Learning from Feedback

Haw-Shiuan Chang

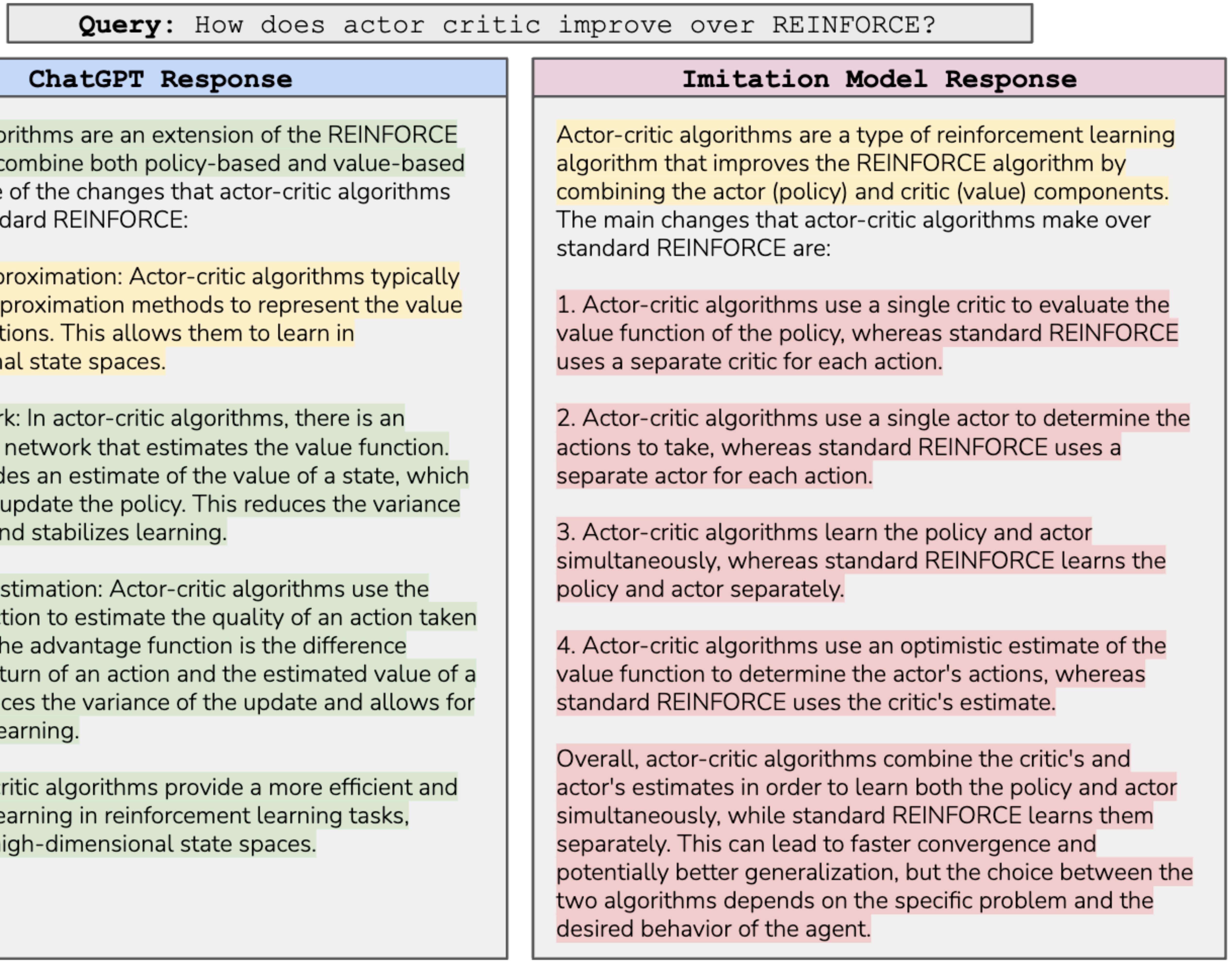
Deadlines

- **https://people.cs.umass.edu/~hschang/cs685/schedule.html**

- **3/14**: HW 1 due
- **3/17:** Quiz 3
- **4/11**: HW 2 due
  - Will be released before the spring break

- **5/9**: Last day to submit extra credit
  - Please check the announcement at Piazza for the recording link

Fine-tuning mostly Changes the Style



Query: How does actor critic improve over REINFORCE?

ChatGPT Response

Actor-critic algorithms are an extension of the REINFORCE algorithm that combine both policy-based and value-based methods. Some of the changes that actor-critic algorithms make over standard REINFORCE:

1. Function Approximation: Actor-critic algorithms typically use function approximation methods to represent the value and policy functions. This allows them to learn in high-dimensional state spaces.

2. Critic Network: In actor-critic algorithms, there is an additional critic network that estimates the value function. The critic provides an estimate of the value of a state, which is then used to update the policy. This reduces the variance of the update and stabilizes learning.

3. Advantage Estimation: Actor-critic algorithms use the advantage function to estimate the quality of an action taken by the policy. The advantage function is the difference between the return of an action and the estimated value of a state. This reduces the variance of the update and allows for more efficient learning.

Overall, actor-critic algorithms provide a more efficient and stable way of learning in reinforcement learning tasks, particularly in high-dimensional state spaces.

Imitation Model Response

Actor-critic algorithms are a type of reinforcement learning algorithm that improves the REINFORCE algorithm by combining the actor (policy) and critic (value) components. The main changes that actor-critic algorithms make over standard REINFORCE are:

1. Actor-critic algorithms use a single critic to evaluate the value function of the policy, whereas standard REINFORCE uses a separate critic for each action.

2. Actor-critic algorithms use a single actor to determine the actions to take, whereas standard REINFORCE uses a separate actor for each action.

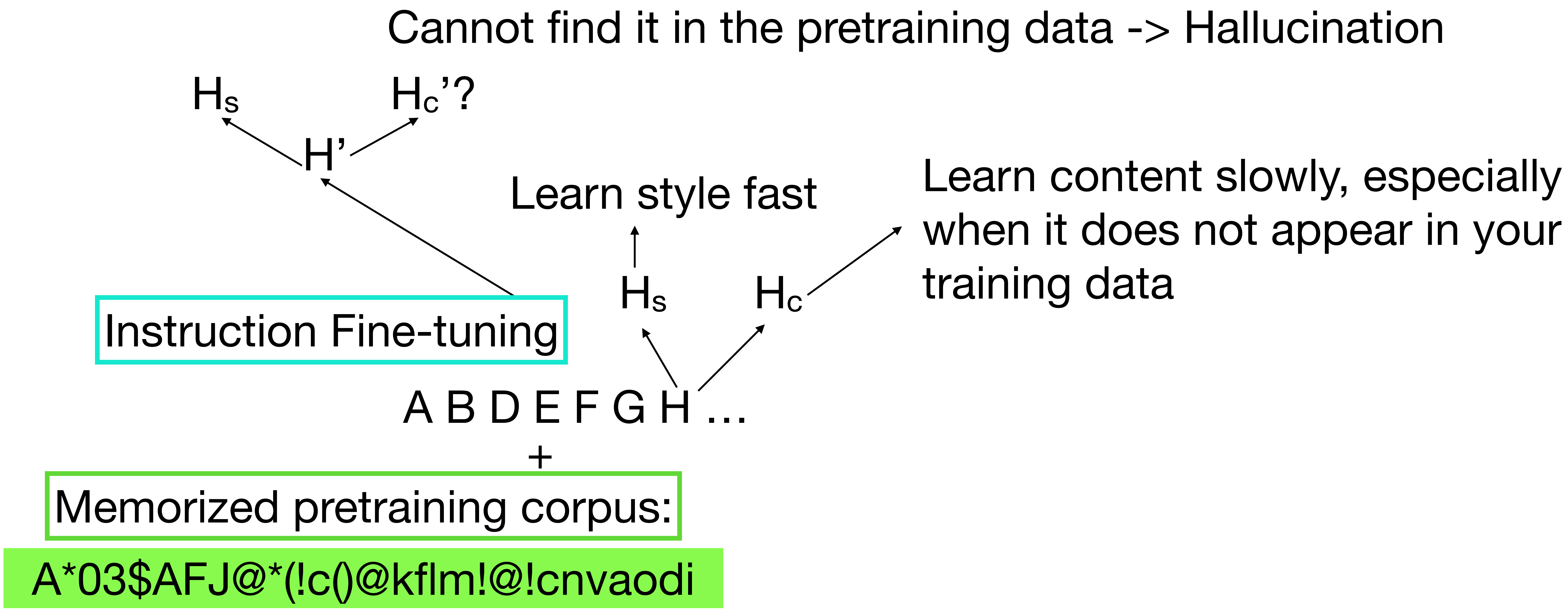
3. Actor-critic algorithms learn the policy and actor simultaneously, whereas standard REINFORCE learns the policy and actor separately.

4. Actor-critic algorithms use an optimistic estimate of the value function to determine the actor's actions, whereas standard REINFORCE uses the critic's estimate.

Overall, actor-critic algorithms combine the critic's and actor's estimates in order to learn both the policy and actor simultaneously, while standard REINFORCE learns them separately. This can lead to faster convergence and potentially better generalization, but the choice between the two algorithms depends on the specific problem and the desired behavior of the agent.

The False Promise of Imitating Proprietary LLMs (https://arxiv.org/pdf/2305.15717)

Limitations of Fine-tuning



Cannot find it in the pretraining data -> Hallucination

H_s H_c'?

