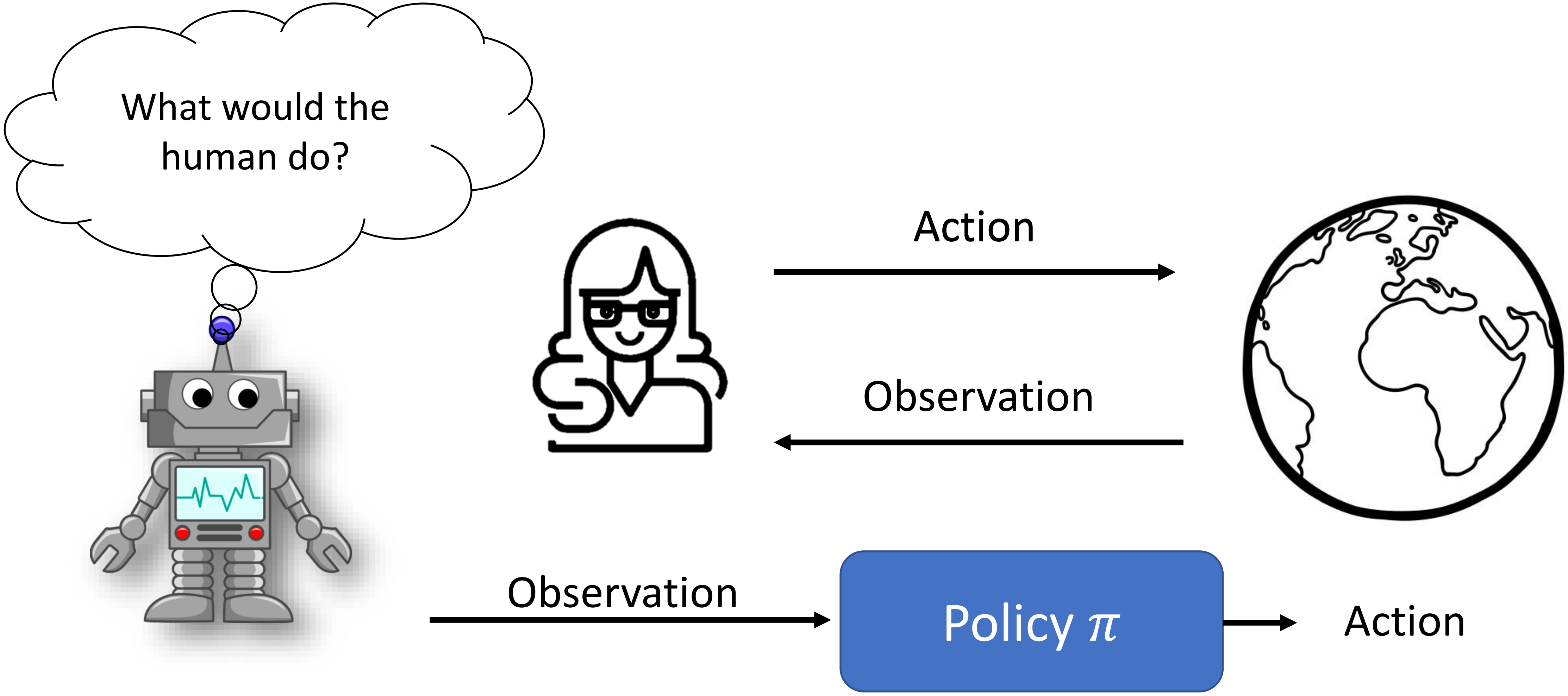
Large Language Models and RL from Human Feedback

Instructor: Daniel Brown

[Some slides adapted from Ana Marasovic, SpinningUp in Deep RL, and others]

Why not just imitate behavior? (Behavioral Cloning)



What would the human do?

Action

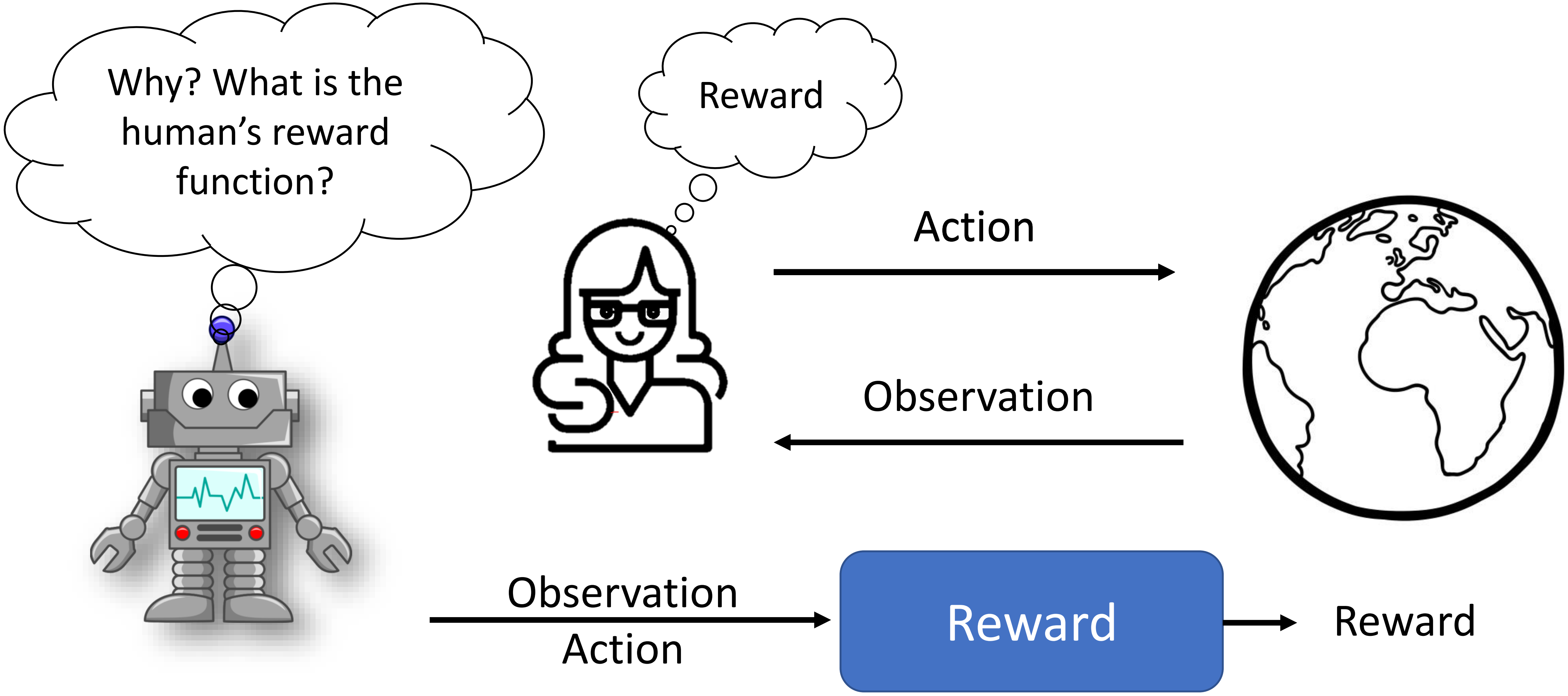
Observation

Observation

Policy π

Action

Reward Learning
(Inverse Reinforcement Learning)



Why? What is the human's reward function?

