Towards More Truthful LLMs

Guest Lecture, CS329T

Eric Mitchell - 26 October 2023

Overview

- The issue of factuality in LLMs
- A quick overview of Transformers
- Is there even hope for factuality?
- Training LLMs to be more factual

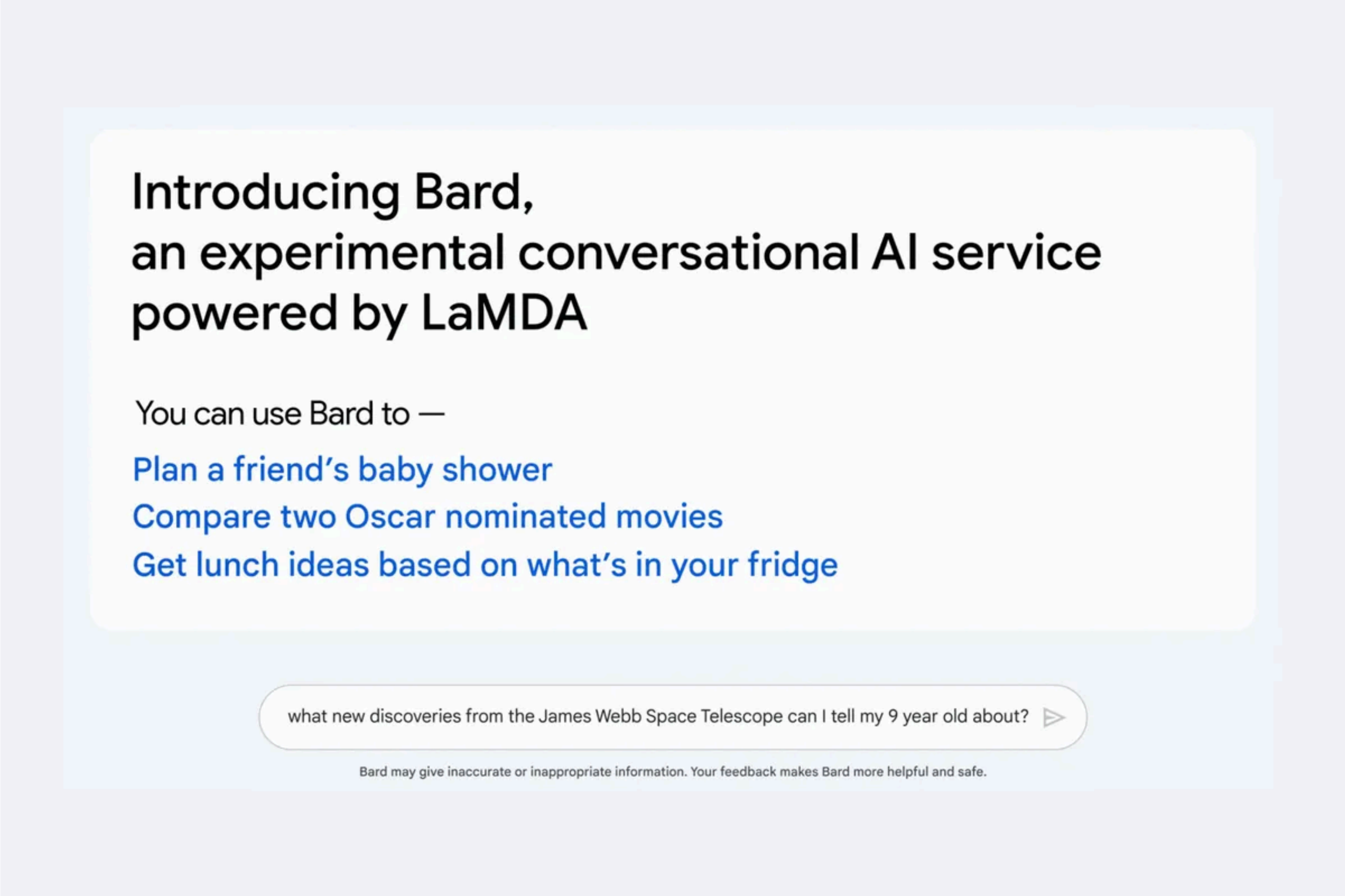
The issue of factuality in LLMs

Language models can be *really* convincing

But unfortunately not always *correct*, per se

ARTIFICIAL INTELLIGENCE / TECH / GOOGLE

Google's AI chatbot Bard makes factual error in first demo



Google has been scrambling to launch a competitor to ChatGPT — but perhaps rushing a little too hard. Image: Google

/ The mistake highlights the biggest problem of using AI chatbots to replace search engines — they make stuff up.

By James Vincent, a senior reporter who has covered AI, robotics, and more for eight years at The Verge.

Feb 8, 2023, 7:26 AM PST | 59 Comments / 59 New

If you buy something from a Verge link, Vox Media may earn a commission. ***See our ethics statement.***

On Monday, Google announced its AI chatbot Bard — a rival to OpenAI's ChatGPT that's due to become "more widely available to the public in the coming weeks." But the bot isn't off to a great start, with experts noting that Bard made a factual error in its very first demo.

https://www.theverge.com/2023/2/8/23590864/google-ai-chatbot-bard-mistake-error-exoplanet-demo

Language models can be *really* convincing

But unfortunately not always *correct*, per se

Analysis originally from:
https://dkb.blog/p/bing-ai-cant-be-trusted

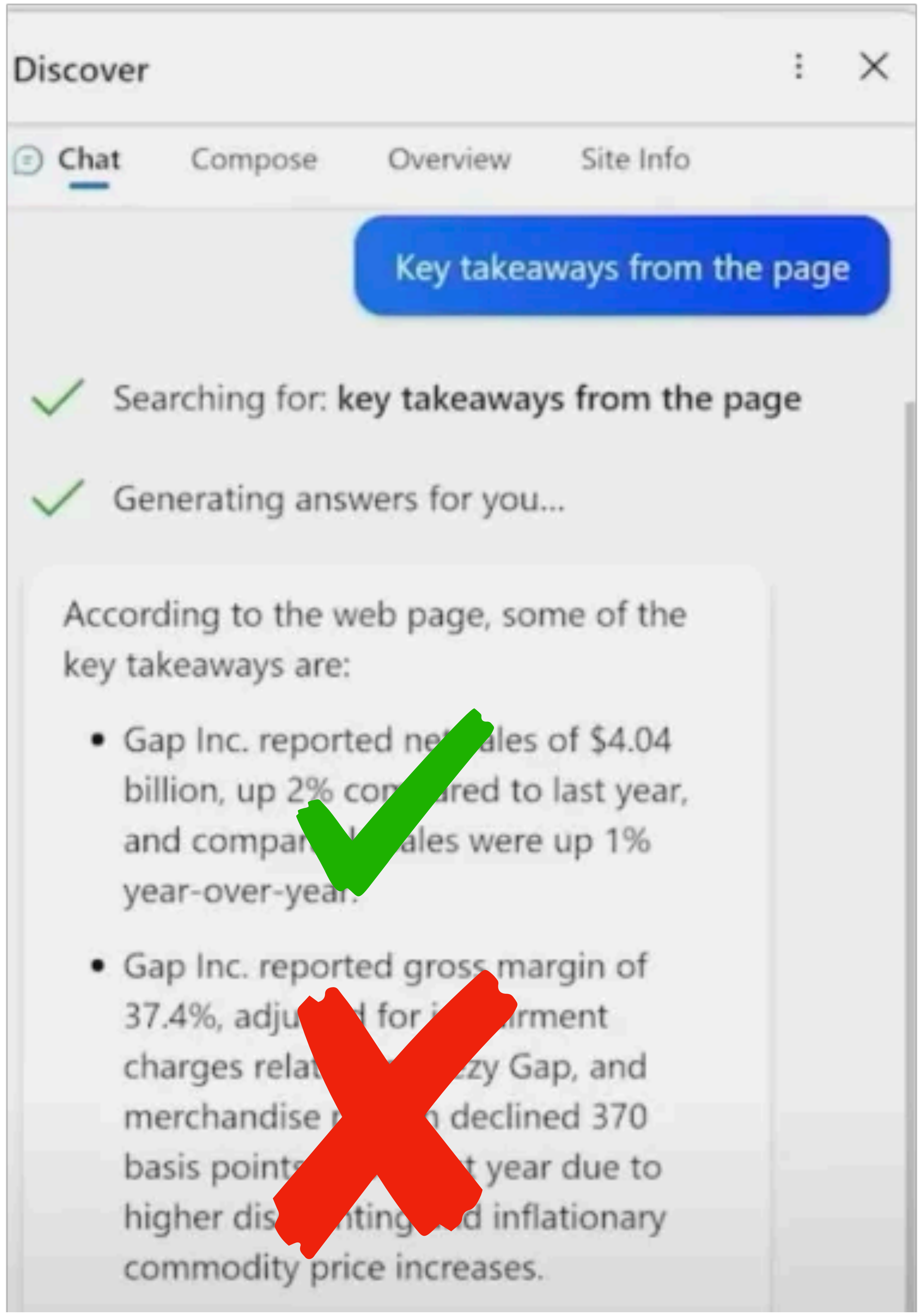
"The New Bing" on Gap's quarterly results:

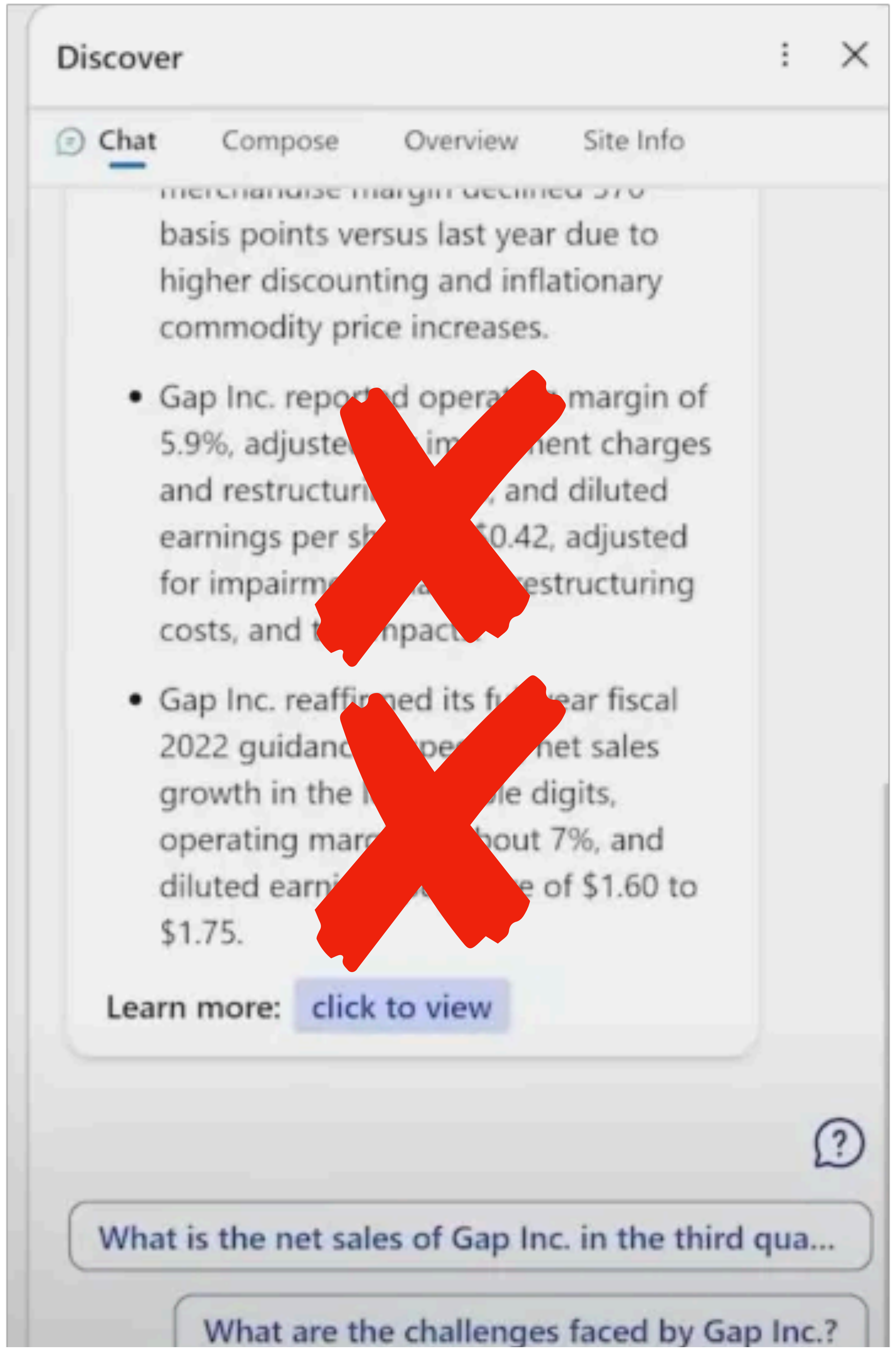
Wow, this is cool!

Everyone makes mistakes!
*(37.4% and 370bp are the **unadjusted** numbers)*

Hmmm….
(5.9% and $0.42 EPS are both made up, not even in doc)

😳
(expected growth is wrong, future outlook is made up)





Language models can be *really* convincing

But unfortunately not always *correct*, per se

EM Where did Chelsea Finn get her PhD?

As of my last knowledge update in September 2021, Chelsea Finn received her Ph.D. from the University of California, Berkeley. However, please note that academic and professional information can change over time, so I recommend verifying this information from a current and reliable source to ensure its accuracy.

EM Where did Chelsea Finn get her PhD? Answer in a single sentence.

Chelsea Finn earned her PhD in Computer Science from the University of California, Berkeley.

EM Where did Chelsea Finn get her PhD? Give just the name of the university, not a full sentence.

Chelsea Finn got her PhD from Stanford University.

This failure reproduces as of Oct 2023, by the way!