H'

Learn style fast

Learn content slowly, especially when it does not appear in your training data

H_s H_c

Instruction Fine-tuning

A B D E F G H …

+

Memorized pretraining corpus:

A*03$AFJ@*(!c()@kflm!@!cnvaodi

Old Fine-Tuning

- Main purpose
  - Learn to do the task
- Data Quality Requirement
  - Usually medium
- Fine-tuning compared to few-shot?
  - Better

Instruction Tuning

- Main purpose
  - Learn to do the task / Learn the output format
- Data Quality Requirement
  - Usually medium
- Fine-tuning compared to few-shot?
  - Better

SFT

- Main purpose
  - Often inducing the high-quality pretraining data
    - (The mechanisms are not fully clear yet)
- Data Quality Requirement
  - High
- Fine-tuning compared to few-shot?
  - Could be worse
    - (contradict to the traditional ML common sense)

Midterm Example Question

- We have an LLM base model (only pretrained). Which of the following fine-tuning data might boost the LLM’s helpfulness as a chatbot?
- (A) 200 expert-written summaries
- (B) 1k expert-written summaries
- (C) 10k stories from Reddit
- (D) 200 most upvoted posts in Stack Exchange
- (E) 100k News in the domains LLM are not familiar with
- (D)

Midterm Example Question

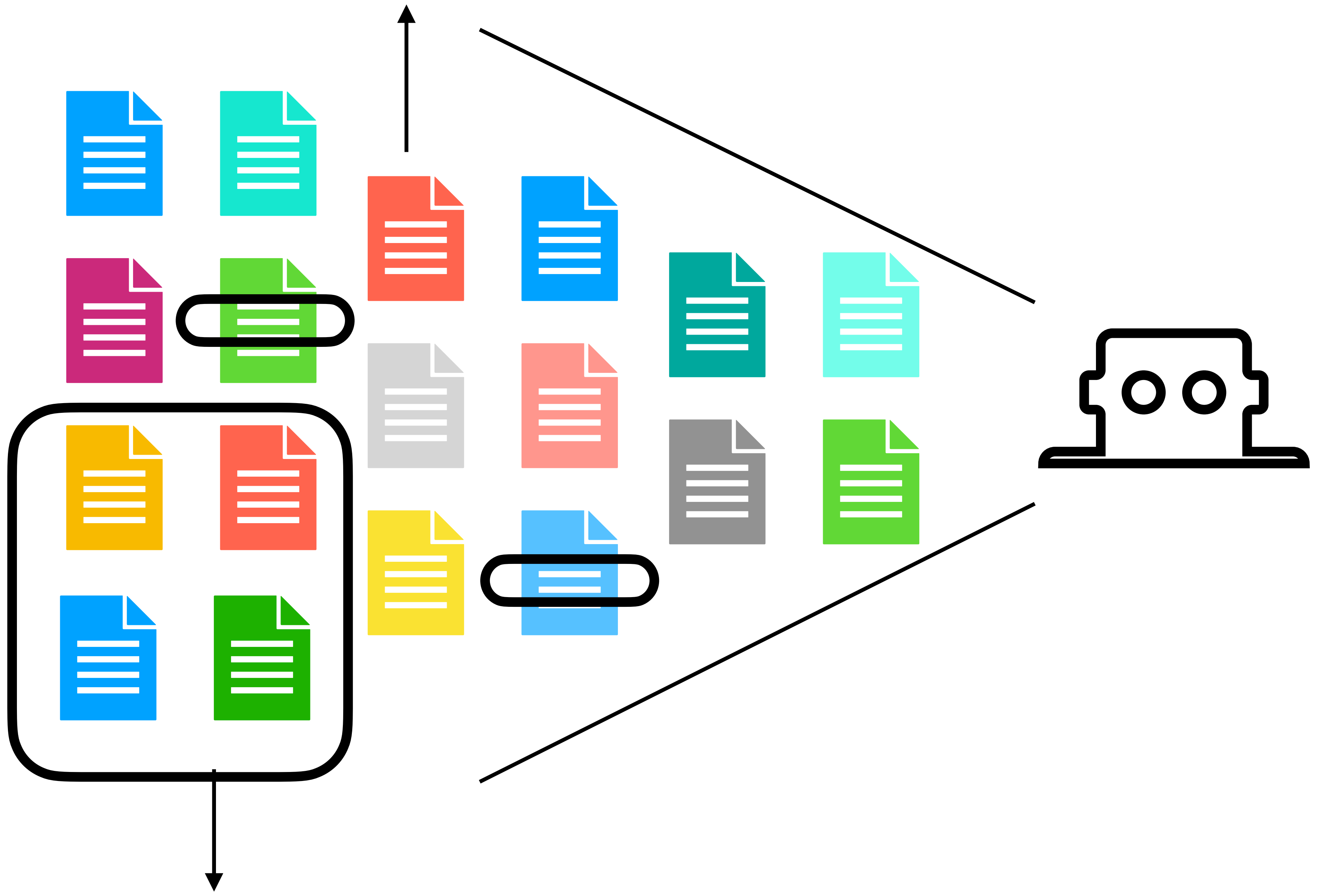
- We have an LLM base model (only pretrained). Which of the following fine-tuning data might boost the LLM's few-shot performance in the corresponding task the most?
- (A) 200 expert-written summaries for summarization
- (B) 1k expert-written summaries for summarization
- (C) 10k stories from Reddit for story generation
- (D) 200 most upvoted posts in StackOverflow for QA in code domain
- (E) 100k News in the domains LLM are not familiar with for news generation
- (D) > (B) > (A) > (E) >? 0 > (C)

Limitations of SFT

- Too expensive
  - Your quality needs to be close to the best response on the Internet
  - Hiring experts is expensive
- Fine-tuning on unfamiliar materials could cause hallucination
- Could easily affect the different tasks
- Do not have negative examples
  - LLM only knows what it should say. It does not know what it should not say
  - Hard to prevent generating harmful/toxic responses

LLM Development

Internet low-quality text (e.g., from trolls or haters)

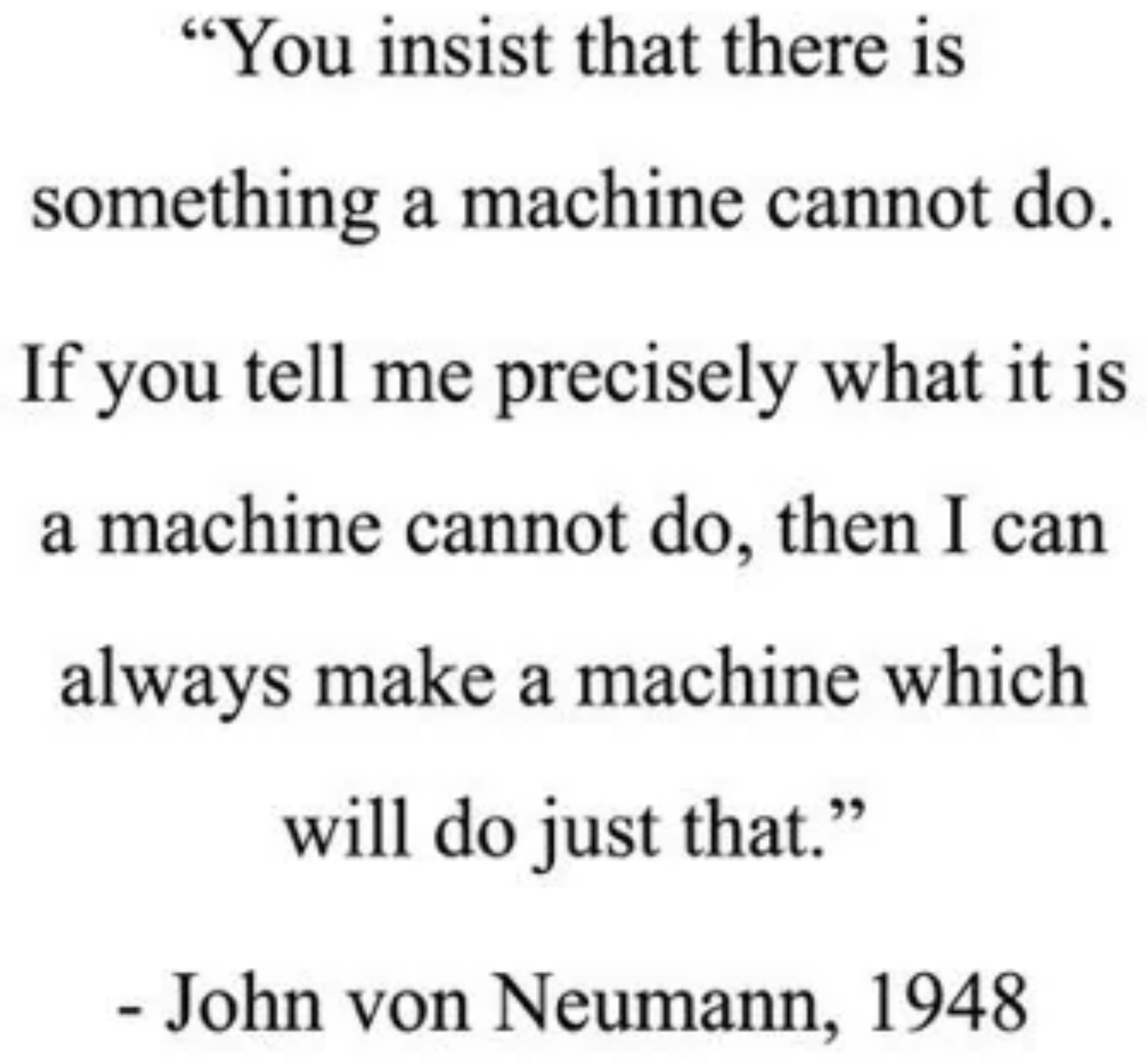


Internet high-quality text

- Architectures
  - MLP
  - RNN
  - Transformer
- Training Stages
  - Pretraining
  - Supervised Fine-tuning (SFT)
  - Alignment
    - **Learning from Human Feedback (LHF)**
    - Reasoning

