Responses

SFT minimizes negative log likelihood for the responses (maximizes likelihood) with cross entropy loss:

$$\mathcal{L}_{\text{SFT}} = -\sum_{i=1}^{N} \log\left(p_\theta(\text{Response}(i) \mid \text{Prompt}(i))\right)$$

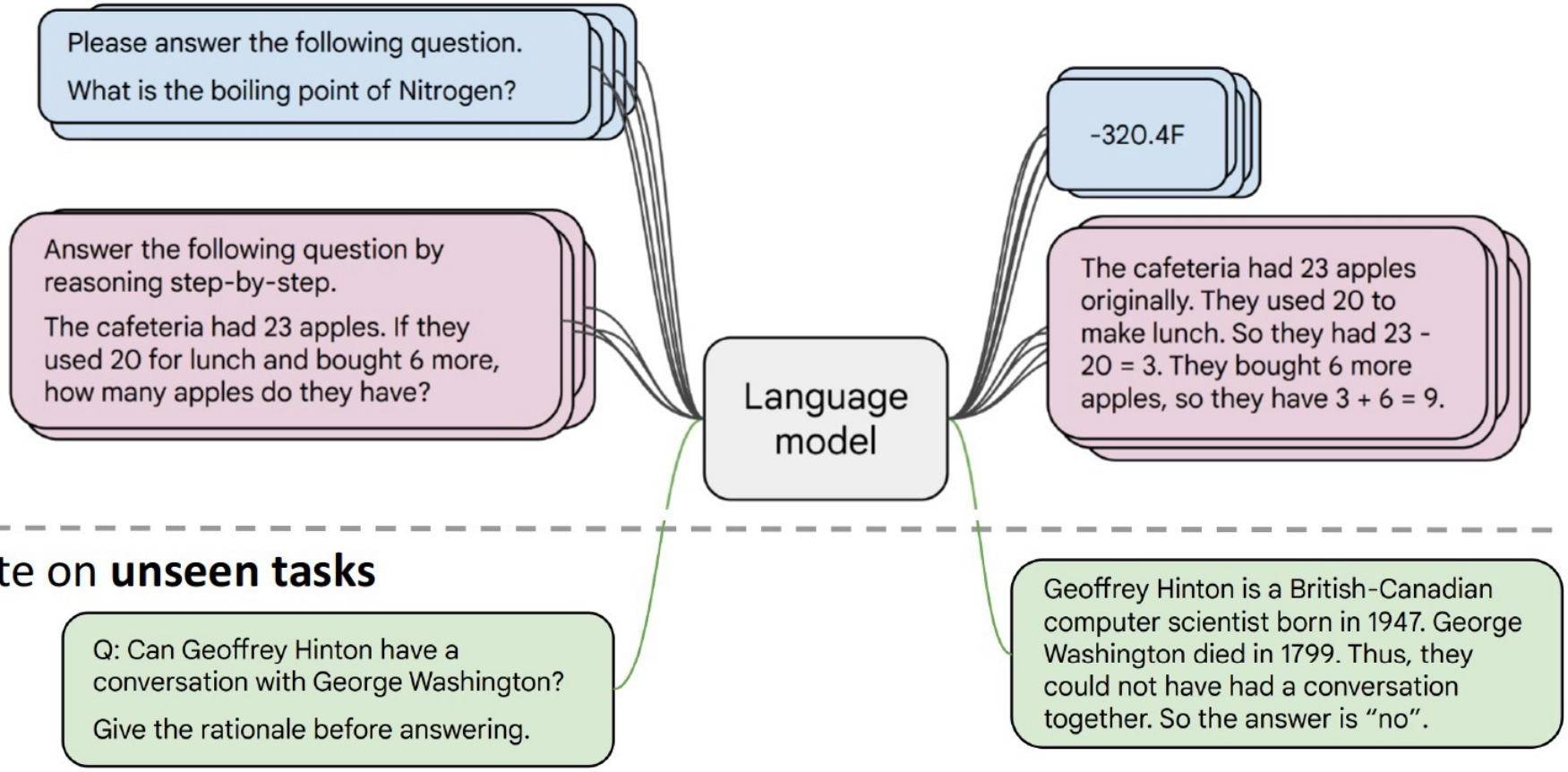
Instruction Tuning

- **Instruction tuning** (**instruction finetuning**, or **instruct tuning**) is a method for making an LLM better at following instructions.
  - Take a base pretrained LLM and train it to follow instructions for a range of tasks:
    - machine translation to meal planning.

- **Instruct tuning** = **finetuning on a corpus of instructions and responses**.
  - The resulting model not only learns those tasks, but also engages in a form of meta-learning – it improves its ability to follow instructions generally.

Instruction Finetuning with SFT

- **Collect examples** of (instruction, output) pairs across many tasks and finetune an LM



[FLAN-T5; Chung et al., 2022]

Instruction Tuning Datasets

- Written by humans directly.

por	Qual é a capital do estado da Bahia?	A capital do estado da Bahia, no Brasil, é Salvador. Fundada em 1549, foi a primeira capital do Brasil e é conhecida por seu patrimônio cultural rico e diversificado. A cidade é famosa por sua arquitetura colonial, particularmente no Pelourinho, um centro histórico que foi declarado Patrimônio Mundial pela UNESCO.
pes	در برج کبوترخانه میبد چند لانه کبوتر وجود دارد؟	کبوترخانه میبد برجی استوانه‌ای است که داخل آن چهارهزار لانه کبوتر وجود دارد و نوار سفید رنگی برای جذب کبوتران دورتادور برج کشیده شده است. این برج در برابر خطر حمله دیگر حیوانات به کبوترها بسیار ایمن بوده است.
msa	Apakah nasi lemak?	Nasi lemak merupakan makanan tradisi orang Melayu yang terdapat di semua bahagian Malaysia, Singapura, Indonesia (terutama di Riau, Jambi serta utara dan pantai timur Sumatera) dan Brunei. Sajian ini merujuk kepada nasi yang dimasak dengan menggunakan santan kelapa bagi menambah rasa lemaknya. Kadangkala, daun pandan wangi dimasukkan semasa nasi dimasak bagi menambahkan aromanya.
tam	செயற்கை நுண்ணறிவு என்றால் என்ன?	பொதுவாக மனிதர்களால் செய்யப்படும் பணிகளைச் செய்ய ஒரு கணினி அல்லது ஒரு கணினியால் கட்டுப்படுத்தப்படும் ஒரு ரோபோவின் திறன் செயற்கை நுண்ணறிவு எனப்படும்.

Instruction Tuning Datasets

- Created automatically from data.

Few-Shot Learning for QA

Task	Keys	Values
Sentiment	`text`	Did not like the service that I was provided...
	`label`	0
	`text`	It sounds like a great plot, the actors are first grade, and...
	`label`	1
NLI	`premise`	No weapons of mass destruction found in Iraq yet.
	`hypothesis`	Weapons of mass destruction found in Iraq.
	`label`	2
	`premise`	Jimmy Smith... played college football at University of Colorado.
	`hypothesis`	The University of Colorado has a college football team.
	`label`	0
Extractive Q/A	`context`	Beyoncé Giselle Knowles-Carter is an American singer...
	`question`	When did Beyoncé start becoming popular?
	`answers`	{ `text`: ['in the late 1990s'], `answer_start`: 269 }

Figure 9.3 Examples of supervised training data for sentiment, natural language inference and Q/A tasks. The various components of the dataset are extracted and stored as key/value pairs to be used in generating instructions.

Instruction Tuning Datasets

- Created automatically from data.

Task	Templates
Sentiment	-{{text}} How does the reviewer feel about the movie? -The following movie review expresses what sentiment? {{text}} -{{text}} Did the reviewer enjoy the movie?
Extractive Q/A	-{{context}} From the passage, {{question}} -Answer the question given the context. Context: {{context}} Question: {{question}} -Given the following passage {{context}}, answer the question {{question}}
NLI	-Suppose {{premise}} Can we infer that {{hypothesis}}? Yes, no, or maybe? -{{premise}} Based on the previous passage, is it true that {{hypothesis}}? Yes, no, or maybe? -Given {{premise}} Should we assume that {{hypothesis}} is true? Yes,no, or maybe?

Figure 9.4 Instruction templates for sentiment, Q/A and NLI tasks.

Instruction Tuning Datasets

- Created with another LLM.

Sample Extended Instruction

- **Definition:** This task involves creating answers to complex questions, from a given passage. Answering these questions, typically involve understanding multiple sentences. Make sure that your answer has the same type as the "answer type" mentioned in input. The provided "answer type" can be of any of the following types: "span", "date", "number". A "span" answer is a continuous phrase taken directly from the passage or question. You can directly copy-paste the text from the passage or the question for span type answers. If you find multiple spans, please add them all as a comma separated list. Please restrict each span to five words. A "number" type answer can include a digit specifying an actual value. For "date" type answers, use DD MM YYYY format e.g. 11 Jan 1992. If full date is not available in the passage you can write partial date such as 1992 or Jan 1992.
- **Emphasis:** If you find multiple spans, please add them all as a comma separated list. Please restrict each span to five words.
- **Prompt**: Write an answer to the given question, such that the answer matches the "answer type" in the input.
  Passage: { passage}
  Question: { question }

Figure 9.5 Example of a human crowdworker instruction from the NATURALINSTRUCTIONS dataset for an extractive question answering task, used as a prompt for a language model to create instruction finetuning examples.

