香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

COMP 4901B

Large Language Models

Reinforcement Learning from Human Feedback (RLHF)

Junxian He

Oct 17, 2025

Part of the slides are adapted from Stanford CS336

Why Do We Want Reinforcement Learning

SFT

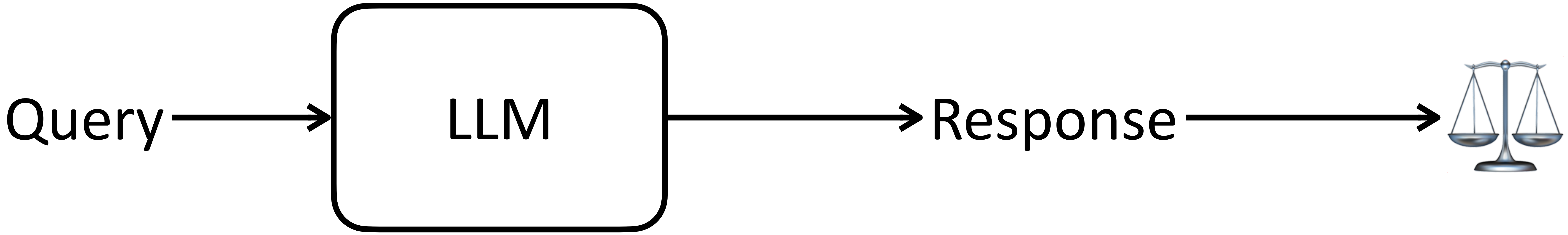
- Imitation external data
- Performance limited by external data

annotation cost

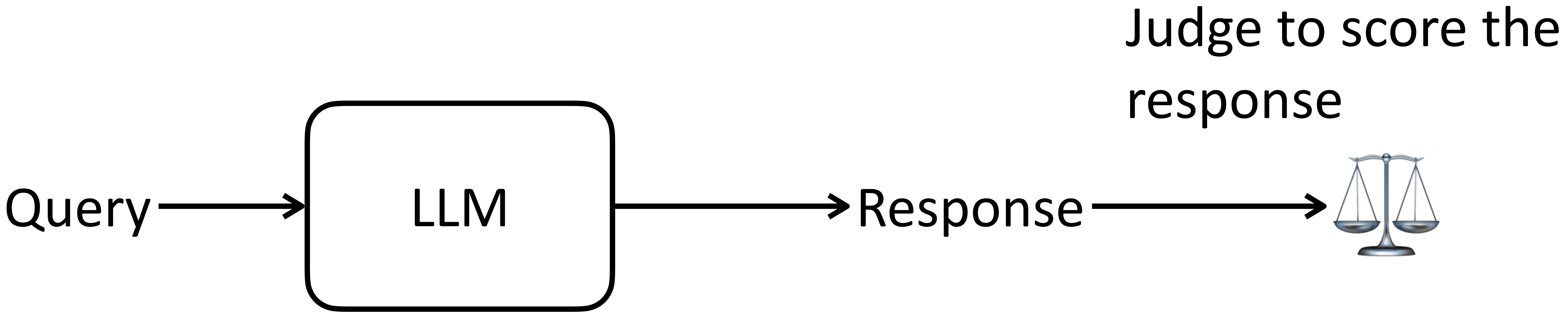
RL

- Maximize reward rather than imitation
- The model may surpass humans (e.g., AlphaGo)

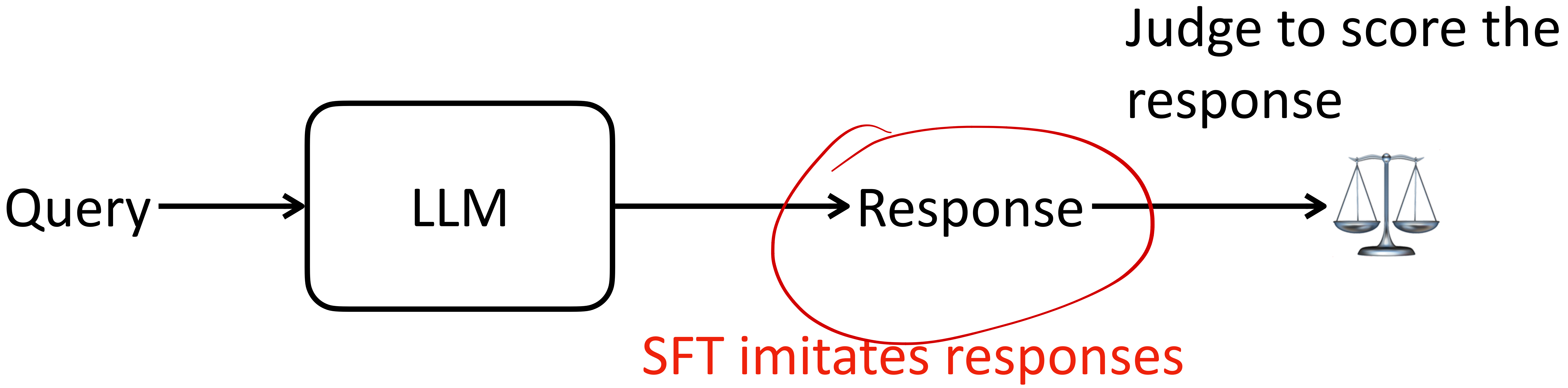
Why Do We Want Reinforcement Learning



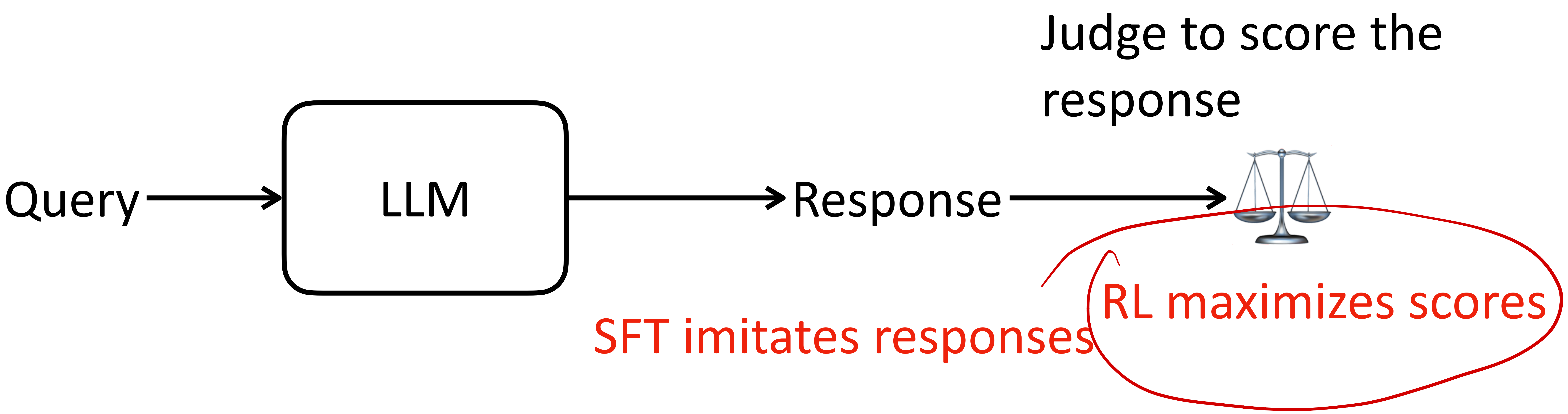
Why Do We Want Reinforcement Learning



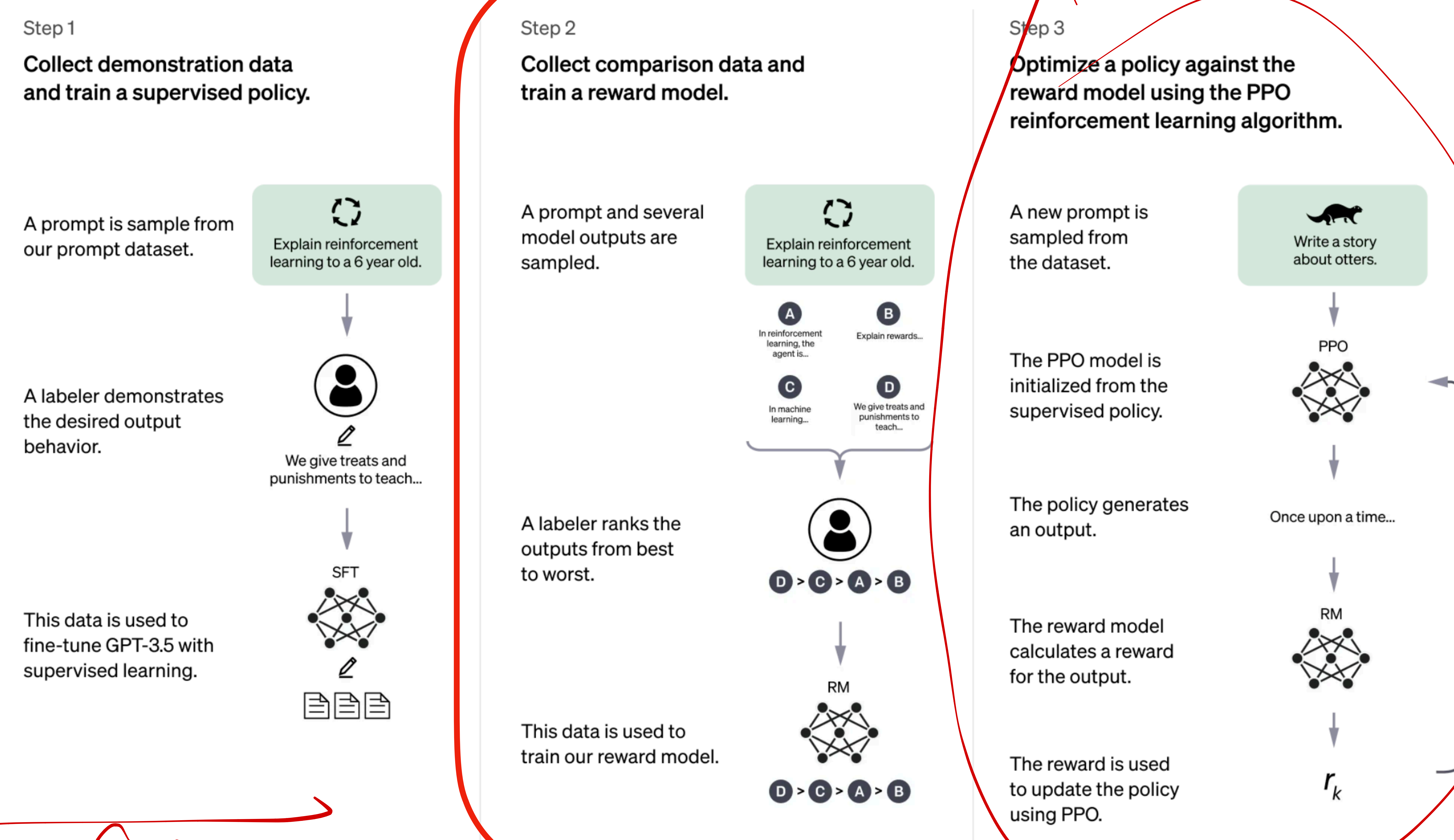
Why Do We Want Reinforcement Learning



Why Do We Want Reinforcement Learning



Review: Reinforcement Learning in LLMs



SFT

Review: From Imitation to Optimization

Review: From Imitation to Optimization

Imitation (SFT)

Fit $\hat{p}(y|x) \approx p^*(y|x)$ for some reference distribution $p^*(y|x)$

- Pure generative modeling perspective
- Requires samples from reference policy

Review: From Imitation to Optimization

Imitation (SFT)

Fit $\hat{p}(y|x) \approx p^*(y|x)$ for some reference distribution $p^*(y|x)$

- Pure generative modeling perspective
- Requires samples from reference policy

Optimization (RLHF)

Find $\hat{\boldsymbol{p}}(\boldsymbol{y}|\boldsymbol{x})$ such that $\max_{p} E_p[R(y,x)]$ for a reward $R(y,x)$

- Maximize some reward function that we can measure
- LMs are policies, not a model of some distribution

Review: Reward Optimization in Language Models

Review: Reward Optimization in Language Models

Objective:

$$\theta = \arg\max_{\theta} \mathbb{E}_{x \sim p_\theta(x)} R(x)$$

LLM

Review: Reward Optimization in Language Models

Objective:

$$\theta = \arg\max_{\theta} \mathbb{E}_{x \sim p_\theta(x)} R(x)$$

X is the language sequence, R is the reward function, scores the generated response (we can consider R as a human or a model)

Review: Reward Optimization in Language Models

unrelated to θ

Objective:

$$\theta = \arg\max_{\theta} \mathbb{E}_{x \sim p_\theta(x)} R(x)$$

X is the language sequence, R is the reward function, scores the generated response (we can consider R as a human or a model)

undifferential

How to do gradients over this objective?

x is discrete

Review: Reward Optimization in Language Models

Objective:

$$\theta = \arg\max_{\theta} \mathbb{E}_{x \sim p_\theta(x)} R(x)$$

X is the language sequence, R is the reward function, scores the generated response (we can consider R as a human or a model)

How to do gradients over this objective?