Post-training stage
(Filtering process)

LLM Evaluation



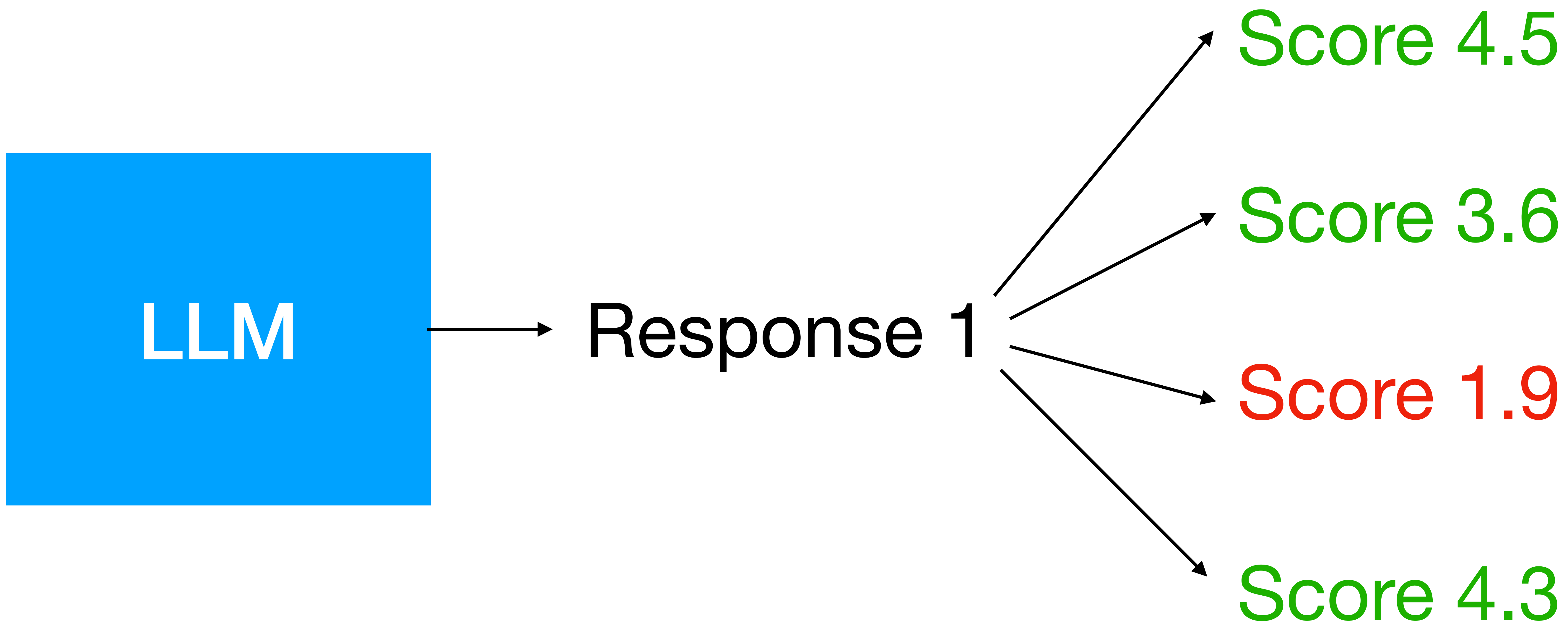
"You insist that there is something a machine cannot do. If you tell me precisely what it is a machine cannot do, then I can always make a machine which will do just that."

- John von Neumann, 1948

https://www.reddit.com/r/singularity/comments/18t02br/john_von_neumann_was_the_first_who_used_the/

Evaluation is Hard, Why?

Human as the Judge



LLM

Response 1

Score 4.5

Score 3.6

Score 1.9

Score 4.3

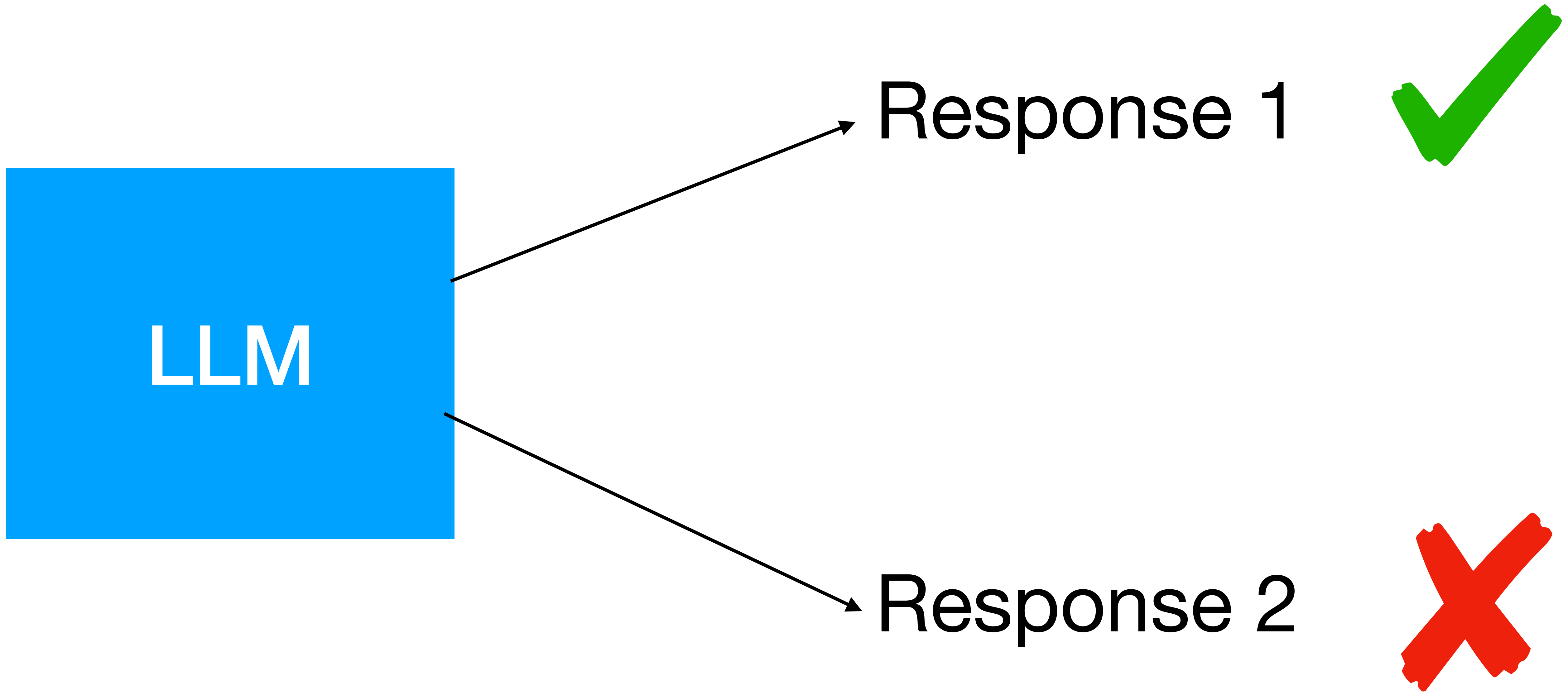
Noisy.

Every person has their preference.