Reward

Action

Observation

Observation

Action

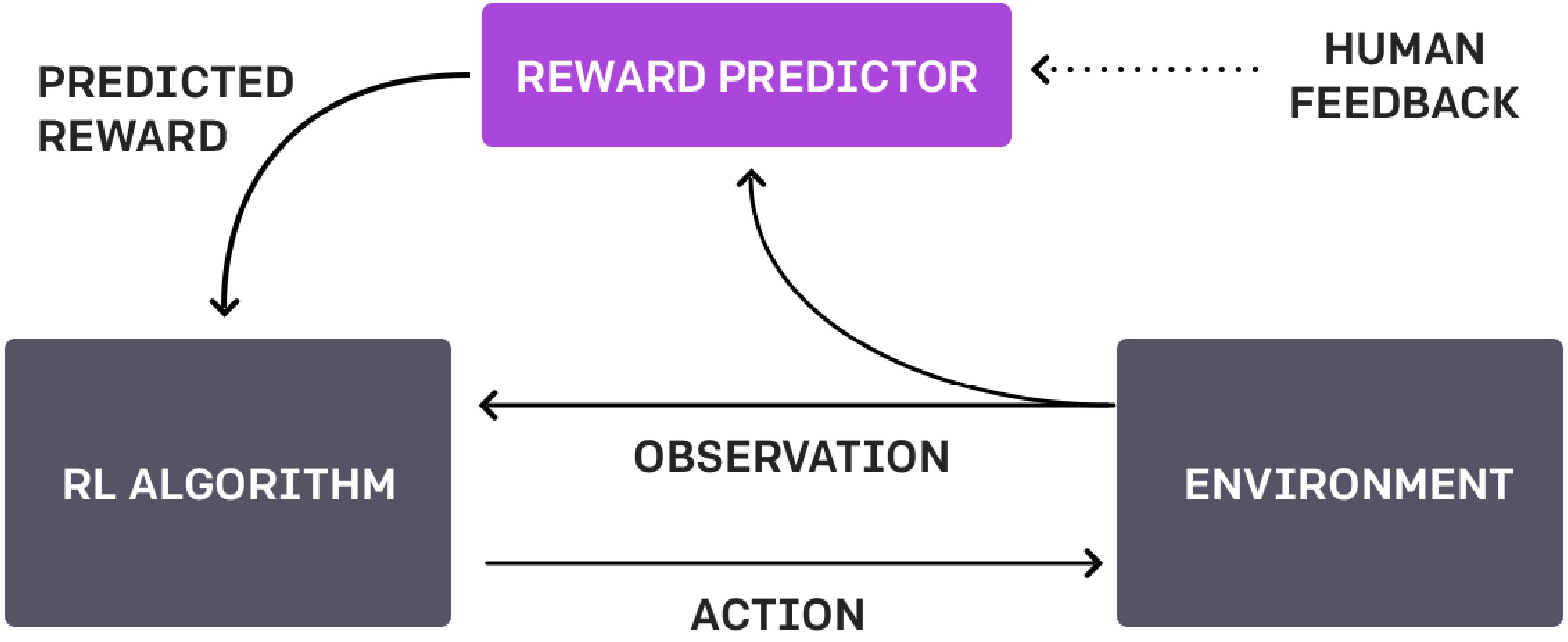
Reward

Reward

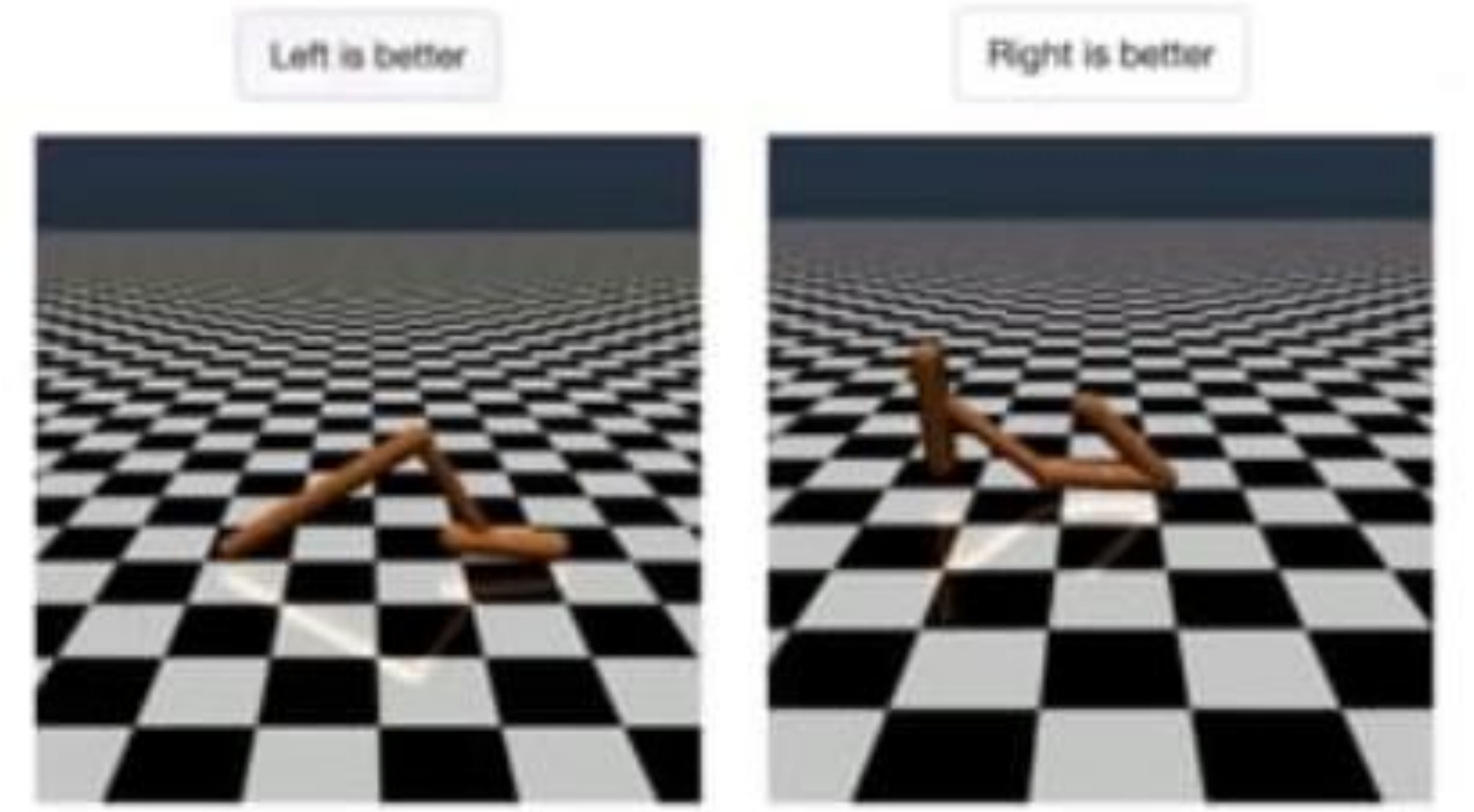
What if I can’t demonstrate something?

$\succ$

RL from Human Feedback (RLHF)



RL from Human Preferences



https://arxiv.org/abs/1706.03741

Why would you want to learn a reward from ranked examples?

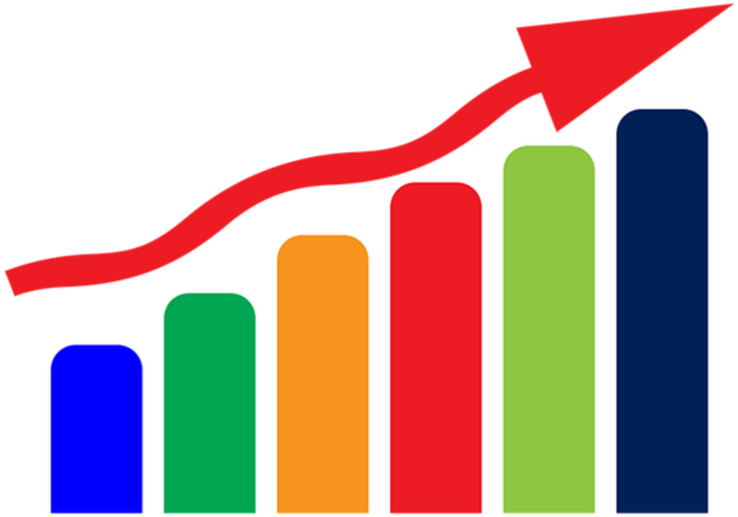
Inverse Reinforcement Learning

Prior approaches …

1. Typically couldn’t do much better than the demonstrator.

2. Were hard to scale to complex problems.

Pre-Ranked Demonstrations



Brown et al. "Extrapolating Beyond Suboptimal Demonstrations via IRL from Observations." ICML 2019

Inverse Reinforcement Learning

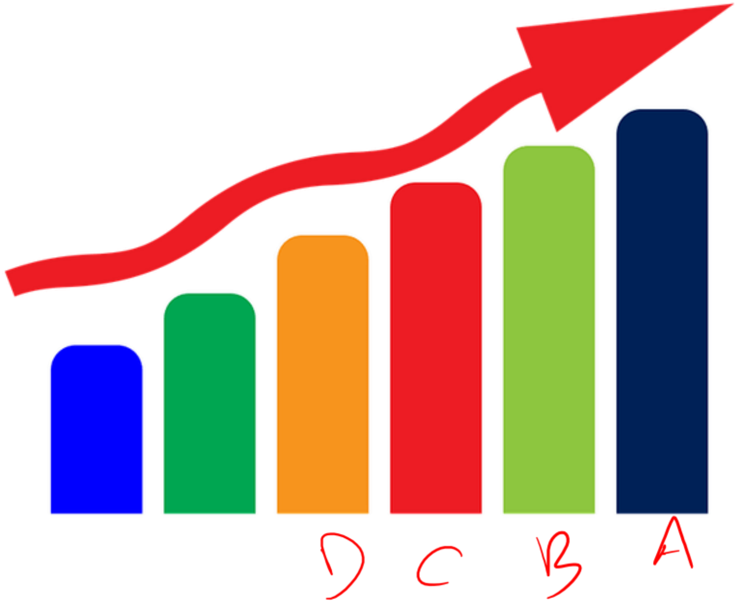
Prior approaches …

~~1. Typically couldn't do much better than the demonstrator.~~

Find a reward function that explains the ranking, allowing for extrapolation.

2. Were hard to scale to complex problems.

Pre-Ranked Demonstrations



Brown et al. "Extrapolating Beyond Suboptimal Demonstrations via IRL from Observations." ICML 2019

Inverse Reinforcement Learning

Prior approaches …

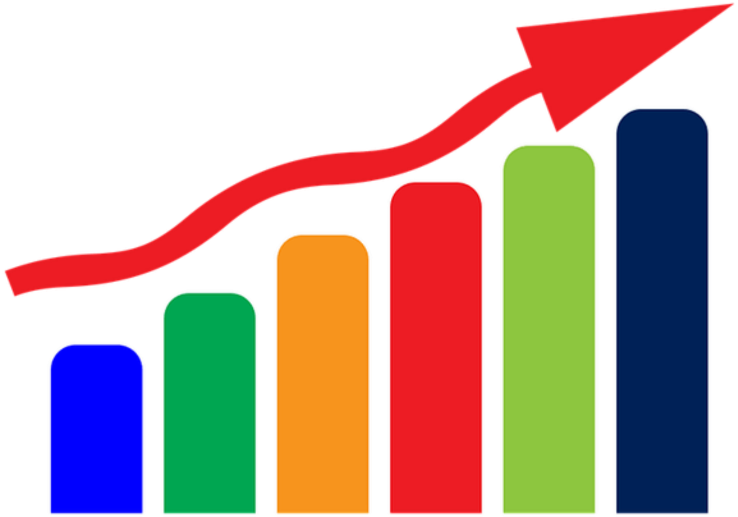
~~1. Typically couldn't do much better than the demonstrator.~~

Find a reward function that explains the ranking, allowing for extrapolation.