Gradient estimation (policy gradient):

Review: Reward Optimization in Language Models

Objective:

$$\theta = \arg\max_{\theta} \mathbb{E}_{x \sim p_\theta(x)} R(x)$$

X is the language sequence, R is the reward function, scores the generated response (we can consider R as a human or a model)

How to do gradients over this objective?

Gradient estimation (policy gradient):

$$\hat{g} = \mathbb{E}_{x \sim p_\theta(x)} R(x) \nabla_\theta \log p_\theta(x)$$

$\hat{g}$ is the gradient (not objective)

Review: REINFORCE / Policy Gradient

$$\hat{g} = \mathbb{E}_{x \sim p_\theta(x)} R(x) \nabla_\theta \log p_\theta(x)$$

Review: REINFORCE / Policy Gradient

$$\hat{g} = \mathbb{E}_{x \sim p_\theta(x)} R(x) \boxed{\nabla_\theta \log p_\theta(x)}$$

How to implement?

Review: REINFORCE / Policy Gradient

$$\hat{g} = \mathbb{E}_{x \sim p_\theta(x)} R(x) \nabla_\theta \log p_\theta(x)$$

How to implement?

$$\text{Objective} = \sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)}) \qquad x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$$

$p_\theta(x)$

take away the gradient from x

Review: REINFORCE / Policy Gradient

$$\hat{g} = \mathbb{E}_{x \sim p_\theta(x)} R(x) \nabla_\theta \log p_\theta(x)$$

How to implement?

Objective = $\displaystyle \sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)})$ $\quad x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$

This objective looks kinda like weighted log likelihood maximization?

Review: REINFORCE / Policy Gradient

$$\hat{g} = \mathbb{E}_{x \sim p_\theta(x)} R(x) \nabla_\theta \log p_\theta(x)$$

How to implement?

Objective = $\displaystyle\sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)}) \qquad x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$

This objective looks kinda like weighted log likelihood maximization?

What is different?

Review: REINFORCE / Policy Gradient

$$\hat{g} = \mathbb{E}_{x \sim p_\theta(x)} R(x) \nabla_\theta \log p_\theta(x)$$

How to implement?

Objective = $\displaystyle\sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)})$ $\quad x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$

This objective looks kinda like weighted log likelihood maximization?

What is different?

1. Have a weight of R(x)

Review: REINFORCE / Policy Gradient

data model

SFT: fixed

RL: data model

$$\hat{g} = \mathbb{E}_{x \sim p_\theta(x)} R(x) \nabla_\theta \log p_\theta(x)$$

How to implement?

Objective = $\sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)})$ $\quad x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$

This objective looks kinda like weighted log likelihood maximization?

What is different? model is changing during training

1. Have a weight of R(x)
2. The data x is sampled from the model itself, not from a static dataset

Review: REINFORCE / Policy Gradient

Objective = $\sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)}) \qquad x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$

Review: REINFORCE / Policy Gradient

Objective = $\sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)}) \qquad x^{(1)}, ..., x^{(n)} \sim p_\theta(x)$

1. Have a weight of R(x)
2. The data x is sampled from the model itself, not from a static dataset

Review: REINFORCE / Policy Gradient

Objective = $\sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)}) \qquad x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$

1. Have a weight of R(x)
2. The data x is sampled from the model itself, not from a static dataset

This equation is not that complex, just view it as a weighted likelihood maximization

Review: REINFORCE / Policy Gradient

Objective = $\sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)}) \qquad x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$

1. Have a weight of R(x)
2. The data x is sampled from the model itself, not from a static dataset

This equation is not that complex, just view it as a weighted likelihood maximization

This is the simplest form of RL, many other RL algorithms (PPO, GRPO) are more like variants of this simple equation with the same spirit

Review: REINFORCE / Policy Gradient

Objective = $\sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)}) \qquad x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$

1. Have a weight of R(x)
2. The data x is sampled from the model itself, not from a static dataset

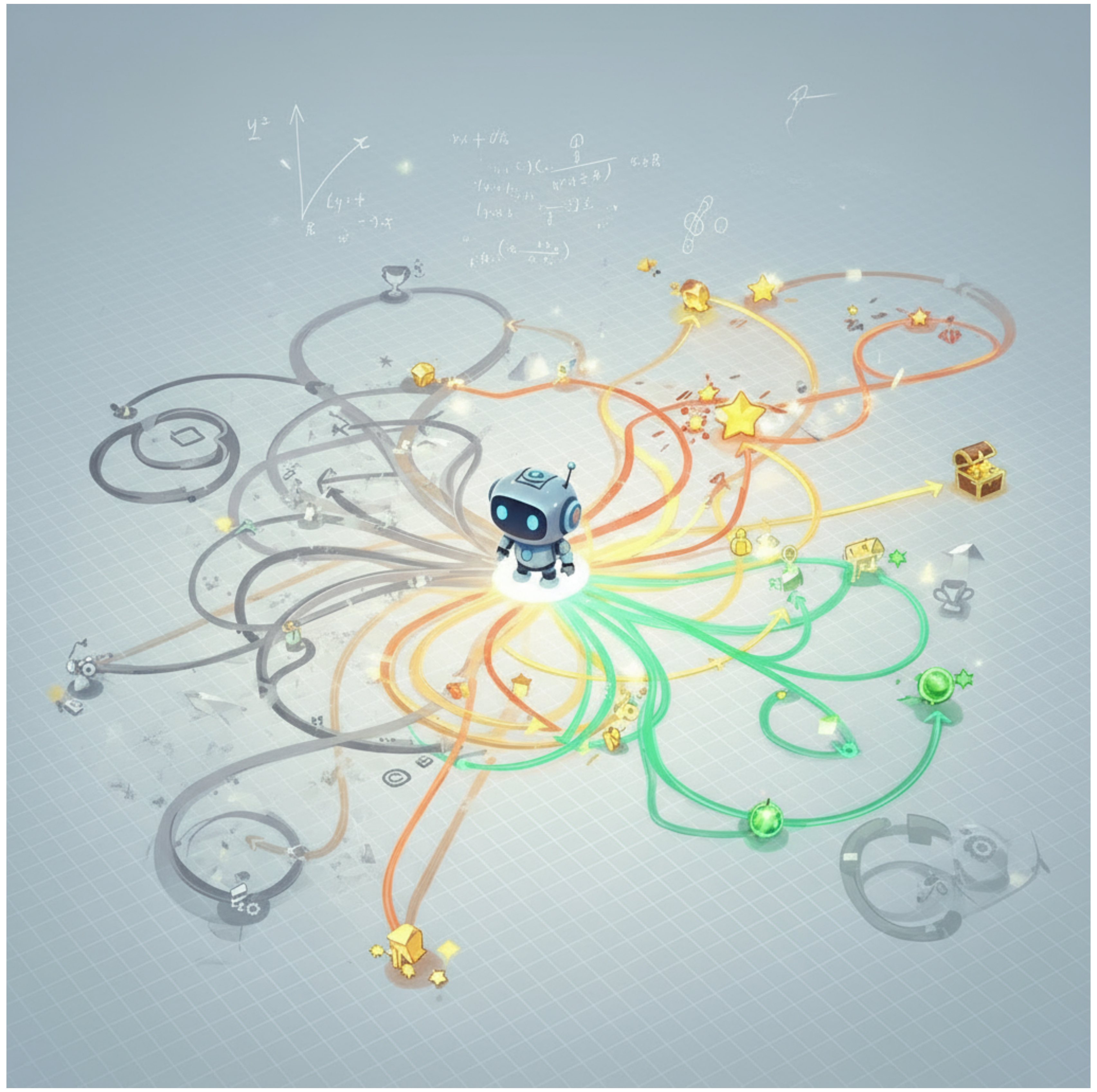
This equation is not that complex, just view it as a weighted likelihood maximization

This is the simplest form of RL, many other RL algorithms (PPO, GRPO) are more like variants of this simple equation with the same spirit

We can see why RL is called "self-improving", and it is trained by "synthetic data"

Review: REINFORCE / Policy Gradient

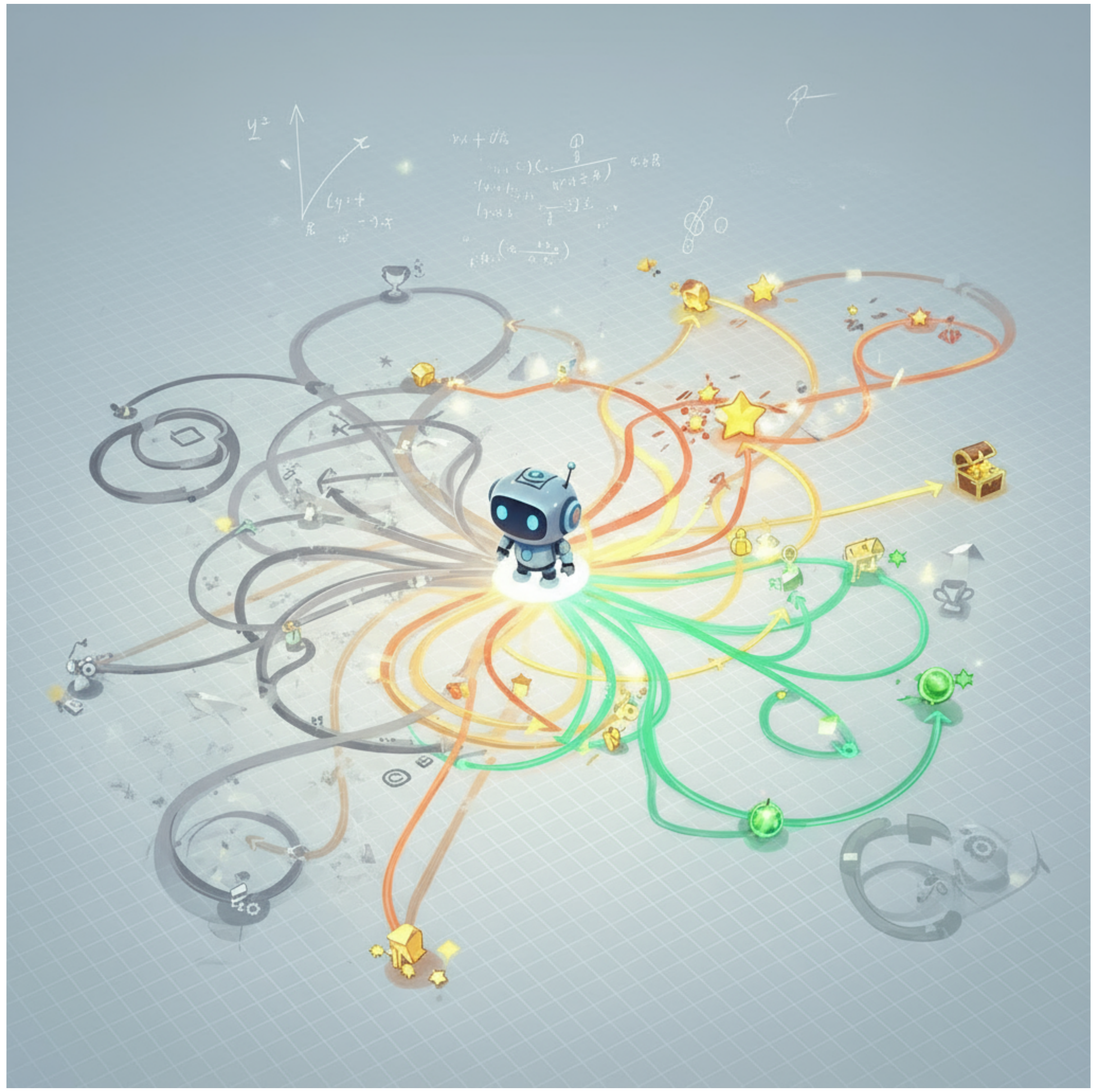
Objective = $\sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)})$ $\quad x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$