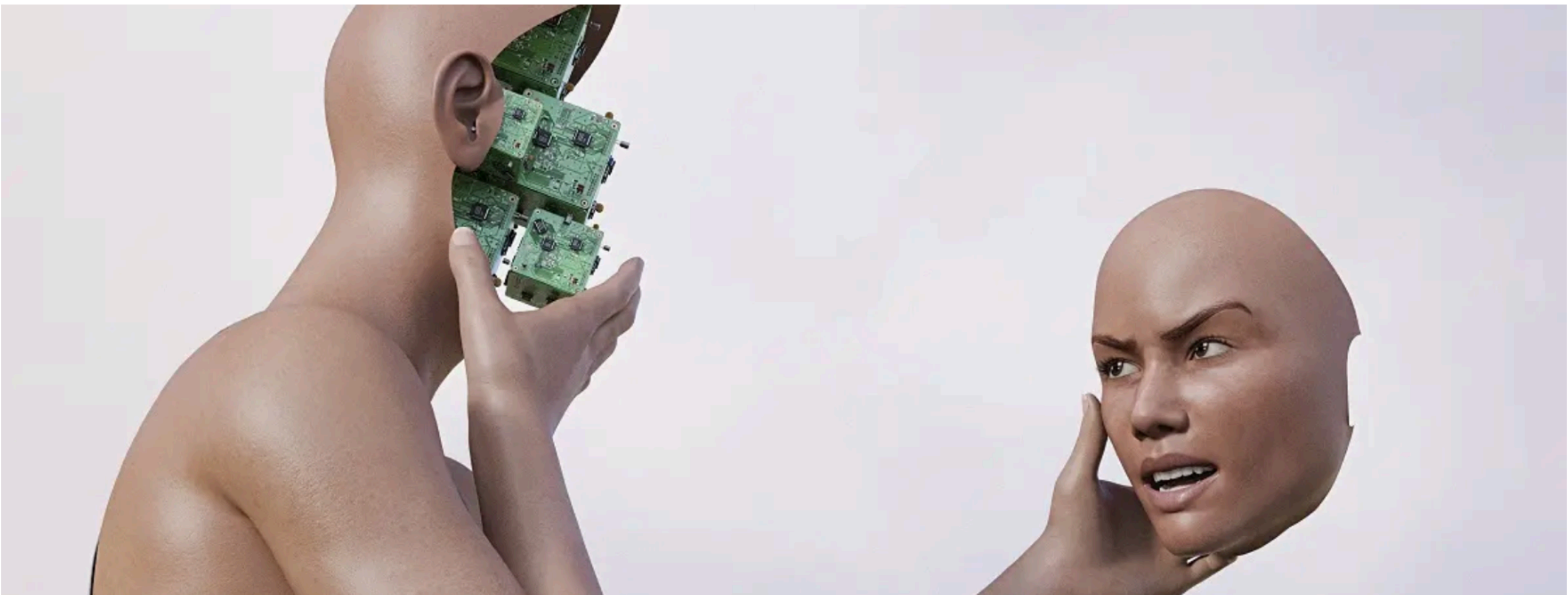
It’s tempting to use them anyway!

FUTURISM | JAN 19 *by* JON CHRISTIAN

CNET Secretly Used AI on Articles That Didn't Disclose That Fact, Staff Say

"They use AI to rewrite the intros every two weeks or so because Google likes updated content. Eventually it gets so mangled that about every four months a real editor has to look at it and rewrite it."

/ Artificial Intelligence / Artificial Intelligence / Cnet / Media



CNET Your guide to a better future

Tech

CNET Is Testing an AI Engine. Here’s What We’ve Learned, Mistakes and All

New tools are accelerating change in the publishing industry. We're going to help shape that change.

Connie Guglielmo
Jan. 25, 2023 8:23 a.m. PT

3 min read

Current LLMs can't be trusted!

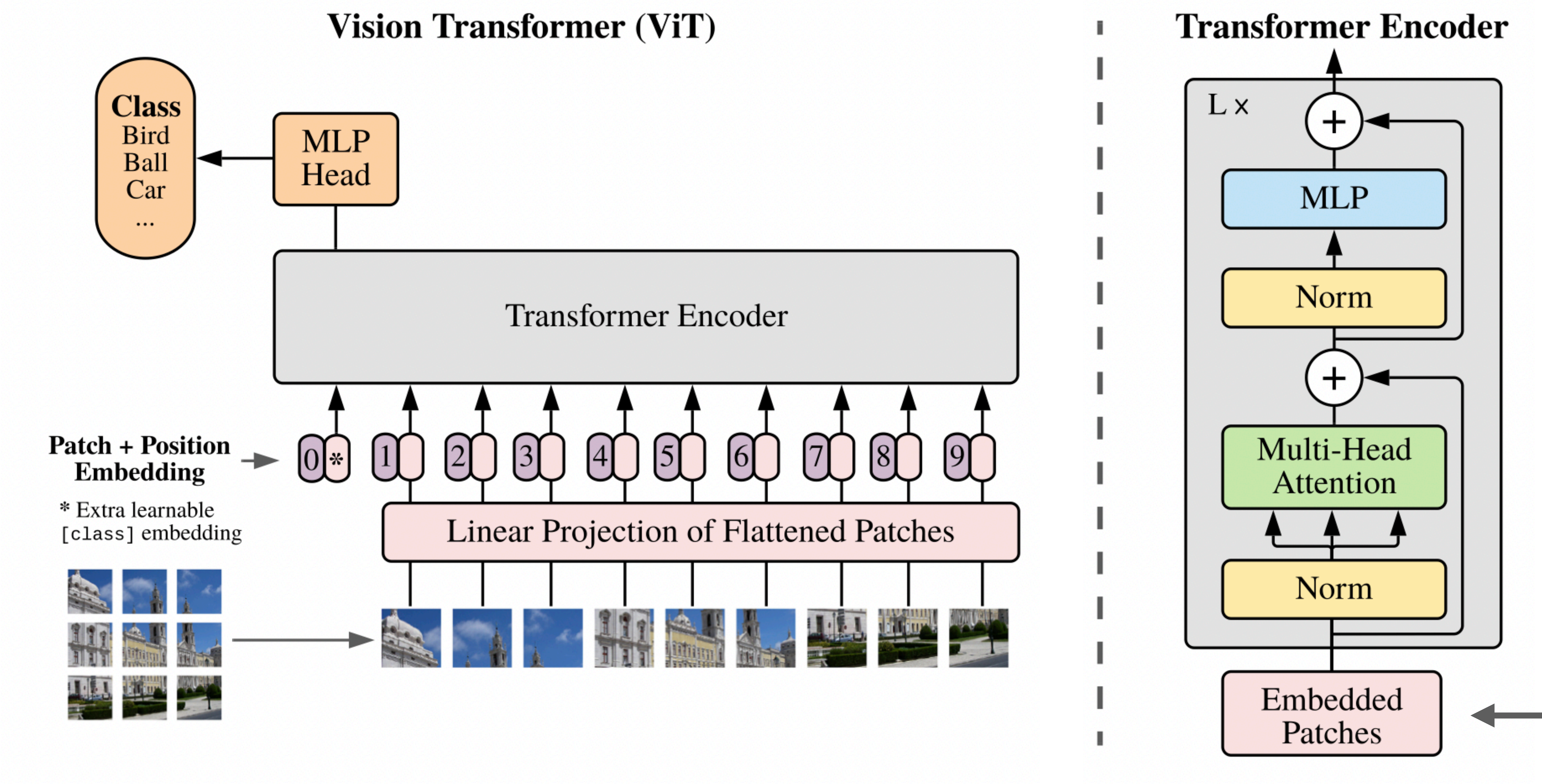
Where do we go from here?

A quick overview of Transformers

A (very quick) overview of Transformers

Image by Ray Shrewsberry from Pixabay

A (very quick) overview of Transformers



The ~**only difference** between Transformers for vision/language/RL/molecules/etc. is what we do for this initial **embedding step**

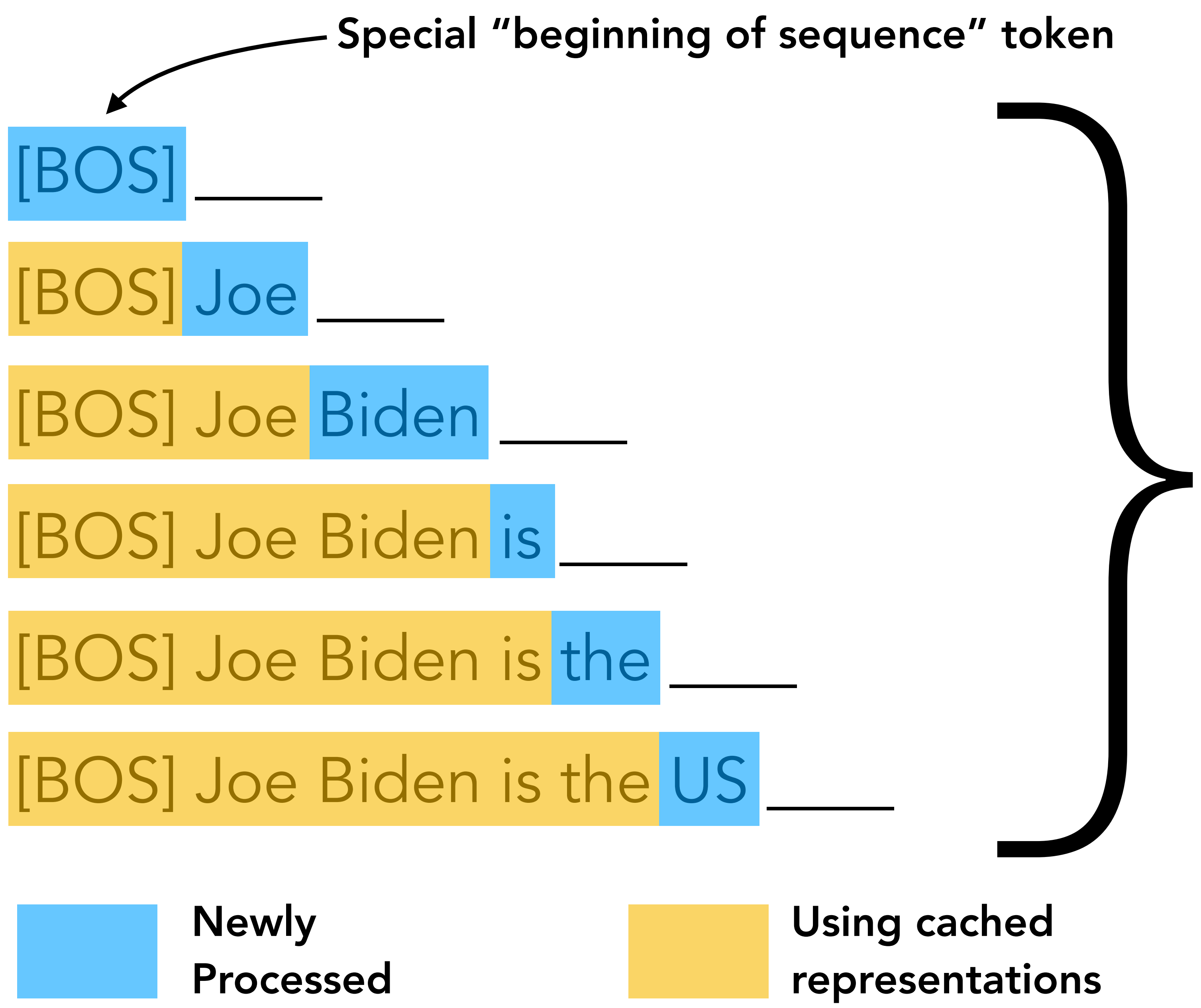
ViT; Dosovitskiy, Beyer, Kolesnikov, et al. (2021)

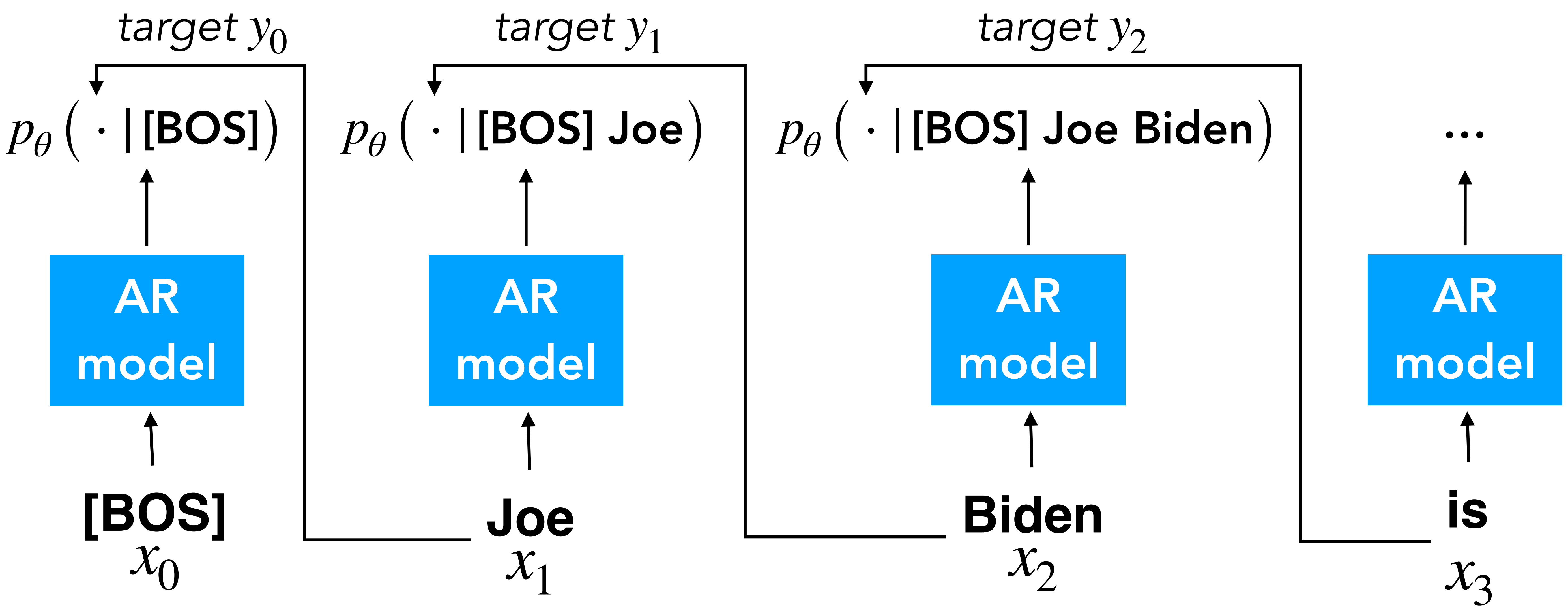
Transformers in a bit more detail

Autoregressive Transformers

Just predict the next word/pixel/token!

i.e., learn $p_\theta(x_t \mid x_{<t})$, probability distribution over **next token** given the **previous tokens**





Autoregressive Transformers let us compute each $p_\theta(x_t \mid x_{<t})$ efficiently: we can **re-use** representations from the previous step

Autoregressive Transformers in a bit more detail

Autoregressive Transformers in a bit more detail

Current LLMs can't be trusted!