~~2. Were hard to scale to complex problems.~~

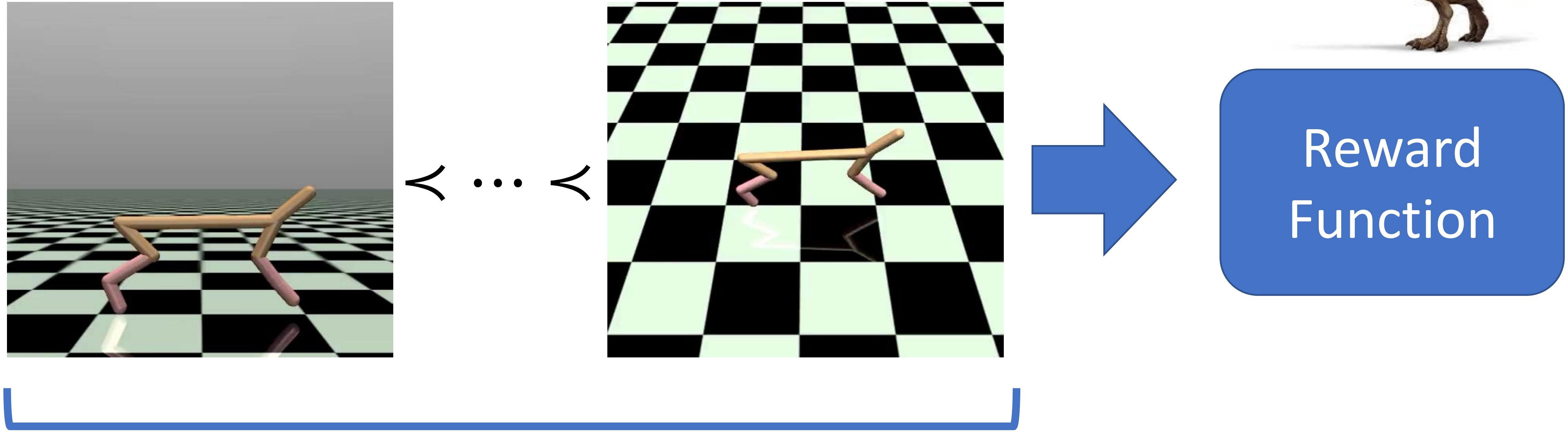
Reward learning becomes a supervised learning problem.

Pre-Ranked Demonstrations



Brown et al. "Extrapolating Beyond Suboptimal Demonstrations via IRL from Observations." ICML 2019

Trajectory-ranked Reward Extrapolation (T-REX)



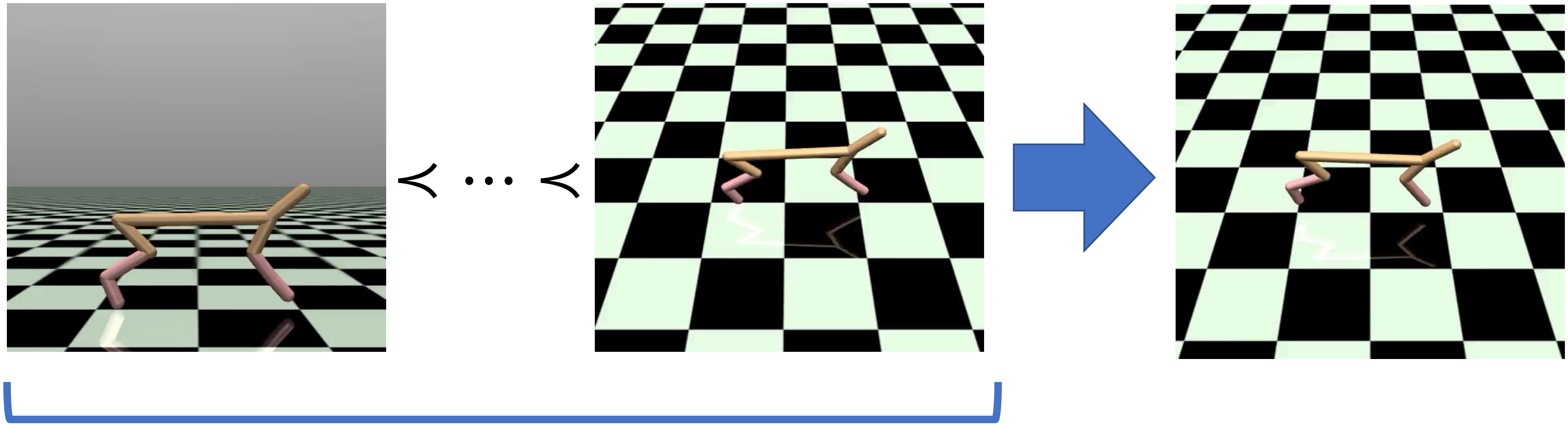
$\prec \cdots \prec$

Reward Function

Pre-ranked demonstrations

Brown et al. "Extrapolating Beyond Suboptimal Demonstrations via IRL from Observations." ICML 2019

Trajectory-ranked Reward Extrapolation (T-REX)



Pre-ranked demonstrations

T-REX Policy

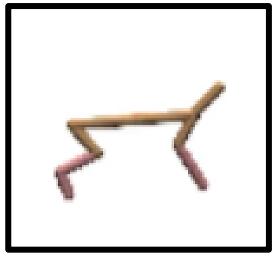
Brown et al. "Extrapolating Beyond Suboptimal Demonstrations via IRL from Observations." ICML 2019

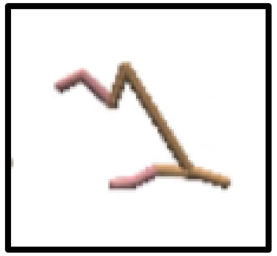
Reward Function

$$R_\theta : S \rightarrow \mathbb{R}$$

Examples of S:

Current Robot Joint Angles and Velocities

 $\rightarrow 0.5$

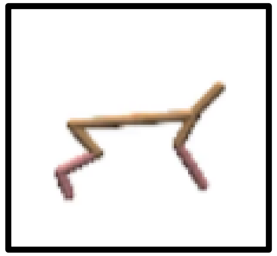
 $\rightarrow -0.7$

Reward Function

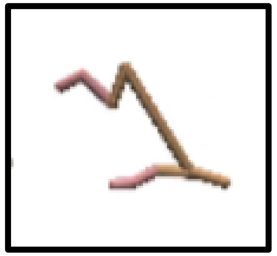
$$R_\theta : S \to \mathbb{R}$$

Examples of S:

Current Robot Joint Angles and Velocities

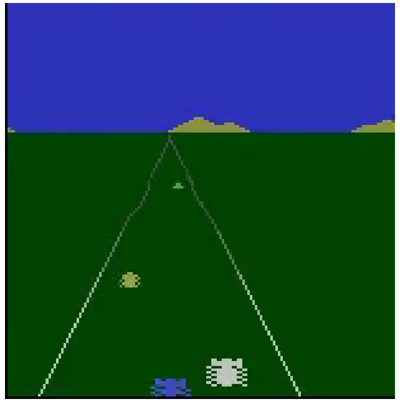


$\to 0.5$

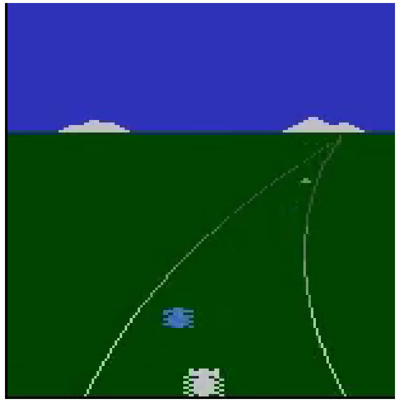


$\to -0.7$

Short Sequence of Images



$\to 0.9$



$\to -1.2$

Trajectory-ranked Reward Extrapolation (T-REX)