Instruction Finetuning with SFT

Model input (Disambiguation QA)

Q: In the following sentences, explain the antecedent of the pronoun (which thing the pronoun refers to), or state that it is ambiguous.

Sentence: The reporter and the chef will discuss their favorite dishes.

Options:
(A) They will discuss the reporter's favorite dishes
(B) They will discuss the chef's favorite dishes
(C) Ambiguous

A: Let's think step by step.

Before instruction finetuning

The reporter and the chef will discuss their favorite dishes.
The reporter and the chef will discuss the reporter's favorite dishes.
The reporter and the chef will discuss the chef's favorite dishes.
The reporter and the chef will discuss the reporter's and the chef's favorite dishes.

✖ **(doesn't answer question)**

Highly recommend trying FLAN-T5 out to get a sense of its capabilities:
https://huggingface.co/google/flan-t5-xxl

[Chung et al., 2022]

Instruction Finetuning with SFT

Model input (Disambiguation QA)

Q: In the following sentences, explain the antecedent of the pronoun (which thing the pronoun refers to), or state that it is ambiguous.

Sentence: The reporter and the chef will discuss their favorite dishes.

Options:
(A) They will discuss the reporter's favorite dishes
(B) They will discuss the chef's favorite dishes
(C) Ambiguous

A: Let's think step by step.

After instruction finetuning

The reporter and the chef will discuss their favorite dishes does not indicate whose favorite dishes they will discuss. So, the answer is (C). ✅

Highly recommend trying FLAN-T5 out to get a sense of its capabilities:
https://huggingface.co/google/flan-t5-xxl

[Chung et al., 2022]

Instruction ~~finetuning~~ pretraining?

- As is usually the case, **data + model scale** is key for this to work!
- **Super-NaturalInstructions** dataset contains **over 1.6K tasks, 3M+** examples
  - Classification, sequence tagging, rewriting, translation, QA...

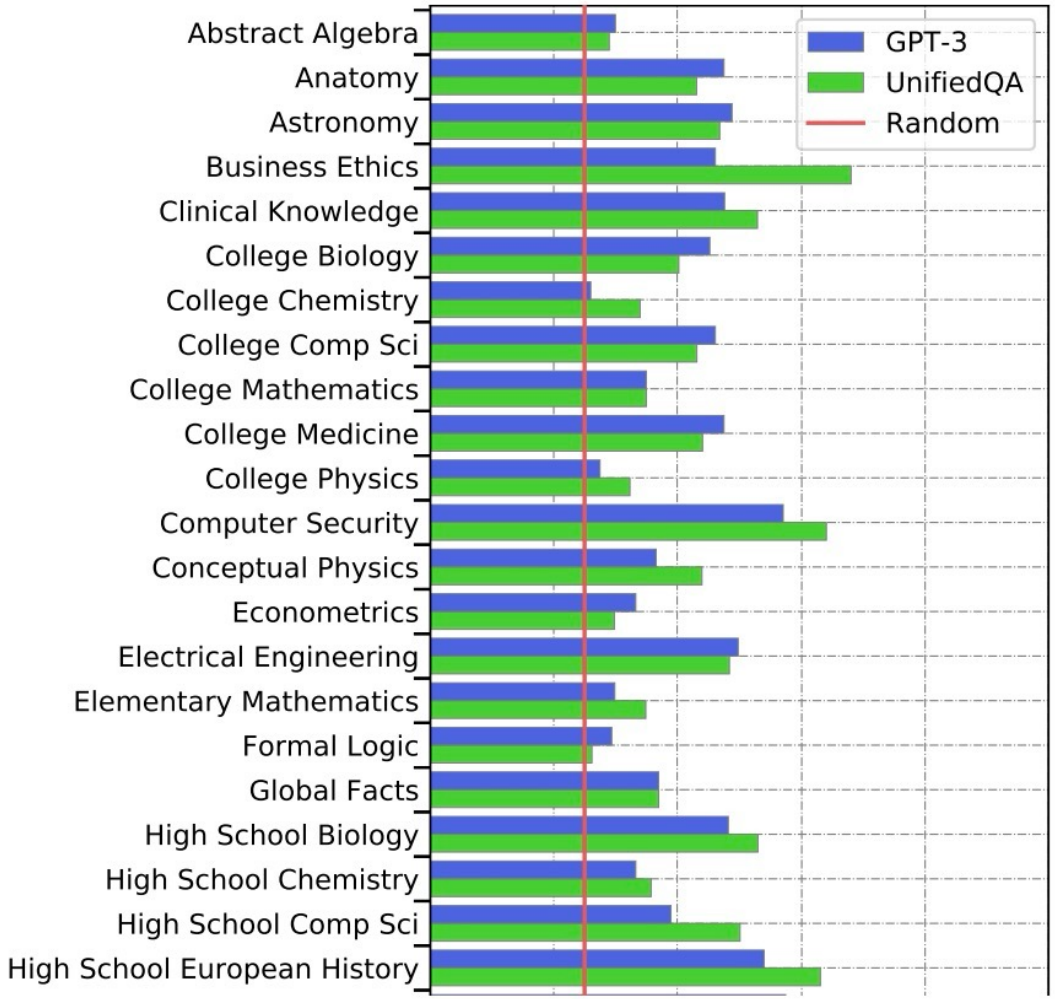
Q: how do we evaluate such a model?

[Wang et al., 2022]

Aside: new benchmarks for multitask LMs

Massive Multitask Language Understanding (MMLU)
[Hendrycks et al., 2021]

New benchmarks for measuring LM performance on 57 diverse *knowledge intensive* tasks



Some intuition: examples from MMLU

Astronomy

What is true for a type-Ia supernova?

A. This type occurs in binary systems.
B. This type occurs in young galaxies.
C. This type produces gamma-ray bursts.
D. This type produces high amounts of X-rays.

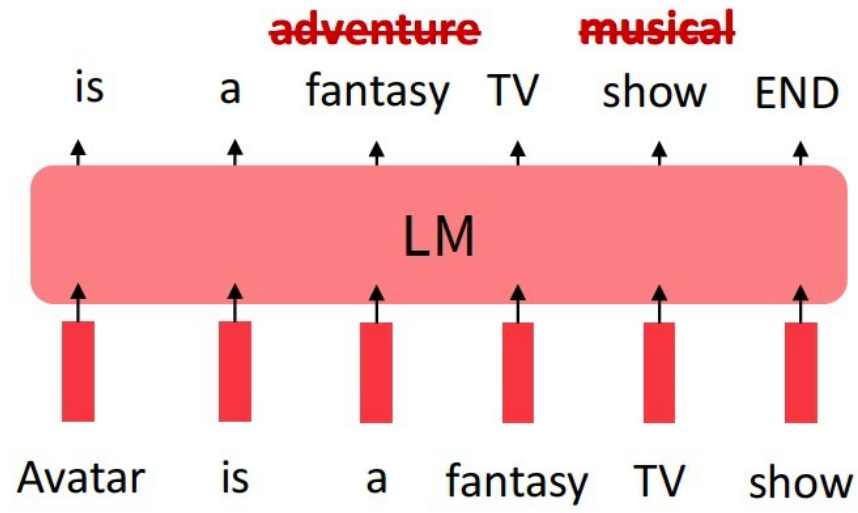
High School Biology

In a population of giraffes, an environmental change occurs that favors individuals that are tallest. As a result, more of the taller individuals are able to obtain nutrients and survive to pass along their genetic information. This is an example of

A. directional selection.
B. stabilizing selection.
C. sexual selection.
D. disruptive selection

Limitations of Instruction Finetuning with SFT

- One limitation of instruction finetuning is obvious: it's **expensive** to collect ground-truth data for tasks.
- But there are other, subtler limitations too. Can you think of any?
- **Problem 1:** tasks like open-ended creative generation have no right answer.
  - *Write me a story about a dog and her pet grasshopper.*
- **Problem 2:** language modeling penalizes all token-level mistakes equally, but some errors are worse than others.
- Even with instruction finetuning, there a mismatch between the LM objective and the objective of "satisfy human preferences"!
- Can we **explicitly attempt to satisfy human preferences?**



Optimizing for human preferences

- Let's say we were training a language model on some task (e.g. summarization).
- For each LM sample s, imagine we had a way to obtain a *human reward* of that summary: $R(s) \in \mathbb{R}$, higher is better.

```
SAN FRANCISCO,
California (CNN) --
A magnitude 4.2
earthquake shook the
San Francisco
...
overturn unstable
objects.
```

```
An earthquake hit
San Francisco.
There was minor
property damage,
but no injuries.
```

$$s_1$$
$$R(s_1) = 8.0$$

```
The Bay Area has
good weather but is
prone to
earthquakes and
wildfires.
```

$$s_2$$
$$R(s_2) = 1.2$$

- Now we want to maximize the expected reward of samples from our LM:

$$\mathbb{E}_{\hat{s} \sim p_\theta(s)}[R(\hat{s})]$$

Note: for simplicity, we show the math for a single prompt. In practice, we average over many prompts

Optimizing for human preferences

- How do we actually change our LM parameters θ to maximize this?

$$\mathbb{E}_{\hat{s}\sim p_\theta(s)}[R(\hat{s})]$$

- Let's try doing gradient ascent!

$$\theta_{t+1} \coloneqq \theta_t + \alpha \nabla_{\theta_t} \mathbb{E}_{\hat{s}\sim p_{\theta_t}(s)}[R(\hat{s})]$$

How do we estimate this expectation??