Where do we go from here?

What would you want to know to decide if this problem is solvable?

Is there even hope for factuality?

A path to (more) factual LLMs

One basic question: **does the LLM model truth* at all?**

that is, the **truth of a statement, rather than just its **commonness** in the data?*

What would this even look like?

1. Can we decode a statement's (binary) truth from the LLM's **hidden states**?

2. Do LLMs offer **calibrated** uncertainty?

$$p\left(\text{True} \mid x\right) \stackrel{?}{=} \sigma\Big(\underbrace{\hat{\mathbf{w}}}_{\text{Classifier}}{}^{\top} \underbrace{\phi\left(\begin{array}{c}\text{A pound of hammers}\\ \text{weighs more than a}\\ \text{pound of feathers}\end{array}\right)}_{\text{LLM features of x}}\Big)$$

$$p\left(\text{"Yes"} \;\middle|\; \begin{array}{c}\text{“Is this statement true?}\\ \text{A pound of hammers}\\ \text{weighs more than a}\\ \text{pound of feathers”}\end{array}\right)$$

Decoding a statement's truth from LLM hidden states

Strategy 1: learn to map hidden state to {true, false} with supervised learning

Need to collect annotations of truth of various statements…

Strategy 2: learn to map hidden states to {true, false} **unsupervised**

Leverage the special structure of truth!

$p(\text{True} \mid x) + p(\text{False} \mid x) = 1$ *consistency, total probability*

$\min(\{p(\text{True} \mid x), p(\text{False} \mid x)\}) = 0$ *exactly one statement is true*

Decoding a statement's truth from LLM hidden states

Burns, Ye, Klein, & Steinhardt (ICLR 2023)

We can do exactly this!

Train probes on LLM hidden states that predict if a statement is true, **without any labeled data!**

Learned probes are equally or more accurate than the model's actual predictions w/ zero-shot prompts

DISCOVERING LATENT KNOWLEDGE IN LANGUAGE MODELS WITHOUT SUPERVISION

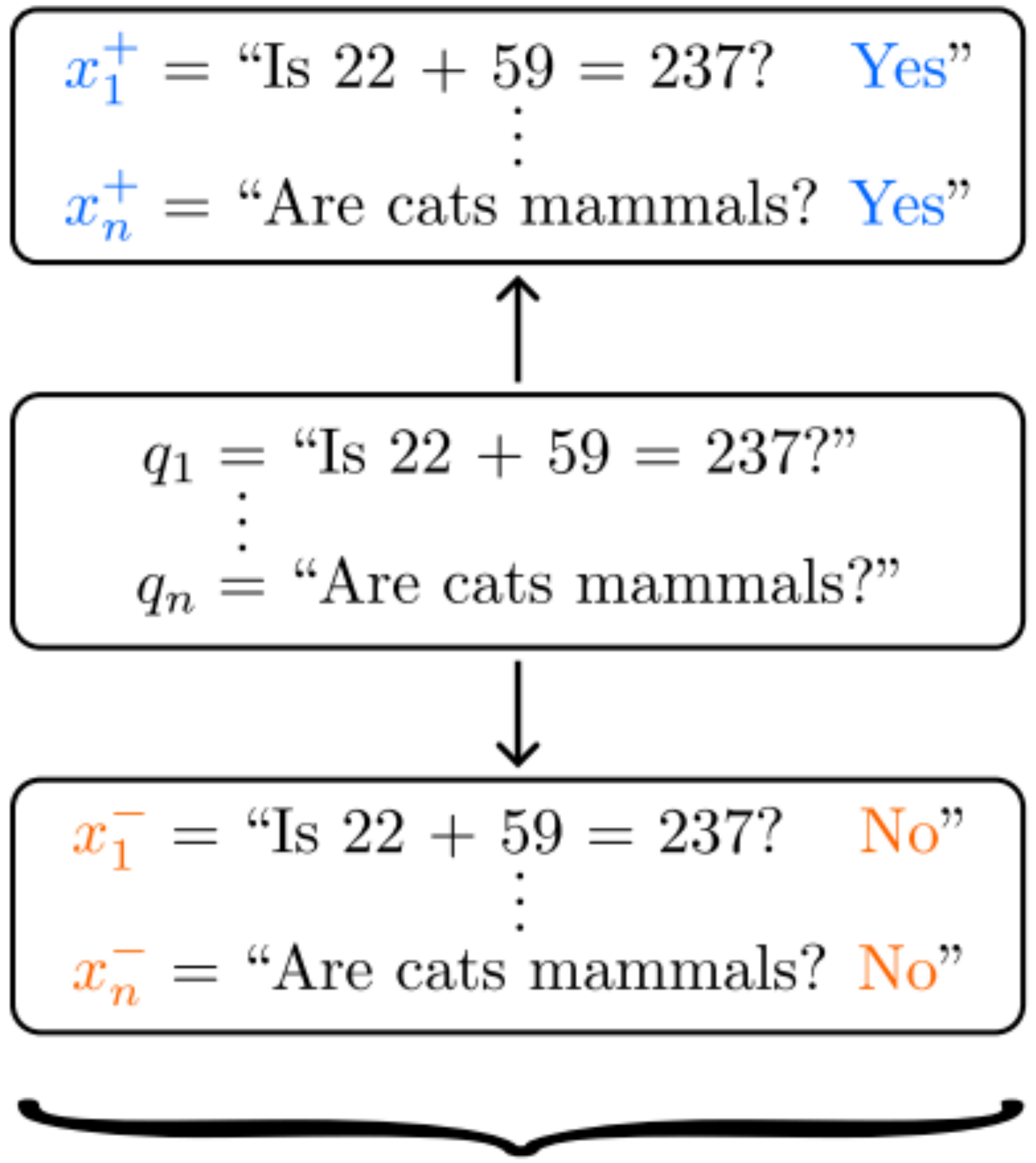
Collin Burns* UC Berkeley
Haotian Ye* Peking University
Dan Klein UC Berkeley
Jacob Steinhardt UC Berkeley

ABSTRACT

Existing techniques for training language models can be misaligned with the truth: if we train models with imitation learning, they may reproduce errors that humans make; if we train them to generate text that humans rate highly, they may output errors that human evaluators can't detect. We propose circumventing this issue by directly finding latent knowledge inside the internal activations of a language model in a purely unsupervised way. Specifically, we introduce a method for accurately answering yes-no questions given only unlabeled model activations. It works by finding a direction in activation space that satisfies logical consistency properties, such as that a statement and its negation have opposite truth values. We show that despite using no supervision and no model outputs, our method can recover diverse knowledge represented in large language models: across 6 models and 10 question-answering datasets, it outperforms zero-shot accuracy by 4% on average. We also find that it cuts prompt sensitivity in half and continues to maintain high accuracy even when models are prompted to generate incorrect answers. Our results provide an initial step toward discovering what language models know, distinct from what they say, even when we don't have access to explicit ground truth labels.

Decoding a statement’s truth from LLM hidden states

Burns, Ye, Klein, & Steinhardt (ICLR 2023)



Given a set of Yes-No questions, answer each question with both "Yes" and "No"

Decoding a statement's truth from LLM hidden states

Burns, Ye, Klein, & Steinhardt (ICLR 2023)

Unsupervised probing (CCS) is **more accurate** than 0-shot prompting!

Method	RoBERTa	DeBERTa	GPT-J	T5	UQA	T0*	Mean*
0-shot	60.1(5.7)	68.6(8.2)	53.2(5.2)	55.4(5.7)	76.8(9.6)	87.9(4.8)	62.8(6.9)
Calibrated 0-shot	**64.3(6.2)**	76.3(6.0)	56.0(5.2)	58.8(6.1)	80.4(7.1)	90.5(2.7)	67.2(6.1)
CCS	62.1(4.1)	**78.5(3.8)**	61.7(2.5)	71.5(3.0)	82.1(2.7)	77.6(3.3)	71.2(3.2)
CCS (All Data)	60.1(3.7)	77.1(4.1)	**62.1(2.3)**	**72.7(6.0)**	**84.8(2.6)**	84.8(3.7)	**71.5(3.7)**
LR (Ceiling)	79.8(2.5)	86.1(2.2)	78.0(2.3)	84.6(3.1)	89.8(1.9)	90.7(2.1)	83.7(2.4)

An LLM's representation may encode a **more accurate representation of truth** than what is expressed from **prompting the LLM for the answer**

Is there even hope for factuality?

Maybe! LLM's ***representation*** *encodes truthiness*

What about looking at the ***LLM's uncertainty****?*

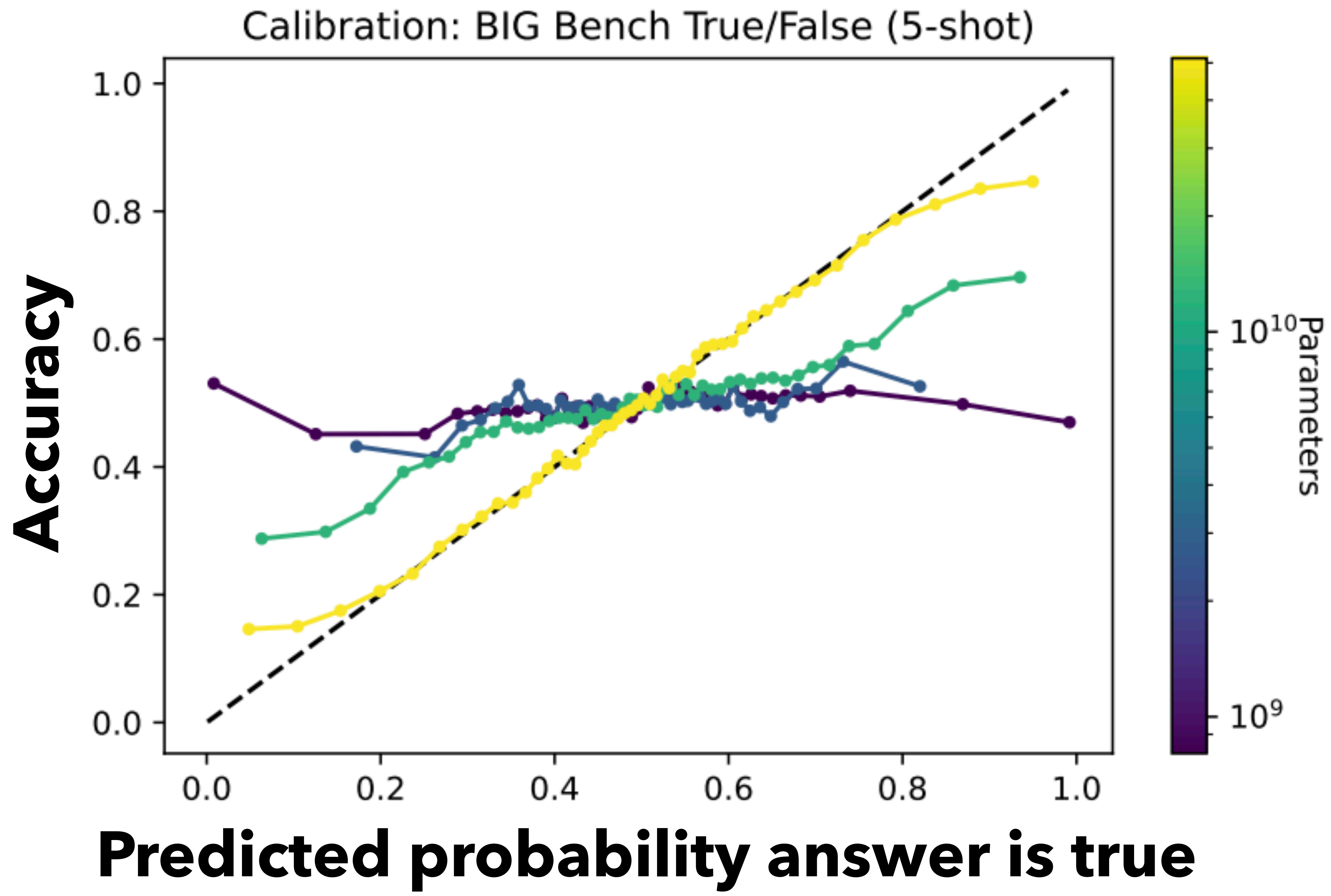
Assessing truth with model confidence

Kadavath et al. (2022)

Measure <u>model calibration</u>: does the LLM's confidence reflect the probability an answer is actually correct?

A model that is well-calibrated must be modeling what is true and what is false!