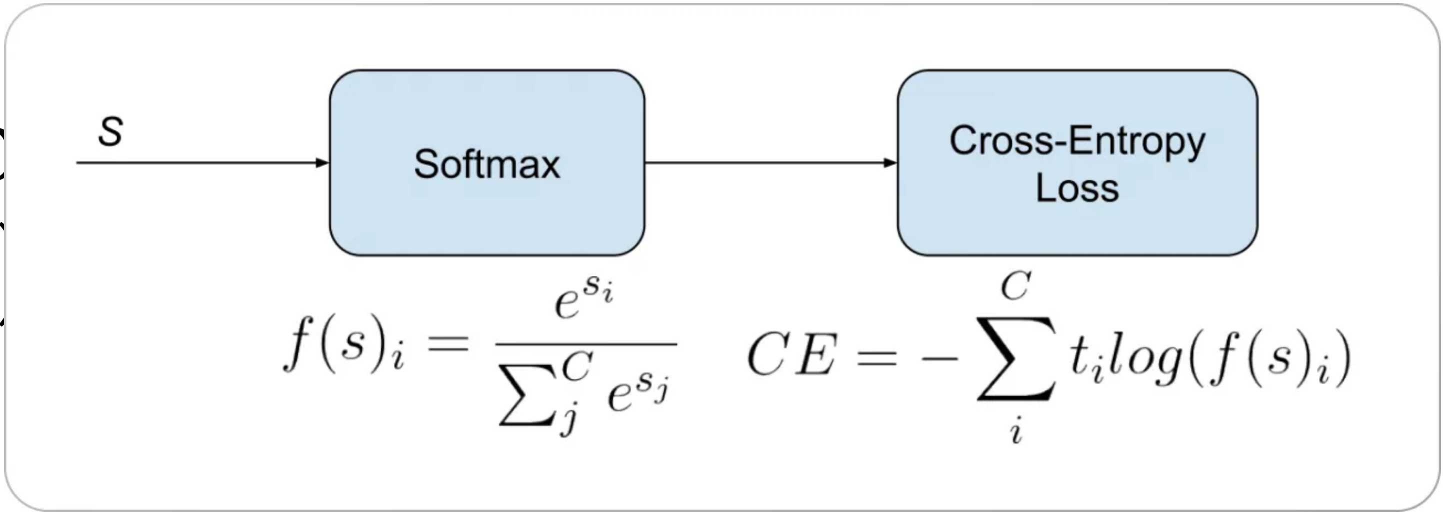
$$\boxed{\tau_1} \prec \boxed{\tau_2} \prec \cdots \prec \tau_T$$

$$\boxed{\sum_{s\in\tau_1} R_\theta(s)} < \boxed{\sum_{s\in\tau_2} R_\theta(s)}$$

Bradley-Terry pairwise ranking loss

$$\mathcal{L}(\theta) = -\sum_{\tau_i\prec\tau_j} \frac{\exp \sum_{s\in\tau_j} R_\theta(s)}{\exp \sum_{s\in\tau_i} R_\theta(s) + \exp \sum_{s\in\tau_j} R_\theta(s)}$$

Trajecto tion (T-REX)



$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}} \qquad CE = -\sum_i^C t_i log(f(s)_i)$$

$$\boxed{\sum_{s \in \tau_1} R_\theta(s)} < \boxed{\sum_{s \in \tau_2} R_\theta(s)}$$

Logits

Minimize cross-entropy loss

$$\mathcal{L}(\theta) = -\sum_{\tau_i \prec \tau_j} \frac{\exp \sum_{s \in \tau_j} R_\theta(s)}{\exp \sum_{s \in \tau_i} R_\theta(s) + \exp \sum_{s \in \tau_j} R_\theta(s)}$$

Trajectory-ranked Reward Extrapolation (T-REX)

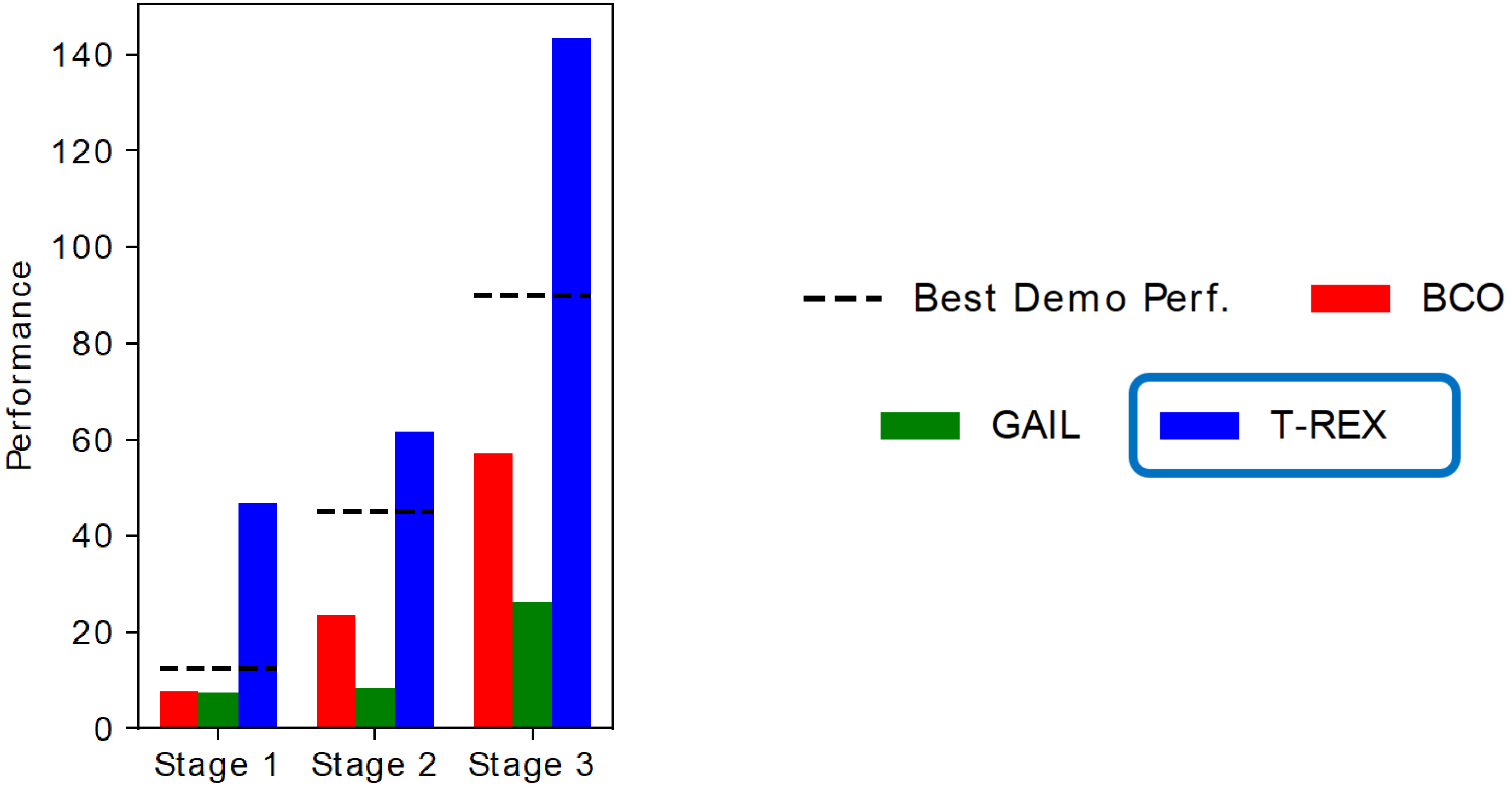
$$\boxed{\tau_1} \prec \boxed{\tau_2} \prec \cdots \prec \tau_T$$

Given pre-ranked demos, reward learning can be formulated as a standard supervised learning task.

Minimize cross-entropy loss

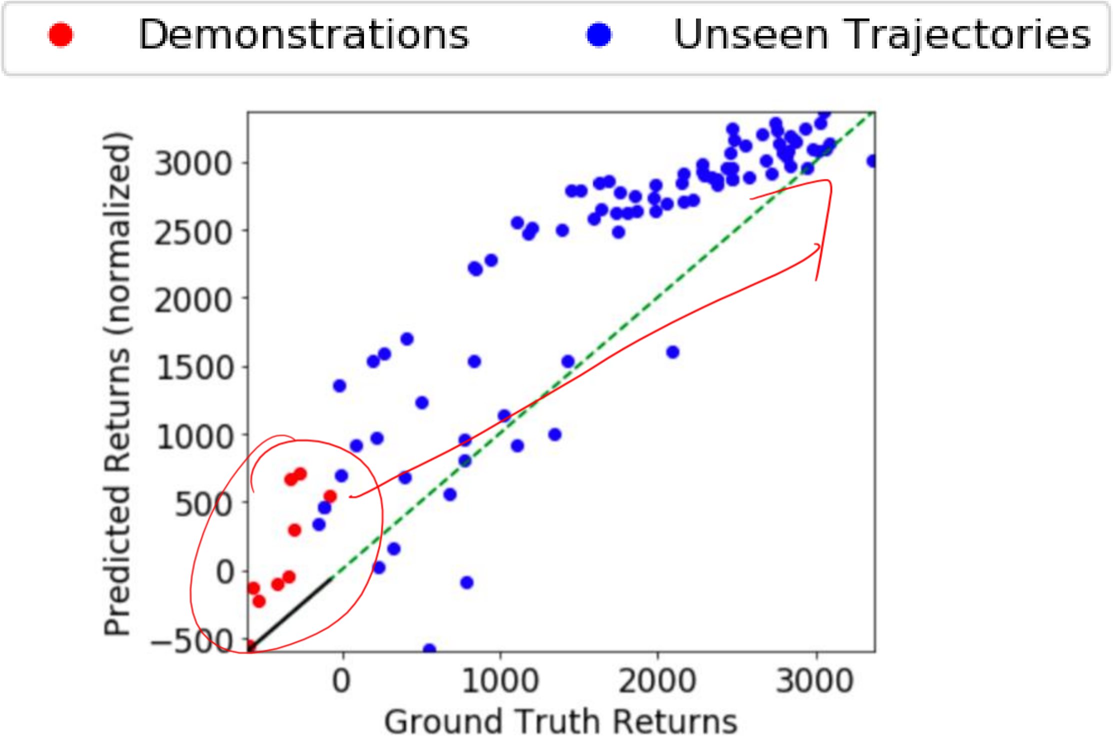
$$\mathcal{L}(\theta) = -\sum_{\tau_i \prec \tau_j} \frac{\exp \sum_{s\in\tau_j} R_\theta(s)}{\exp \sum_{s\in\tau_i} R_\theta(s) + \exp \sum_{s\in\tau_j} R_\theta(s)}$$