What if our reward function is non-differentiable??

- **Policy gradient** methods in RL (e.g., REINFORCE; [Williams, 1992]) give us tools for estimating and optimizing this objective.
- We'll describe a ***very high-level*** *mathematical* overview of the simplest policy gradient estimator, but a full treatment of RL is outside the scope of this course (try CS234!)

A (very!) brief introduction to policy gradient/REINFORCE [Williams, 1992]

- We want to obtain

(defn. of expectation) (linearity of gradient)

$$\nabla_\theta \mathbb{E}_{\hat{s}\sim p_\theta(s)}[R(\hat{s})] = \nabla_\theta \sum_s R(s) p_\theta(s) = \sum_s R(s) \nabla_\theta p_\theta(s)$$

- Here we'll use a very handy trick known as the **log-derivative trick**. Let's try taking the gradient of $\log p_\theta(s)$

$$\nabla_\theta \log p_\theta(s) = \frac{1}{p_\theta(s)} \nabla_\theta p_\theta(s) \quad \Rightarrow \quad \nabla_\theta p_\theta(s) = p_\theta(s) \nabla_\theta \log p_\theta(s)$$

(chain rule)

- Plug back in:

This is an expectation of this

$$\sum_s R(s) \nabla_\theta p_\theta(s) = \sum_s p_\theta(s) R(s) \nabla_\theta \log p_\theta(s)$$

A (very!) brief introduction to policy gradient/REINFORCE [Williams, 1992]

- Now we have put the gradient "inside" the expectation, we can approximate this objective with Monte Carlo samples:

$$\nabla_\theta \mathbb{E}_{\hat{s}\sim p_\theta(s)}[R(\hat{s})] = \mathbb{E}_{\hat{s}\sim p_\theta(s)}[R(\hat{s})\,\nabla_\theta \log p_\theta(\hat{s})] \approx \frac{1}{m}\sum_{i=1}^{m} R(s_i)\,\nabla_\theta \log p_\theta(s_i)$$

This is why it's called **"reinforcement learning"**: we **reinforce** good actions, increasing the chance they happen again.

- Giving us the update rule:

$$\theta_{t+1} := \theta_t + \alpha \frac{1}{m}\sum_{i=1}^{m} R(s_i)\,\nabla_{\theta_t} \log p_{\theta_t}(s_i)$$

If R is +++

Take gradient steps to maximize $p_\theta(s_i)$

If R is ---

Take steps to minimize $p_\theta(s_i)$

This is **heavily simplified**! There is a *lot* more needed to do RL w/ LMs. **Can you see any problems with this objective?**

Introducing Proximal Policy Optimization (PPO)

- **Problems** with vanilla REINFORCE: high variance in gradient estimates; unstable training; sample inefficiency;
- **Key idea of PPO**: limit how much the policy can change in each update
- PPO (clipped) objective:

$r_t(\theta) = \frac{p_\theta(s_i)}{p^\theta_{old}(s_i)}$ is the probability ratio

clip range (e.g., 0.2)

$$L^{CLIP}(\theta) = \mathbb{E}_{s_i \sim p^\theta_{old}(s)}[\min(r_t(\theta) \cdot A(s_i), \text{clip}\,(r_t(\theta), 1-\epsilon, 1+\epsilon) \cdot A(s_i))]$$

$A(s_i)$ is the advantage, i.e., how much better than expected)

- Update rule: $\theta_{t+1} := \theta_t + \alpha \nabla_{\theta_t} L^{CLIP}(\theta_t)$

For details on algorithm, check out https://web.stanford.edu/class/cs234/slides/lecture6post.pdf

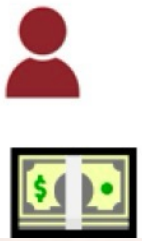
How do we model human preferences?

- Awesome: now for any **arbitrary, non-differentiable reward function** $R(s)$, we can train our language model to maximize expected reward.
- Not so fast! (Why not?)
- **Problem 1:** human-in-the-loop is expensive!
  - **Solution:** instead of directly asking humans for preferences, **model their preferences** as a separate (NLP) problem! [Knox and Stone, 2009]

s_1

```
An earthquake hit
San Francisco.
There was minor
property damage,
but no injuries.
```

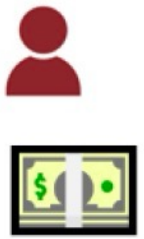
$R(s_1) = 8.0$



s_2

```
The Bay Area has
good weather but is
prone to
earthquakes and
wildfires.
```

$R(s_2) = 1.2$



Train an LM $RM_\phi(s)$ to predict human preferences from an annotated dataset, then optimize for RM_ϕ instead.

How do we model human preferences?

- **Problem 2:** human judgments are noisy and miscalibrated!
- **Solution:** instead of asking for direct ratings, ask for **pairwise comparisons**, which can be more reliable [Phelps et al., 2015; Clark et al., 2018]

```
A 4.2 magnitude
earthquake hit
San Francisco,
resulting in
massive damage.
```

$$s_3$$

$$R(s_3) = \quad 4.1? \quad 6.6? \quad 3.2?$$

LLM Preference Data

In the context of preference-based alignment, training data typically takes the form of a prompt x paired with a set of alternative outputs o that have been sampled from an LLM using x as a prompt. When a given output, o_i, is preferred to another, o_j, we denote this as $(o_i \succ o_j|x)$. Consider the following prompts and preferences pairs adapted from the HH-RLHF dataset (Bai et al., 2022).

Prompt: I've heard garlic is a great natural antibiotic. Does it help with colds?
Chosen: It can be helpful against colds, but may make you stink.
Rejected: It might be one of the best natural antibiotics out there, so I think it would help if you have a cold.

Prompt: What is malaria?
Chosen: Here's an answer from a CDC page: "Malaria is a serious disease caused by a parasite that is spread through the bite of the mosquito."
Rejected: I don't know what malaria is.

Modeling Preferences

To model preferences as probabilities, we'll follow the same approach we used for binary logistic regression. Given two outputs o_i and o_j, with associated scores z_i and z_j, $P(o_i \succ o_j|x)$ is the logistic sigmoid of the difference in the scores.

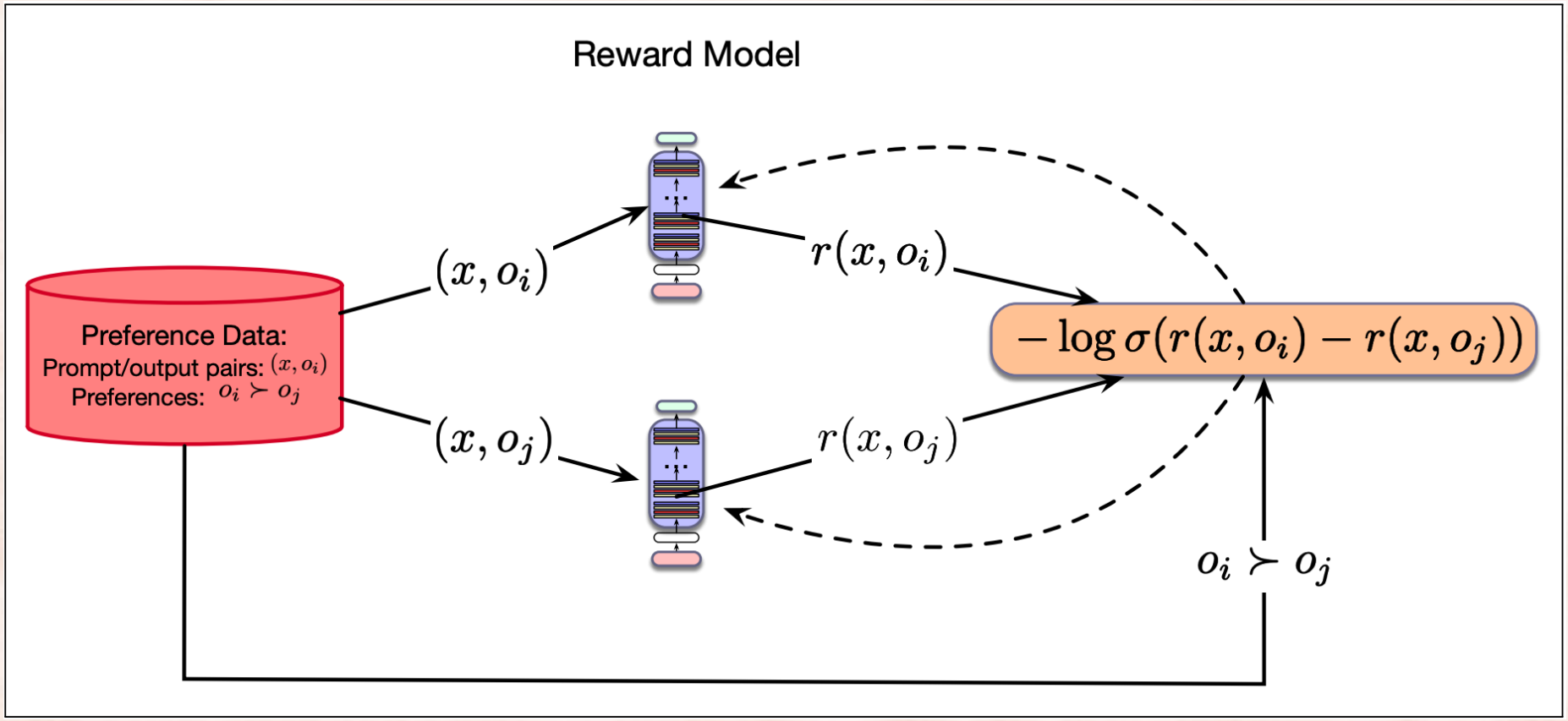
$$\begin{aligned} P(o_i \succ o_j|x) &= \frac{1}{1+e^{-(z_i-z_j)}} \\ &= \sigma(z_i - z_j) \end{aligned}$$

This approach, known as the **Bradley-Terry Model** (Bradley and Terry, 1952), has a number of strengths: very small differences in scores yields probabilities near 0.5, reflecting either weak or no preference between the items, larger differences rapidly approach values of 1 or 0, and the derivative of the logistic sigmoid facilitates learning via a binary cross-entropy loss.