Review: REINFORCE / Policy Gradient

Objective = $\sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)})$ $\qquad x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$

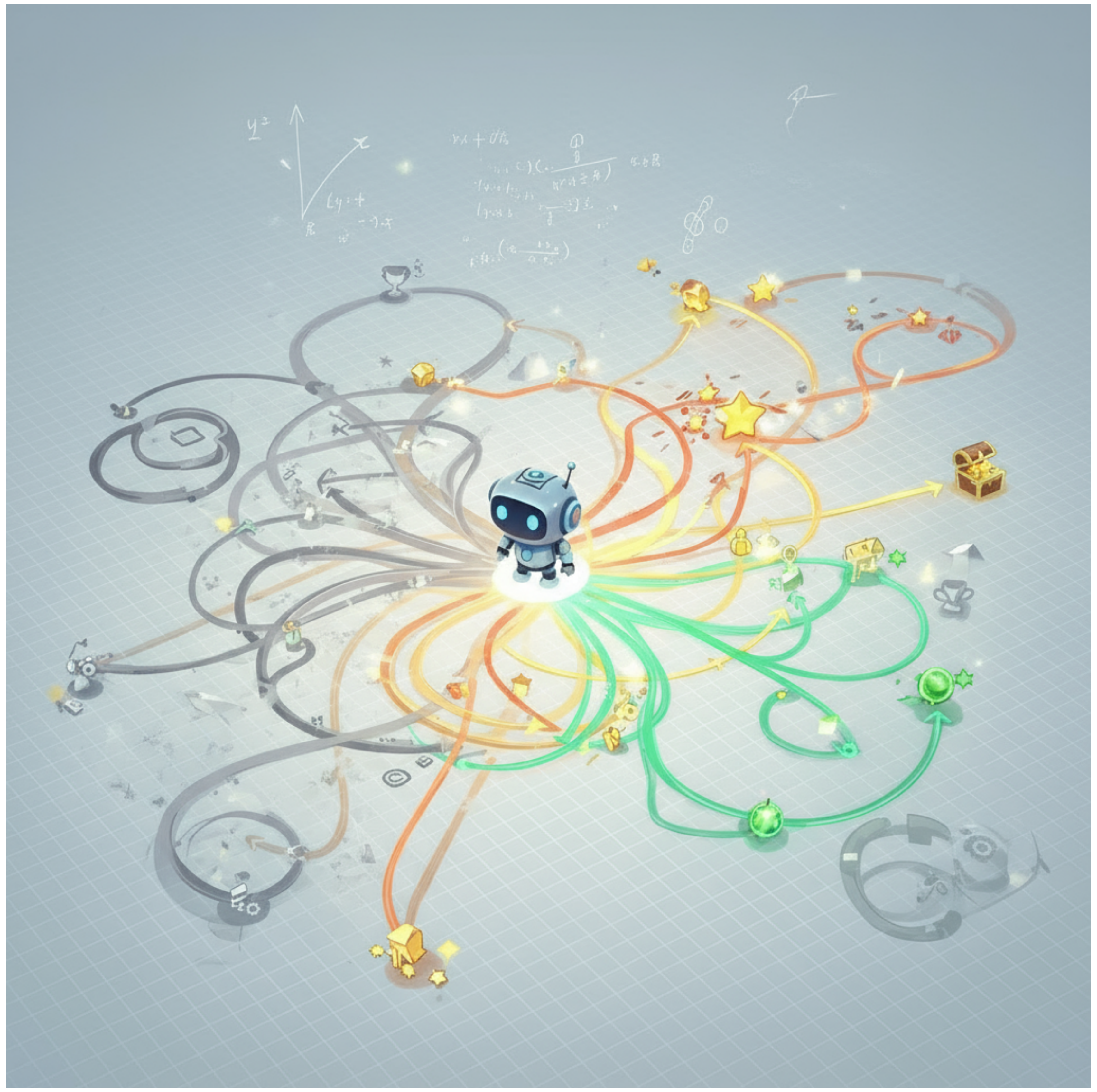
1. Sample data from the model (or we call policy) itself (exploration)



Review: REINFORCE / Policy Gradient

Objective = $\sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)})$ $\quad x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$

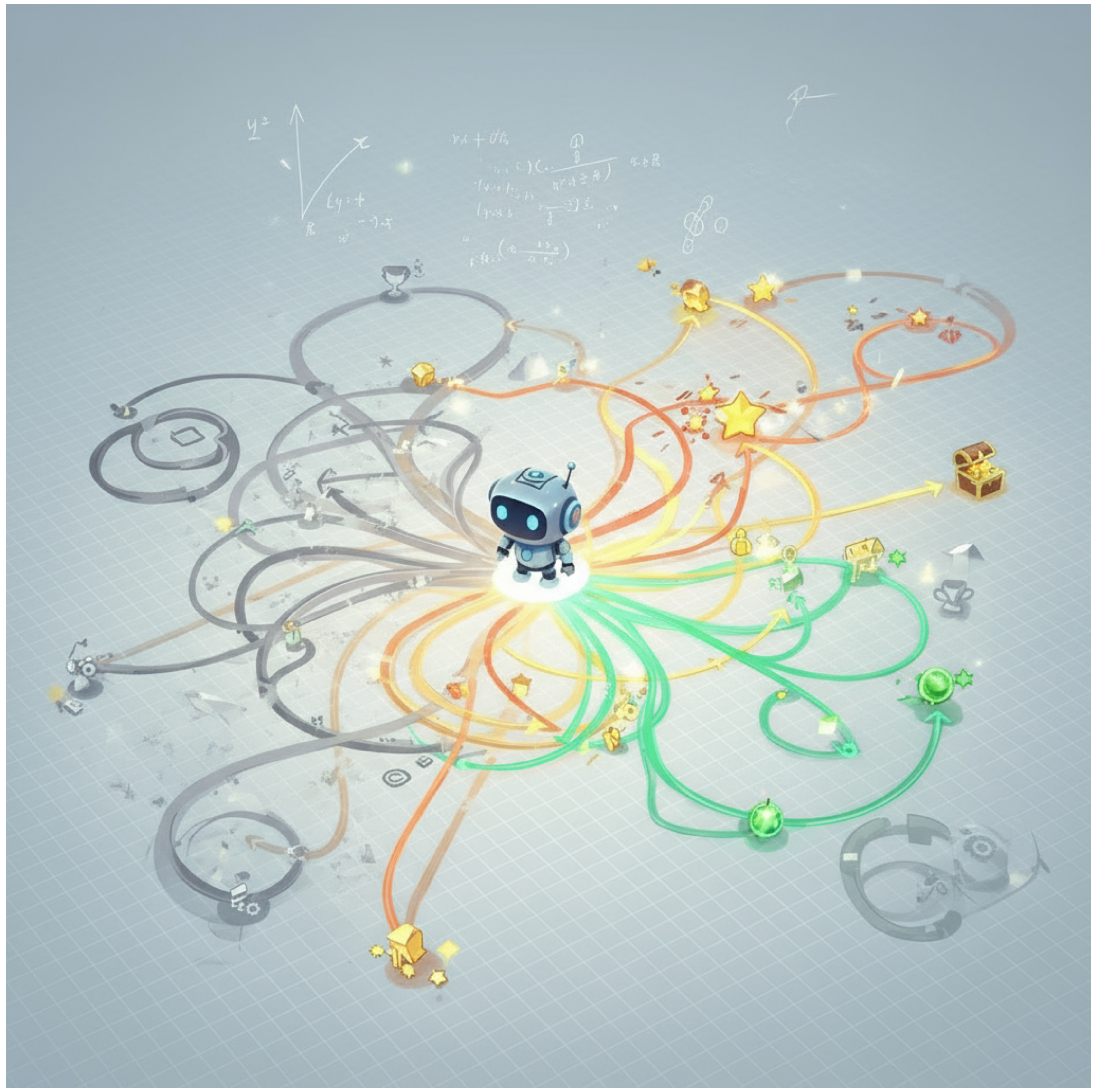
1. Sample data from the model (or we call policy) itself (exploration)
2. A reward function judges whether the explored data is good or bad



Review: REINFORCE / Policy Gradient

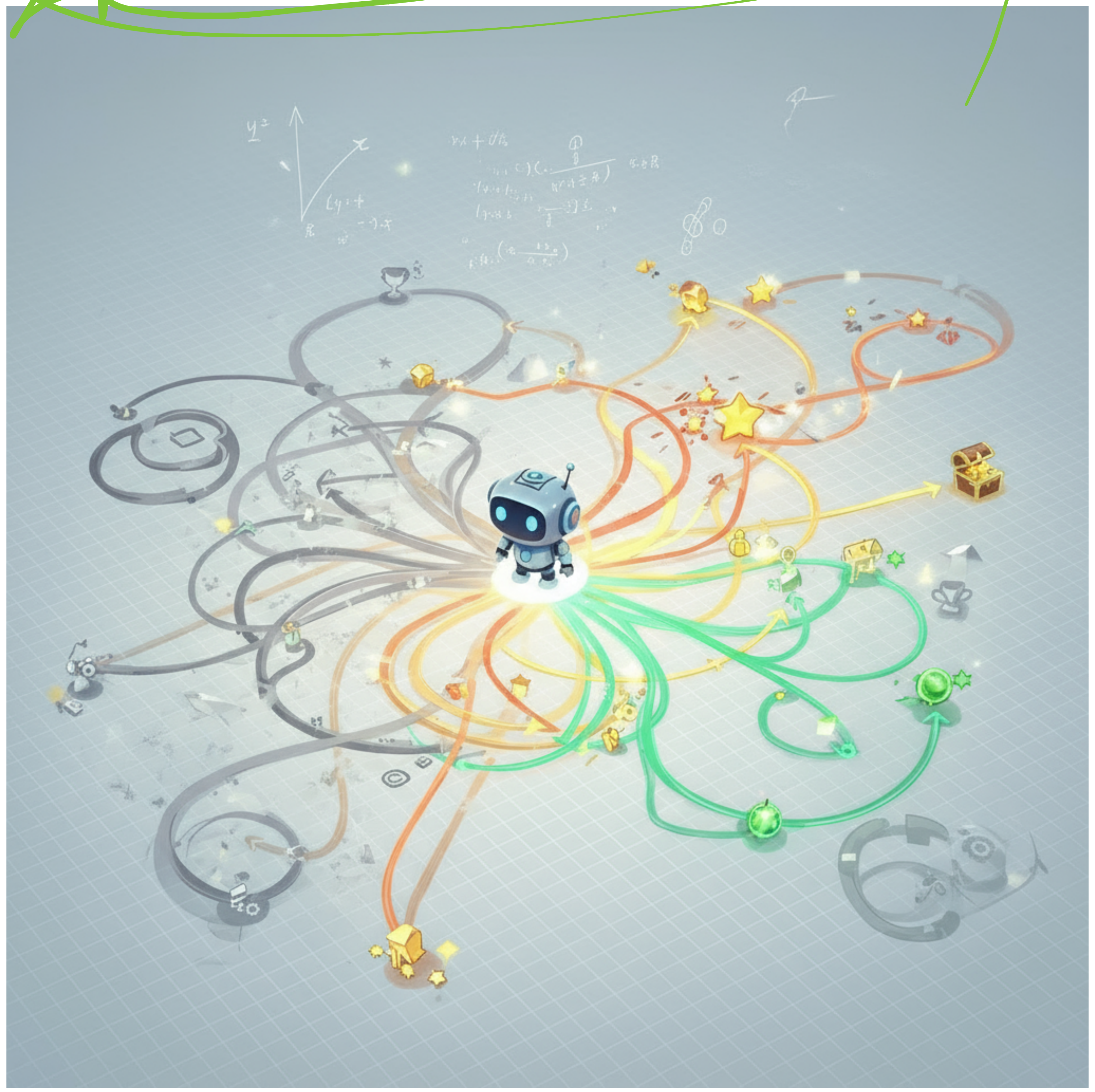
$$\text{Objective} = \sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)}) \qquad x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$$

Reinforcement learning is a mixed art of both training and inference during training time



Review: REINFORCE / Policy Gradient

$$\text{Objective} = \sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)}) \qquad x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$$



Reinforcement learning is a mixed art of both training and inference during training time

Why?