Scores are easily affected by the context

Evaluation -> Loss

Human as the Judge /
LLM as the Judge



LLM

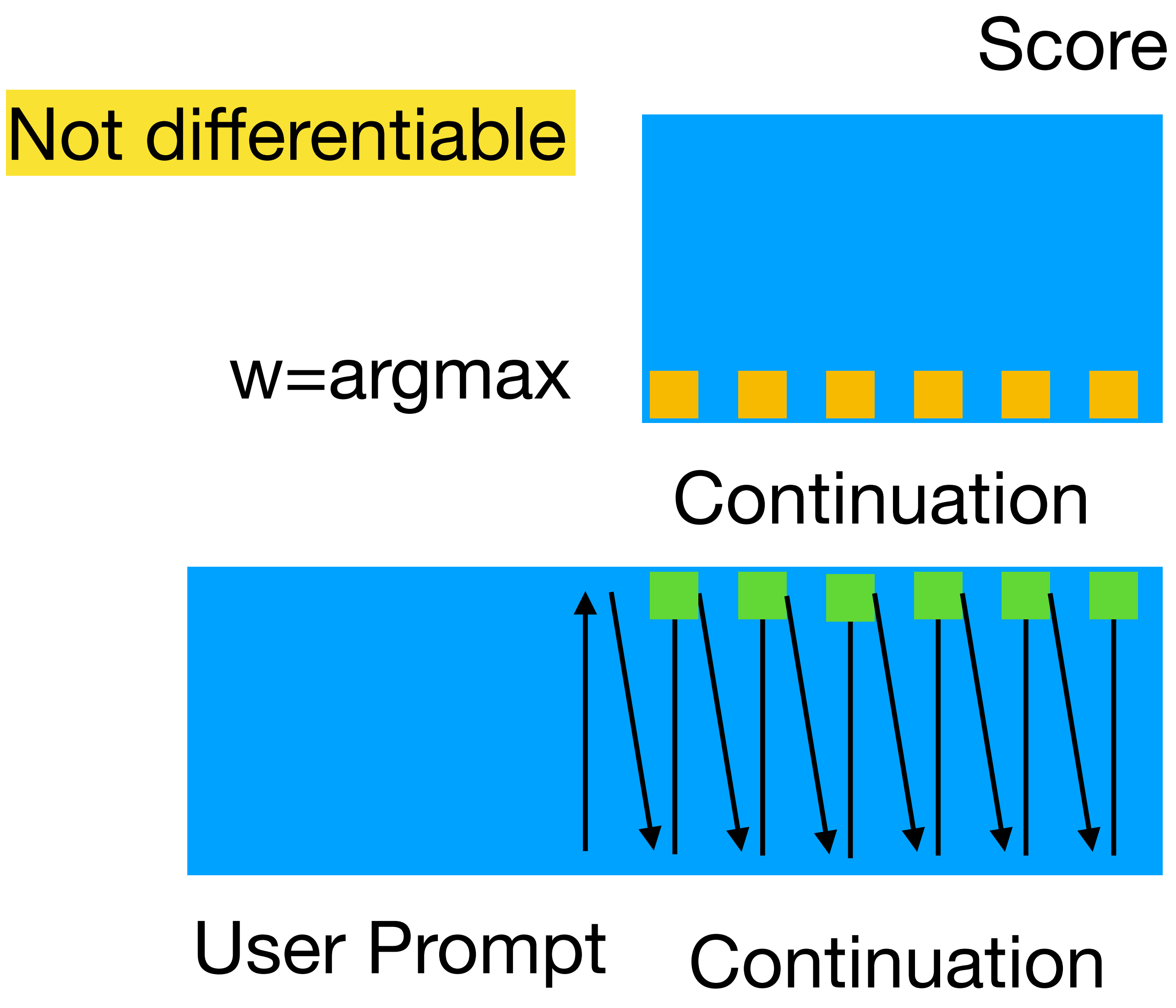
Response 1 ✓

Response 2 ✗

We Should Encourage the
LLM to output more of this

Differentiable Review

- Gradient descent is the easiest and most stable way
- If you change a little, the output cannot change a little.
  - That is not differentiable



Score

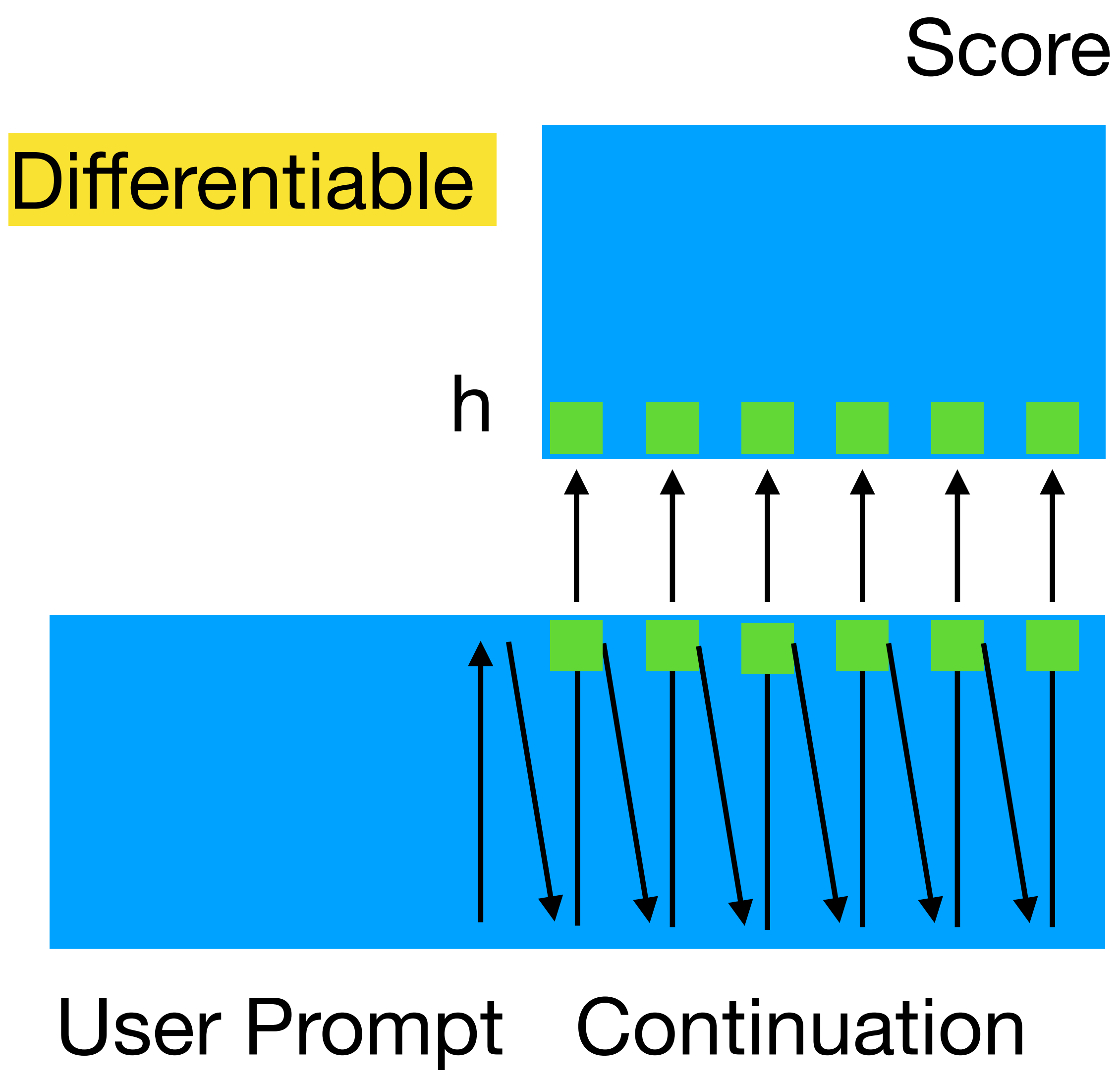
Not differentiable

w=argmax

Continuation

User Prompt

Continuation



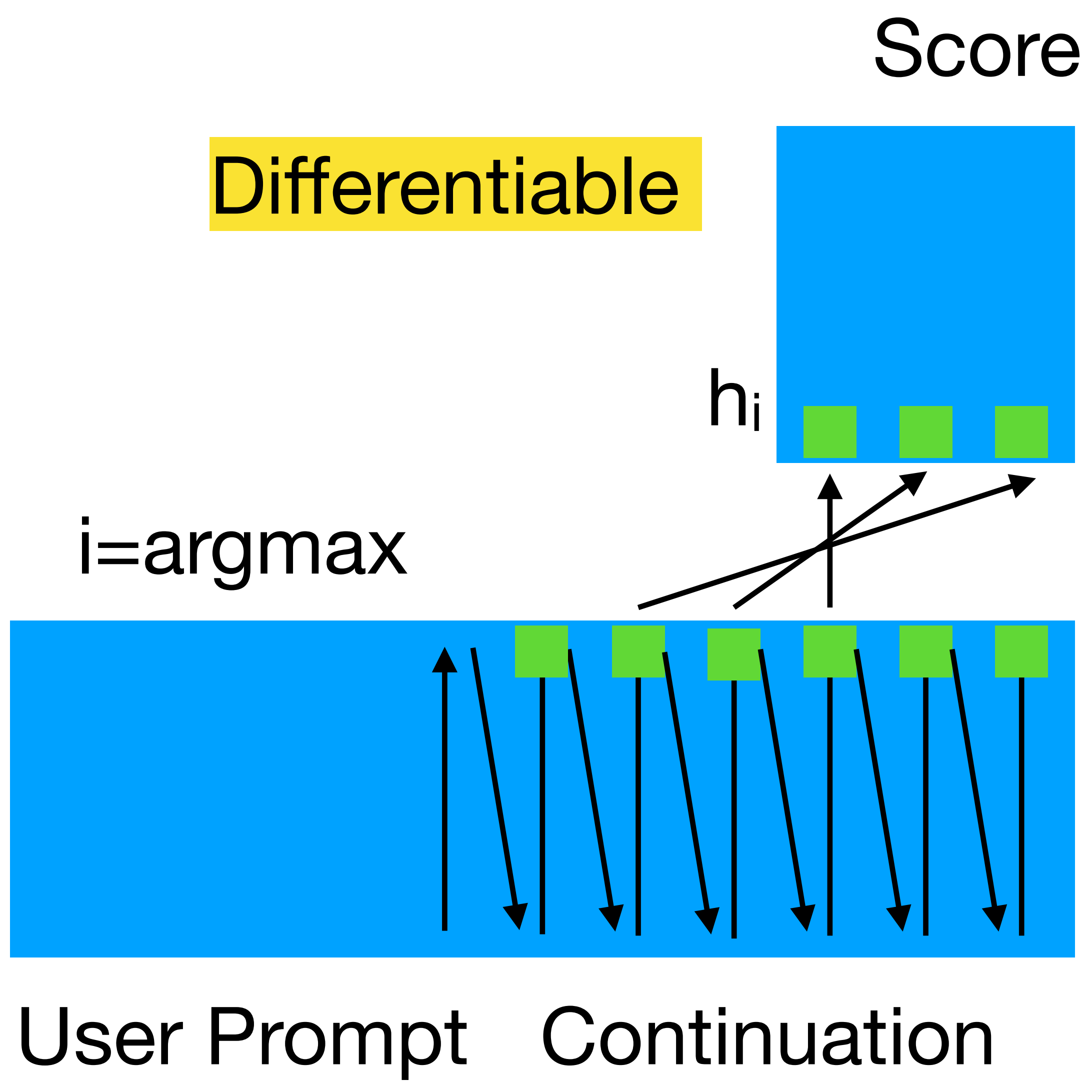
Score

Differentiable

h

User Prompt

Continuation



Score

Differentiable

h_i

i=argmax

User Prompt

Continuation

Last Year Note

Typos

al align
nodel

$$\max_{\pi} \underset{x,y}{E} \left[\underbrace{r(x,y)}_{\text{reward}} - \beta \underbrace{D_{KL}\left(\pi(y|x) \,||\, \pi_{ref}(y|x)\right)}_{\text{KL penalty to prevent huge deviations from } \pi_{ref}} \right]$$