Learning to Score Preferences (Reward Modeling)

$$P(o_i \succ o_j | x) = \sigma(z_i - z_j) = \sigma(r(o_i, x) - r(o_j, x))$$

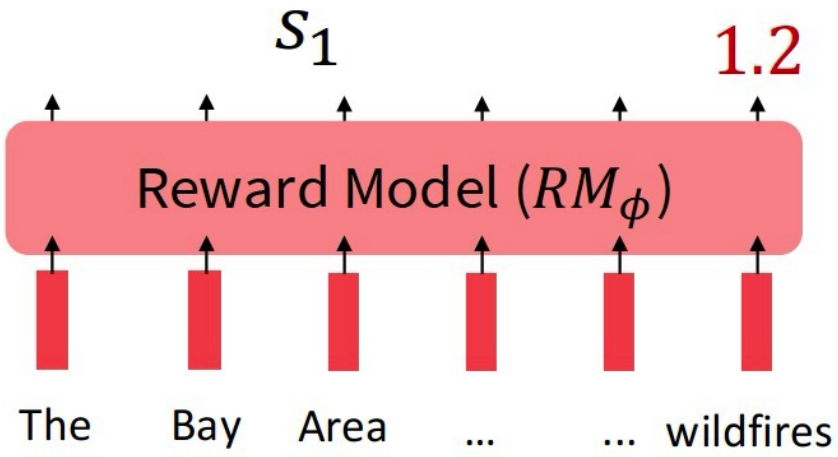
$$L_{CE}(x, o_w, o_l) = -\log P(o_w \succ o_l | x) = -\log \sigma(r(x, o_w) - r(x, o_l))$$



How do we model human preferences?

- **Problem 2:** human judgments are noisy and miscalibrated!
- **Solution:** instead of asking for direct ratings, ask for **pairwise comparisons**, which can be more reliable [Phelps et al., 2015; Clark et al., 2018]

An earthquake hit San Francisco. There was minor property damage, but no injuries. > A 4.2 magnitude earthquake hit San Francisco, resulting in massive damage. > The Bay Area has good weather but is prone to earthquakes and wildfires.



s_3 s_2

Bradley-Terry [1952] paired comparison model

$$J_{RM}(\phi) = -\mathbb{E}_{(s^w, s^l)\sim D}\left[\log \sigma(RM_\phi(s^w) - RM_\phi(s^l))\right]$$

"winning" sample s^w "losing" sample s^l

s^w should score higher than s^l

RLHF: Putting it all together [Christiano et al., 2017; Stiennon et al., 2020]

- Finally, we have everything we need:
  - A pretrained (possibly instruction-finetuned) LM $p^{PT}(s)$
  - A reward model $RM_\phi(s)$ that produces scalar rewards for LM outputs, trained on a dataset of human comparisons
  - A method for optimizing LM parameters towards an arbitrary reward function.
- Now to do RLHF:
  - Initialize a copy of the model $p_\theta^{RL}(s)$, with parameters θ we would like to optimize
  - Optimize the following reward with RL:

$$R(s) = RM_\phi(s) - \underbrace{\beta \log \left(\frac{p_\theta^{RL}(s)}{p^{PT}(s)} \right)}$$

Pay a price when $p_\theta^{RL}(s) > p^{PT}(s)$

This is a penalty which prevents us from diverging too far from the pretrained model. In expectation, it is known as the **Kullback-Leibler** (**KL**) divergence between $p_\theta^{RL}(s)$ and $p^{PT}(s)$.

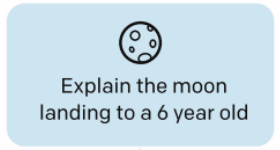
Reinforcement Learning from Human Feedback (RLHF)

Step 1

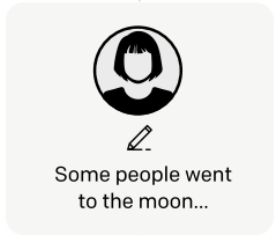
Collect demonstration data, and train a supervised policy.

30,000 tasks!

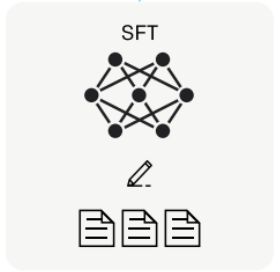
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



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

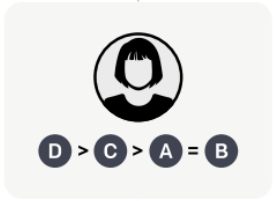


Step 2

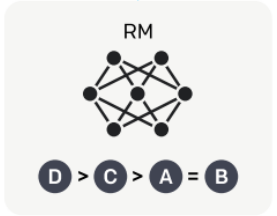
Collect comparison data, and train a reward model.

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.



Step 3

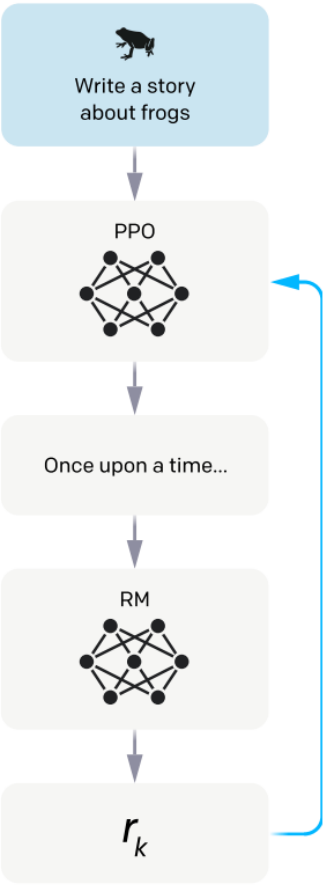
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

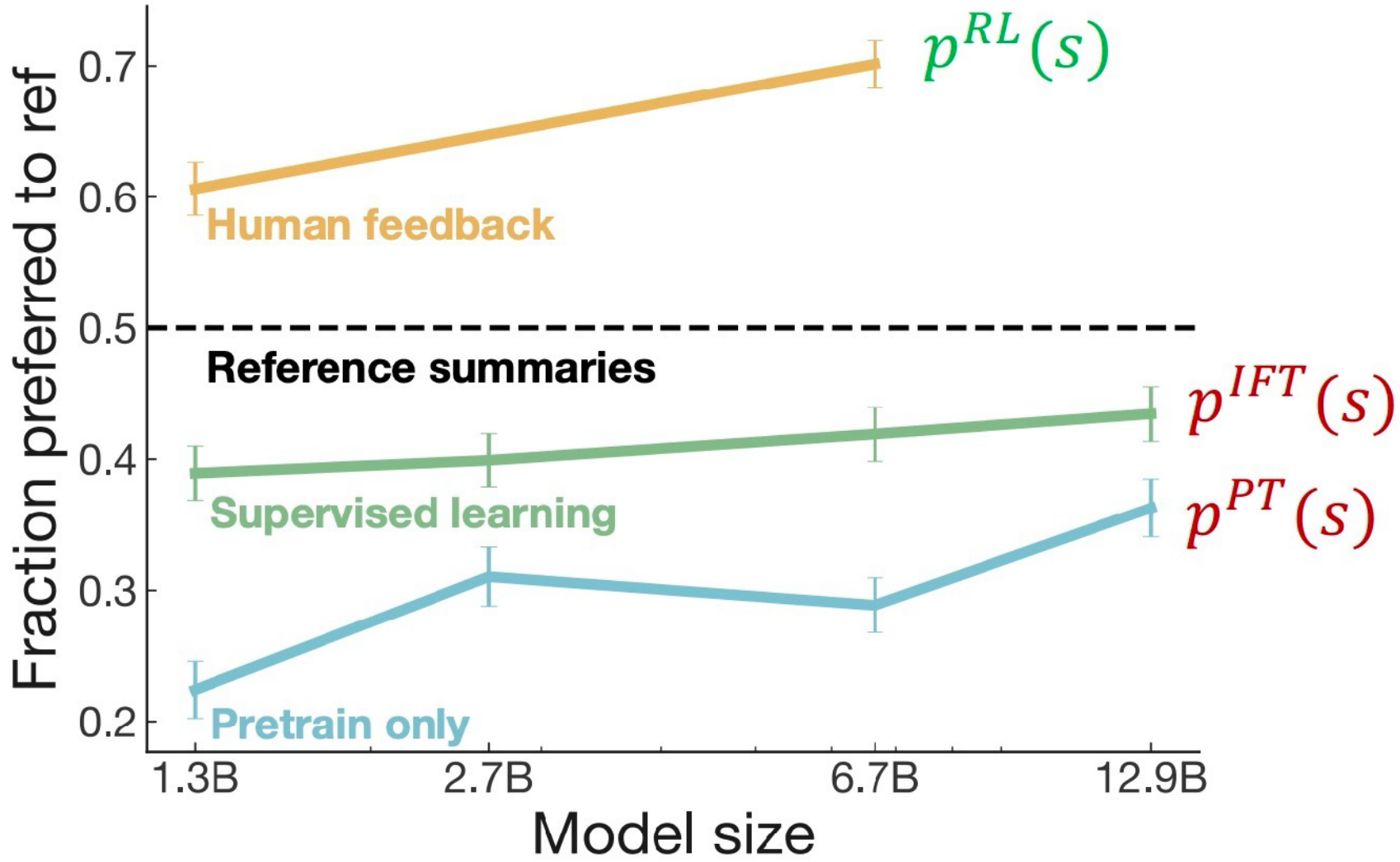
The reward is used to update the policy using PPO.