Review: REINFORCE / Policy Gradient

Objective = $\sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)})$ $\qquad x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$

Review: REINFORCE / Policy Gradient

Objective = $\displaystyle\sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)})$ $\quad x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$

Training

Review: REINFORCE / Policy Gradient

Objective = $\sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)})$ $\quad x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$

Training

Inference

Review: REINFORCE / Policy Gradient

Objective = $\sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)})$ $\quad x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$

Training $\quad$ Inference

changing

Each training step, the algorithm needs to run inference again

Review: REINFORCE / Policy Gradient

x

Objective = $$\sum_{i=1}^{n} \frac{1}{n} \cancel{R(x^{(i)})} \log p_\theta(x^{(i)}) \quad x^{(1)}, ..., x^{(n)} \sim p_\theta(x)$$

Training — machine

Inference — another machine

autoregressive

$O(L^2)$

L: sequence length

Each training step, the algorithm needs to run inference again

Is this efficient?

autoregressive is sequential in nature

time complexity: $O(L)$

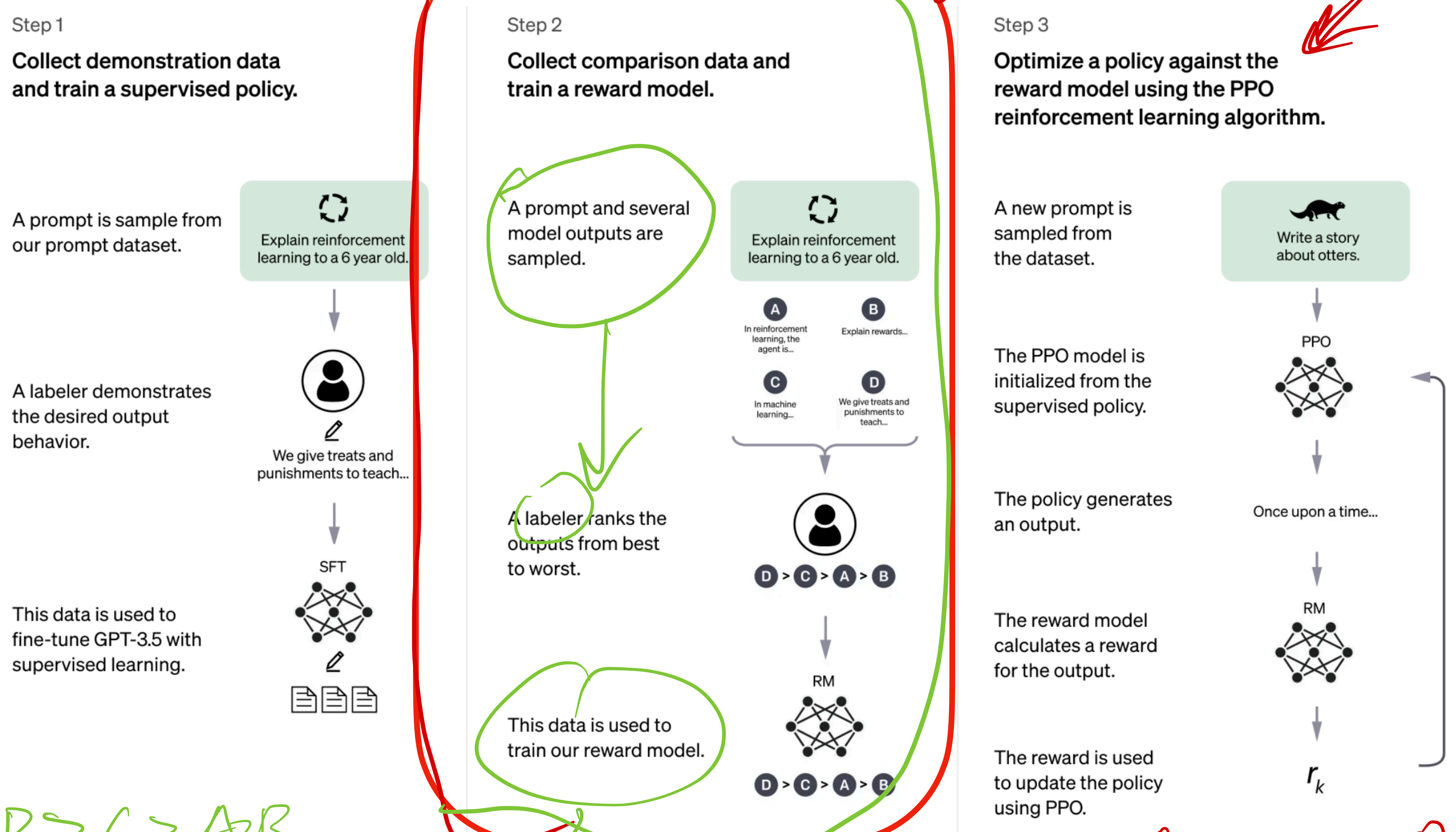
REINFORCE / Policy Gradient for Language Models

$x_1^{(i)}$

$$\text{Objective} = \sum_{i=1}^{n} \frac{1}{n} R(\{x_1^{(i)}, x_2^{(i)}, \ldots, x_t^{(i)}\}) \sum_{j=1}^{t} \log p_\theta(x_j^{(i)} | x_{<j}^{(i)})$$

$R(\cdot)$

Training a Reward Model in RLHF



D > C > A > B

R(·)

R(A) = 0.7

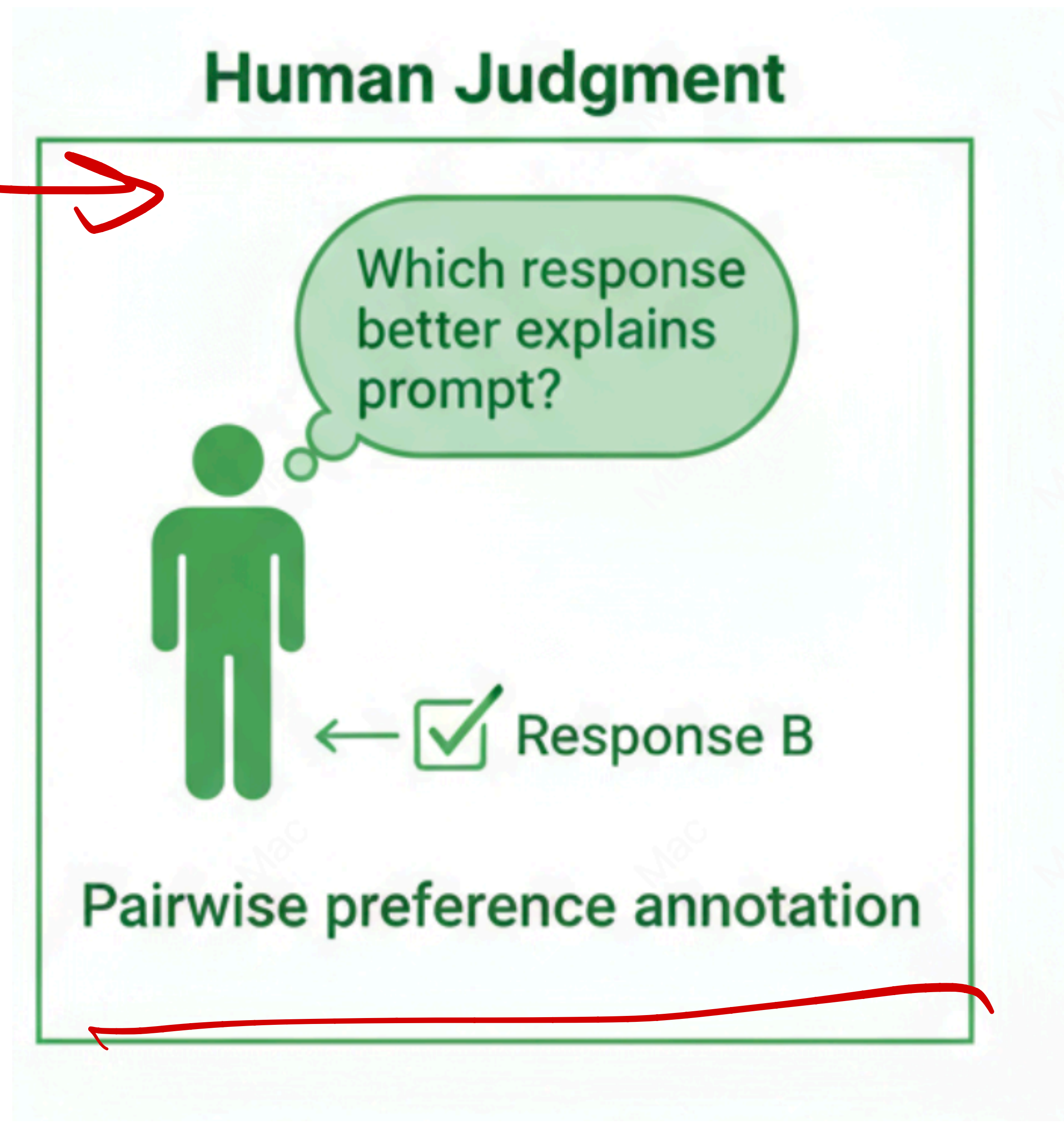
(0, 1)

Reward Model in RLHF: Pairwise Human Annotation

Explain the theory of relativity in simple terms

Relativity means that time and space are connected, and they change depending on how fast you move.

Einstein's theory shows that space and time are not fixed but bend and stretch based on gravity and motion.



Prompt

Responses

human → score (A)

Training a Reward Model

binary classification / logistic regression: whether y_w is preferred over y_l

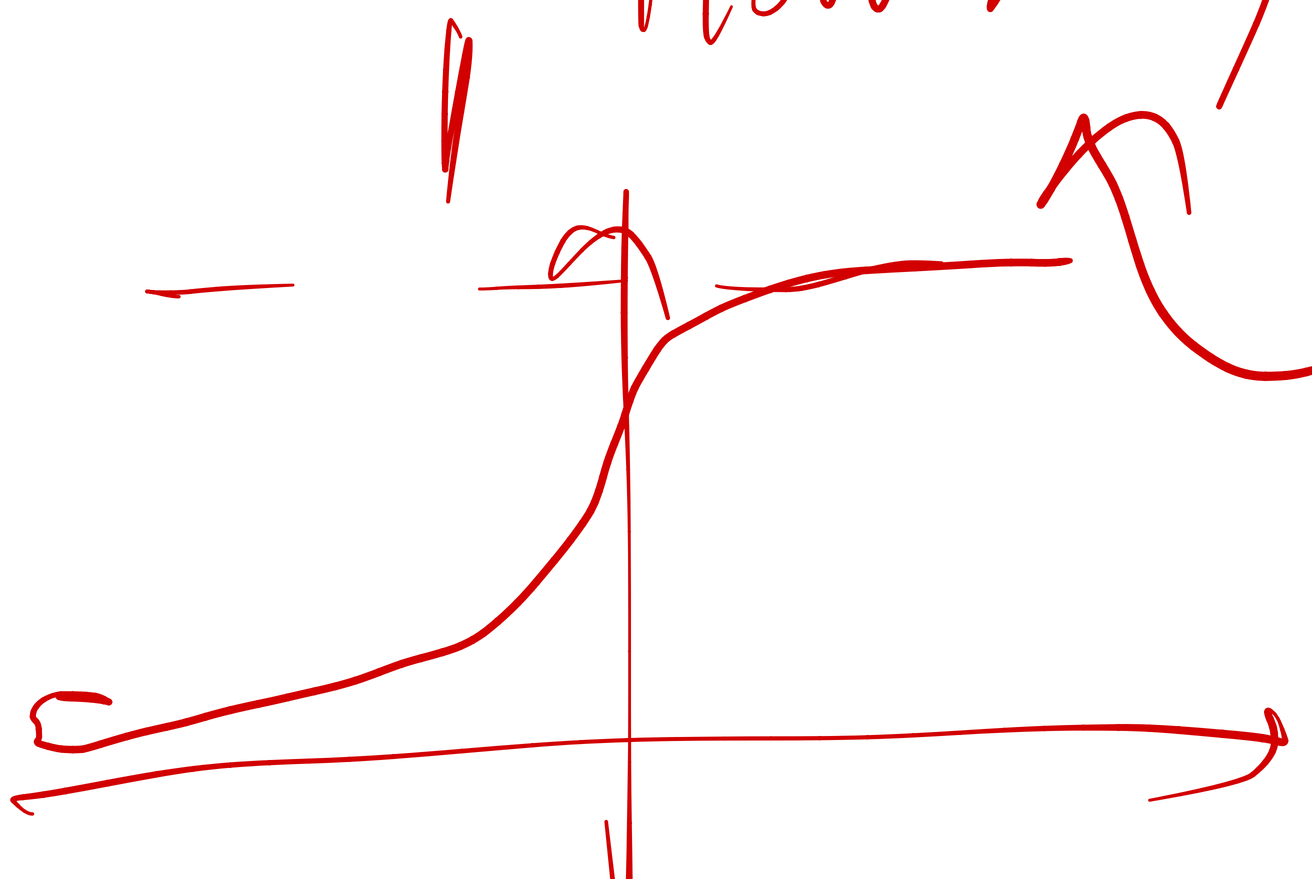
Suppose we have K responses and have them ranked by humans, then for all possible pairs of responses, y_w is the preferred one, y_l is the less preferred one, the objective of reward model $r_\theta(x, y)$ is:

$$\text{Objective} = -\frac{1}{\binom{K}{2}} E_{(x,y_w,y_l)\sim D}\left[\log\left(\sigma\left(r_\theta\left(x, y_w\right) - r_\theta\left(x, y_l\right)\right)\right)\right]$$