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

$$D_{KL}\left(\pi(y|x) \,||\, \pi_{ref}(y|x)\right) = \log \frac{\pi(w_i | w_{1 \dots i-1}, x)}{\pi_{ref}(w_i | w_{1 \dots i-1}, x)}$$

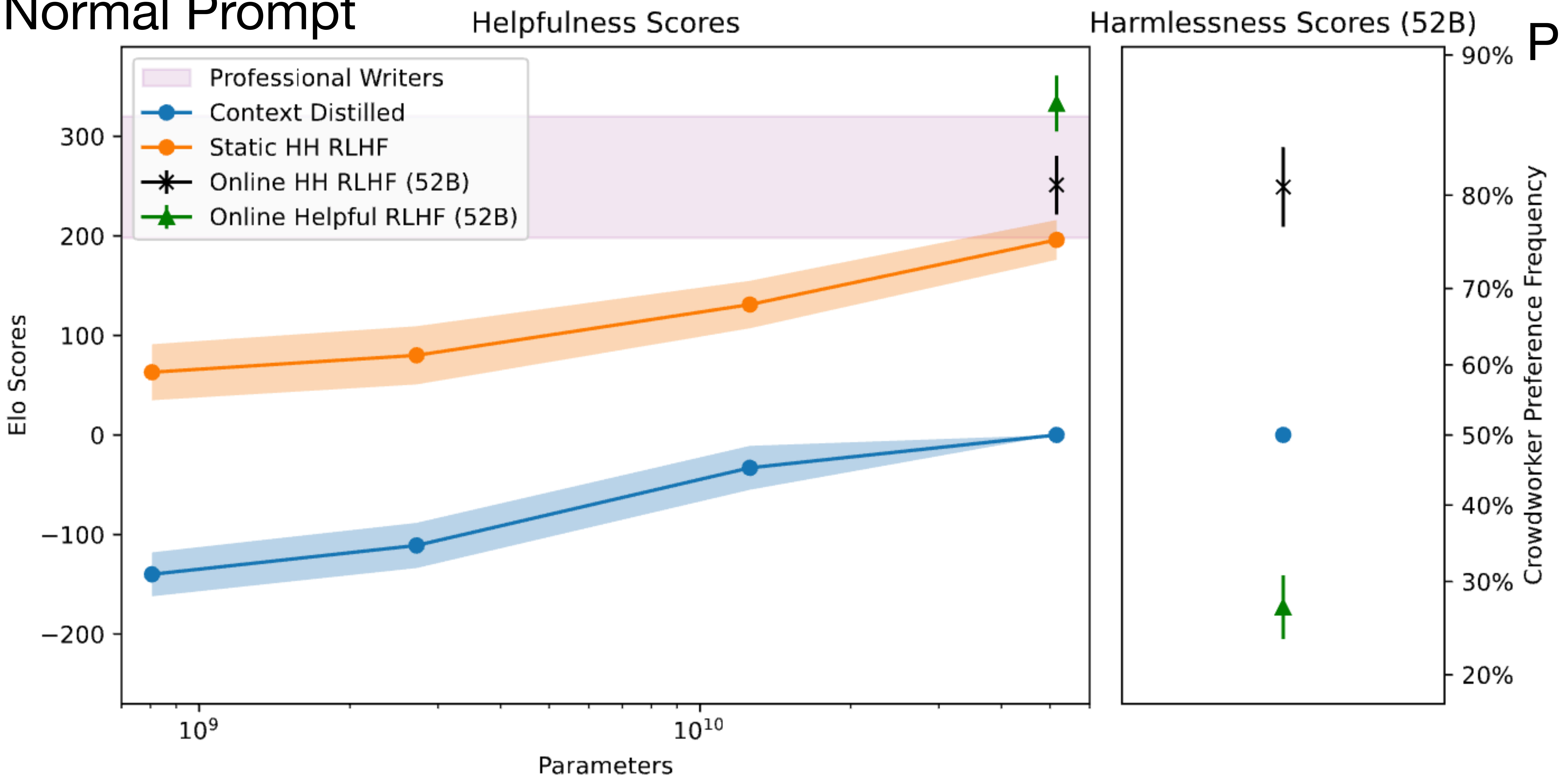
$P_{\text{FINAL RLHF MODEL}}$

$$\pi(w_i | w_{1 \dots i-1}, x)$$

Helpfulness vs Harmlessness

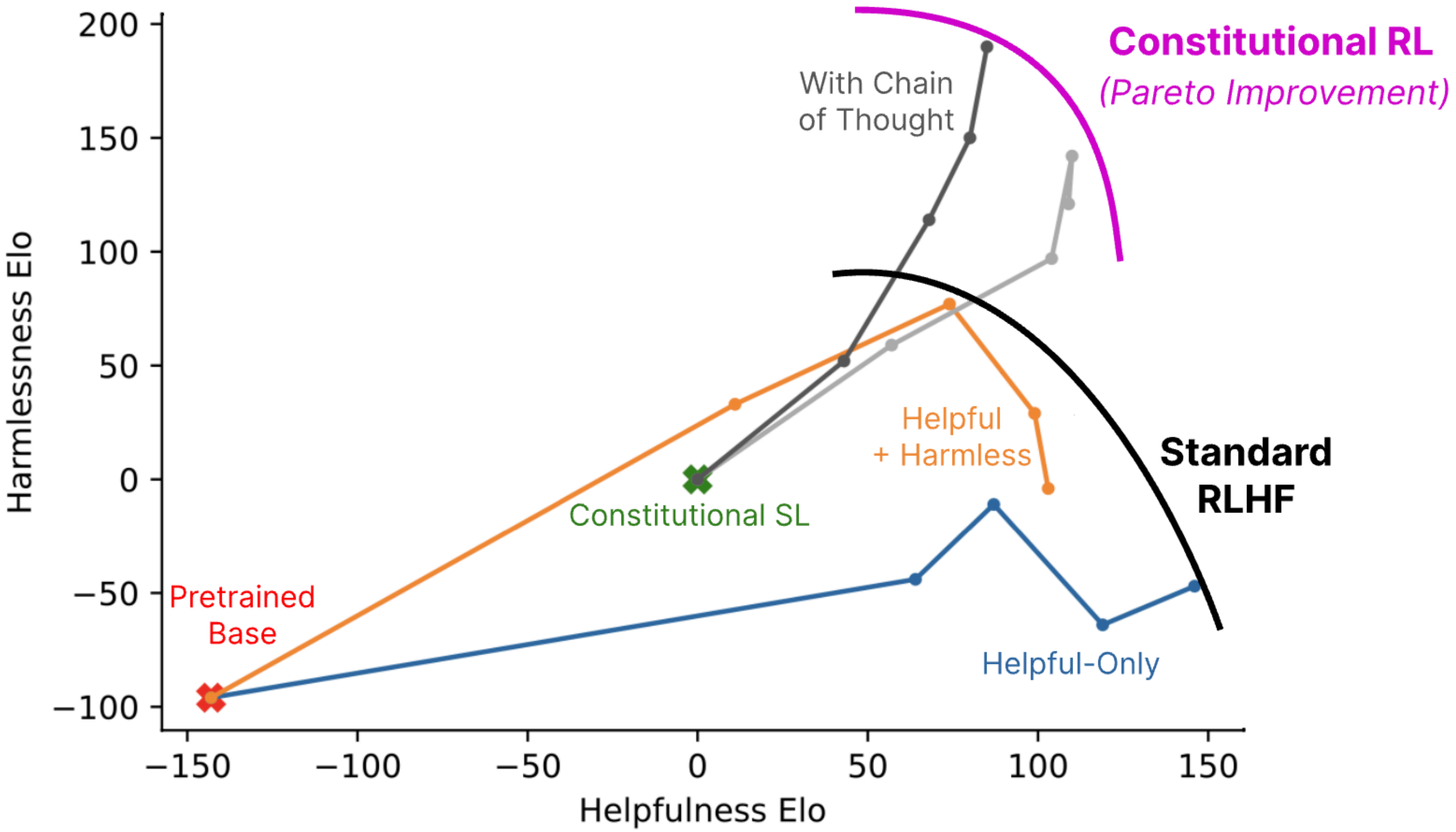
Normal Prompt

Adverserial Prompt



Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback (https://arxiv.org/abs/2204.05862)