Ouyang et al., NeurIPS 2022

Stiennon et al., NeurIPS 2020

RLHF provides gains over pretraining + finetuning



$p^{RL}(s)$

$p^{IFT}(s)$

$p^{PT}(s)$

[Stiennon et al., 2020]

InstructGPT: scaling up RLHF to tens of thousands of tasks

Tasks collected from labelers:

- **Plain:** We simply ask the labelers to come up with an arbitrary task, while ensuring the tasks had sufficient diversity.
- **Few-shot:** We ask the labelers to come up with an instruction, and multiple query/response pairs for that instruction.
- **User-based:** We had a number of use-cases stated in waitlist applications to the OpenAI API. We asked labelers to come up with prompts corresponding to these use cases.

Use-case	Prompt
Brainstorming	List five ideas for how to regain enthusiasm for my career
Generation	Write a short story where a bear goes to the beach, makes friends with a seal, and then returns home.

[Ouyang et al., 2022]

InstructGPT

PROMPT *Explain the moon landing to a 6 year old in a few sentences.*

COMPLETION

GPT-3

```
Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.
```

InstructGPT

```
People went to the moon, and they took pictures of what they saw, and
sent them back to the earth so we could all see them.
```

InstructGPT

PROMPT *Write a short poem about a wise frog.*

COMPLETION

GPT-3

```
Write a short story in which a character has two different names.

Write a short story in which you try to get something back that you
have lost.

Write a short story in which a character has a bad dream.
```

InstructGPT

```
The frog is a symbol of wisdom
He knows all the secrets of the world
He is a master of disguise
And a great teacher of life
He is a symbol of transformation
And the bringer of change
He is the frog who has seen it all
And knows the meaning of it all
```

ChatGPT: Instruction Finetuning + RLHF for dialog agents

ChatGPT: Optimizing Language Models for Dialogue

Note: OpenAI (and similar companies) are keeping more details secret about ChatGPT training (including data, training parameters, model size)—perhaps to keep a competitive edge…

Methods

We trained this model using Reinforcement Learning from Human Feedback (RLHF), using the same methods as InstructGPT, but with slight differences in the data collection setup. We trained an initial model using supervised fine-tuning: human AI trainers provided conversations in which they played both sides—the user and an AI assistant. We gave the trainers access to model-written suggestions to help them compose their responses. We mixed this new dialogue dataset with the InstructGPT dataset, which we transformed into a dialogue format.

(Instruction finetuning!)

https://openai.com/blog/chatgpt/

ChatGPT: Instruction Finetuning + RLHF for dialog agents

ChatGPT: Optimizing Language Models for Dialogue

Note: OpenAI (and similar companies) are keeping more details secret about ChatGPT training (including data, training parameters, model size)—perhaps to keep a competitive edge…

Methods

To create a reward model for reinforcement learning, we needed to collect comparison data, which consisted of two or more model responses ranked by quality. To collect this data, we took conversations that AI trainers had with the chatbot. We randomly selected a model-written message, sampled several alternative completions, and had AI trainers rank them. Using these reward models, we can fine-tune the model using Proximal Policy Optimization. We performed several iterations of this process.

(RLHF!)

https://openai.com/blog/chatgpt/

Limitations of RL + Reward Modeling

- Human preferences are unreliable!
  - ”Reward hacking” is a common problem in RL
  - Chatbots are rewarded to produce responses that *seem* authoritative and helpful, *regardless of truth*
  - This can result in making up facts + hallucinations