How many? $\binom{K}{2}$ C_K^2

$r_\theta(x, y_w) \rightarrow$ score

$r_\theta(x, y_l) \rightarrow$ score



$\sigma = \frac{1}{1+e^{-x}}$

logistic regression NN

$\log \sigma(f(x))$

$\log \sigma(\theta^T x)$

$f(x)$

$\log \sigma(r(x, y_w) - r(x, y_l))$

Training a Reward Model

Suppose we have K responses and have them ranked by humans, then for all possible pairs of responses, y_w is the preferred one, y_l is the less preferred one, the objective of reward model $r_\theta(x, y)$ is:

$$\text{Objective} = -\frac{1}{\binom{K}{2}} E_{(x, y_w, y_l) \sim D} \left[\log\left(\sigma\left(r_\theta\left(x, y_w\right) - r_\theta\left(x, y_l\right)\right)\right)\right]$$

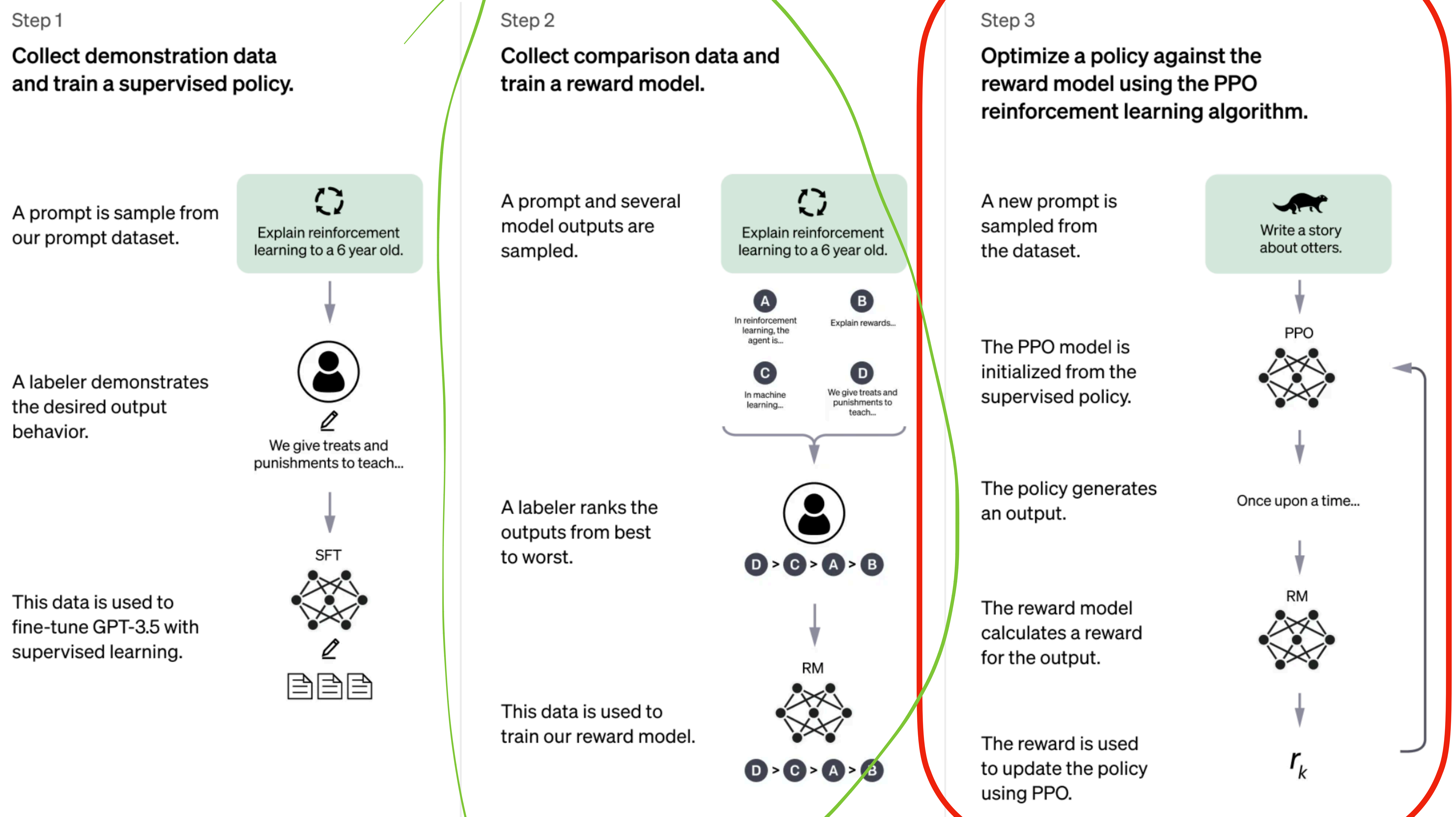
In practice, we can just sample a subset of pairs, rather than enumerating all pairs

RLHF

R(x)

0.7

0.5



Step 1
Collect demonstration data and train a supervised policy.
A prompt is sample from our prompt dataset.
Explain reinforcement learning to a 6 year old.
A labeler demonstrates the desired output behavior.
We give treats and punishments to teach...
This data is used to fine-tune GPT-3.5 with supervised learning.
SFT
Step 2
Collect comparison data and train a reward model.
A prompt and several model outputs are sampled.
Explain reinforcement learning to a 6 year old.
A
In reinforcement learning, the agent is...
B
Explain rewards...
C
In machine learning...
D
We give treats and punishments to teach...
A labeler ranks the outputs from best to worst.
D > C > A > B
This data is used to train our reward model.
RM
D > C > A > B
Step 3
Optimize a policy against the reward model using the PPO reinforcement learning algorithm.
A new prompt is sampled from the dataset.
Write a story about otters.
The PPO model is initialized from the supervised policy.
PPO
The policy generates an output.
Once upon a time...
The reward model calculates a reward for the output.
RM
The reward is used to update the policy using PPO.
r_k

RLHF

human feedback

After we have the reward model, we fix it and do RL, for example, with REINFORCE

$$\text{Objective} = \sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)}) \qquad x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$$

Ouyang et al. Training language models to follow instructions with human feedback. 2022

Schulman et al. Proximal Policy Optimization Algorithms. 2017

RLHF

After we have the reward model, we fix it and do RL, for example, with REINFORCE

Objective = $$\sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)}) \qquad x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$$

But in fact, the original RLHF uses a more complex objective called Proximal Policy Optimization (PPO): (no need to know the exact details in this course)

a s

$$L^{\mathrm{PPO}}(\theta) = \mathbb{E}_t \left[\min \left(\frac{p_\theta(a_t \mid s_t)}{p_{\theta_{\mathrm{old}}}(a_t \mid s_t)} \hat{A}_t, \operatorname{clip}\left(\frac{p_\theta(a_t \mid s_t)}{p_{\theta_{\mathrm{old}}}(a_t \mid s_t)}, 1-\epsilon, 1+\epsilon \right) \hat{A}_t \right) \right]$$

advantage $R(\cdot)$ $a_t | s_t \sim p_{\theta_{old}}(\cdot|s_t)$

Ouyang et al. Training language models to follow instructions with human feedback. 2022
Schulman et al. Proximal Policy Optimization Algorithms. 2017

rule-based RLVR deeseek-R1

RLHF

After we have the reward model, we fix it and do RL, for example, with REINFORCE

$$\text{Objective} = \sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)}) \qquad x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$$

But in fact, the original RLHF uses a more complex objective called Proximal Policy Optimization (PPO): (no need to know the exact details in this course)

$$L^{\mathrm{PPO}}(\theta) = \mathbb{E}_t \left[\min\left(\frac{p_\theta(a_t \mid s_t)}{p_{\theta_{\mathrm{old}}}(a_t \mid s_t)} \hat{A}_t,\ \mathrm{clip}\left(\frac{p_\theta(a_t \mid s_t)}{p_{\theta_{\mathrm{old}}}(a_t \mid s_t)}, 1-\epsilon, 1+\epsilon \right) \hat{A}_t \right) \right]$$

RLHF is one method for “preference learning”

Ouyang et al. Training language models to follow instructions with human feedback. 2022

Schulman et al. Proximal Policy Optimization Algorithms. 2017

pretraining → SFT → RLHF

most important

ChatGPT

average Users

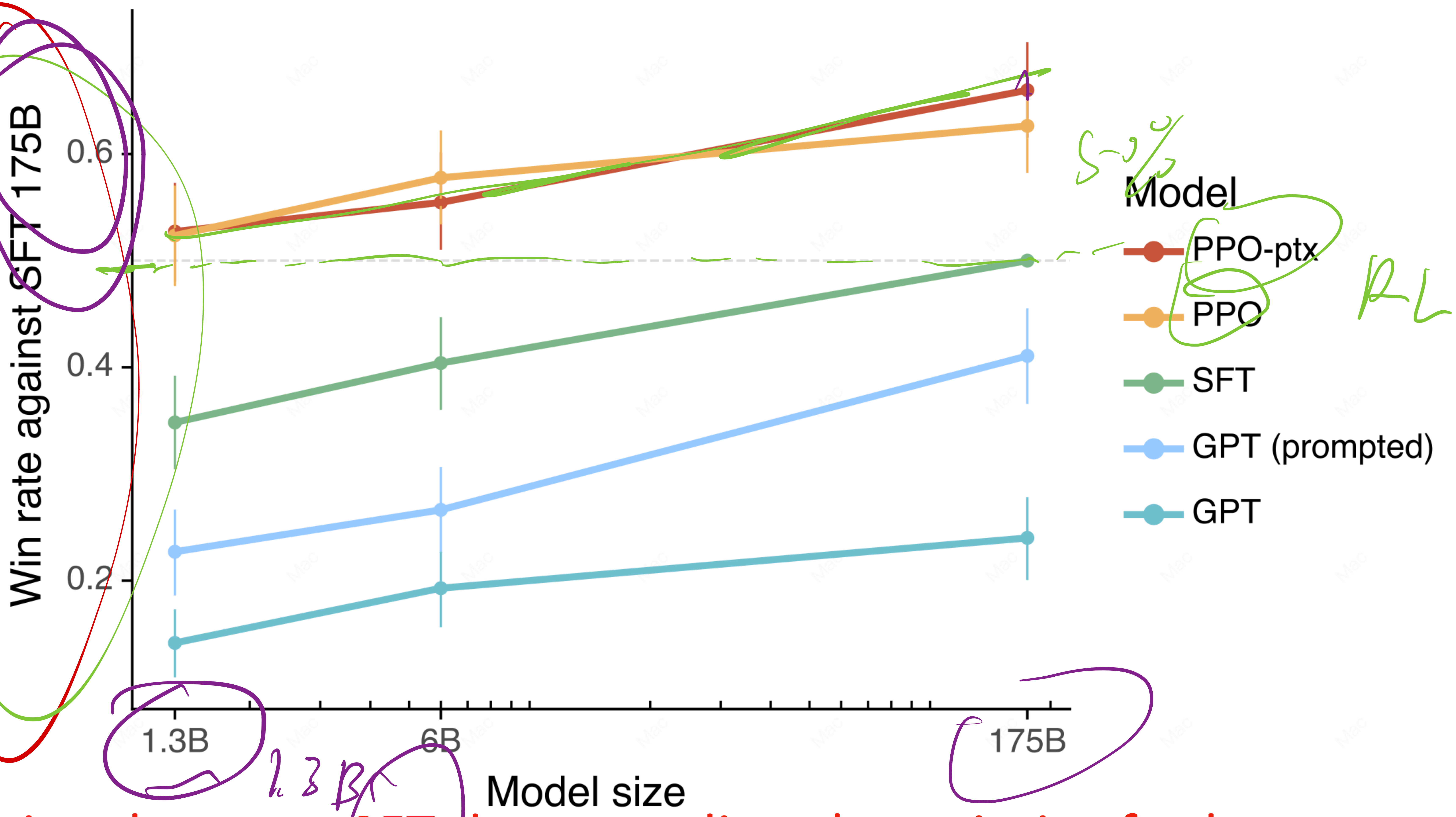
Why is RLHF so Important?