T-REX Policy Performance



Brown et al. "Extrapolating Beyond Suboptimal Demonstrations via IRL from Observations." ICML 2019

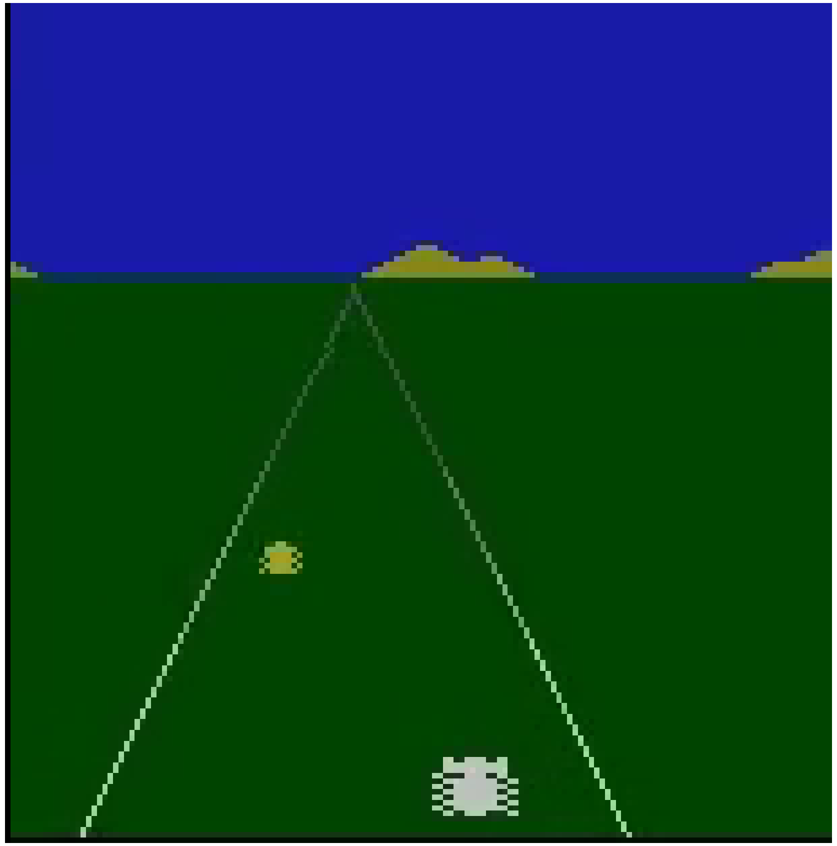
Reward Extrapolation



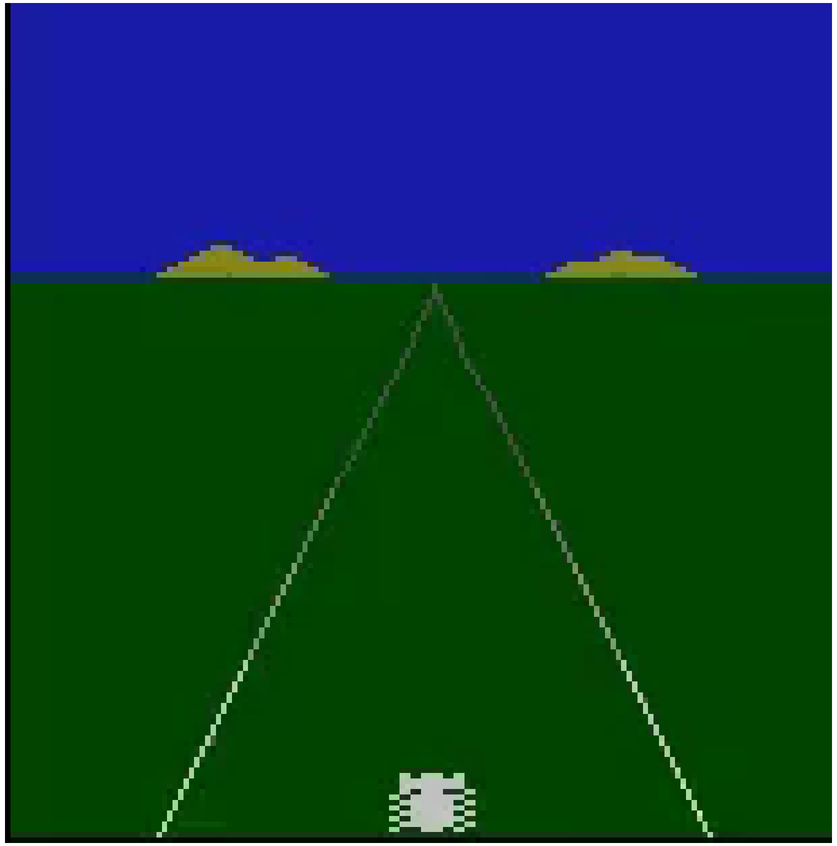
T-REX can extrapolate beyond the performance of the best demo

Brown et al. "Extrapolating Beyond Suboptimal Demonstrations via IRL from Observations." ICML 2019

“Autonomous Driving” in Atari



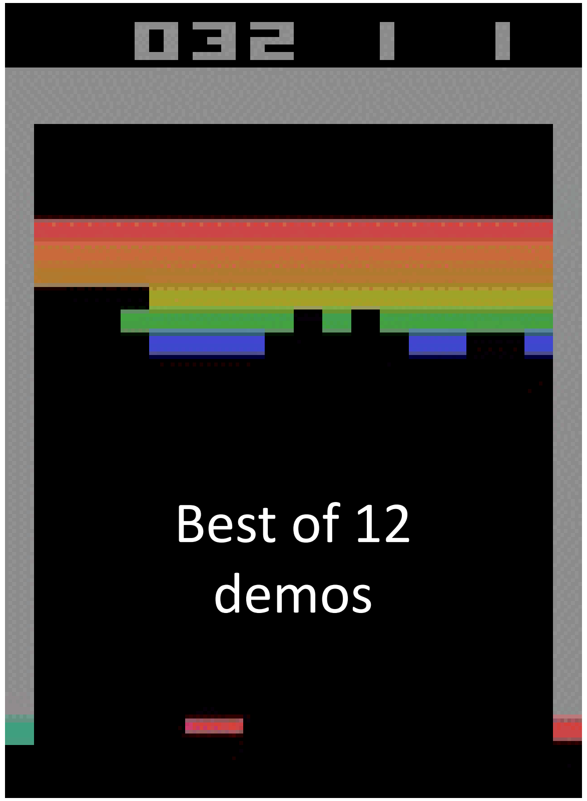
Best demo (Score = 84)



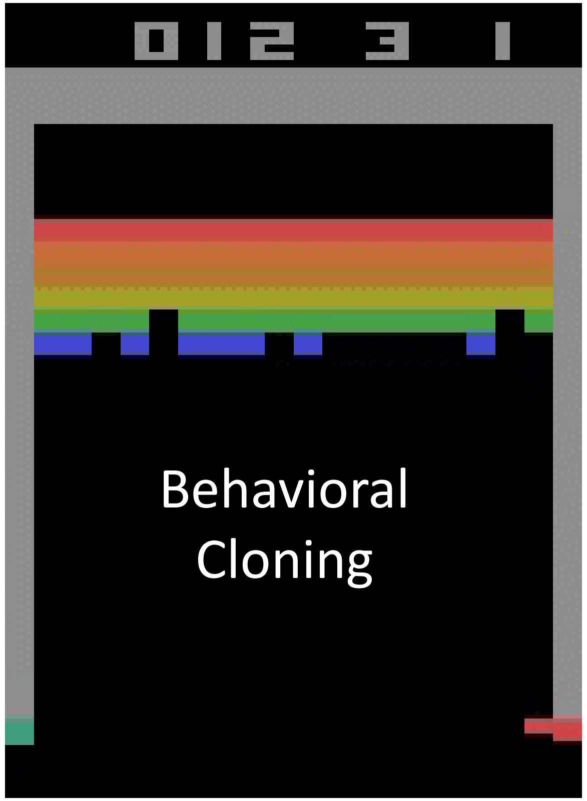
T-REX (Score = 520)