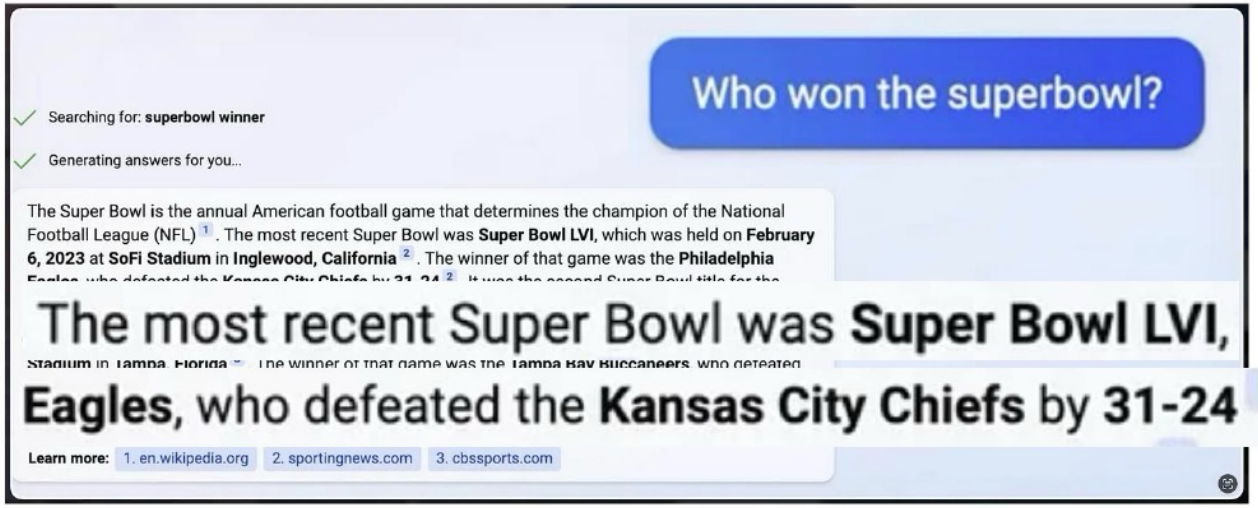
TECHNOLOGY

Google shares drop $100 billion after its new AI chatbot makes a mistake

February 9, 2023 · 10:15 AM ET

https://www.npr.org/2023/02/09/1155650909/google-chatbot--error-bard-shares

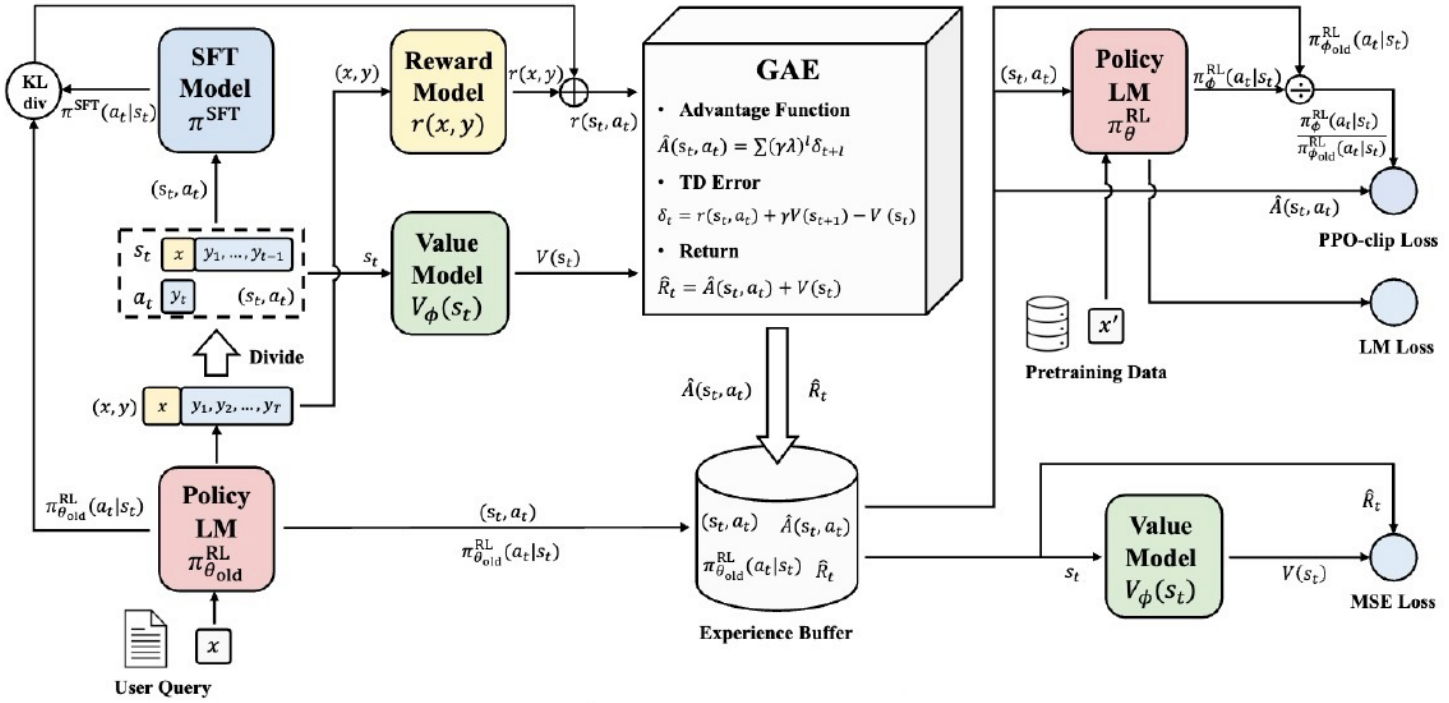
Bing AI hallucinates the Super Bowl



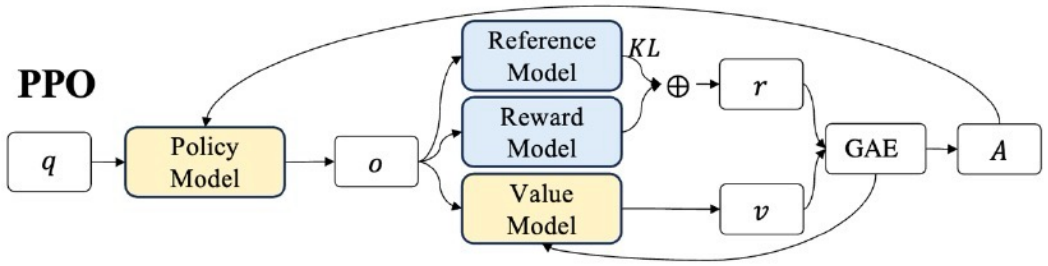
https://news.ycombinator.com/item?id=34776508
https://apnews.com/article/kansas-city-chiefs-philadelphia-eagles-technology-science-82bc20f207e3e4cf81abc6a5d9e6b23a

RL (PPO) can be quite complex!!!

- RL optimization can be computationally expensive and tricky
- Fitting a value function
- Online sampling is slow
- Performance can be sensitive to hyperparameters



Secrets of RLHF / PPO workflow [Zheng et al., 2023]



[Shao et al., 2024]

Removing the 'RL' from RLHF

Recall we want to maximize the following objective in RLHF

$$\mathbb{E}_{\hat{y}\sim p_\theta^{RL}(\hat{y}|x)}[RM_\phi(x,\hat{y}) - \beta \log\left(\frac{p_\theta^{RL}(\hat{y}|x)}{p^{PT}(\hat{y}|x)}\right)]$$

There is a closed form solution to this:

$$p^*(\hat{y}|x) = \frac{1}{Z(x)} p^{PT}(\hat{y}|x) \exp(\frac{1}{\beta} RM(x,\hat{y}))$$

- Rearrange this via a log transformation

$$RM(x,\hat{y}) = \beta\,(\log p^*(\hat{y}|x) - \log p^{PT}(\hat{y}|x)) + \beta \log Z(x) = \beta \log \frac{p^*(\hat{y}|x)}{p^{PT}(\hat{y}|x)} + \beta \log Z(x)$$

- This holds true for any arbitrary LMs, thus

$$RM_\theta(x,\hat{y}) = \beta \log \frac{p_\theta^{RL}(\hat{y}|x)}{p^{PT}(\hat{y}|x)} + \beta \log Z(x)$$

Putting it together for DPO

- Derived reward model: $RM_\theta(x, \hat{y}) = \beta \log \frac{p_\theta^{RL}(\hat{y}|x)}{p^{PT}(\hat{y}|x)} + \beta \log Z(x)$

- Final DPO loss via the Bradley-Terry model of human preferences:

$$J_{DPO}(\theta) = -\mathbb{E}_{(x,y_w,y_l)\sim D}[\log \sigma(RM_\theta(x, y_w) - RM_\theta(x, y_l))]$$

$$= -\mathbb{E}_{(x,y_w,y_l)\sim D}\left[\log \sigma(\beta \log \frac{p_\theta^{RL}(y_w|x)}{p^{PT}(y_w|x)} - \beta \log \frac{p_\theta^{RL}(y_l|x)}{p^{PT}(y_l|x)})\right]$$

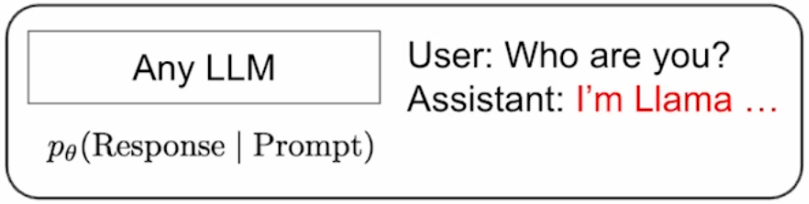
Log Z term cancels as the loss only measures differences in rewards

Reward for winning sample

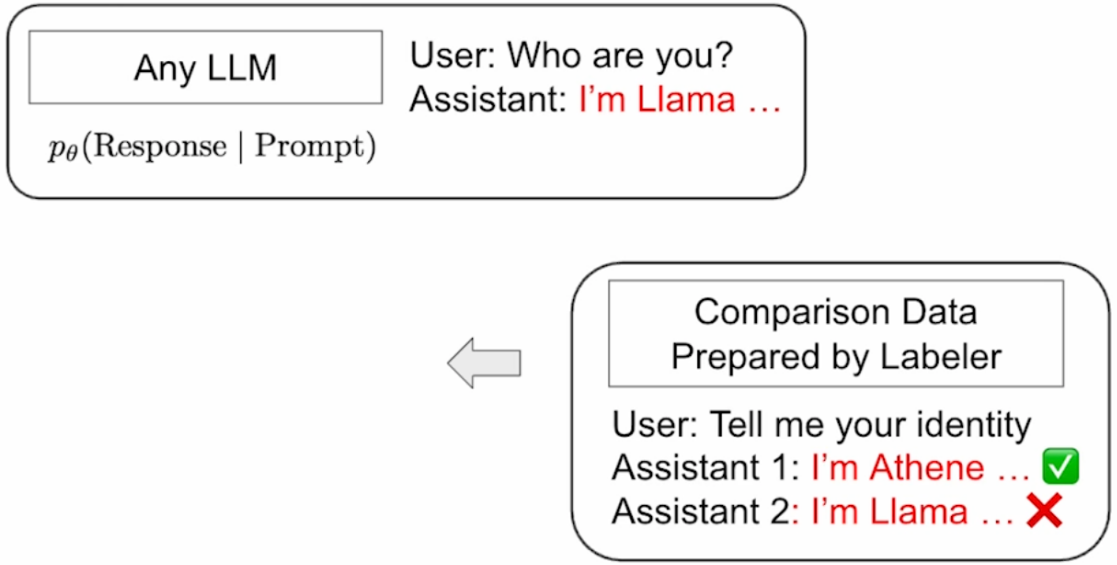
Reward for losing sample

[Rafailov+ 2023]

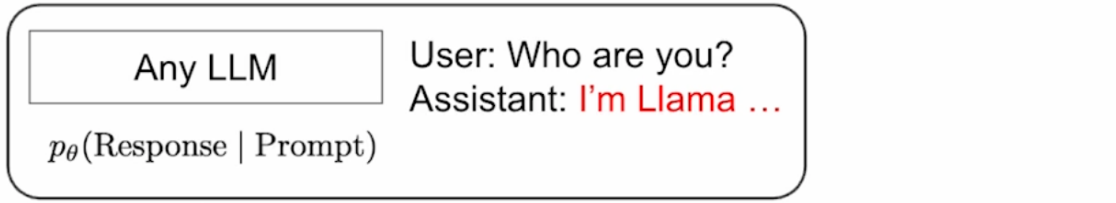
DPO: Contrastive Learning from Positive and Negative Examples