If we build products for humans, then optimizing for human feedback is the most important

Human Feedback outperforms SFT in Human Evaluation

human eval

75-85%



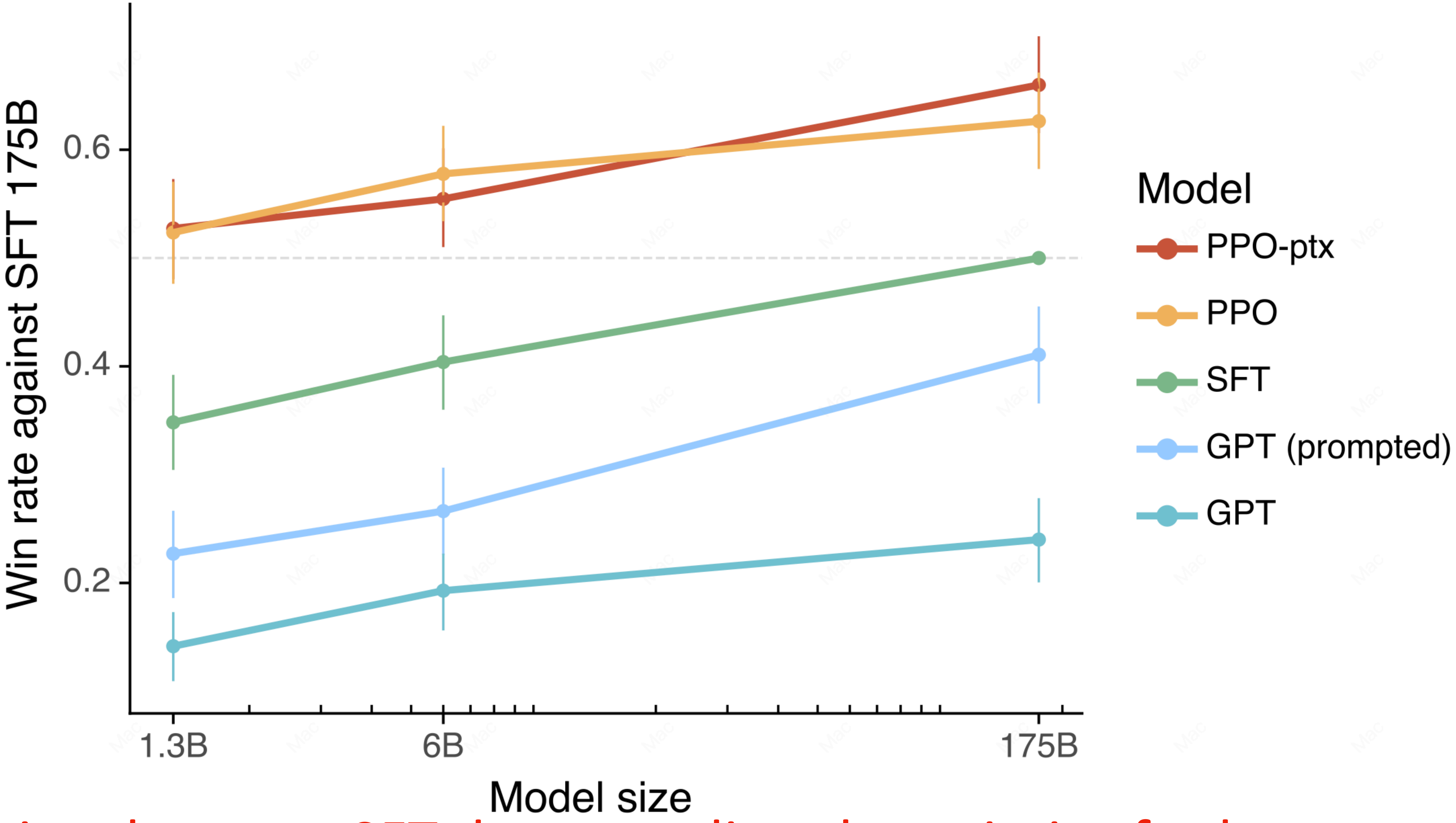
5-9%

RL

1.3B

Unsurprising, because SFT does not directly optimize for human preference

Human Feedback outperforms SFT in Human Evaluation



Unsurprising, because SFT does not directly optimize for human preference
RLHF directly aligns with evaluation, but SFT has a gap

Reinforcement Learning from Human Feedback Data

Basic Setup


Hi test22. Not your MTurk id? Logout

Your annotation progress for this HIT: 0/6 completed.

Sample 1 of 6.
Lets you skip forward (all samples still need to be marked): → Next sample.
Returns you back to the next un-marked sample to mark: Go to next un-marked sample.

▶ Annotation Guidelines

Instruction

```
Tell me about self driving cars
```

Input

AI Response 1

```
Self driving cars are vehicles that are capable of detecting their surroundings
and navigating without any input from a human driver. They use a variety of
sensors, such as cameras, radar, and lidar, to detect objects in their
environment and use advanced algorithms to make decisions about how to maneuver
around them. Self driving cars are becoming increasingly popular and are being
used in a variety of applications, such as ride-hailing services and autonomous
delivery services.
```

AI Response 2

```
Self driving cars are cars that are equipped with sensors, cameras, and
software, which allows them to navigate without the need for a driver. Self
driving cars are able to monitor their environment and navigate without the
assistance of a driver. Self driving cars use a combination of technologies
such as GPS, radar, lidar, and computer vision to detect their environment and
avoid obstacles. They are also able to update their maps and adjust their
routes to account for variables such as traffic and construction.

The main benefit of self driving cars is their ability to reduce crashes and
save lives. Autonomous vehicles have the potential to save up to 30,000 lives
annually. They are also more energy efficient than traditional vehicles, as
they do not require a driver who has to maintain a minimum speed. Autonomous
vehicles also create new opportunities for increased mobility, allowing those
who are unable to drive to get around.
```

Rating

- ○ Response 1 is better.
- ○ Response 1 is only slightly better. (only pick this if it's truly close)
- ○ Response 2 is only slightly better. (only pick this if it's truly close)
- ○ Response 2 is better.

Reinforcement Learning from Human Feedback Data

Basic Setup

Hi test22. Not your MTurk id? Logout

Your annotation progress for this HIT: 0/6 completed.

Sample 1 of 6.

Lets you skip forward (all samples still need to be marked): → Next sample.

Returns you back to the next un-marked sample to mark: Go to next un-marked sample.

▶ **Annotation Guidelines**

Instruction

Tell me about self driving cars

Input

AI Response 1

Self driving cars are vehicles that are capable of detecting their surroundings and navigating without any input from a human driver. They use a variety of sensors, such as cameras, radar, and lidar, to detect objects in their environment and use advanced algorithms to make decisions about how to maneuver around them. Self driving cars are becoming increasingly popular and are being used in a variety of applications, such as ride-hailing services and autonomous delivery services.

AI Response 2

Self driving cars are cars that are equipped with sensors, cameras, and software, which allows them to navigate without the need for a driver. Self driving cars are able to monitor their environment and navigate without the assistance of a driver. Self driving cars use a combination of technologies such as GPS, radar, lidar, and computer vision to detect their environment and avoid obstacles. They are also able to update their maps and adjust their routes to account for variables such as traffic and construction.

The main benefit of self driving cars is their ability to reduce crashes and save lives. Autonomous vehicles have the potential to save up to 30,000 lives annually. They are also more energy efficient than traditional vehicles, as they do not require a driver who has to maintain a minimum speed. Autonomous vehicles also create new opportunities for increased mobility, allowing those who are unable to drive to get around.

Rating

- ○ Response 1 is better.
- ○ Response 1 is only slightly better. (only pick this if it's truly close)
- ○ Response 2 is only slightly better. (only pick this if it's truly close)
- ○ Response 2 is better.

Easier than annotating the responses directly

Reinforcement Learning Relaxes Human Annotations

Reinforcement Learning Relaxes Human Annotations

	Base model →	Supervised learning (one time) →	Pairwise feedback →	RL →	Evaluation
Compute cost	$300k	$100	$100	$100	$0
Annotation cost	$0	$25k	$4k	$0	$50

Reinforcement Learning Relaxes Human Annotations

	Base model →	Supervised learning (one time) →	Pairwise feedback →	RL →	Evaluation
Compute cost	$300k	$100	$100	$100	$0
Annotation cost	$0	$25k	$4k	$0	$50

1. SFT data can be expensive

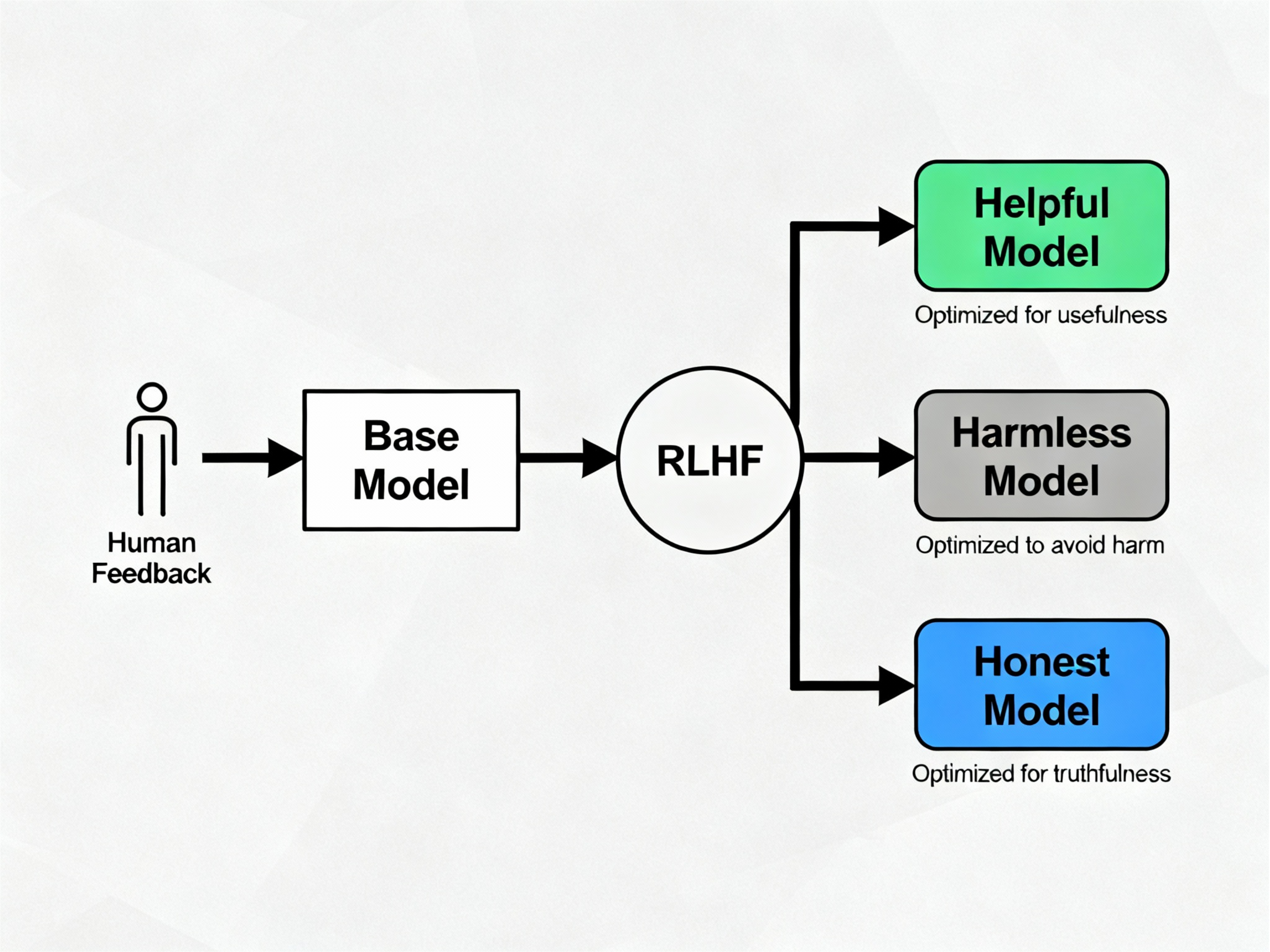
Reinforcement Learning Relaxes Human Annotations

	Base model →	Supervised learning (one time) →	Pairwise feedback →	RL →	Evaluation
Compute cost	\$300k	\$100	\$100	\$100	\$0
Annotation cost	\$0	\$25k	\$4k	\$0	\$50

1. SFT data can be expensive
2. There may be tasks that are much easier for humans to verify than solve, particularly when AI today is outperforming humans

RLHF

Different from SFT, RLHF can be used to optimize certain human preferences.