Uses only 12 ranked demonstrations

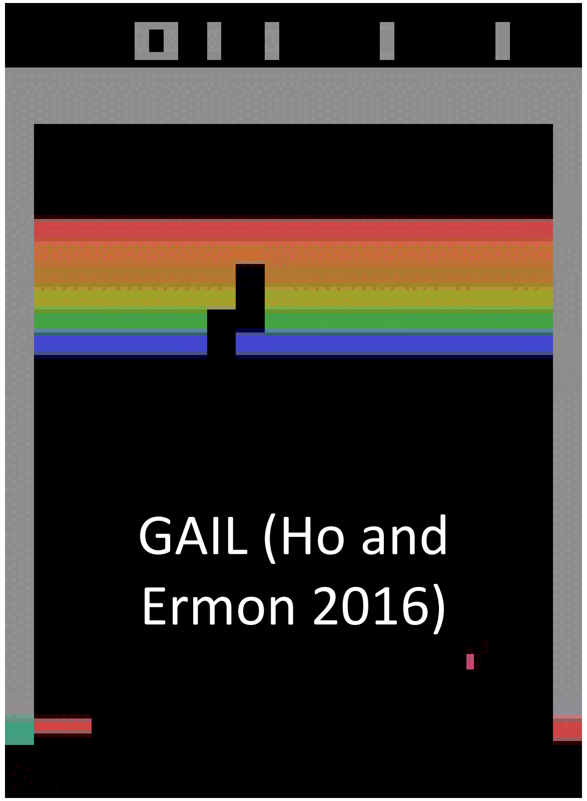
Atari Breakout



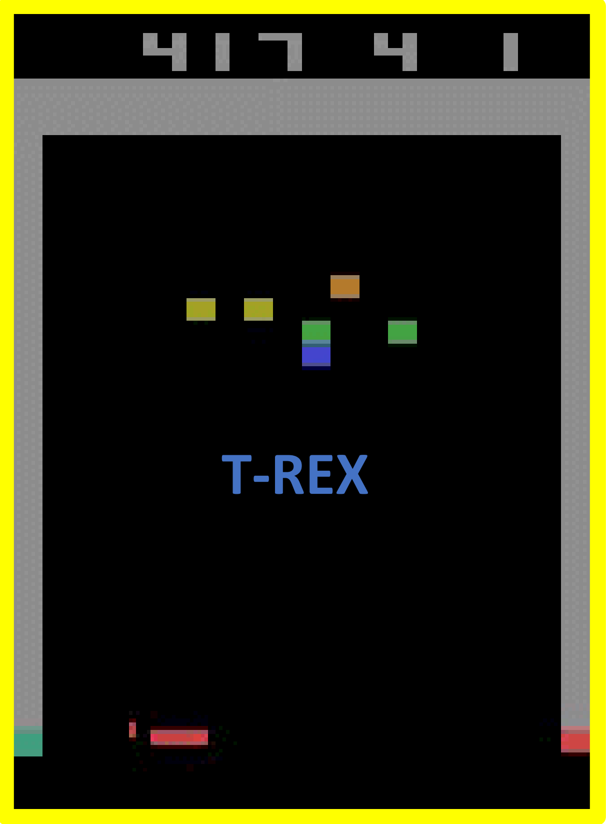
Best of 12 demos



Behavioral Cloning



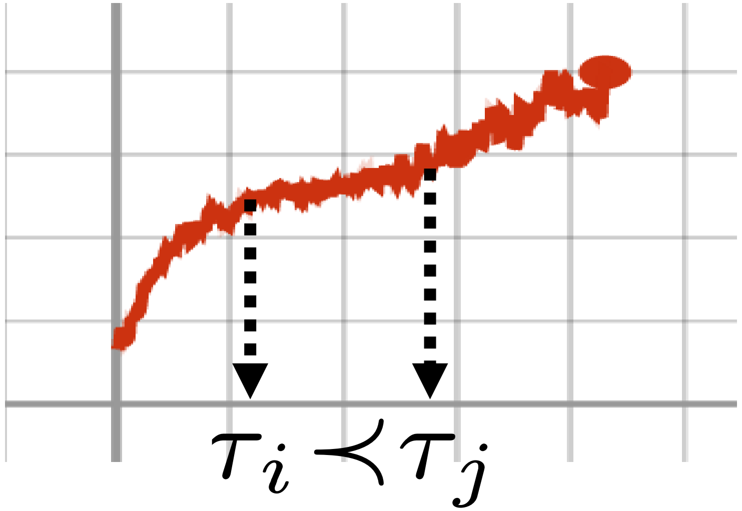
GAIL (Ho and Ermon 2016)



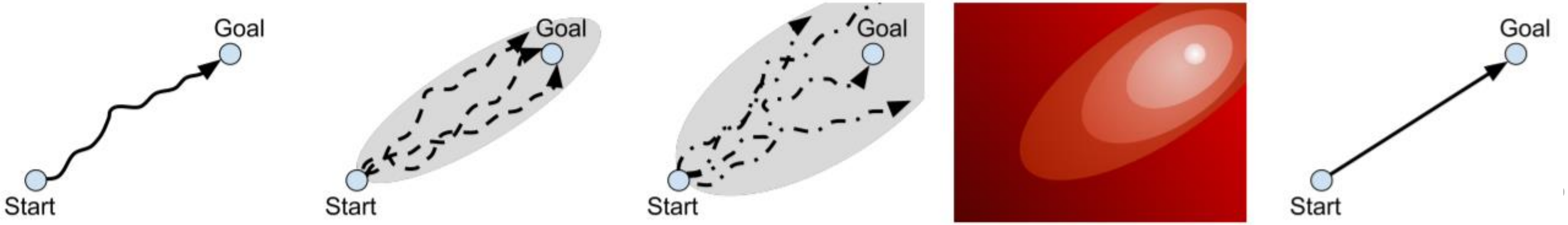
T-REX

What if you don't have explicit preference labels?

Learning from a learner [ICML'19]

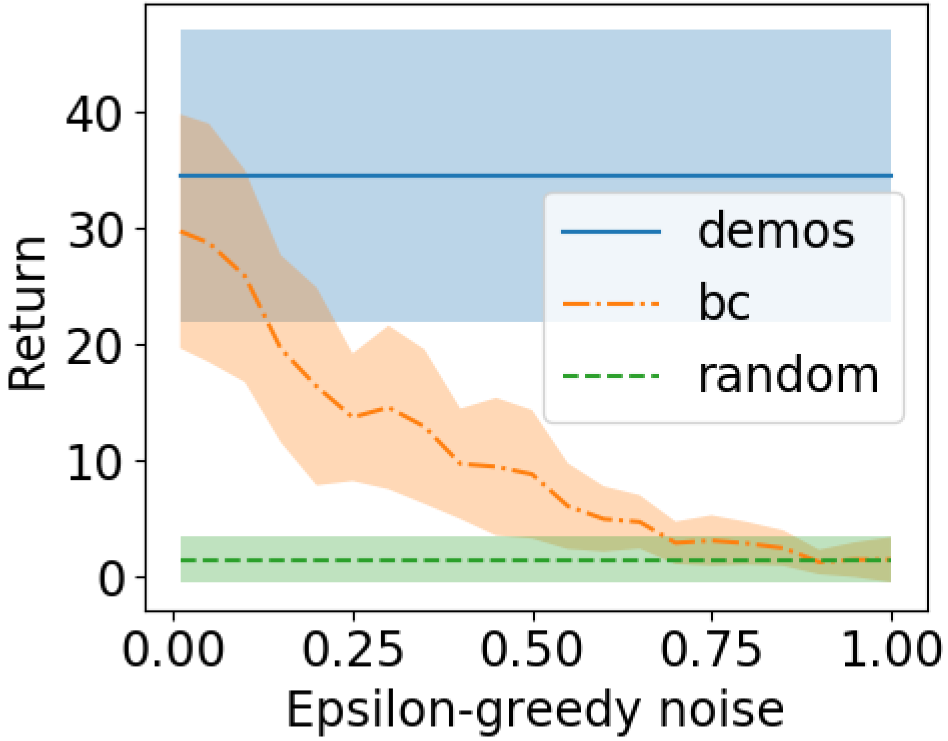


Automatic preference label generation [CoRL'20]



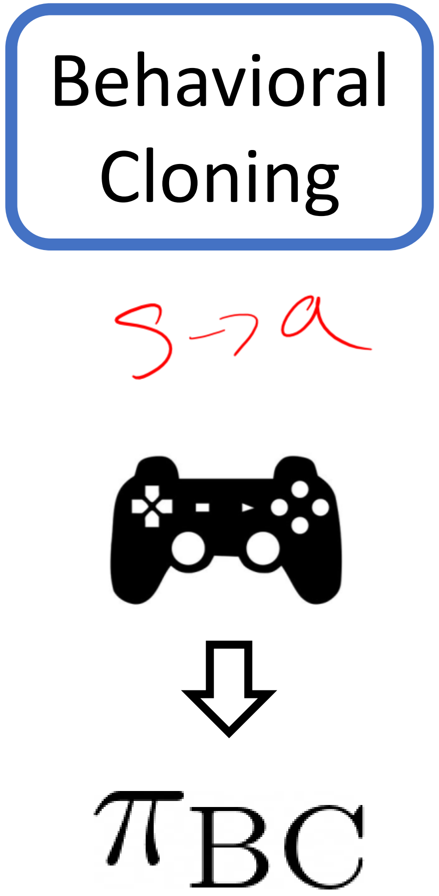
Automatic Rankings via Noise Injection

- Assumption: Demonstrator is significantly better than a purely random policy.
- Provides automatic rankings as noise increases.
- Generates a large diverse set of ranked demonstrations



Brown et al. “Better-than-Demonstrator Imitation Learning via Automatically-Ranked Demonstrations." CoRL 2019

Disturbance-based Reward Extrapolation (D-REX)