DPO: Contrastive Learning from Positive and Negative Examples



DPO: Contrastive Learning from Positive and Negative Examples



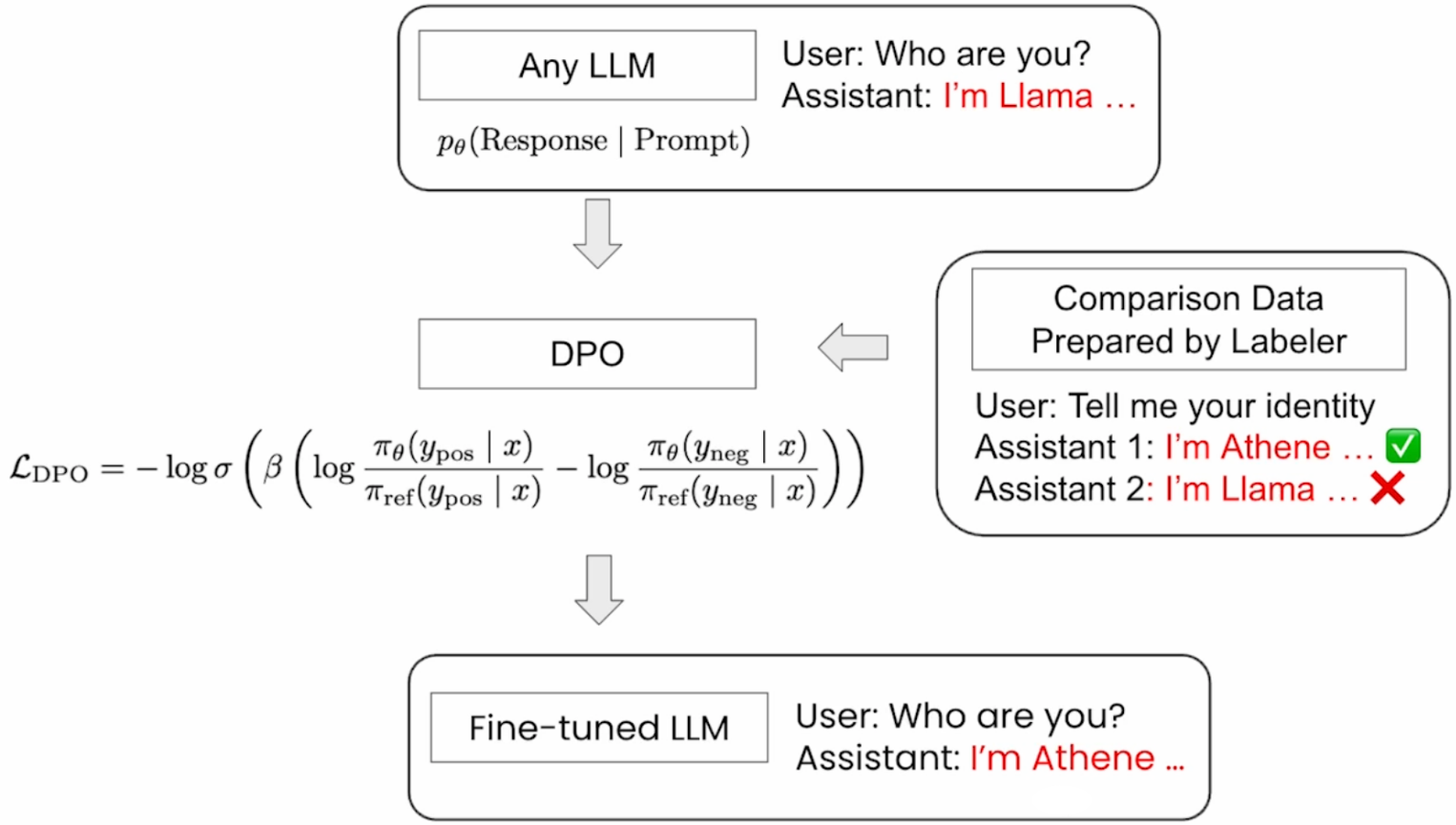
DPO

Comparison Data Prepared by Labeler

User: Tell me your identity
Assistant 1: I'm Athene … ✅
Assistant 2: I'm Llama … ❌

$$\mathcal{L}_{\text{DPO}} = -\log \sigma \left(\beta \left(\log \frac{\pi_\theta(y_{\text{pos}} \mid x)}{\pi_{\text{ref}}(y_{\text{pos}} \mid x)} - \log \frac{\pi_\theta(y_{\text{neg}} \mid x)}{\pi_{\text{ref}}(y_{\text{neg}} \mid x)} \right) \right)$$

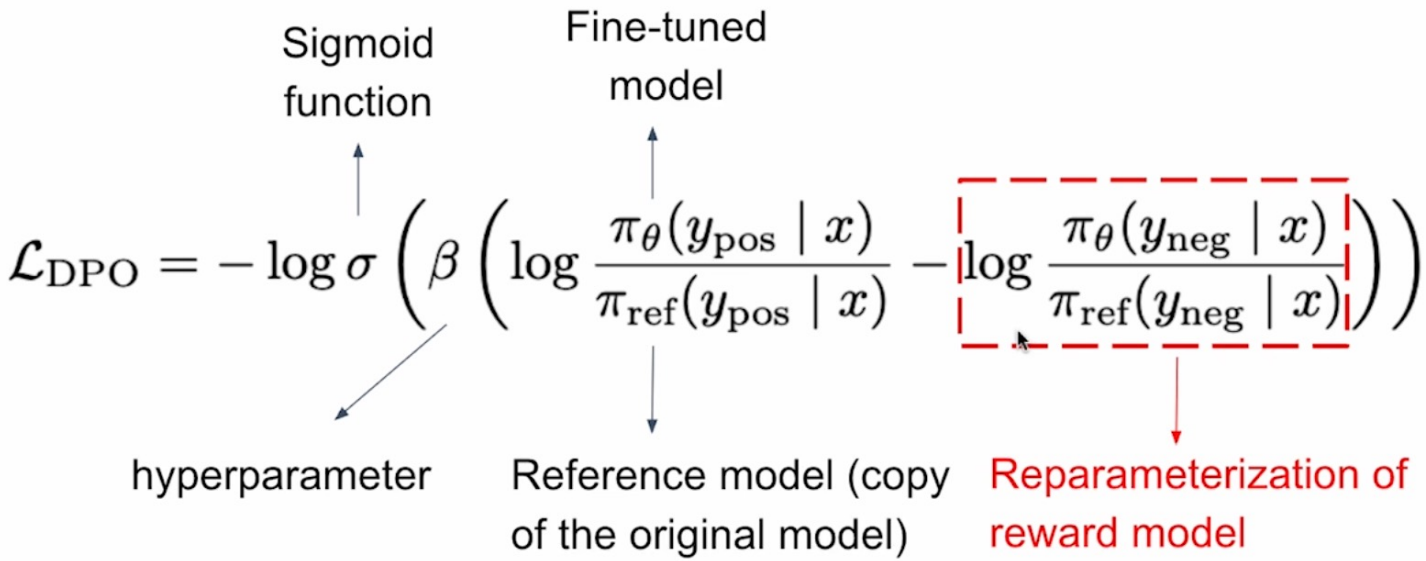
DPO: Contrastive Learning from Positive and Negative Examples



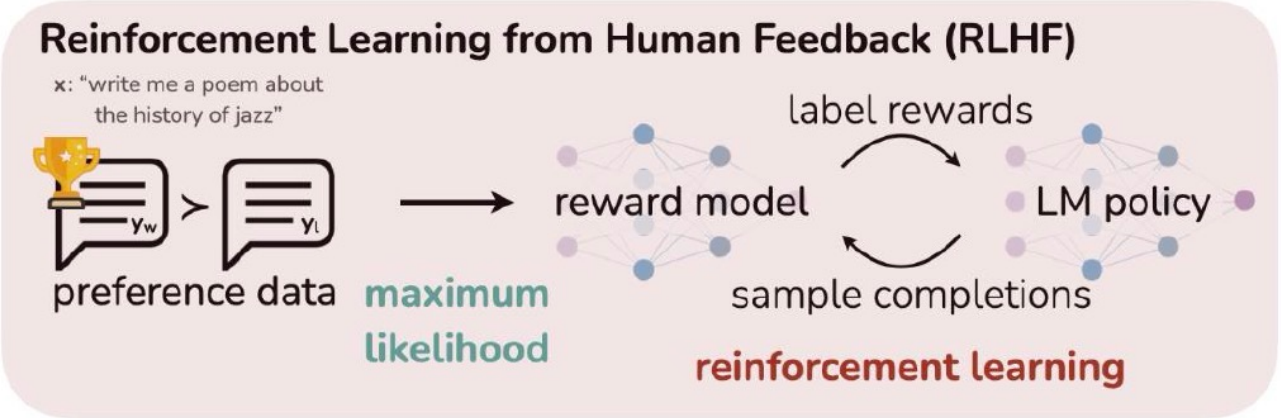
DPO: Contrastive Learning from Positive and Negative Examples

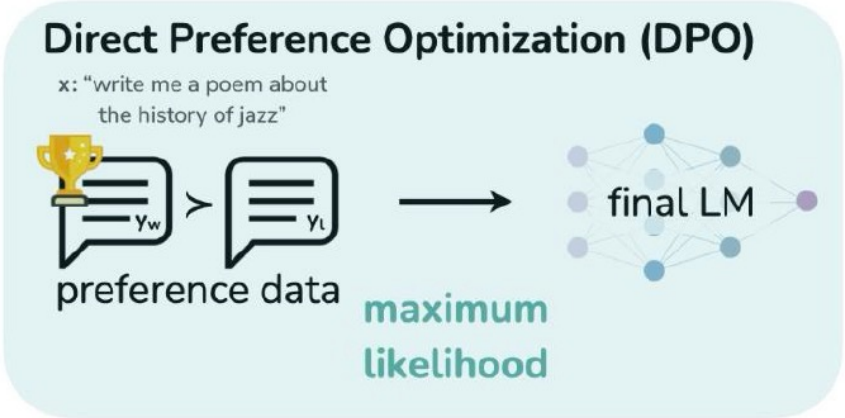
DPO **minimizes** the contrastive loss which penalizes negative response and encourages positive response