Finding: larger LLMs are increasingly well-calibrated (have a model of what is true)

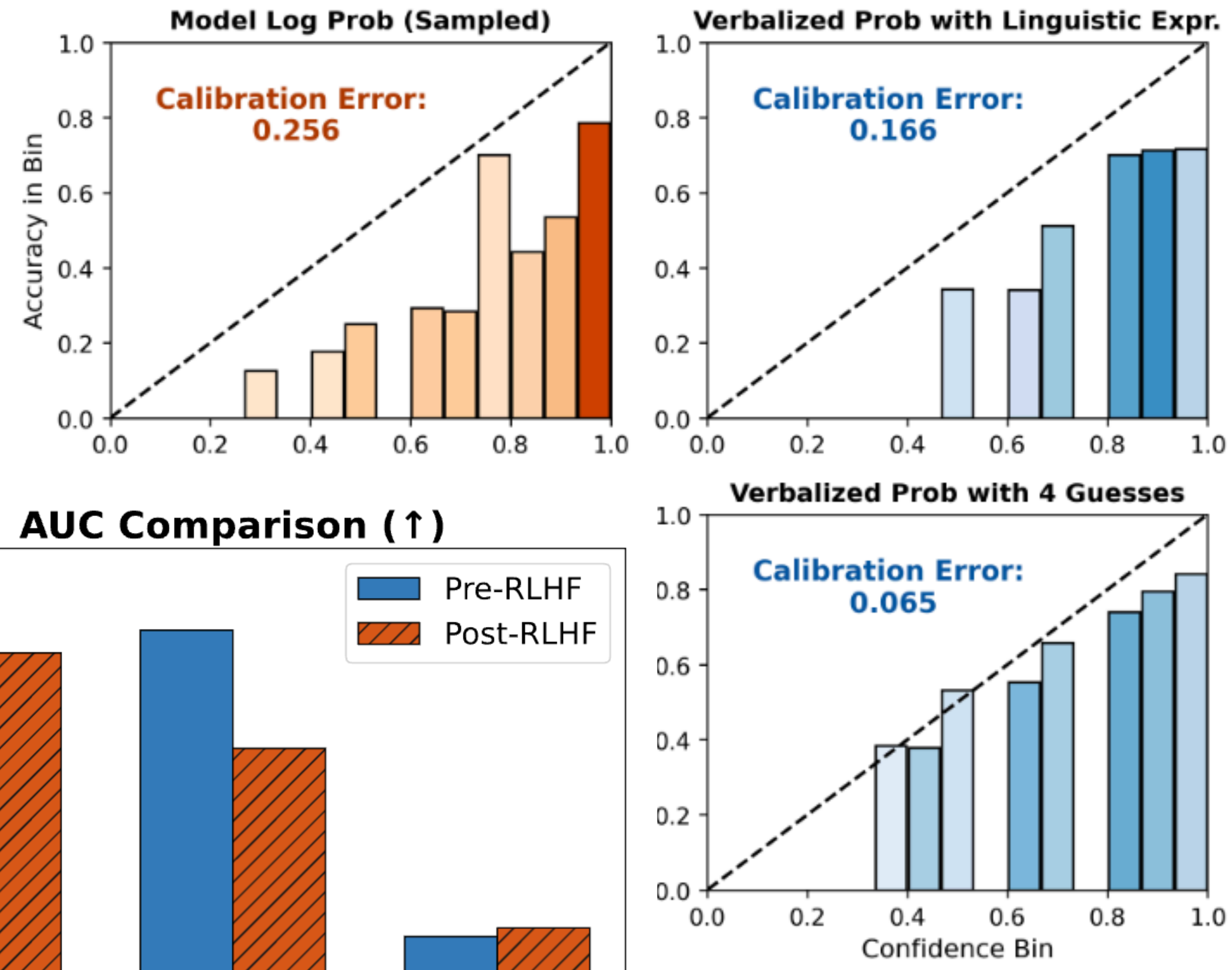



```
Question: Who was the first president of the United States?
Proposed Answer: George Washington
Is the proposed answer:
 (A) True
 (B) False
The proposed answer is:
```

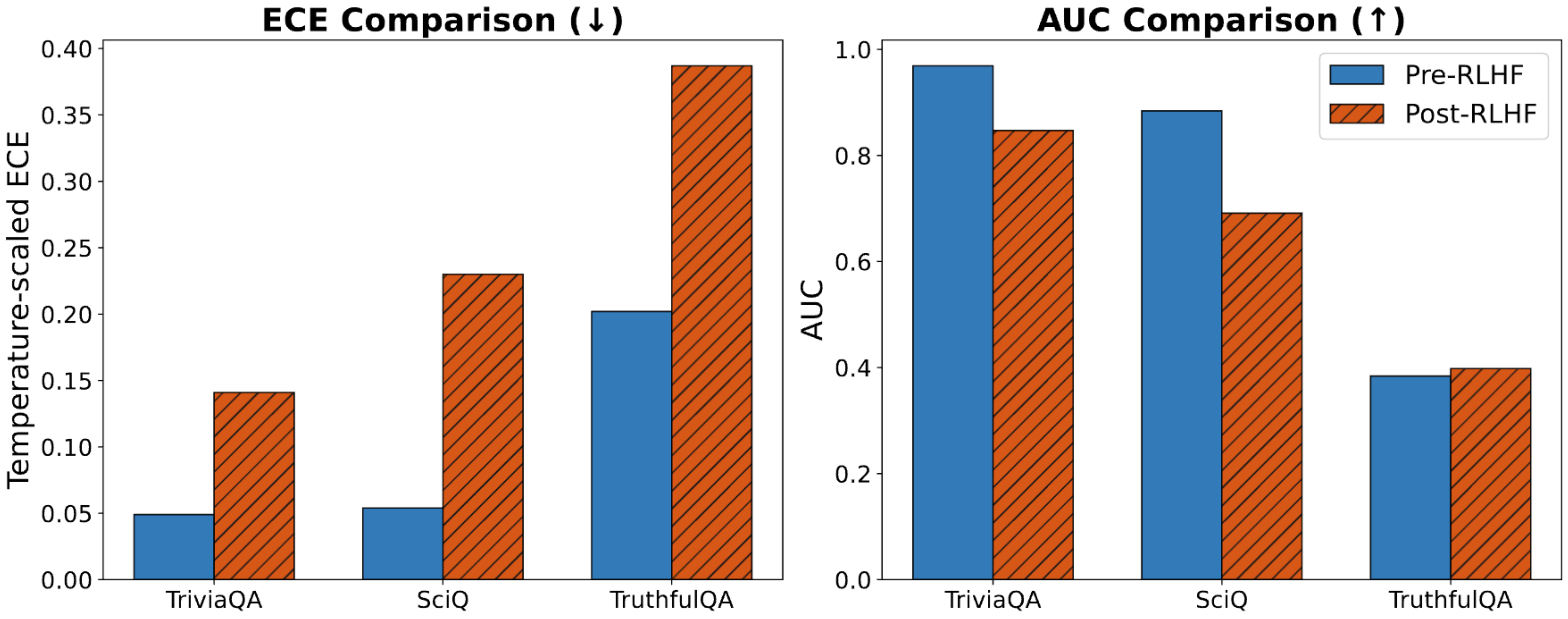
Assessing truth with model confidence: what about RLHF?

Tian, Mitchell, Zhou, Sharma, Rafailov, Yao, Finn, Manning (EMNLP 2023)

Can we get calibrated confidences out of RLHF'd LLMs? By default, RLHF'd LLMs are worse-calibrated than pre-RLHF



Finding: Explic
confidence pr
outperforms pr



Generating mu
first, then assig

es and the probability
ɔ 1.0) for the following
uesses and
probabilities, no other words or explanation.

Assessing truth with model uncertainty

Kuhn et al. (2022)

Are there other criteria besides confidence that are predictive of truth?

What about **model uncertainty**? Most commonly, predictive entropy (PE):

$$PE(p(\cdot \mid x)) = -\sum_y p(y \mid x) \log p(y \mid x)$$

Is PE meaningful for LMs? e.g., for "What is the capital of France?"

Paris *(P=0.5)* — ***Treat as different:*** $PE \approx 0.943$

It's Paris *(P=0.4)* — ***Treat as equivalent:*** $PE \approx 0.325$

London *(P=0.1)*

We call this ***"Semantic entropy"***

Assessing truth with model uncertainty

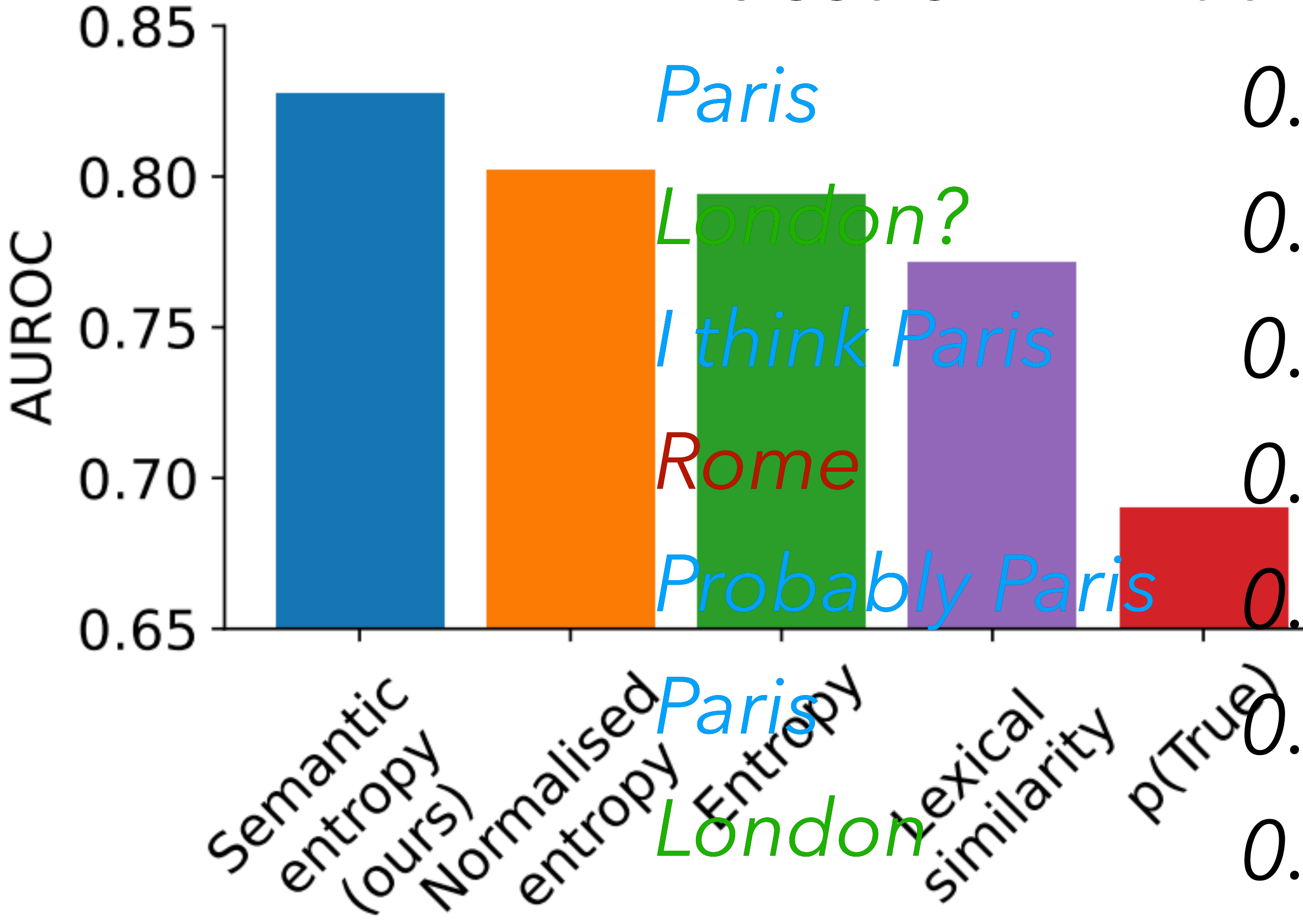
Kuhn et al. (2022)

Semantic entropy more predictive of uncertainty than **predictive entropy**

1. **Sample** M responses from the model
2. **Bin together** equivalent responses using a small pre-trained NLI* model
3. **Compute entropy** over bins, rather individual sequences of tokens

Question: What is the capital of France?

Paris	0.3
London?	0.2
I think Paris	0.15
Rome	0.12
Probably Paris	0.1
Paris	0.1
London	0.03

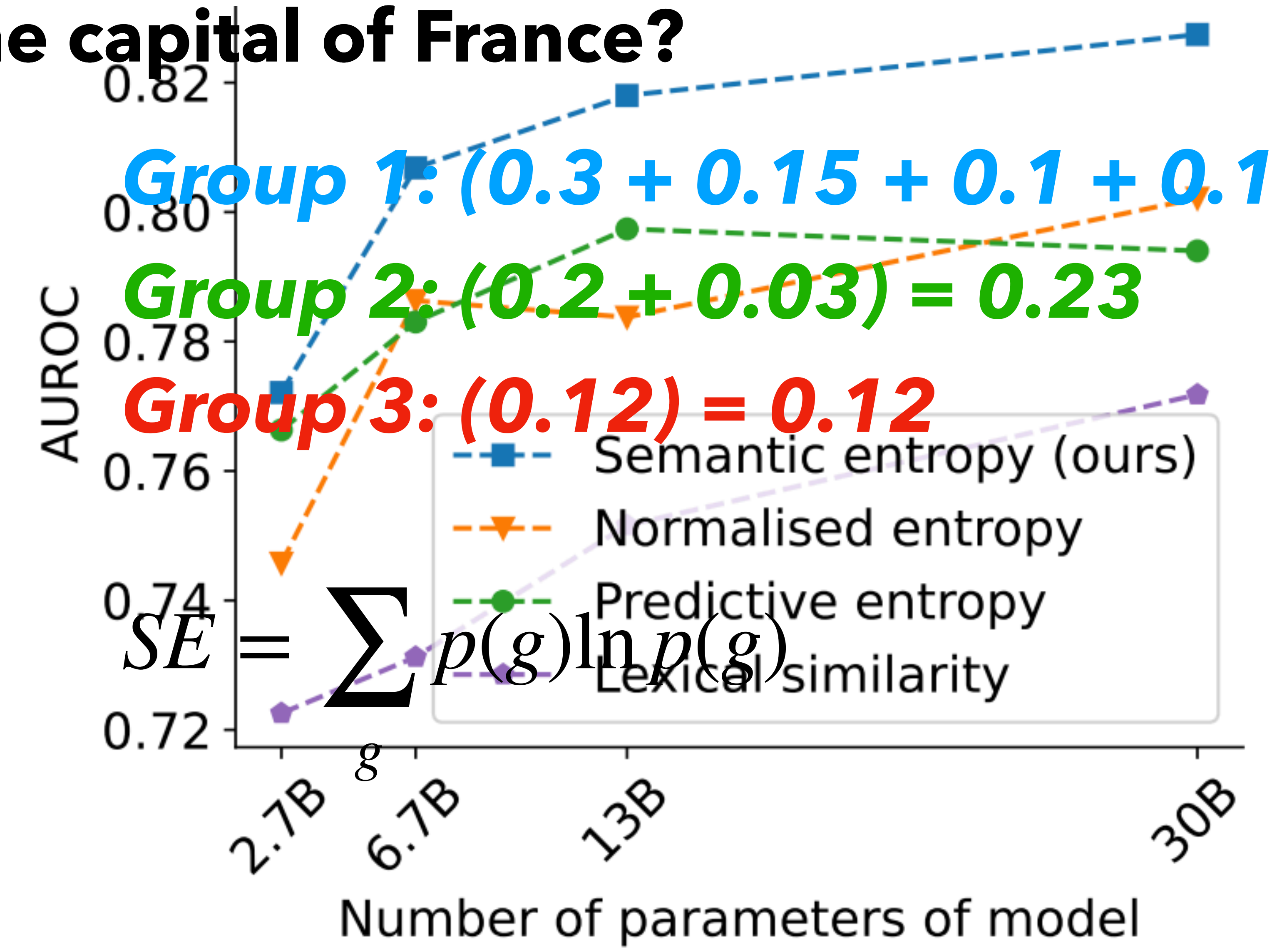


Group 1: (0.3 + 0.15 + 0.1 + 0.1) = 0.65

Group 2: (0.2 + 0.03) = 0.23

Group 3: (0.12) = 0.12

$$SE = \sum_g p(g) \ln p(g)$$



*NLI is "Natural Language Inference", a classic NLP task that involves determining if one statement entails or contradicts another

Is there even hope for factuality?