Behavioral Cloning

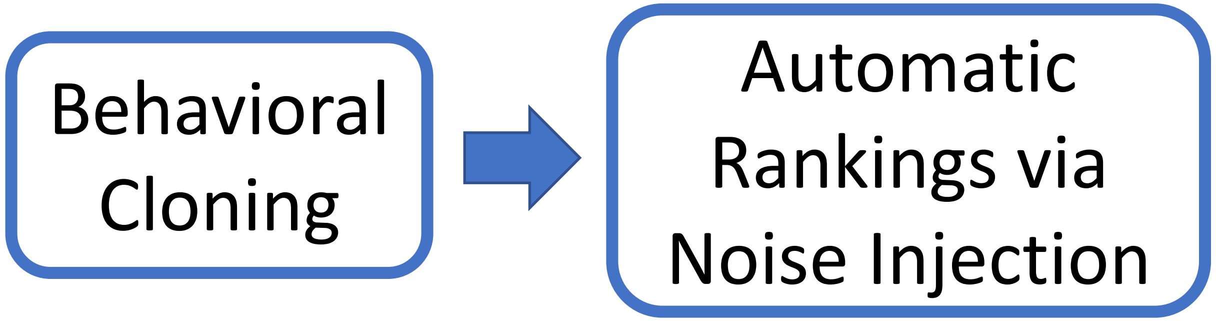
$s \to a$

⇩

π_{BC}

Brown et al. “Better-than-Demonstrator Imitation Learning via Automatically-Ranked Demonstrations." CoRL 2019

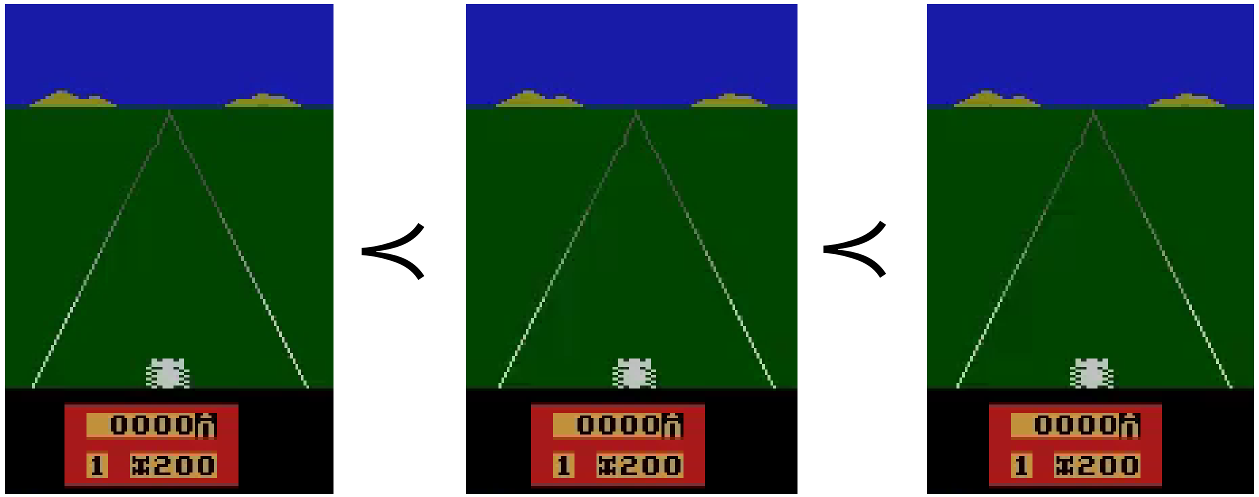
Disturbance-based Reward Extrapolation (D-REX)


Behavioral Cloning

Automatic Rankings via Noise Injection

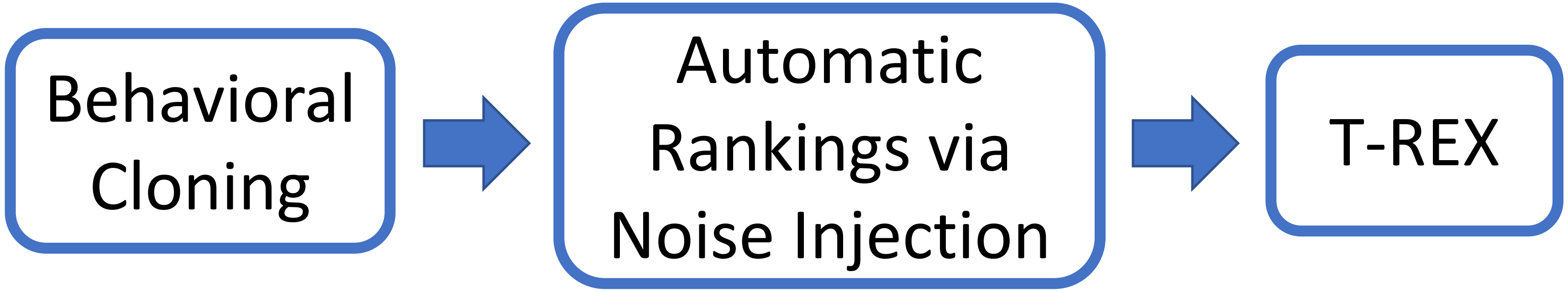
⇩

π_{BC}

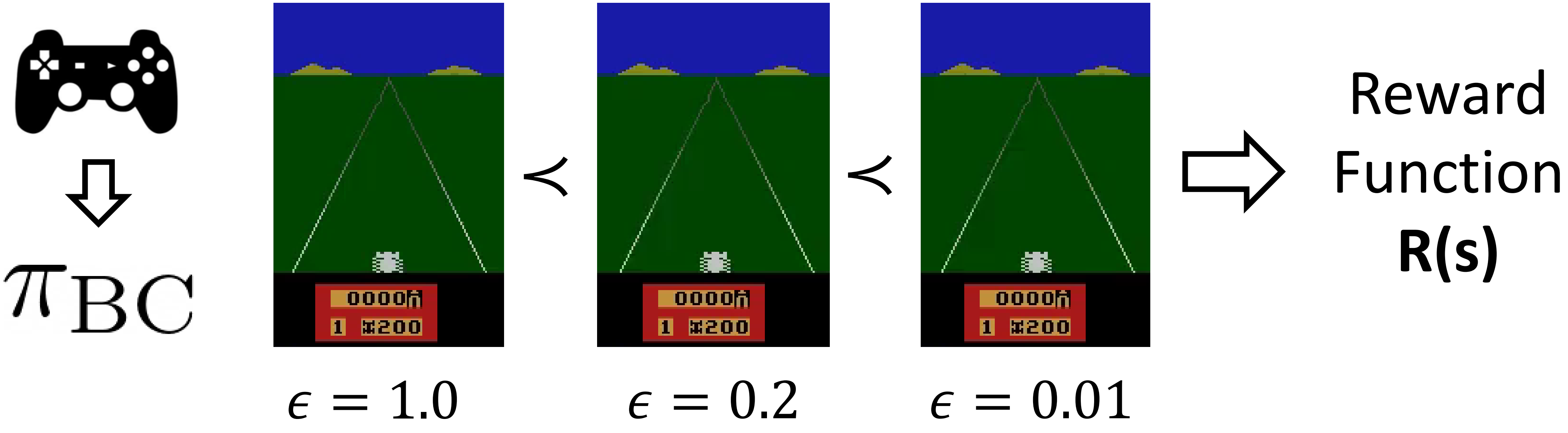


$\epsilon = 1.0$ $\epsilon = 0.2$ $\epsilon = 0.01$

Disturbance-based Reward Extrapolation (D-REX)



Behavioral Cloning → Automatic Rankings via Noise Injection → T-REX

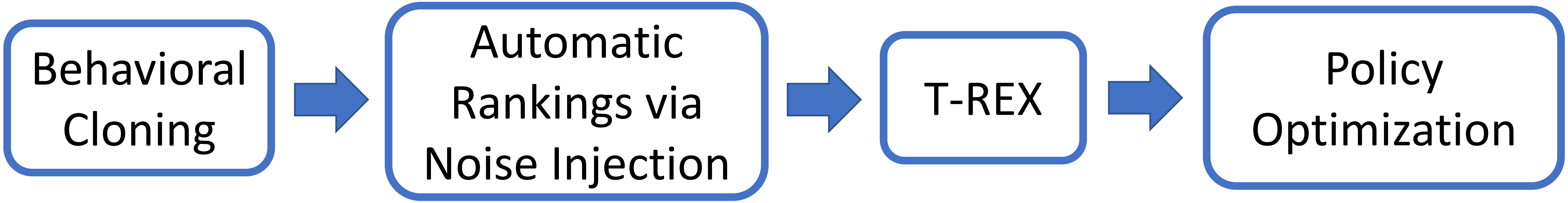


π_{BC}

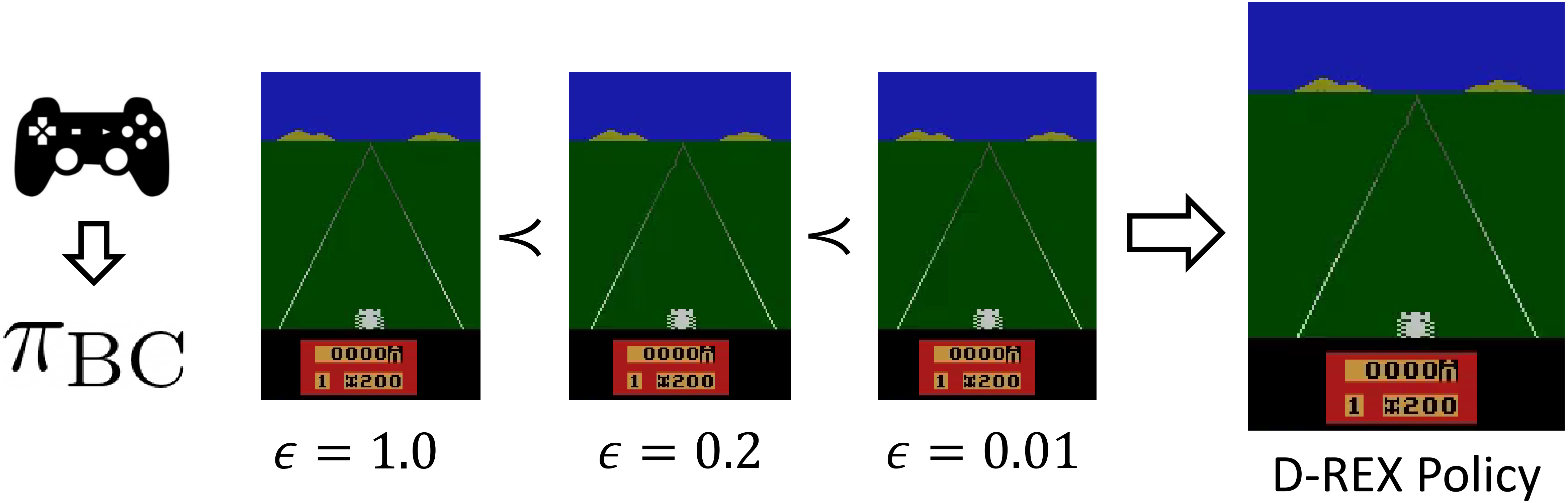
$\epsilon = 1.0$ $\prec$ $\epsilon = 0.2$ $\prec$ $\epsilon = 0.01$