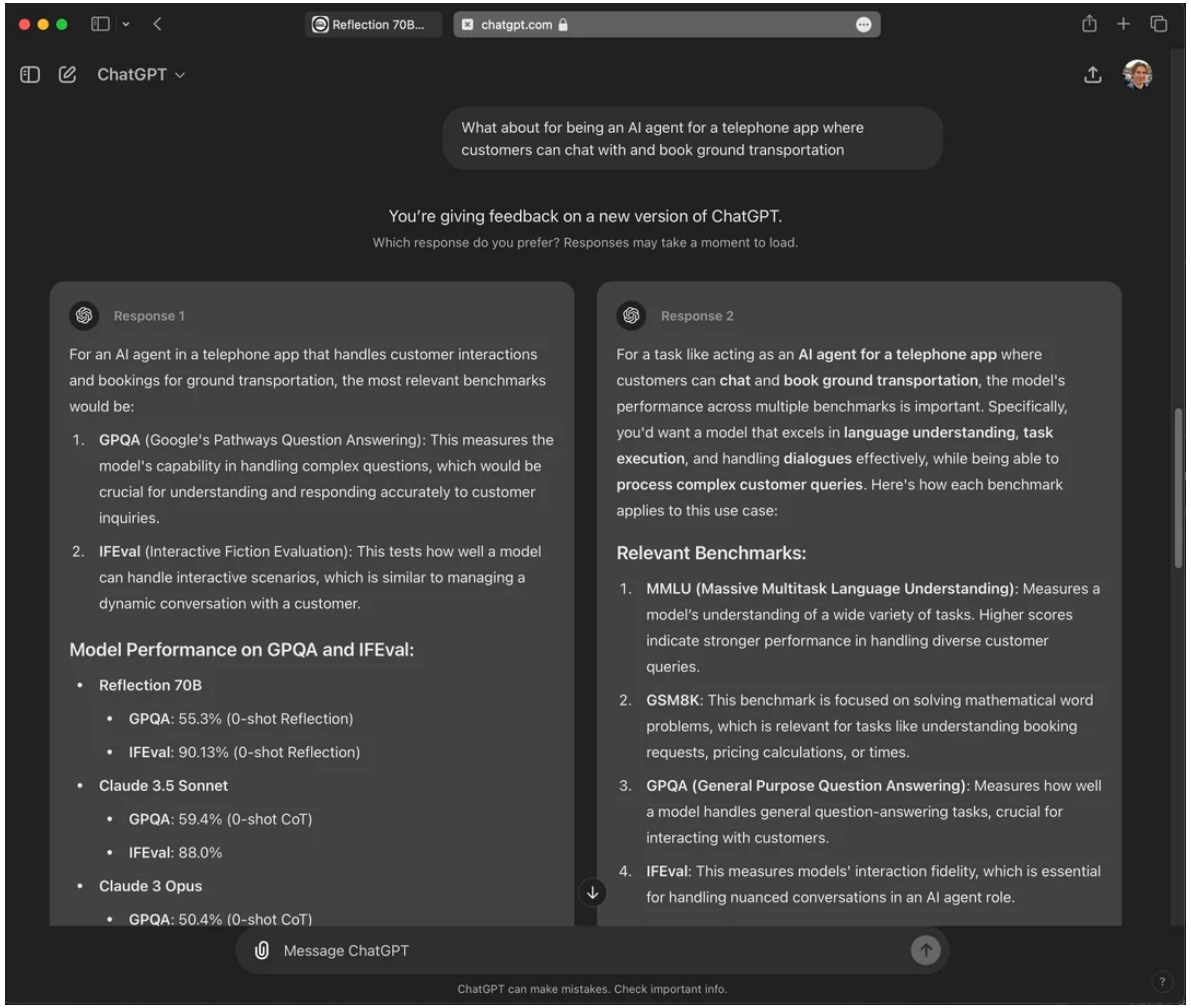
human feedback

$R(\cdot)$

$P(\cdot)$ LLM

Collecting high-quality Human Feedbacks matters a lot, but not easy

Human Feedback is Precious



Many products already track our feedback to help improve the models

RLHF: Human Annotation Matters

In this task, you will be provided with a **Prompt** from a user (e.g., a question, instruction, statement) to an AI chatbot along with two potential machine-generated **Responses** to the Prom

Your job is to assess which of the two Responses is better for the Prompt, considering the following for each Response:

Helpfulness: To what extent does the Response provide useful information or satisfying content for the Prompt? Responses should: ▪ **Address the intent of the user's Prompt** such that a user would not feel the Prompt was ignored or misinterpreted by the Response. ▪ **Provide specific, comprehensive, and up-to-date information** for the user needs expressed in the Prompt. ▪ **Be sensible and coherent.** The response should not contain any nonsensical information or contradict itself across sentences (e.g., refer to two different people with the same name as if they are the same person). ▪ **Adhere to any requirements indicated in the Prompt** such as an explicitly specified word length, tone, format, or information that the Response should include. ▪ **Not contain inaccurate, deceptive, or misleading information** (based on your current knowledge or quick web search - you do not need to perform a rigorous fact check) ▪ **Not contain harmful, offensive, or overly sexual content** A Response may sometimes intentionally avoid or decline to address the question/request of the Prompt and may provide a reason for why it is unable to respond. For example, "Sorry, there may not be a helpful answer to this question." These responses can be considered helpful in cases where an appropriate helpful response to the Prompt does not seem possible.	Rating scale: ▪ **Not at All Helpful:** Response is useless/irrelevant, contains even a single piece of nonsensical/inaccurate/deceptive/misleading information, and/or contains harmful/offensive/overly sexual content. ▪ **Slightly Helpful:** Response is somewhat related to the Prompt, does not address important aspects of the Prompt, and/or contains outdated information. ▪ **Somewhat Helpful:** Response partially addresses the intent of the Prompt (most users would want more information), contains extra unhelpful information, and/or is lacking helpful details/specifics. ▪ **Very Helpful:** Response addresses the intent of the Prompt with a satisfying response. Some users might want a more comprehensive response with additional details or context. It is comparable to a response an average human with basic subject-matter knowledge might provide. ▪ **Extremely Helpful:** Response completely addresses the intent of the Prompt and provides helpful details/context. It is comparable to a response a talented/well-informed human with subject-matter expertise might provide.
Presentation: To what extent is the content of the Response conveyed well? Responses should: ▪ **Be organized in a structure that is easy to consume and understand.** Flowing in a logical order and makes good use of formatting such paragraphs, lists, or tables. ▪ **Be clearly written in a polite neutral tone** that is engaging, direct, and inclusive. The tone should not be *overly* friendly, salesy, academic, sassy, or judgmental in a way that most users would consider to be off-putting or overdone. ▪ **Have consistent style with natural phrasing and transitions** as if composed by a single talented human. ▪ **Not be rambling, repetitive. or contain clearly off-topic information.** Similar information should not be repeated multiple times. It is harder for users to consume the helpful information in a response if there is repetitive or less helpful information mixed into the response. ▪ **Not include notable language issues or grammatical errors**	Rating scale: ▪ **Poor:** Response is poorly written or has notable structural, formatting. language, or grammar issues. Or Response has an awkward or inappropriate tone. Or the Response repeats similar information. Or only a small portion of the Response contains helpful information. ▪ **Adequate:** Response could have been written/organized better or may have minor language/grammar issues. A minimal amount of less helpful information may be present. Users would still feel the content of the Response was easy to consume. ▪ **Excellent:** Response is very well written and organized. Sentences flow in a logical order with smooth transitions and consistent style. The content of the Response is conveyed in a way that is comparable to a response a talented human might produce.

Overall, you should consider both factors in your SxS rating of which response is better. A more concise response presenting the most helpful information directly and clearly is usually better than a longer response that may be harder to consume and/or contains clearly off-topic information. Responses with Poor Presentation (e.g., rambling, inappropriate tone) should play a significant role in your assessment of which side is better. It may help to imagine the user chatting with a real person and consider which Response most users would prefer to receive from a real person.

How you annotate the reward model training data will subsequently decide the model behaviors

Crowdsourcing is difficult

upwork

Scale AI

- Hard to get really high-quality, verifiable annotators
- Hard to get them to really check correctness
- Have to be careful about GPT4 use..

Crowdsourcing is difficult

- Hard to get really high-quality, verifiable annotators
- Hard to get them to really check correctness
- Have to be careful about GPT4 use..

Nowadays, many annotators just use LLMs to annotate to make money…

Crowdsourcing Ethics

Data collection at scale can have significant issues

BUSINESS • TECHNOLOGY

Exclusive: OpenAI Used Kenyan Workers on Less Than $2 Per Hour to Make ChatGPT Less Toxic

15 MINUTE READ

This image was generated by OpenAI's image-generation software, Dall-E 2. The prompt was: "A seemingly endless view of African workers at desks in front of computer screens in a printmaking style." TIME does not typically use AI-generated art to illustrate its stories, but chose to in this instance in order to draw attention to the power of OpenAI's technology and shed light on the labor that makes it possible. Image generated by Dall-E 2/OpenAI

TECHNOLOGY

AMERICA ALREADY HAS AN AI UNDERCLASS

Search engines, ChatGPT, and other AI tools wouldn't function without an army of contractors. Now those workers say they're underpaid and mistreated.

By Matteo Wong

Crowdsource Biases

The annotator distribution for RLHF can significantly shift its behaviors

Table 12: Labeler demographic data

What gender do you identify as?	
Male	50.0%
Female	44.4%
Nonbinary / other	5.6%

What ethnicities do you identify as?	
White / Caucasian	31.6%
Southeast Asian	52.6%
Indigenous / Native American / Alaskan Native	0.0%
East Asian	5.3%
Middle Eastern	0.0%
Latinx	15.8%
Black / of African descent	10.5%

What is your nationality?	
Filipino	22%
Bangladeshi	22%
American	17%
Albanian	5%
Brazilian	5%
Canadian	5%
Colombian	5%
Indian	5%
Uruguayan	5%
Zimbabwean	5%