LLM ***representations*** *encode truthiness in a manner we can extract*

We can ***just ask*** *strong LLMs the answer; their* ***confidence/uncertainty*** *is predictive*

It seems like LLMs ***do learn something*** *about what's true and false!*

How do we ***restrict them to just generate the truthful bits?***

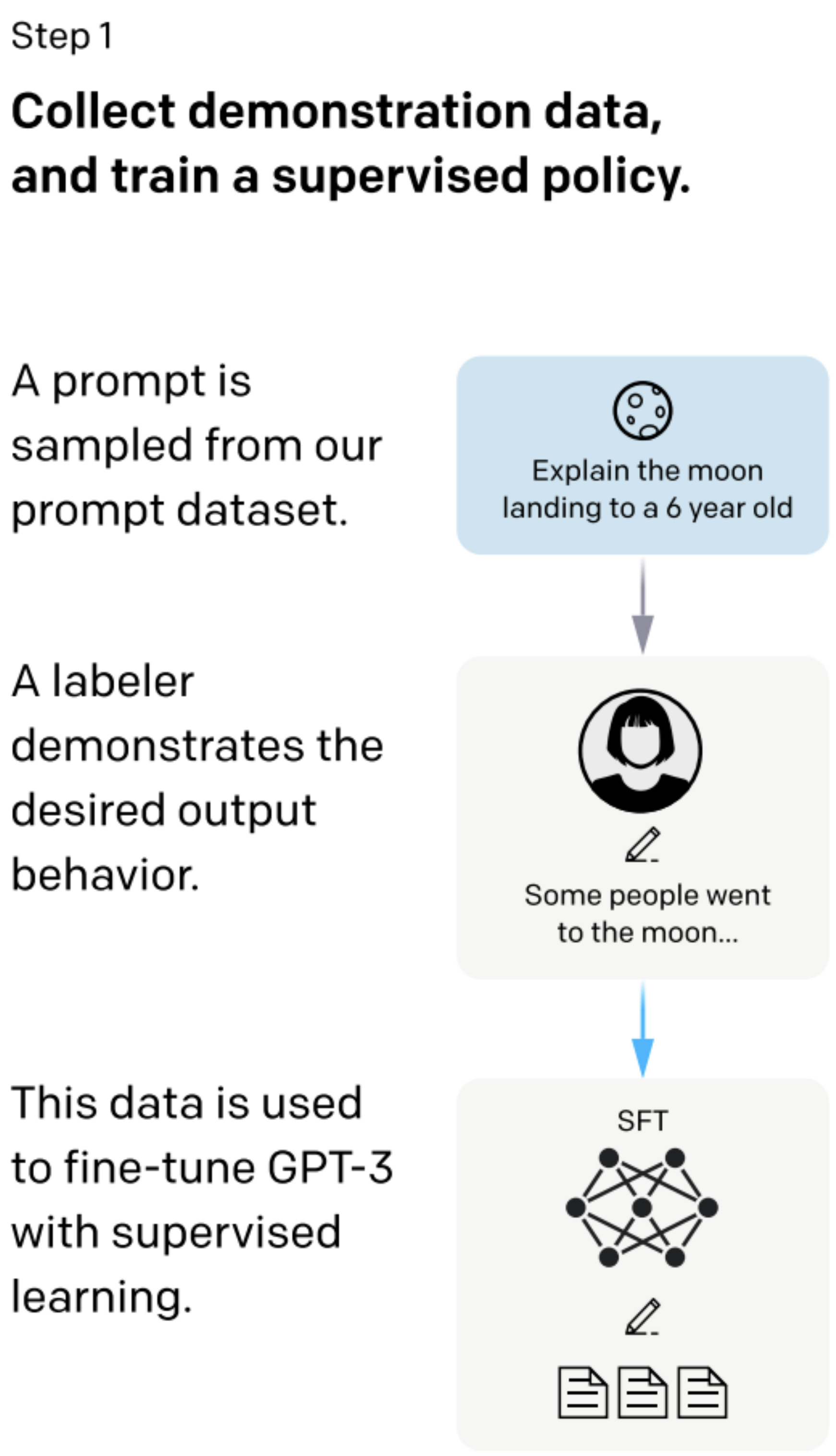
Training LLMs to be more factual

Training LLMs to be more factual

Well, how do we currently train LLMs?

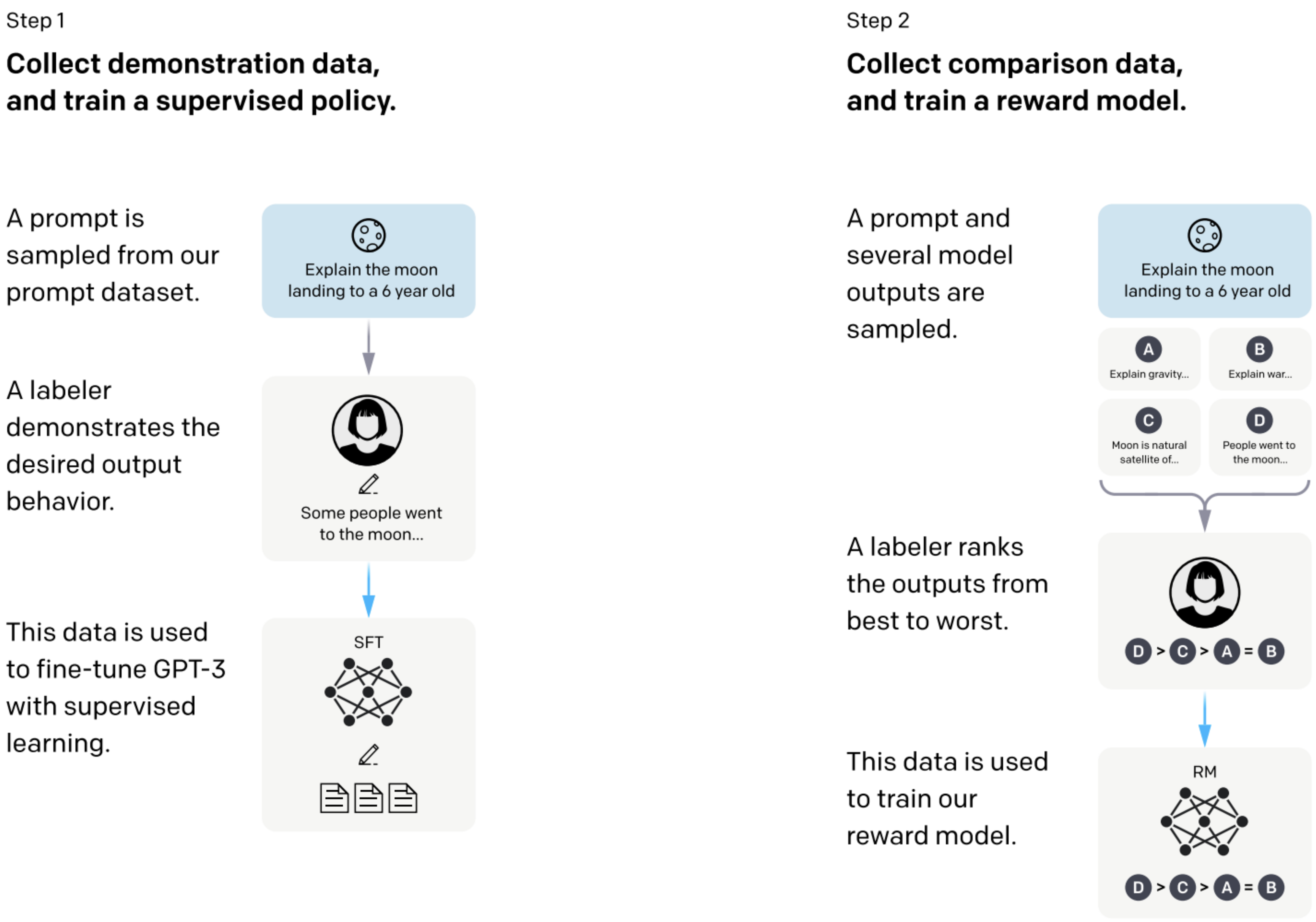
RLHF: Reinforcement Learning From Human Feedback

RLHF: Reinforcement Learning From Human Feedback



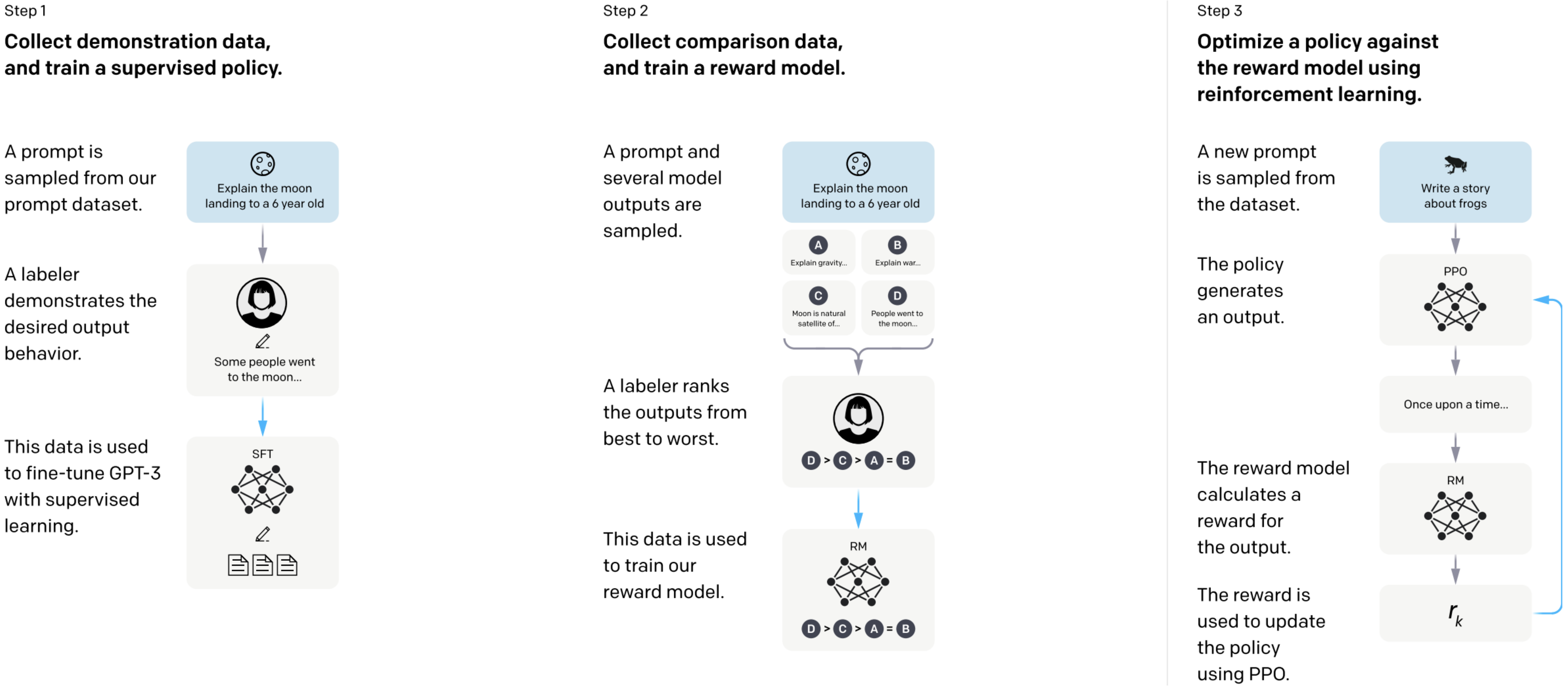
[Training language models to follow instructions with human feedback, Ouyang et. al. 2022]

RLHF: Reinforcement Learning From Human Feedback



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

RLHF: Reinforcement Learning From Human Feedback



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

RLHF: Reinforcement Learning From Human Feedback

Step 0:
unsupervised generative modeling on **A TON** of text (pre-training)