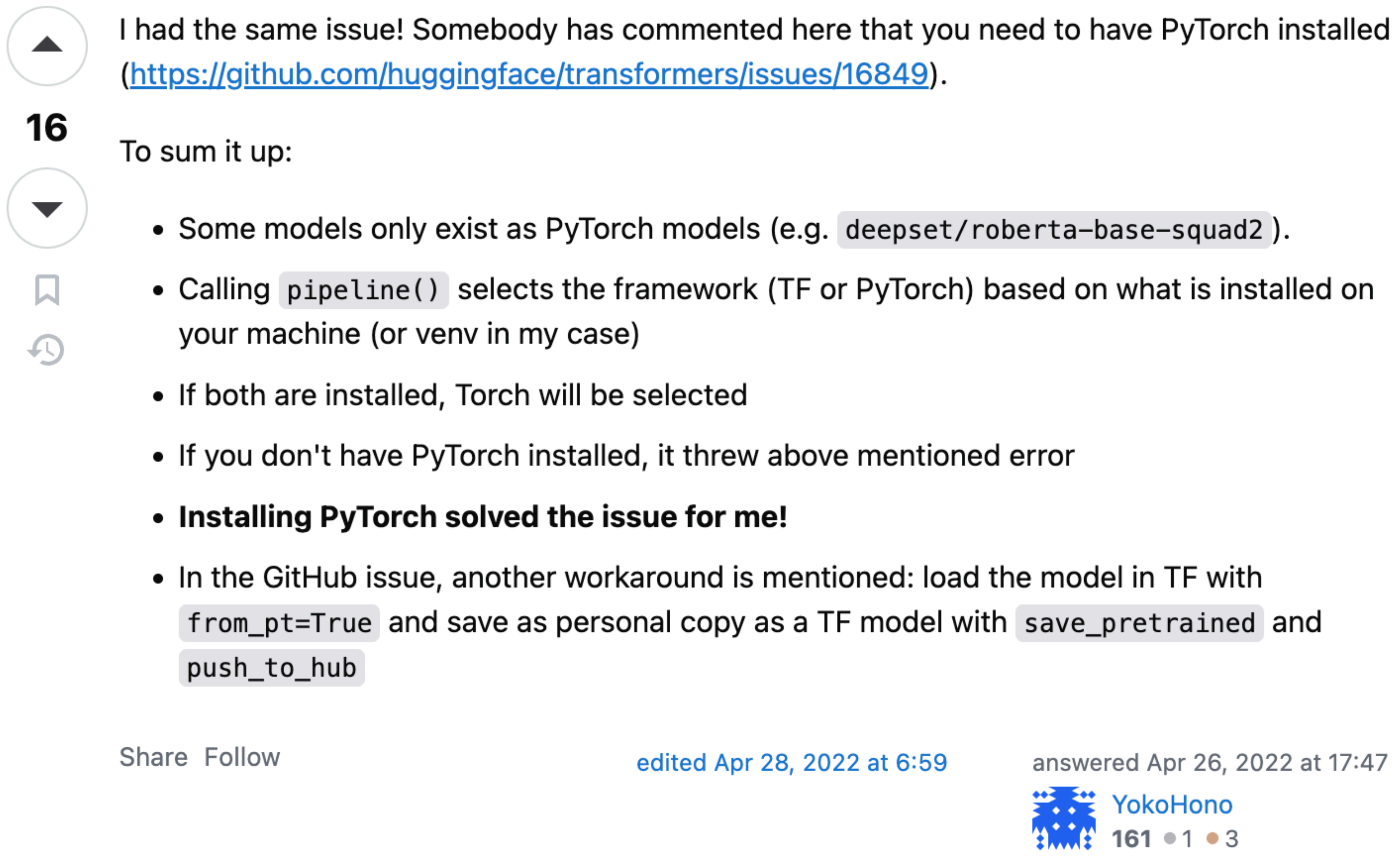
RLAIF



Constitutional AI: Harmlessness from AI Feedback (https://arxiv.org/abs/2212.08073)

Why can LLM be a Judge?

- Some websites have higher quality
- Different quality of text inducing different kinds of responses



I had the same issue! Somebody has commented here that you need to have PyTorch installed (https://github.com/huggingface/transformers/issues/16849).

To sum it up:

- Some models only exist as PyTorch models (e.g. `deepset/roberta-base-squad2`).
- Calling `pipeline()` selects the framework (TF or PyTorch) based on what is installed on your machine (or venv in my case)
- If both are installed, Torch will be selected
- If you don't have PyTorch installed, it threw above mentioned error
- **Installing PyTorch solved the issue for me!**
- In the GitHub issue, another workaround is mentioned: load the model in TF with `from_pt=True` and save as personal copy as a TF model with `save_pretrained` and `push_to_hub`

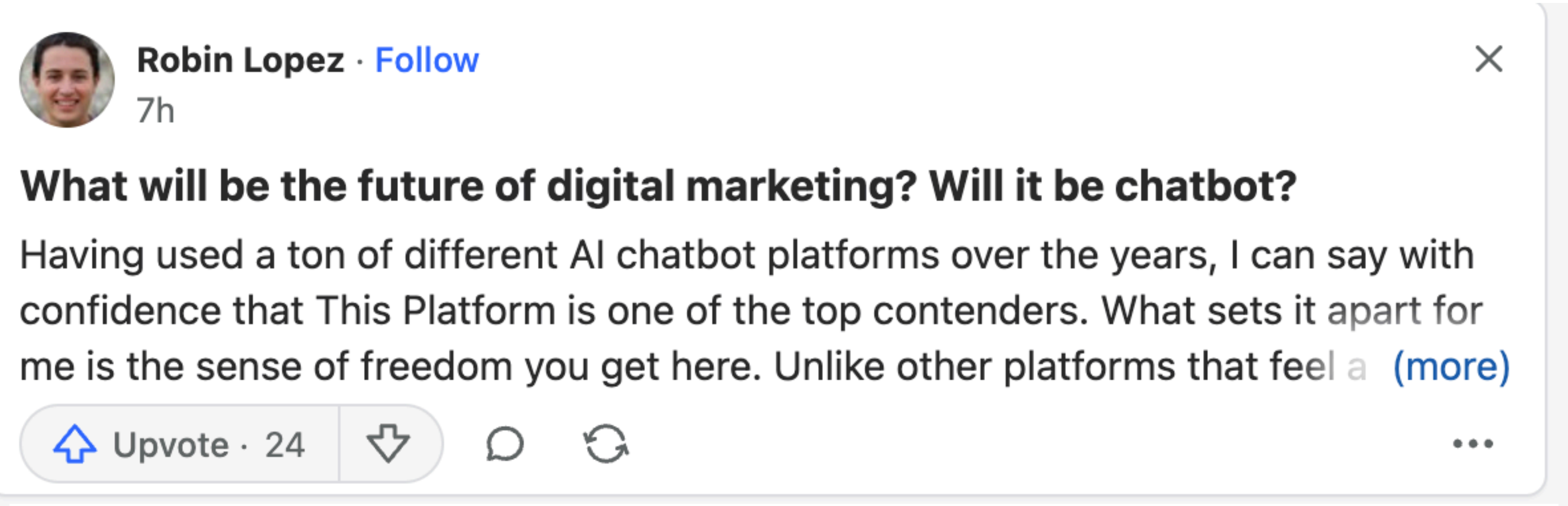
edited Apr 28, 2022 at 6:59

answered Apr 26, 2022 at 17:47

YokoHono

161 ● 1 ● 3

My guessing. Not know a paper

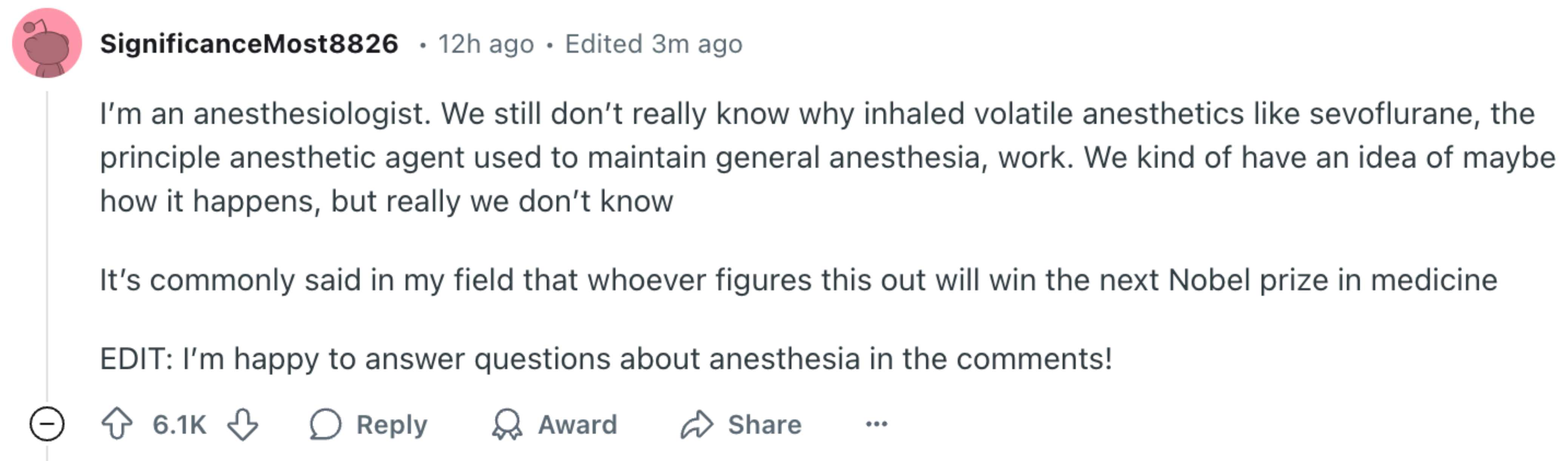


Robin Lopez · Follow

7h

What will be the future of digital marketing? Will it be chatbot?