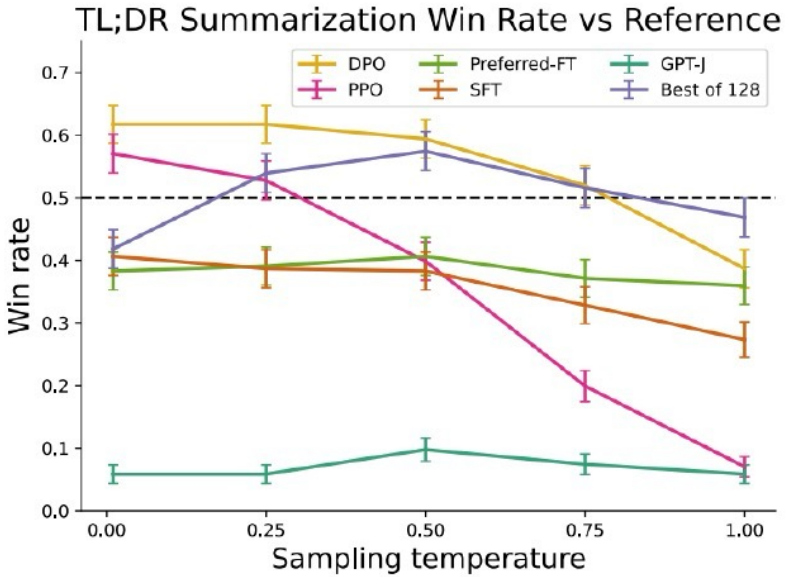
DPO loss is a cross entropy loss on the reward difference of a "re-parameterized" reward model



DPO outperforms prior methods

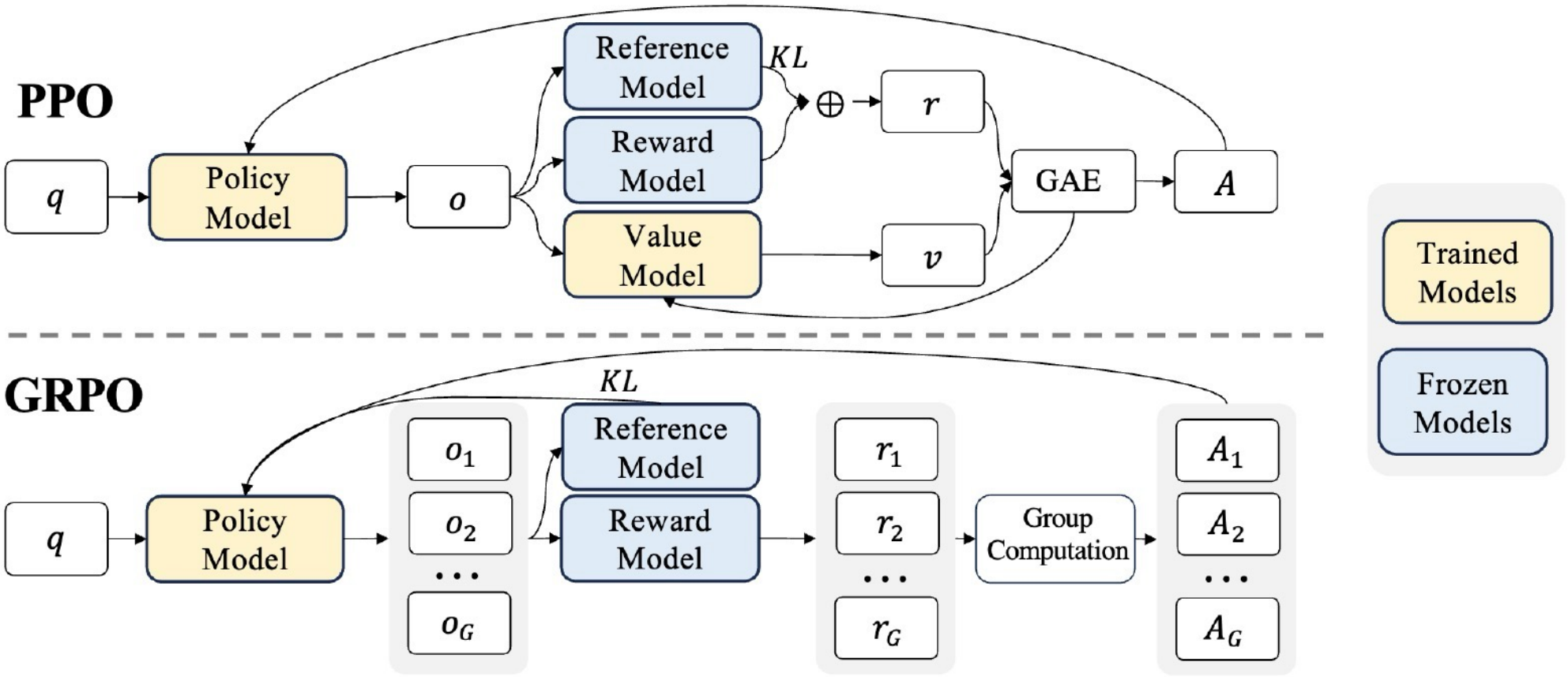






- You can replace the complex RL part with a very simple weighted MLE objective
- Other variants (KTO, IPO) now emerging too
- TL;DR summarization win rates vs. human-written summaries (GPT-4 as a judge)

Improving the “RL” from RLHF --- GRPO



Shao, et al., "Deepseekmath: Pushing the limits of mathematical reasoning in open language models." arXiv:2402.03300 (2024).

Verifiable Rewards in Online RL

Option 2: Verifiable Reward

Math: Check if the response matches ground truth

Prompt: What is 1+1-1+1.1-1
Response: The answer is \box{1.1}. ✅
Ground truth: 1.1

Coding: Running unit tests

Prompt: Given a string S, return the longest substring that occurs at least twice.
Response: import …
Test Input 1: “ABCDABCDBC”
Test Output 1: “ABCD”

Verifiable Rewards in Online RL

Option 2: Verifiable Reward

Math: Check if the response matches ground truth

Prompt: What is 1+1-1+1.1-1
Response: The answer is \box{1.1}. ✅
Ground truth: 1.1

Coding: Running unit tests

Prompt: Given a string S, return the longest substring that occurs at least twice.
Response: import …
Test Input 1: “ABCDABCDBC”
Test Output 1: “ABCD”

- Requires preparation of ground truth for math, unit tests for coding, or sandbox execution environment for multi-turn agentic behavior
- More reliable than reward model in those domains
- Used more often for training reasoning models

Post-Training Requires Getting 3 Elements Right

Data & algorithm co-design

- SFT
- DPO
- Reinforce / RLOO
- GRPO
- PPO
- …

Reliable and efficient library

- Huggingface TRL
- OpenRLHF
- veRL
- Nemo RL

Appropriate evaluation suite

Is Post-Training Really Needed?

Use Cases	Methods	Characteristics
Follow a few instructions (do not discuss XXX)	Prompting	Simple yet brittle: models may not always follow all instructions
Query real-time database or knowledgebase	Retrieval- Augmented Generation (RAG) or Search	Adapt to rapidly-changing knowledgebase
Create a medical LLM / Cybersecurity LLM	Continual Pre-training + Post-training	Inject large-scale domain knowledge (>1B tokens) not seen during pre-training
Follow 20+ instructions tightly; Improve targeted capabilities ("Create a strong SQL / function calling / reasoning model")	Post-training	Reliably change model behavior & improve targeted capabilities; May degrade other capabilities if not done right

Is Post-Training Really Needed?

Use Cases	Methods	Characteristics
Follow a few instructions (do not discuss XXX)	Prompting	Simple yet brittle: models may not always follow all instructions
Query real-time database or knowledgebase	Retrieval- Augmented Generation (RAG) or Search	Adapt to rapidly-changing knowledgebase
Create a medical LLM / Cybersecurity LLM	Continual Pre-training + Post-training	Inject large-scale domain knowledge (>1B tokens) not seen during pre-training
Follow 20+ instructions tightly; Improve targeted capabilities ("Create a strong SQL / function calling / reasoning model")	Post-training	Reliably change model behavior & improve targeted capabilities; May degrade other capabilities if not done right

Is Post-Training Really Needed?

Use Cases	Methods	Characteristics
Follow a few instructions (do not discuss XXX)	Prompting	Simple yet brittle: models may not always follow all instructions
Query real-time database or knowledgebase	Retrieval- Augmented Generation (RAG) or Search	Adapt to rapidly-changing knowledgebase
Create a medical LLM / Cybersecurity LLM	Continual Pre-training + Post-training	Inject large-scale domain knowledge (>1B tokens) not seen during pre-training
Follow 20+ instructions tightly; Improve targeted capabilities ("Create a strong SQL / function calling / reasoning model")	Post-training	Reliably change model behavior & improve targeted capabilities; May degrade other capabilities if not done right

Is Post-Training Really Needed?

Use Cases	Methods	Characteristics
Follow a few instructions (do not discuss XXX)	Prompting	Simple yet brittle: models may not always follow all instructions
Query real-time database or knowledgebase	Retrieval- Augmented Generation (RAG) or Search	Adapt to rapidly-changing knowledgebase
Create a medical LLM / Cybersecurity LLM	Continual Pre-training + Post-training	Inject large-scale domain knowledge (>1B tokens) not seen during pre-training
Follow 20+ instructions tightly; Improve targeted capabilities ("Create a strong SQL / function calling / reasoning model")	Post-training	Reliably change model behavior & improve targeted capabilities; May degrade other capabilities if not done right

Supplemental Material

- Chapter 9 on Post-Training from the J&M textbook:
- PPO for LLMs: A Guide for Normal People, Cameron R. Wolfe, Oct 2025.
- Direct Preference Optimization: Your Language Model is Secretly a Reward Model, Rafailov et al., NeurIPS 2023.
- Spinning Up in Deep RL, OpenAI, 2018.