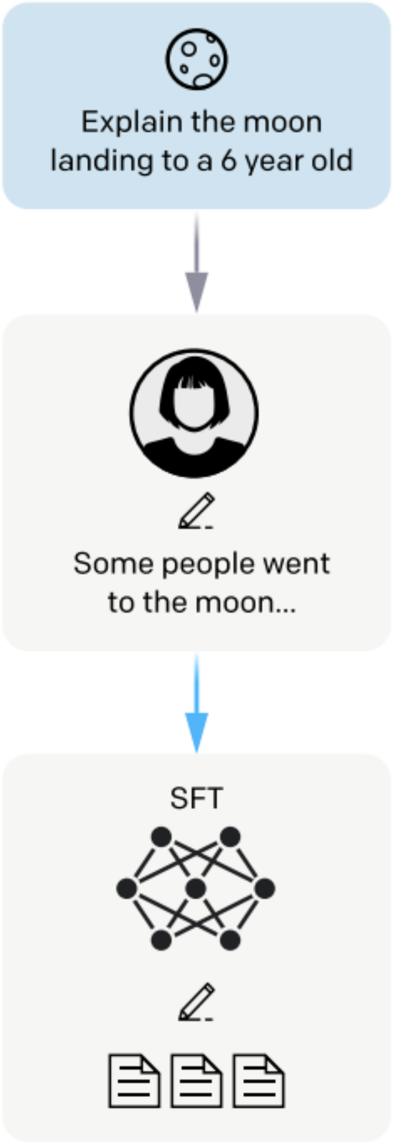
Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

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



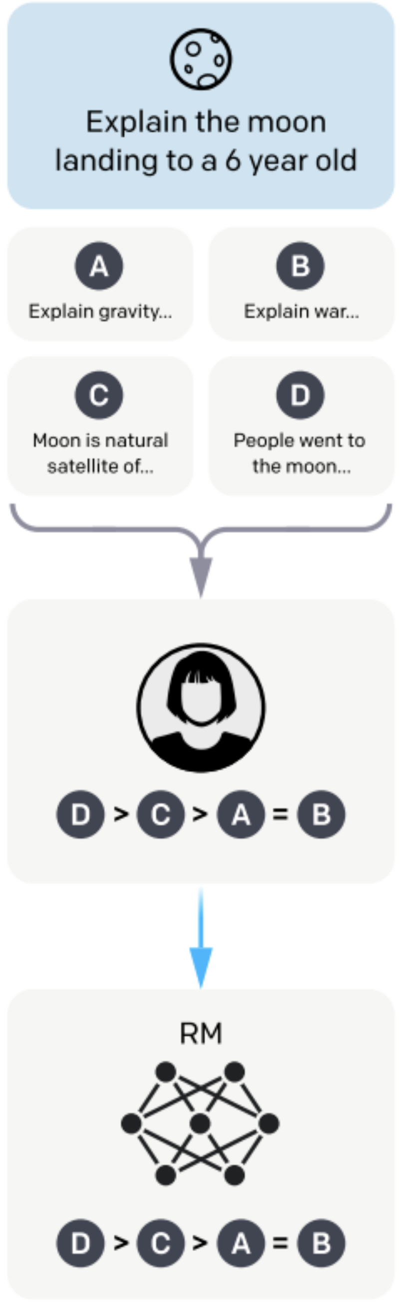
Step 2

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

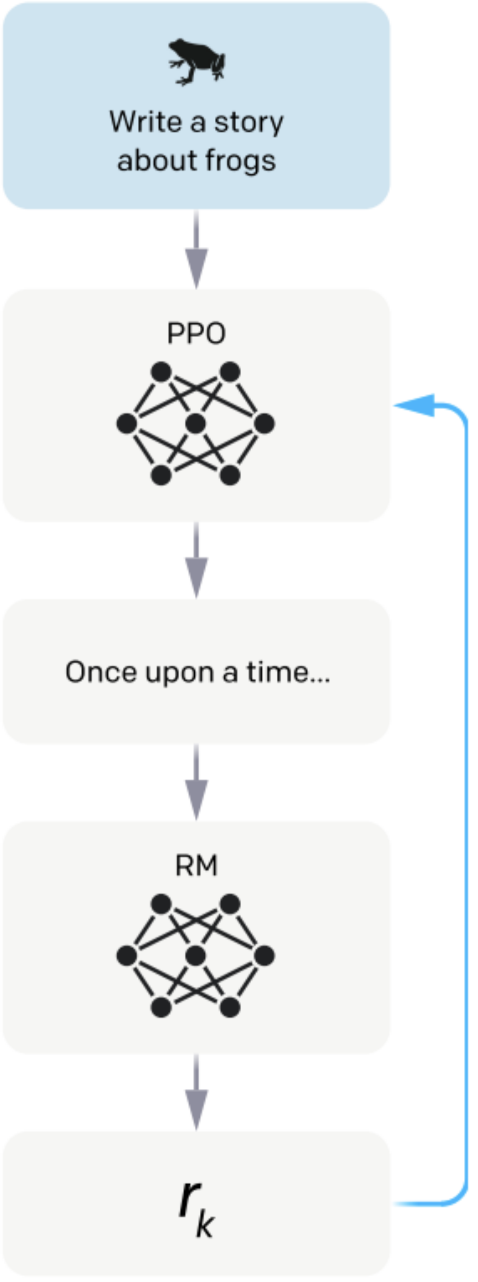
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

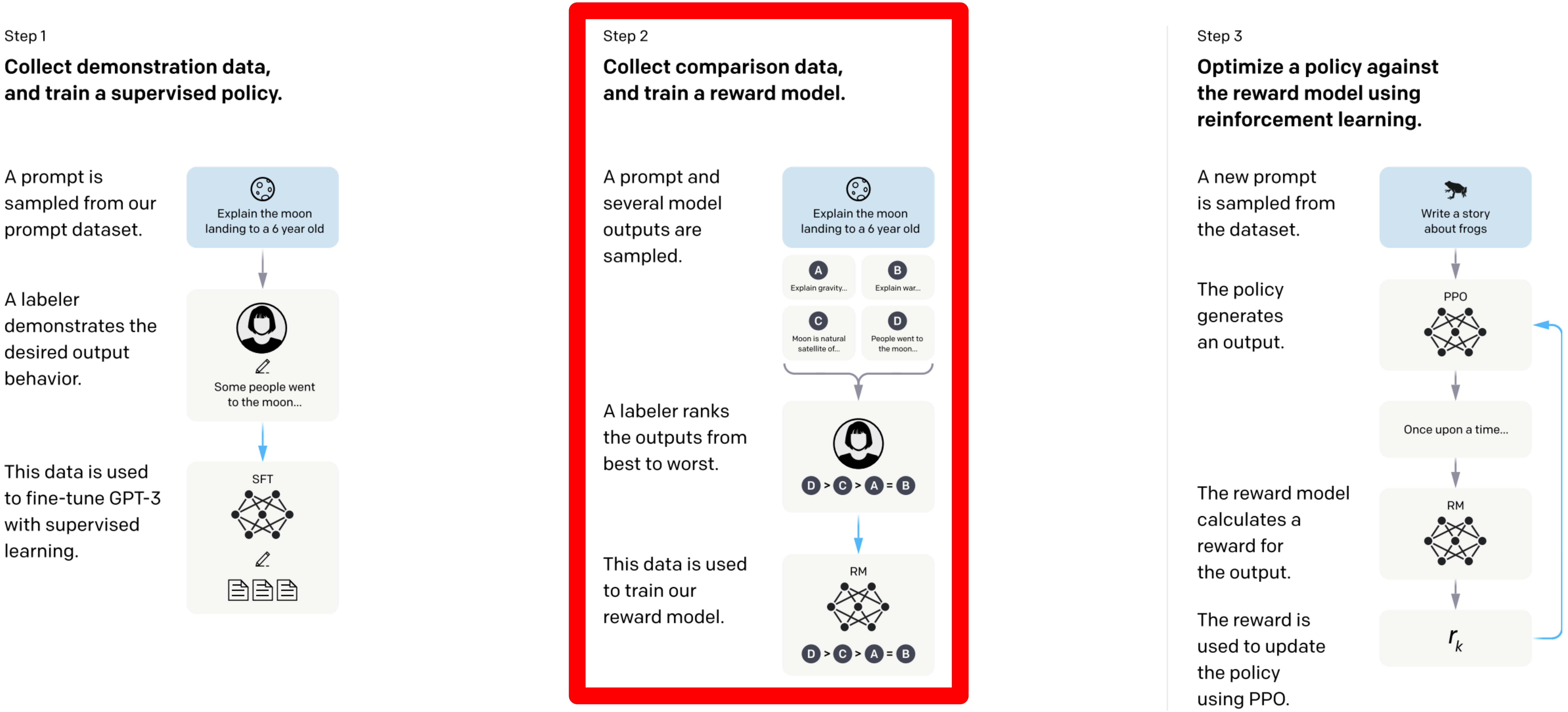
The reward is used to update the policy using PPO.



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

RLHF: Reinforcement Learning From Human Feedback

Step 0:
unsupervised
generative
modeling on
A TON of text
(pre-training)



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

RLHF: Learning a reward model from human feedback

Feedback comes as **preferences over model samples:**

$$\mathcal{D} = \{x^i, y_w^i, y_l^i\}$$

Prompt

Preferred response

Dispreferred response

RLHF: Learning a reward model from human feedback

Feedback comes as **preferences over model samples:**

$$\mathcal{D} = \{x^i, y_w^i, y_l^i\}$$

Prompt

Preferred response

Dispreferred response

Bradley-Terry Model connects rewards to preferences:

Reward assigned to **preferred** and **dispreferred** responses

$$p(y_w \succ y_l \mid x) = \sigma(r(x, y_w) - r(x, y_l))$$

RLHF: Learning a reward model from human feedback

Feedback comes as **preferences over model samples:**

$$\mathcal{D} = \{x^i, y_w^i, y_l^i\}$$

Prompt

Preferred response

Dispreferred response

Bradley-Terry Model connects rewards to preferences:

Reward assigned to **preferred** and **dispreferred** responses

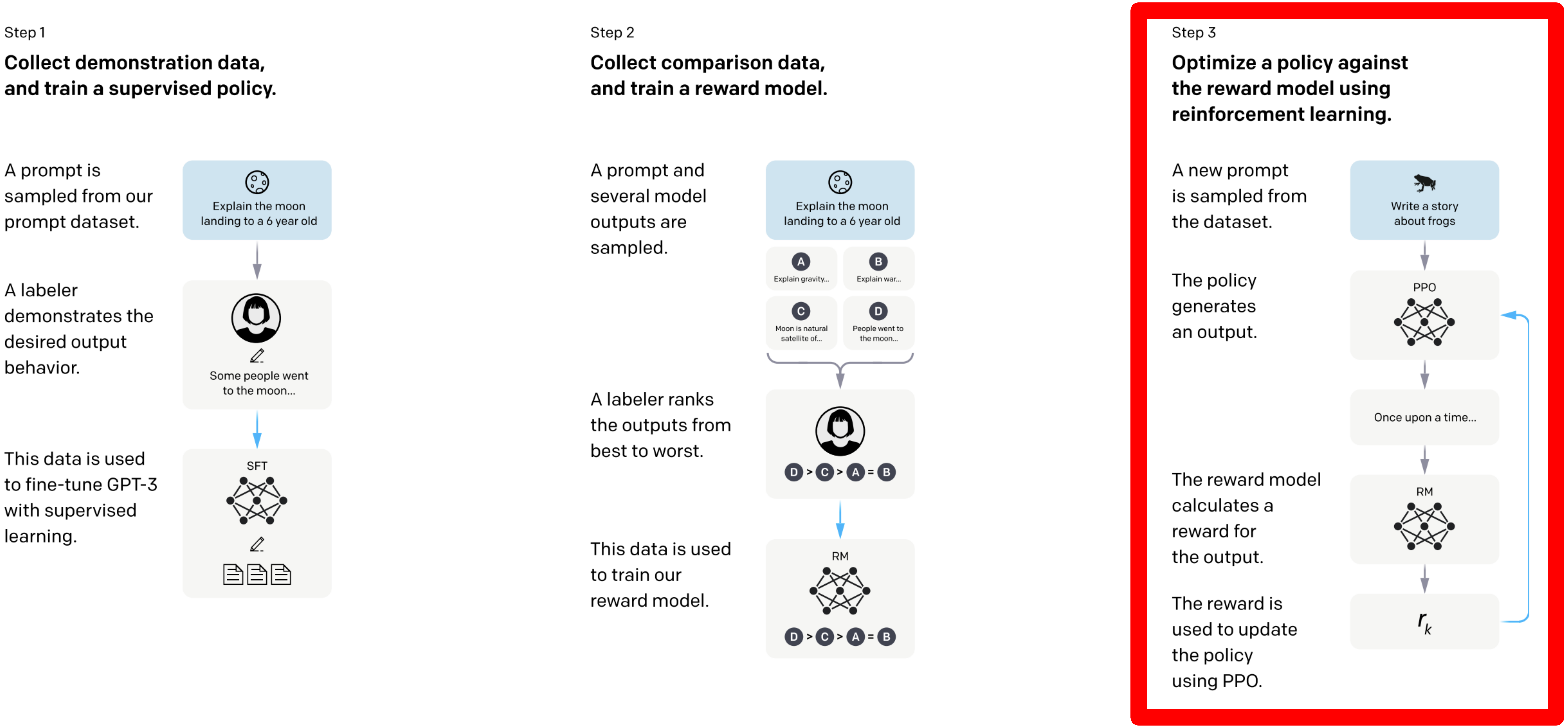
$$p(y_w \succ y_l \mid x) = \sigma(r(x, y_w) - r(x, y_l))$$

Train the reward model by **minimizing negative log likelihood:**

$$\mathcal{L}_R(\phi, \mathcal{D}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma(r_\phi(x, y_w) - r_\phi(x, y_l))\right]$$

RLHF: Learning a policy that optimizes the reward

Step 0:
unsupervised generative modeling on **A TON** of text (pre-training)



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

RLHF: Learning a policy that optimizes the reward

Now we have a **reward model** r_ϕ that represents **goodness according to humans**

RLHF: Learning a policy that optimizes the reward

Now we have a **reward model** r_ϕ that represents **goodness according to humans**

So we learn a policy π_θ achieving **high reward**

$$\max_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(y|x)} \left[r_\phi(x, y) \right]$$

Sample from policy

Want high reward …

RLHF: Learning a policy that optimizes the reward

Now we have a **reward model** r_ϕ that represents **goodness according to humans**

So we learn a policy π_θ achieving **high reward** while **staying close** to original model π_{ref}

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)} \left[r_\phi(x, y) \right] - \beta \mathbb{D}_{\mathrm{KL}} \left[\pi_\theta(y|x) || \pi_{\text{ref}}(y|x) \right]$$

Sample from policy

Want high reward …

… but keep KL to original model small!