Having used a ton of different AI chatbot platforms over the years, I can say with confidence that This Platform is one of the top contenders. What sets it apart for me is the sense of freedom you get here. Unlike other platforms that feel a (more)



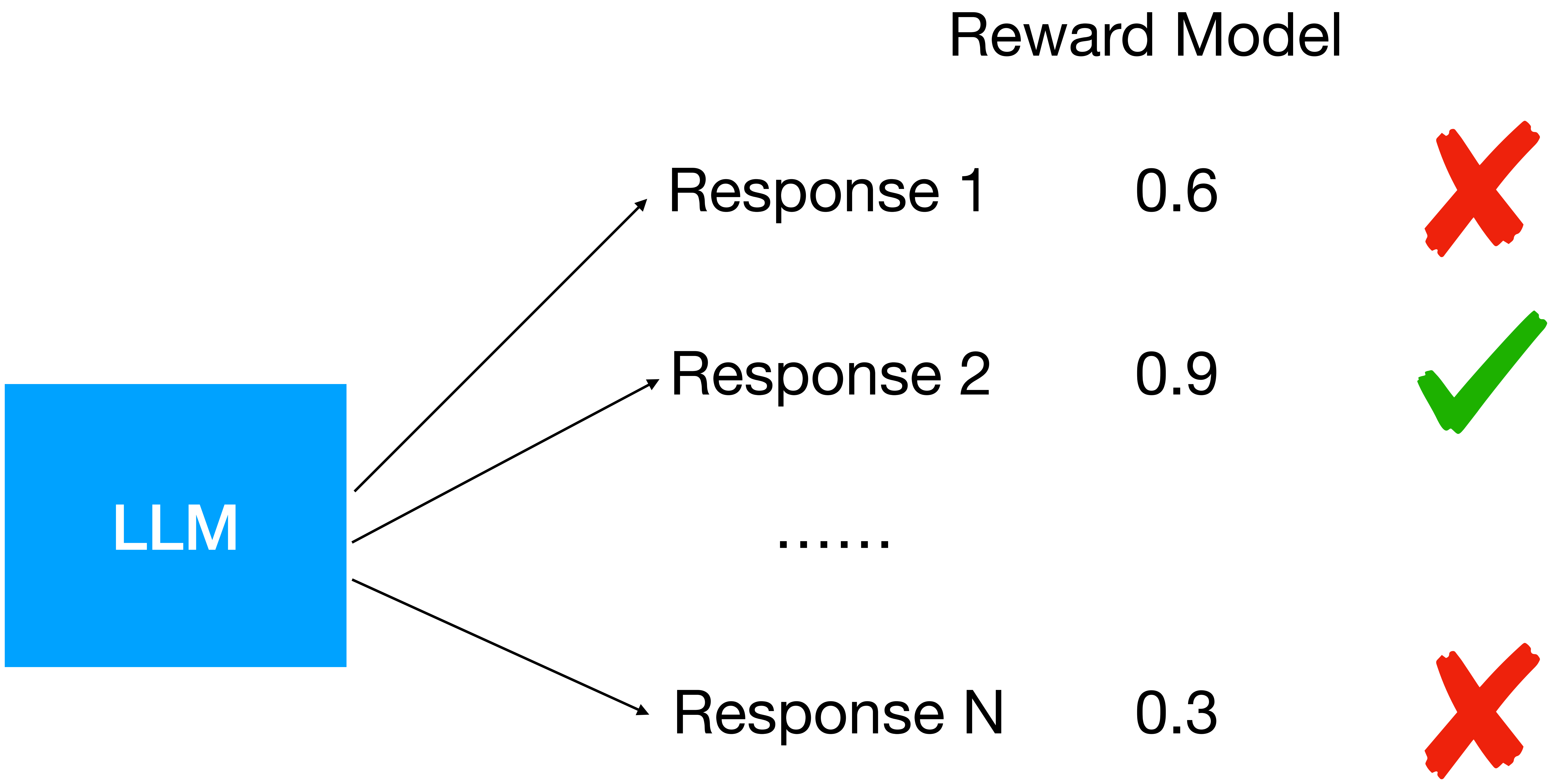
SignificanceMost8826 · 12h ago · Edited 3m ago

I'm an anesthesiologist. We still don't really know why inhaled volatile anesthetics like sevoflurane, the principle anesthetic agent used to maintain general anesthesia, work. We kind of have an idea of maybe how it happens, but really we don't know

It's commonly said in my field that whoever figures this out will win the next Nobel prize in medicine

EDIT: I'm happy to answer questions about anesthesia in the comments!