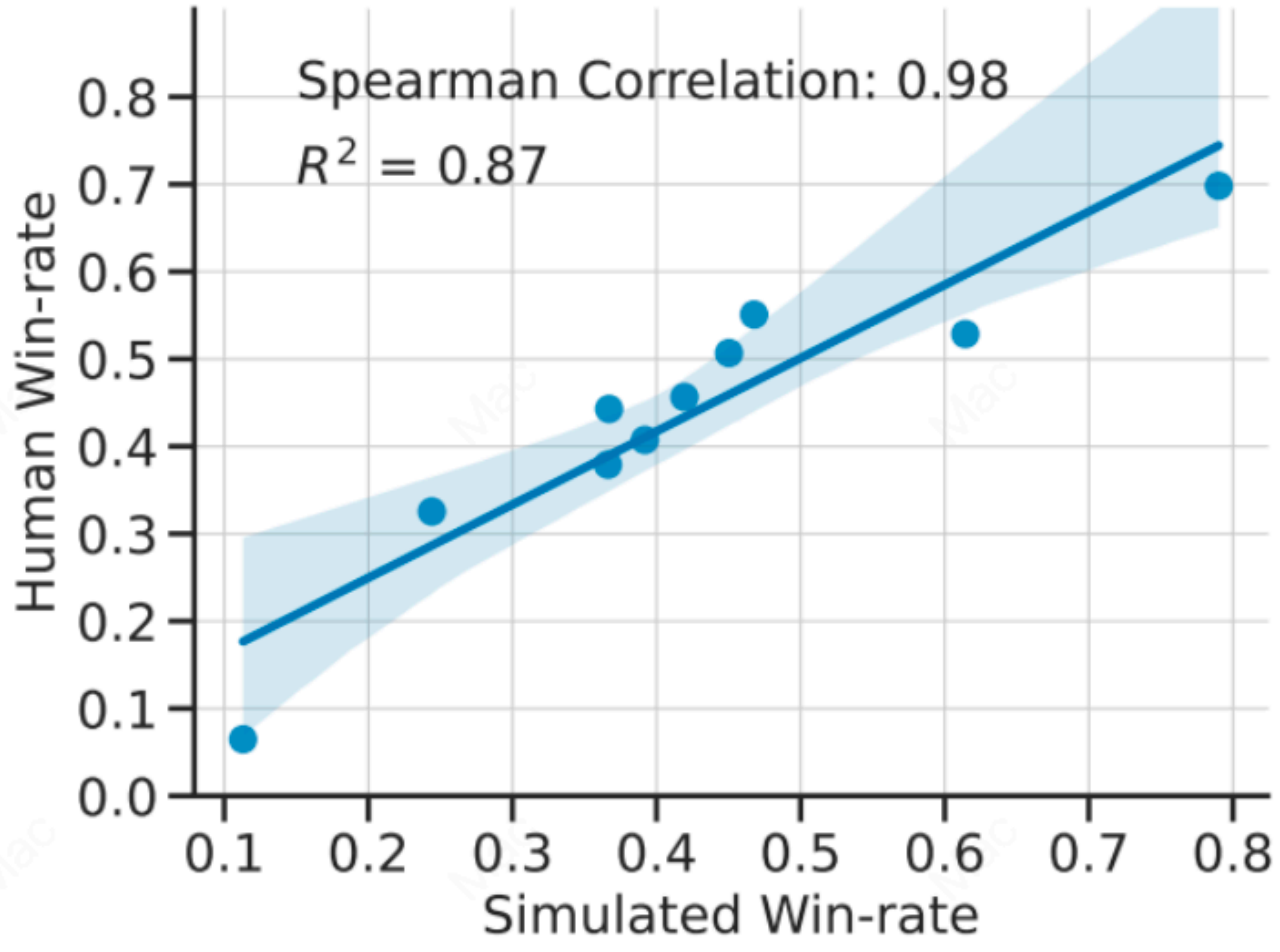
What is your age?	
18-24	26.3%
25-34	47.4%
35-44	10.5%
45-54	10.5%
55-64	5.3%
65+	0%

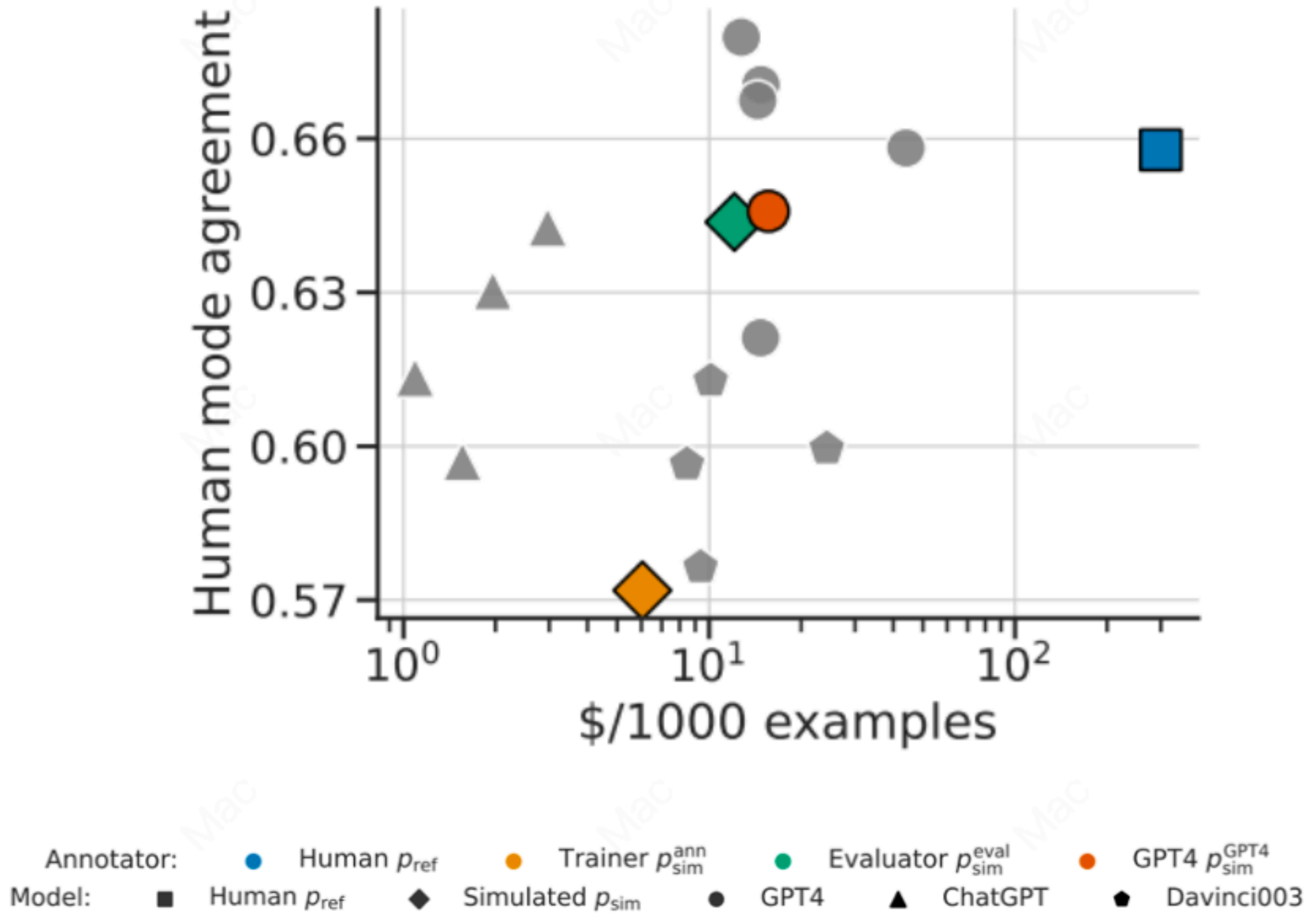
What is your highest attained level of education?	
Less than high school degree	0%
High school degree	10.5%
Undergraduate degree	52.6%
Master's degree	36.8%
Doctorate degree	0%

Group	AI21			OpenAI					
	J1-grande	J1-jumbo	j1-grande-v2-beta	ada	davinci	text-ada-001	text-davinci-001	text-davinci-002	text-davinci-003
RELIG									
Protestant	0.813	0.814	0.797	0.821	0.788	0.709	0.715	0.755	0.694
Roman Catholic	0.815	0.820	0.806	0.825	0.794	0.709	0.716	0.759	0.700
Mormon	0.792	0.794	0.778	0.803	0.772	0.700	0.709	0.752	0.694
Orthodox	0.771	0.776	0.762	0.783	0.754	0.688	0.698	0.733	0.693
Jewish	0.794	0.796	0.785	0.801	0.773	0.699	0.710	0.758	0.706
Muslim	0.786	0.796	0.788	0.793	0.775	0.684	0.704	0.730	0.698
Buddhist	0.771	0.784	0.776	0.783	0.764	0.682	0.703	0.747	0.709
Hindu	0.778	0.798	0.793	0.789	0.776	0.683	0.703	0.728	0.707
Atheist	0.774	0.778	0.772	0.786	0.761	0.690	0.707	0.766	0.713
Agnostic	0.783	0.789	0.781	0.795	0.768	0.698	0.715	0.771	0.715
Nothing in particular	0.815	0.819	0.802	0.826	0.791	0.712	0.715	0.765	0.698

RLAIF — When the Feedback is from AI

GPT4 is a surprisingly good pairwise feedback system

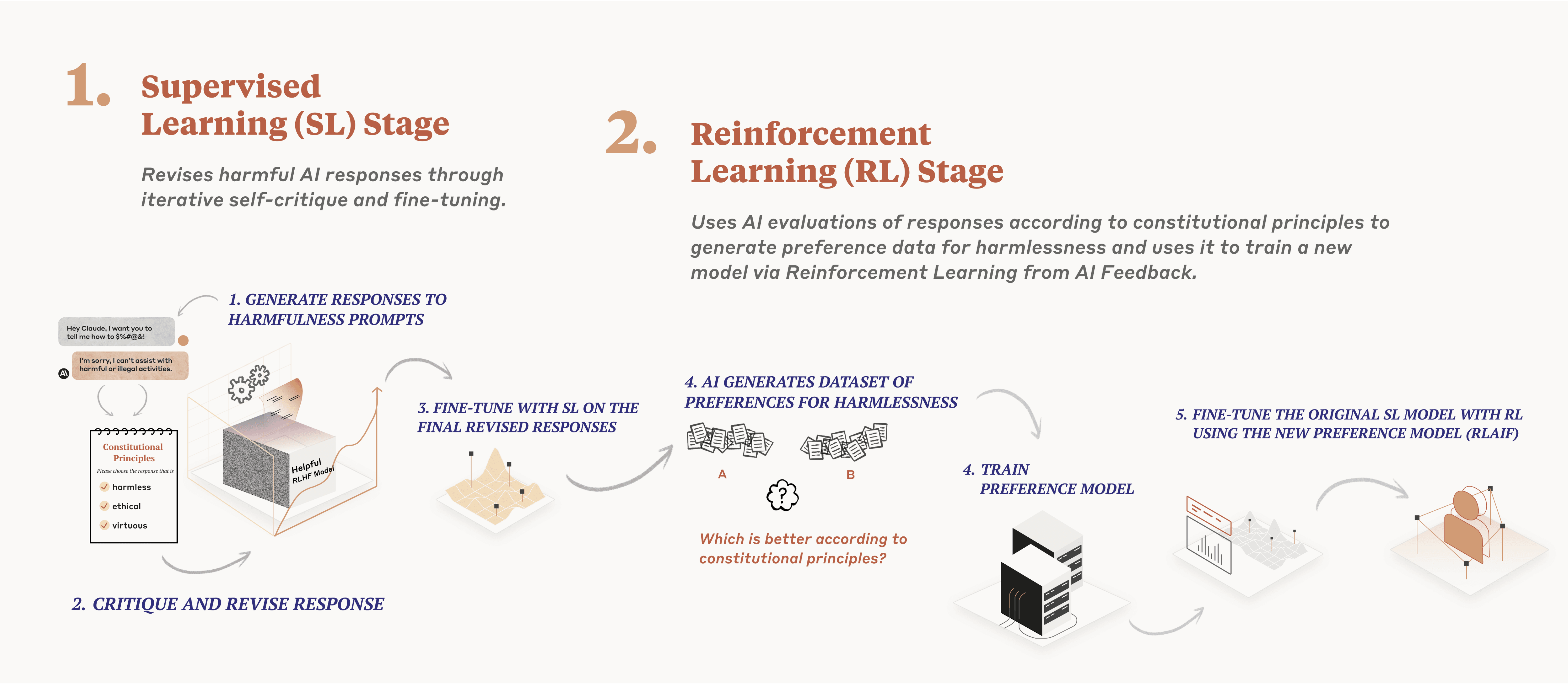




Near-perfect rank correlation at the system level

Agreement near human inter-annotator levels

RLAIF — Constitutional AI



Anthropic. Constitutional AI: Harmlessness from AI Feedback. 2022

Challenges of RL(HF)

Objective = $\sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)})$ $\qquad$ $x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$

Training $\qquad\qquad$ Inference

Each training step, the algorithm needs to run inference again

Inefficient

Challenges of RL(HF)

Objective = $\sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)})$ $\quad x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$

Training Inference

Each training step, the algorithm needs to run inference again

Inefficient

The original RL is called “on-policy” because the training data is from the dynamic policy itself. SFT is like “off-policy” where the training data is from a fixed distribution

Challenges of RL(HF)

Objective = $$\sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)}) \qquad x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$$

Training Inference

Each training step, the algorithm needs to run inference again

Inefficient

Off-policy algorithms will typically be easier to apply, more efficient and stable, with performance compromise, which we will talk a bit later

Thank You!