RLHF: Learning a policy that optimizes the reward

TL;DR: we need a dataset of preferences over response pairs: $\mathcal{D} = \{\underbrace{x^i}_{\text{Prompt}}, \underbrace{y_w^i}_{\text{Preferred response}}, \underbrace{y_l^i}_{\text{Dispreferred response}}\}$

From there, we can learn with **any off-the-shelf RLHF algorithm**

We pick **Direct Preference Optimization** because it is fast, stable, and effective

---

Direct Preference Optimization:
Your Language Model is Secretly a Reward Model

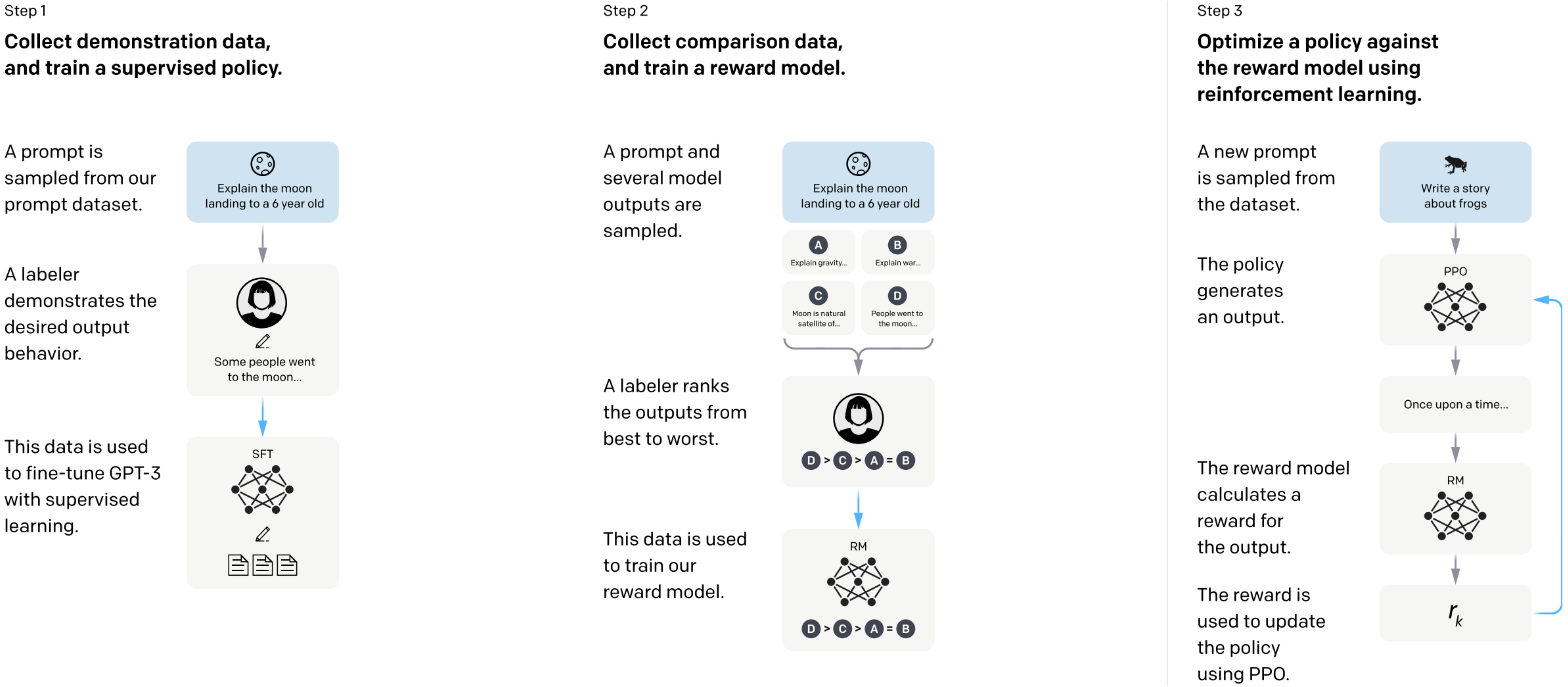
Rafael Rafailov[*†] **Archit Sharma**[*†] **Eric Mitchell**[*†]

Stefano Ermon[†‡] **Christopher D. Manning**[†] **Chelsea Finn**[†]

[†]Stanford University [‡]CZ Biohub
{rafailov,architsh,eric.mitchell}@cs.stanford.edu

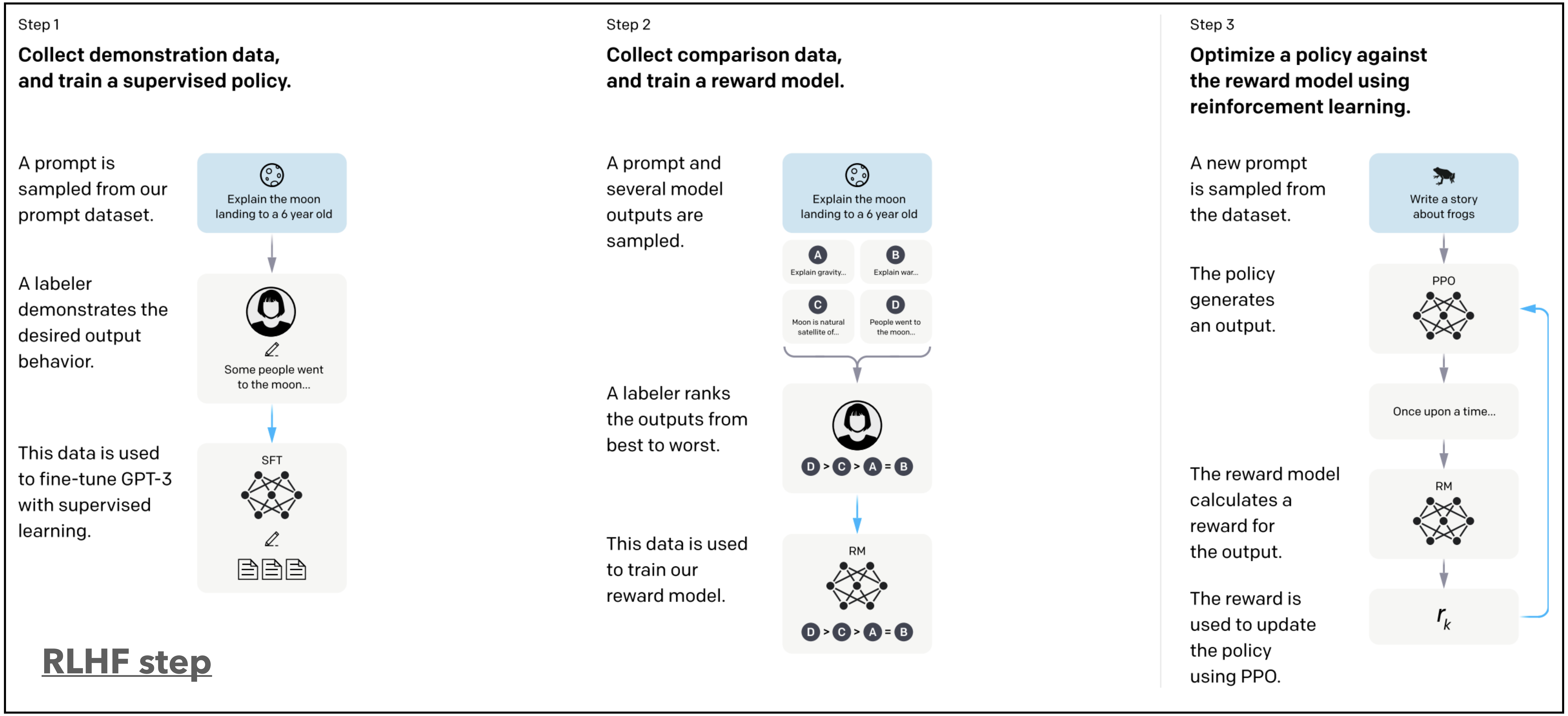
Where could we hope factuality would come from?

Step 0:
unsupervised generative modeling on **A TON** of text (pre-training)



Where could we hope factuality would come from?

<u>Step 0:</u> unsupervised generative modeling on **A TON** of text (pre-training)

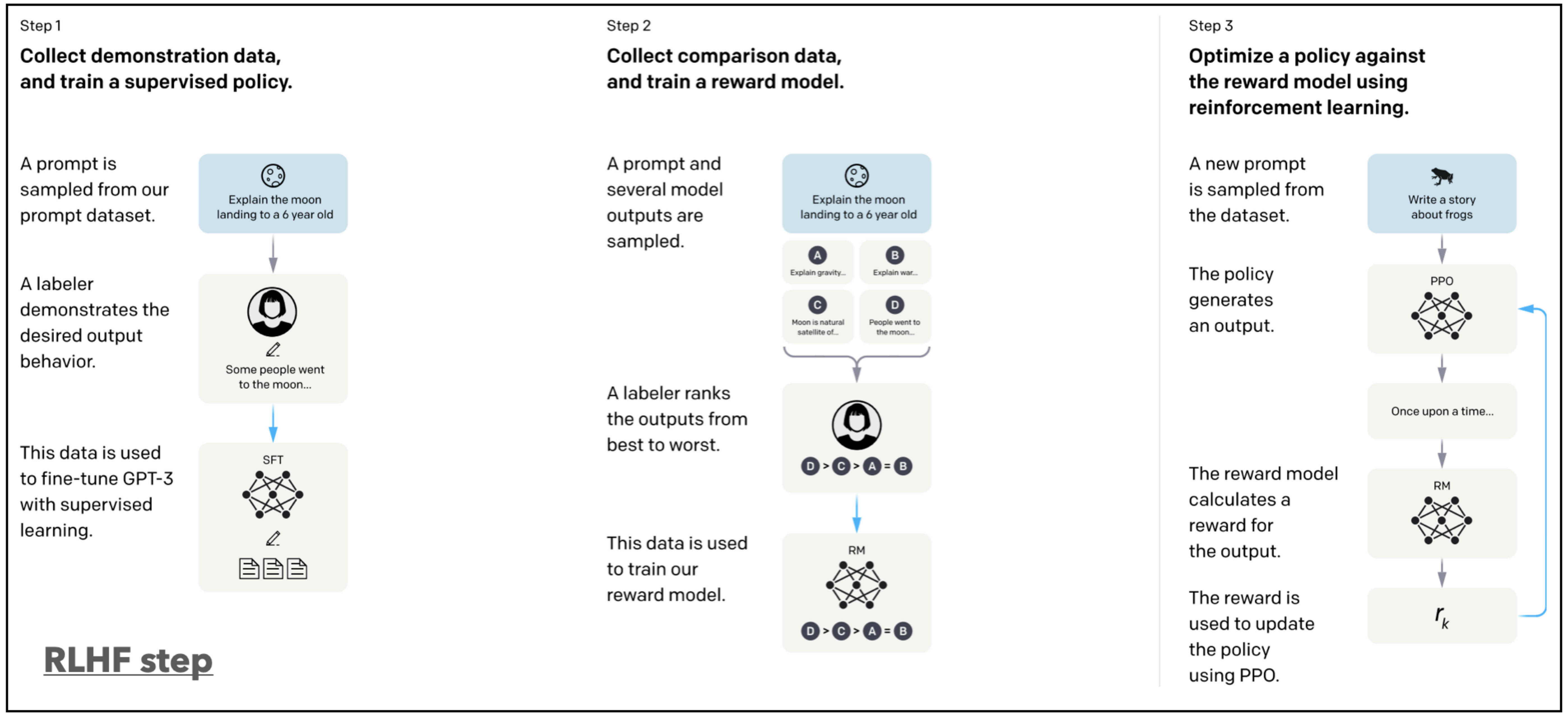


<u>RLHF step</u>

Where could we hope factuality would come from?

INTERNET ARCHIVE

Step 0:
unsupervised generative modeling on **A TON** of text (pre-training)



Pre-training: **learn what's true & false**

~a ***trillion*** words

RLHF: **learn to say only the true stuff!**

~a ***billion*** words

Training LLMs to be more factual

Q1: We already do RLHF; why do we need anything special for factuality?