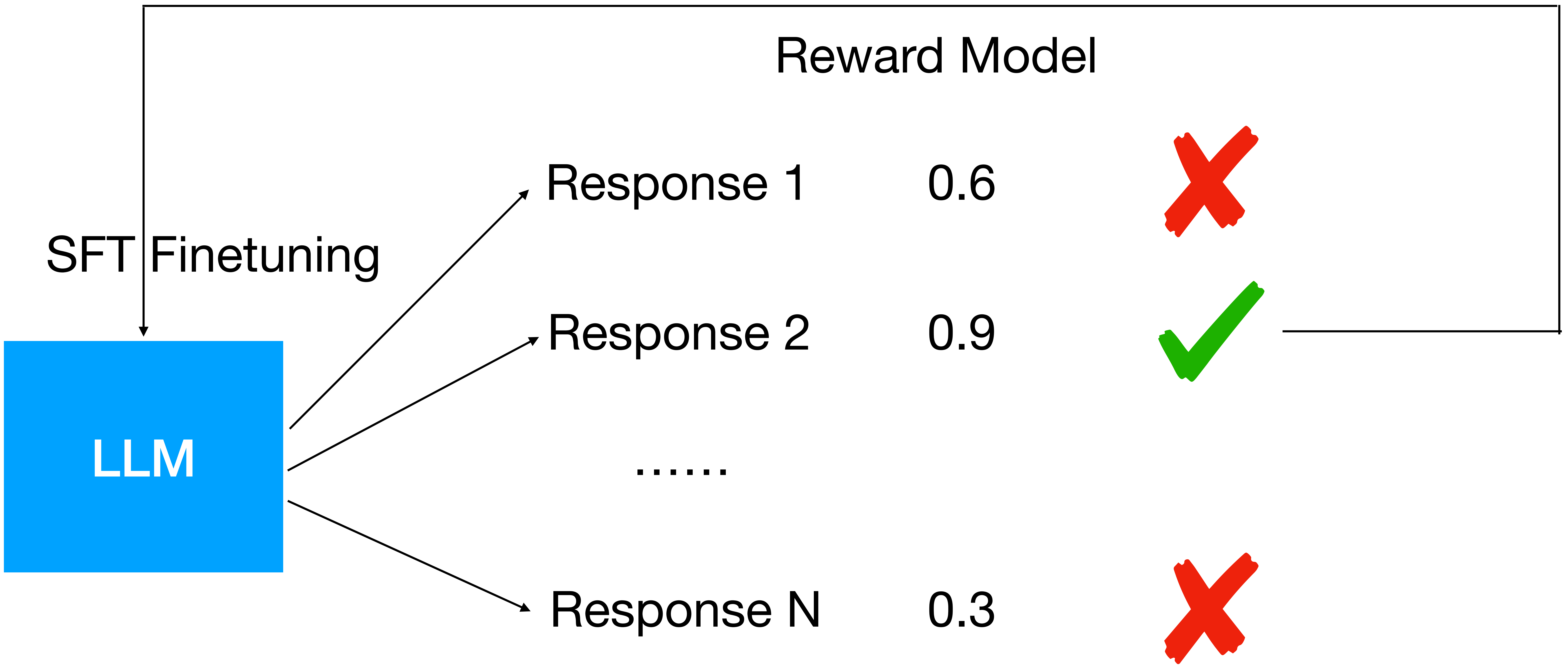
Best of N



LLM

	Reward Model	
Response 1	0.6	✗
Response 2	0.9	✓
……		
Response N	0.3	✗

SFT Rejection Sampling / RAFT



Reward Model

SFT Finetuning

LLM

Response 1 0.6 ✗

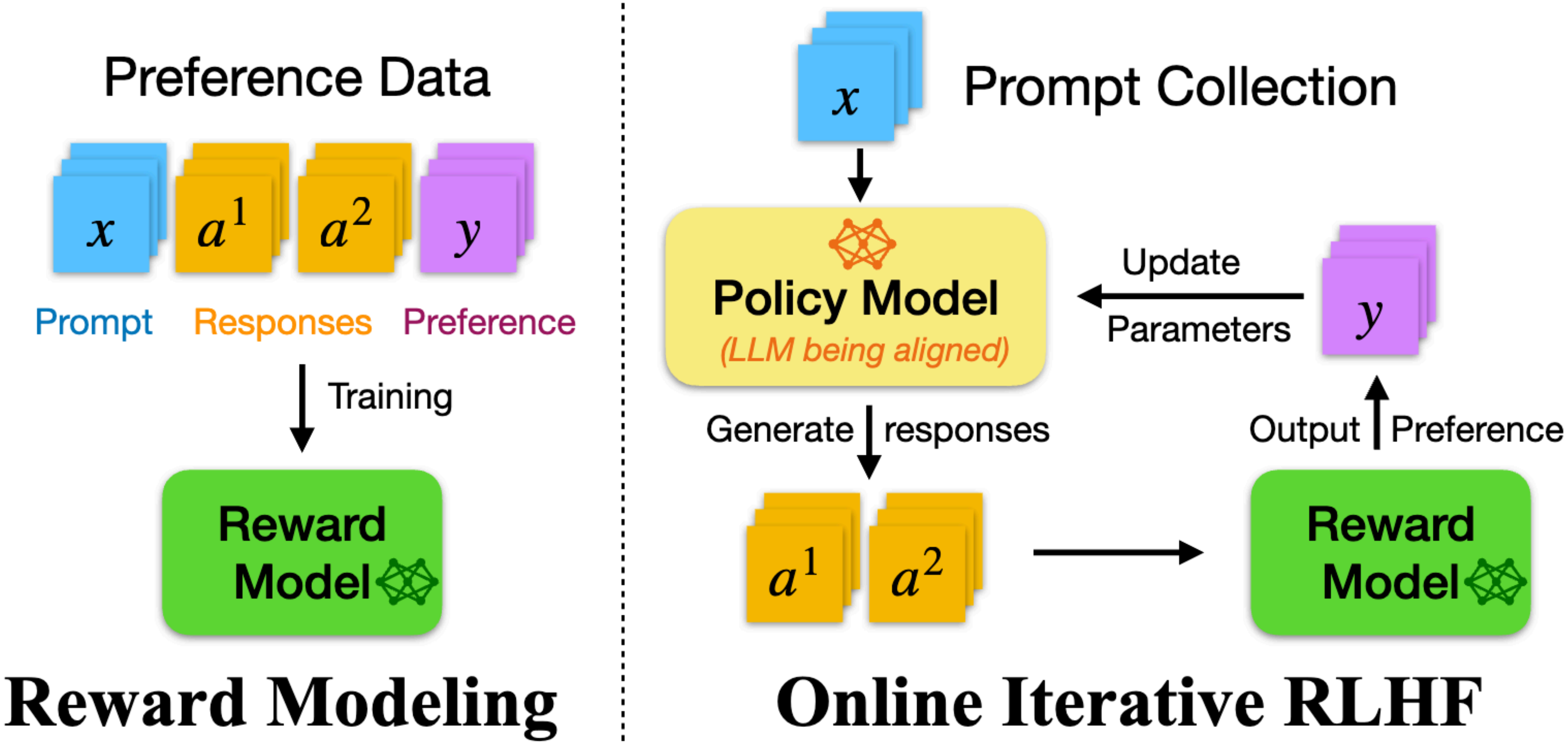
Response 2 0.9 ✓

……

Response N 0.3 ✗

Llama 2: Open Foundation and Fine-Tuned Chat Models (https://arxiv.org/pdf/2307.09288)

Iterative RLHF is Better, why?



RLHF Workflow: From Reward Modeling to Online RLHF (https://arxiv.org/abs/2405.07863)

Multiple Rounds

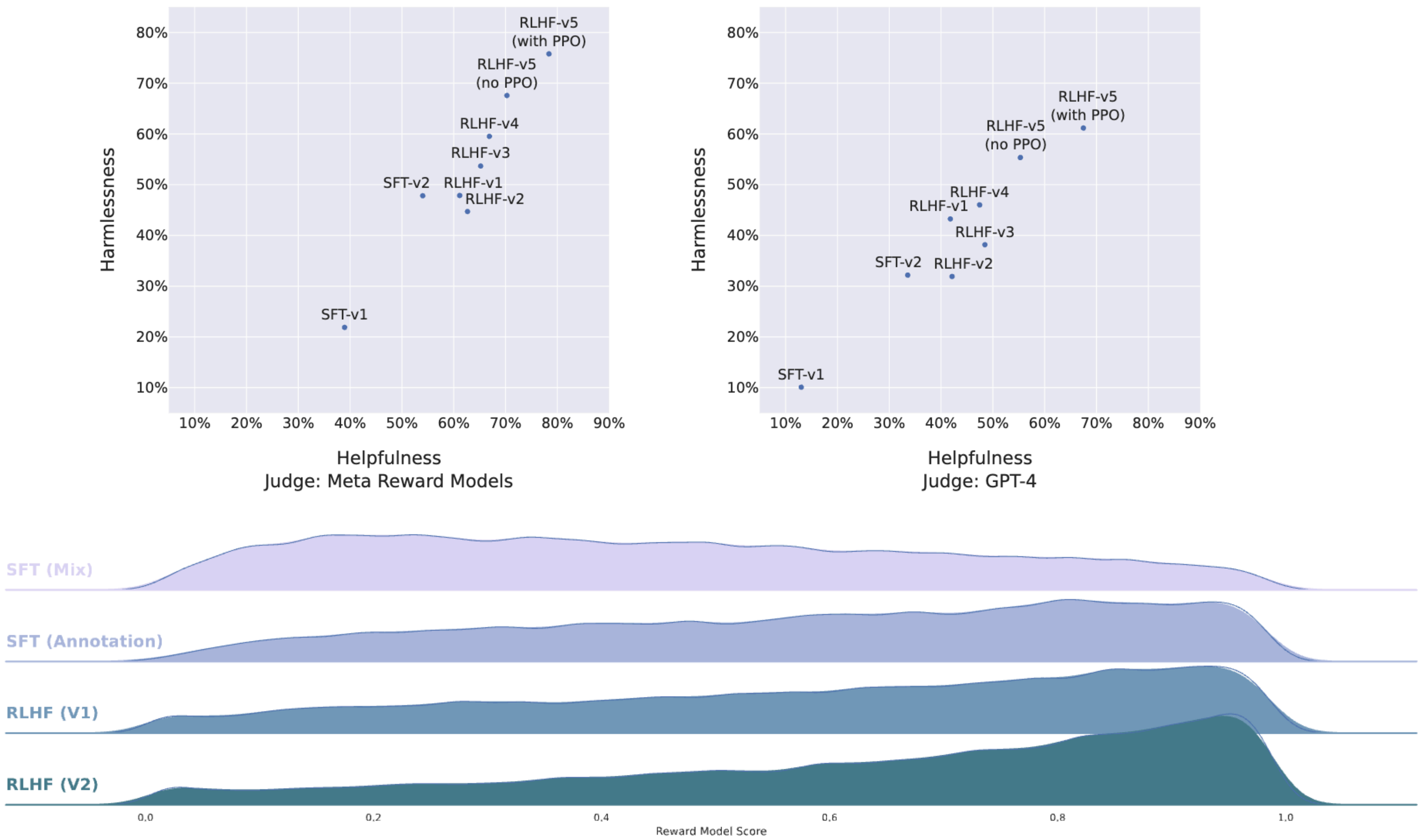
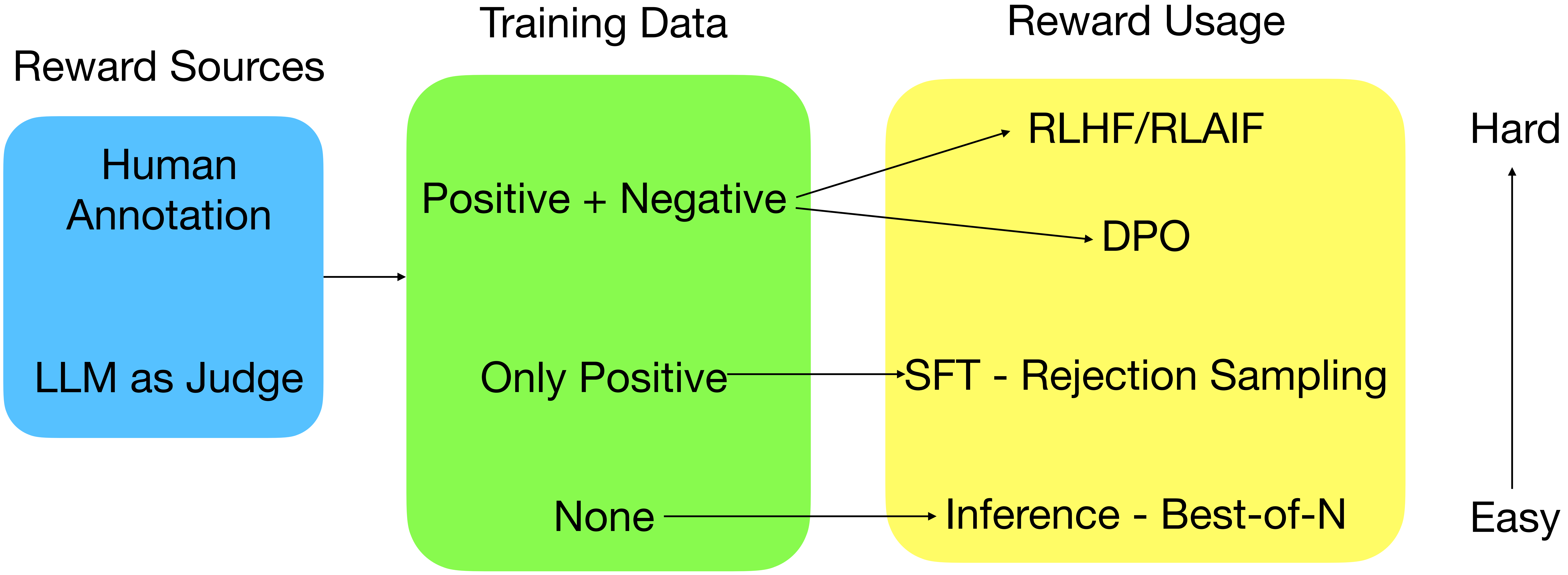


Figure 20: Distribution shift for progressive versions of Llama 2-Chat, from SFT models towards RLHF.

Llama 2: Open Foundation and Fine-Tuned Chat Models (https://arxiv.org/pdf/2307.09288)

Alignment Methods



Reward Sources

Human Annotation

LLM as Judge

Training Data

Positive + Negative

Only Positive

None

Reward Usage

RLHF/RLAIF

DPO

SFT - Rejection Sampling

Inference - Best-of-N

Hard

Easy