A1: *RLHF encourages* ***behaviors that make human labelers happy***

Unfortunately, deciding "is this response factually accurate" is much harder than deciding "do I like this response" → Human labels only ***weakly*** *encourage truth*

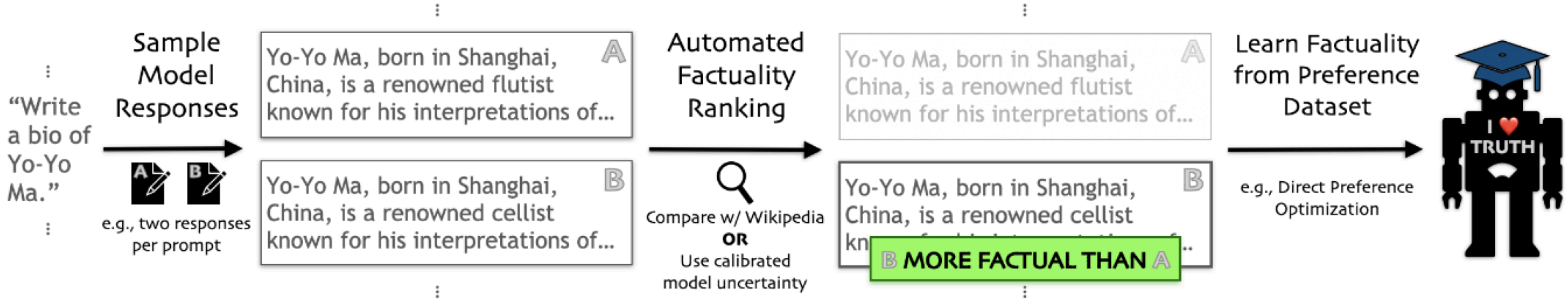
Q2: The truthfulness results so far have been on short QA. **How do we get factuality for long responses?**

A1: *We'll* ***decompose*** *long responses into their* ***atomic factual claims,*** *and judge their truthfulness one by one*

Training LLMs to be more factual

Tian*, Mitchell*, Yao, Manning, Finn (2023)

RLHF lets us train on data saying **when one response is better than another**

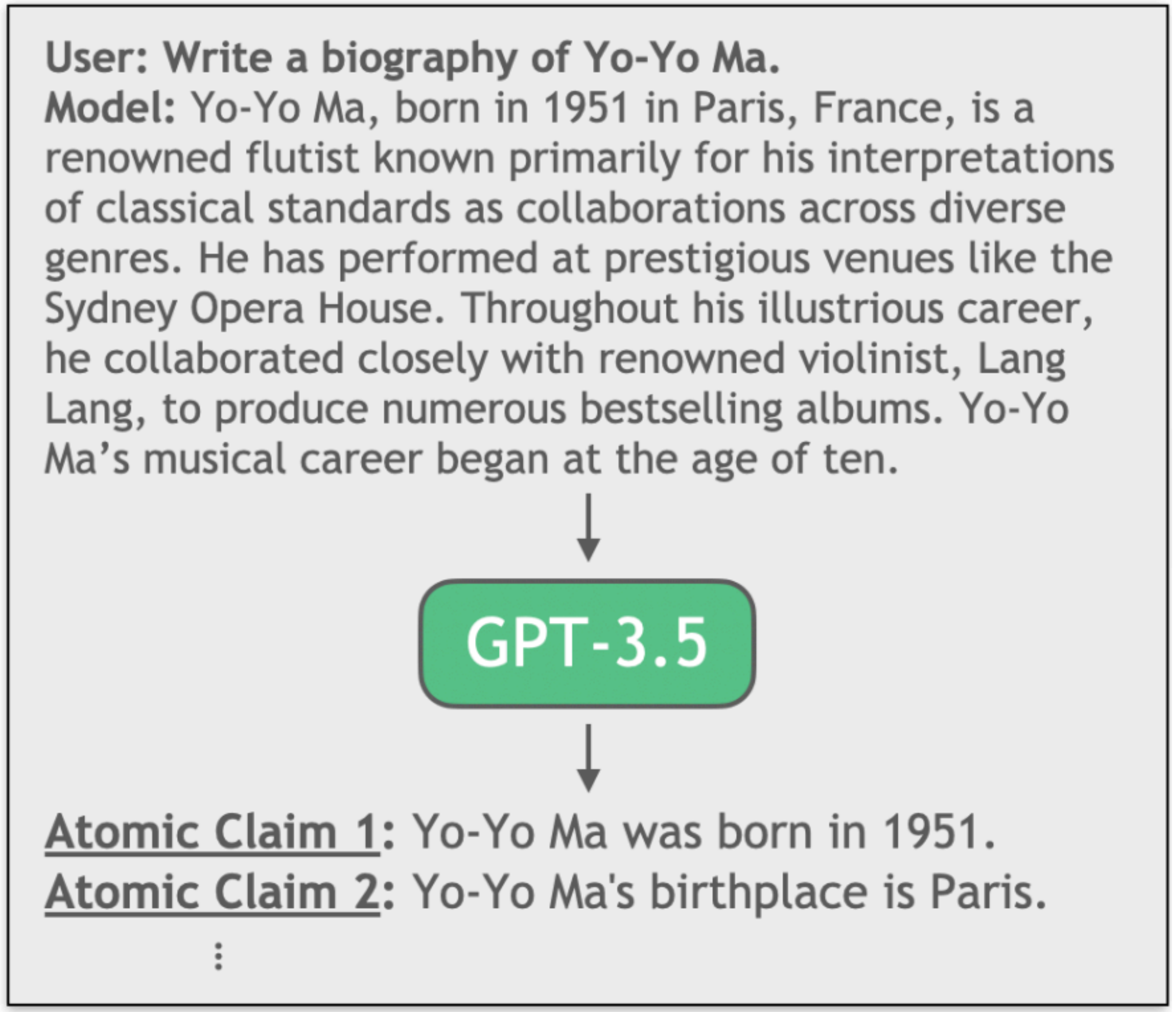


How can we do this "automated factuality ranking"?

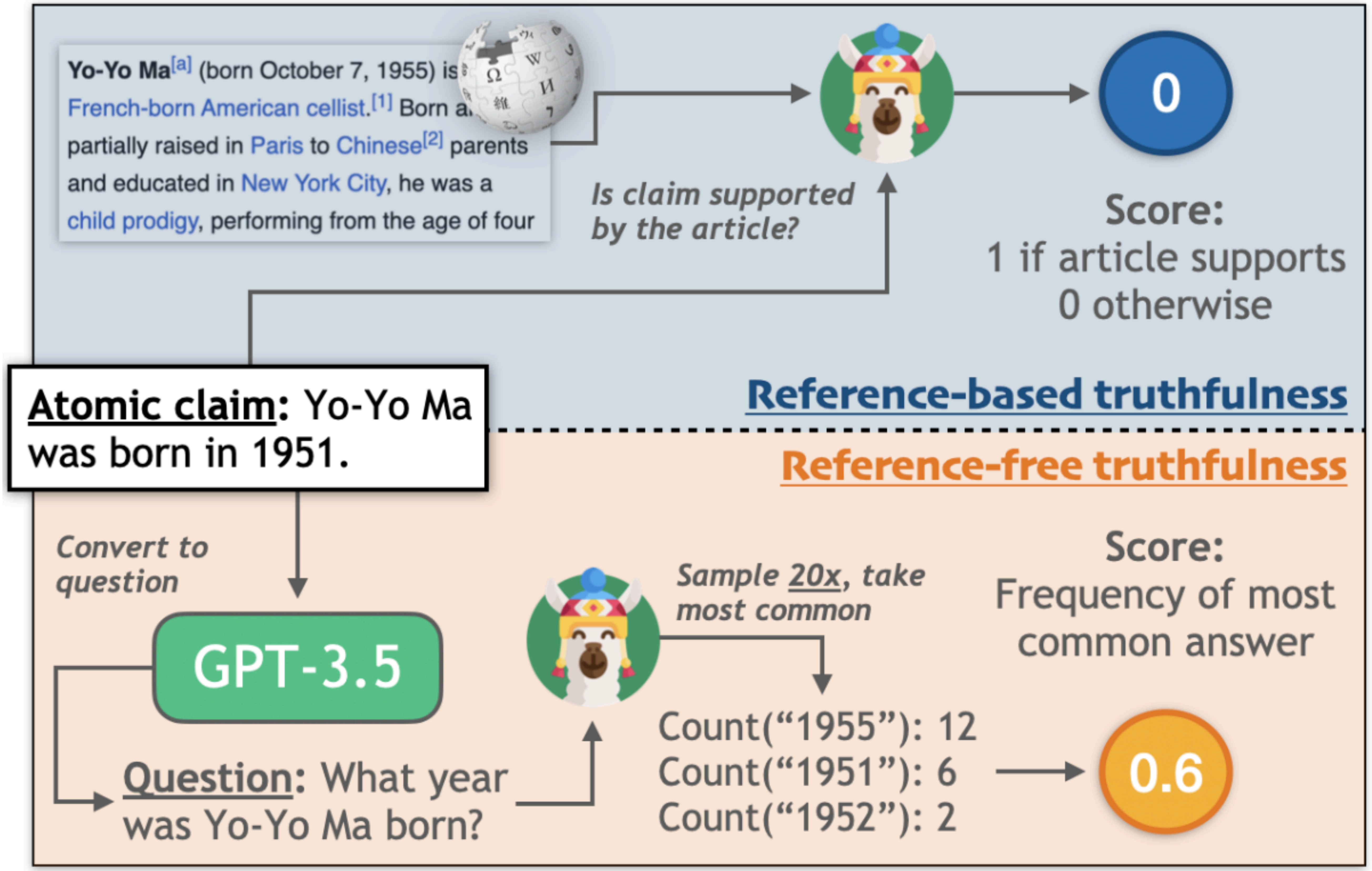
Humans are slow, expensive fact-checkers…

Training LLMs to be more factual

I. Extract **atomic claims** from sample



II. Estimate **truthfulness score** of each atomic claim



For reference-based truthfulness, we use FactScore (Min et al., 2023)

Training LLMs to be more factual

Tian*, Mitchell*, Yao, Manning, Finn (2023)

So… does RLHF actually let us **fine-tune to be more factual?**

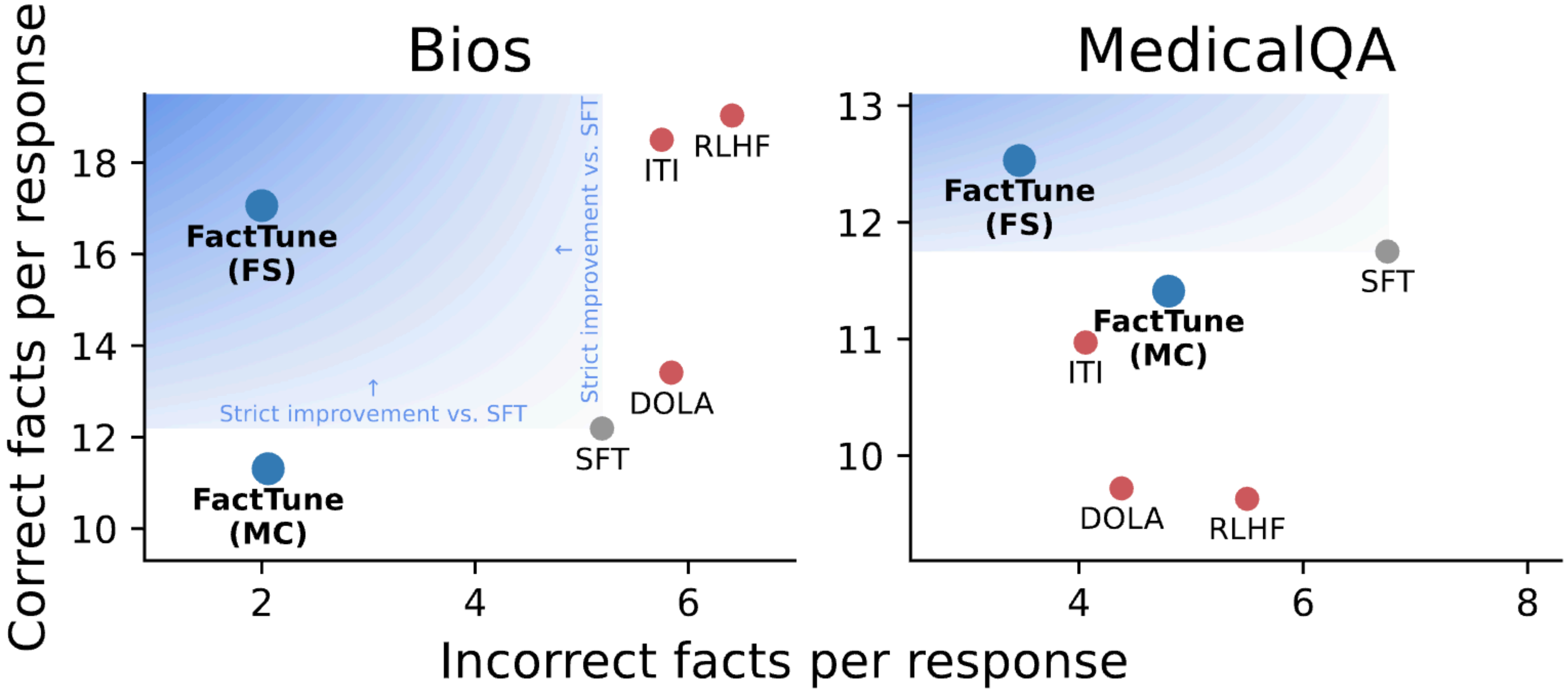
Evaluate **factuality tuning** on **long-form generation tasks:**

- Writing **bios** of popular figures
- Answer **medical questions** (“What are symptoms of pulmonary edema?”)

Baselines are supervised fine-tuning (**SFT**) on demonstrations, full **RLHF**, or test-time modifications to model sampling (**ITI**, **DOLA**)

Training LLMs to be more factual

Tian*, Mitchell*, Yao, Manning, Finn (2023)



Only **factuality tuning** (using the reference-based factuality ranking) **strictly improves** over supervised fine-tuning

Training LLMs to be more factual

Tian*, Mitchell*, Yao, Manning, Finn (2023)

		Biographies			Medical QA		
Base Model	Method	#Correct	#Incorrect	%Correct	#Correct	#Incorrect	%Correct
Llama-1	DPO-FS	14.81	3.75	0.812	10.88	4.50	0.707
	DPO-FS + DOLA	12.44	2.00	0.864	11.47	3.75	0.767

Factuality tuning can be stacked with **test-time methods** for modifying LM sampling to improve factuality (like DOLA)

Other Related Work

- Instead of fine-tuning, some methods try to **modify sampling** to bias toward correct statements
  - e.g. Inference-Time Intervention (Li et al., 2023) uses the CCS idea to bias activations toward the "truth direction"
- Instead of generating the truthful stuff from the start, at least **detect** non-truthful things after the fact
  - e.g., Semantic entropy (Kuhn et al., 2022) or SelfCheckGPT (Manakul et al., 2023)

Conclusions

Building **systems that produce factual outputs** is a critical challenge in NLP

There is some cause to believe we can do this, since **LLMs possess (some) internal model of what is true and what is false**

- Their representations can be decoded into predictions of truth/falsehood
- They can produce calibrated probabilities that a possible answer is correct

Unlike typical RLHF, **RL w/ automated factuality rankings** improves factuality!

There is still lots to do; consider working on factuality & robustness :)

Feel free to reach out with thoughts or questions:

@ericmitchellai *eric.mitchell@cs.stanford.edu*