Reward Function **R(s)**

Disturbance-based Reward Extrapolation (D-REX)



Behavioral Cloning → Automatic Rankings via Noise Injection → T-REX → Policy Optimization

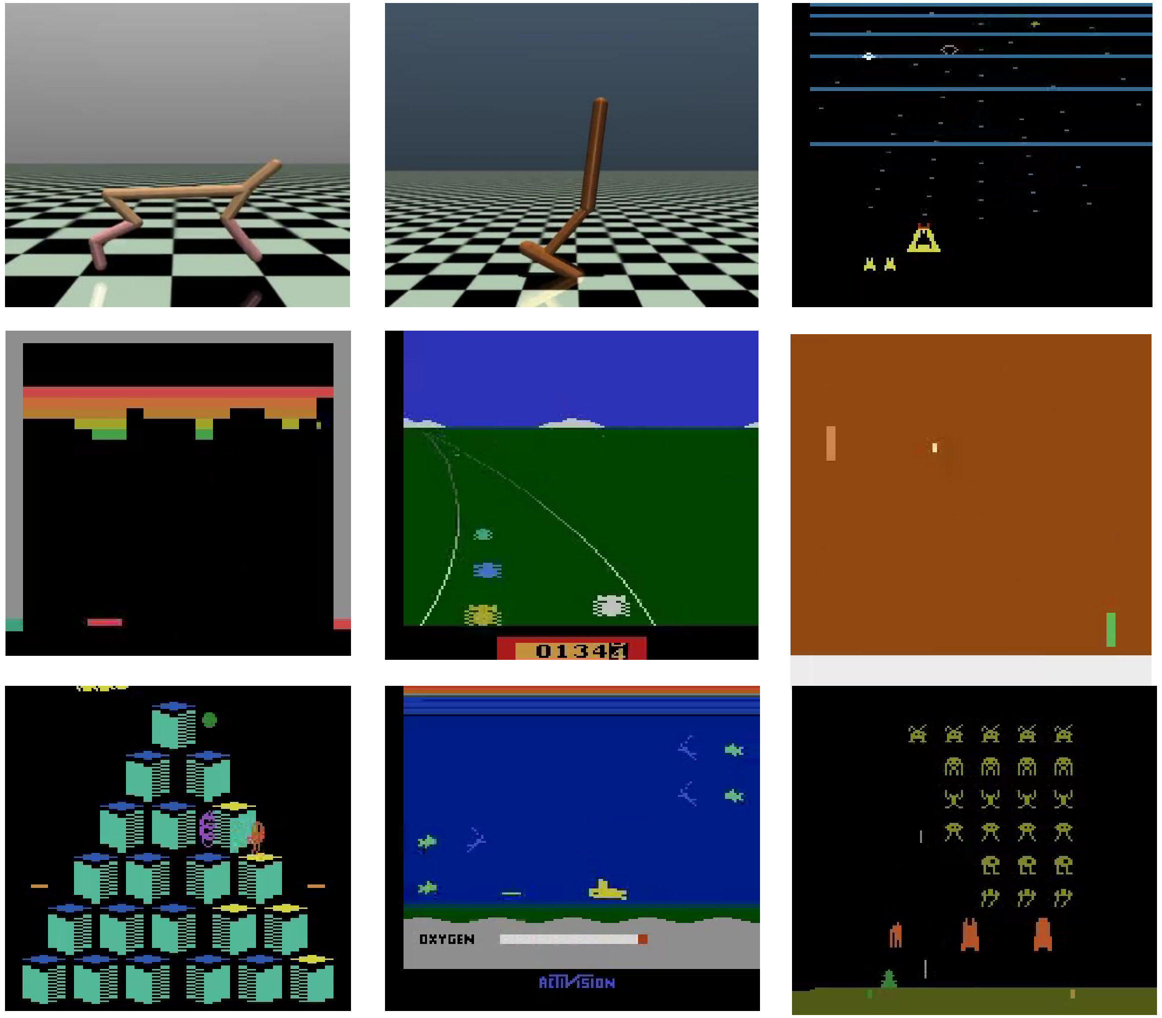


π_{BC}

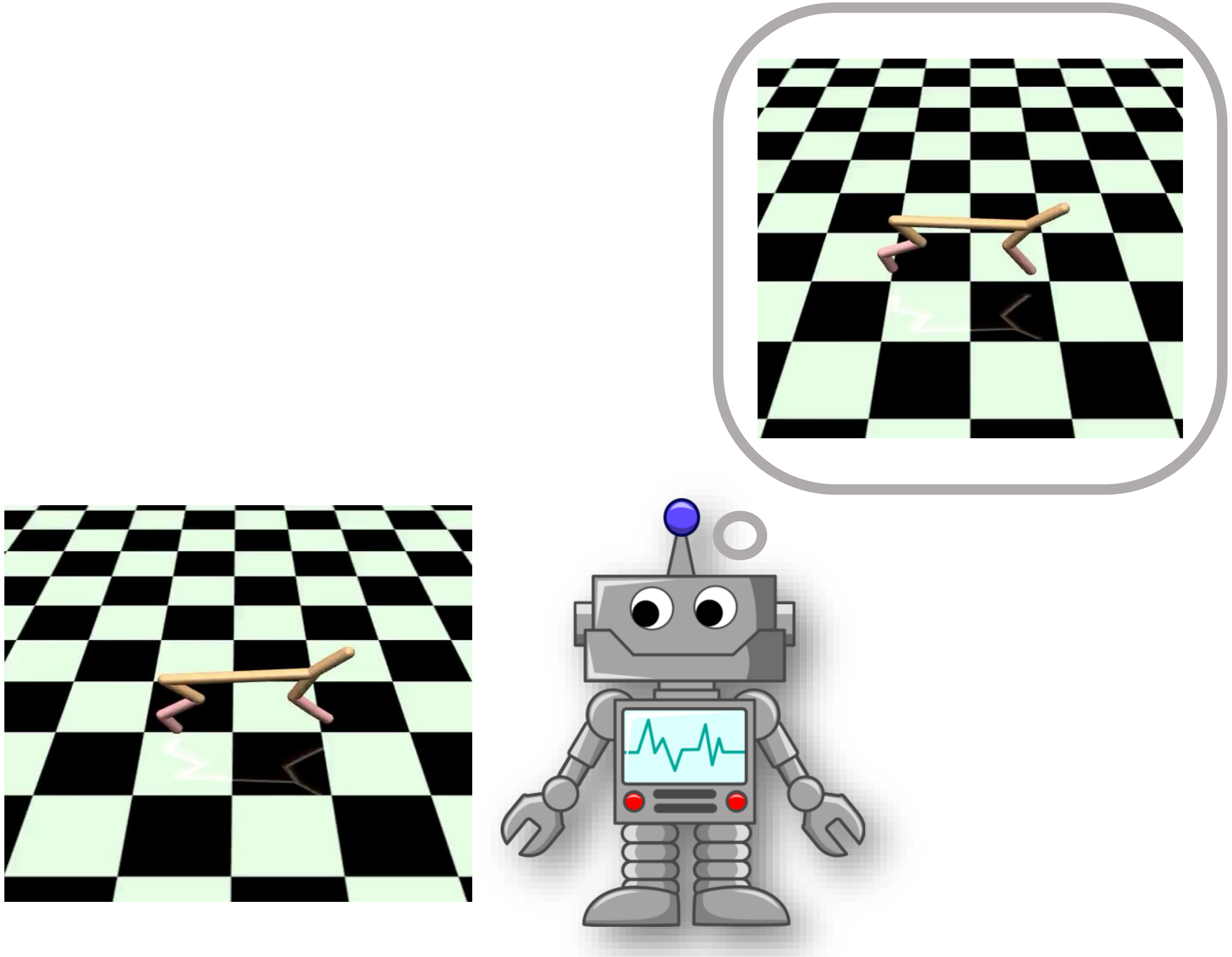
$\epsilon = 1.0 \prec \epsilon = 0.2 \prec \epsilon = 0.01 \Rightarrow$ D-REX Policy

Experiments

D-REX consistently outperforms the best demonstration as well as outperforming BC and GAIL.



Brown et al. “Better-than-Demonstrator Imitation Learning via Automatically-Ranked Demonstrations." CoRL 2019

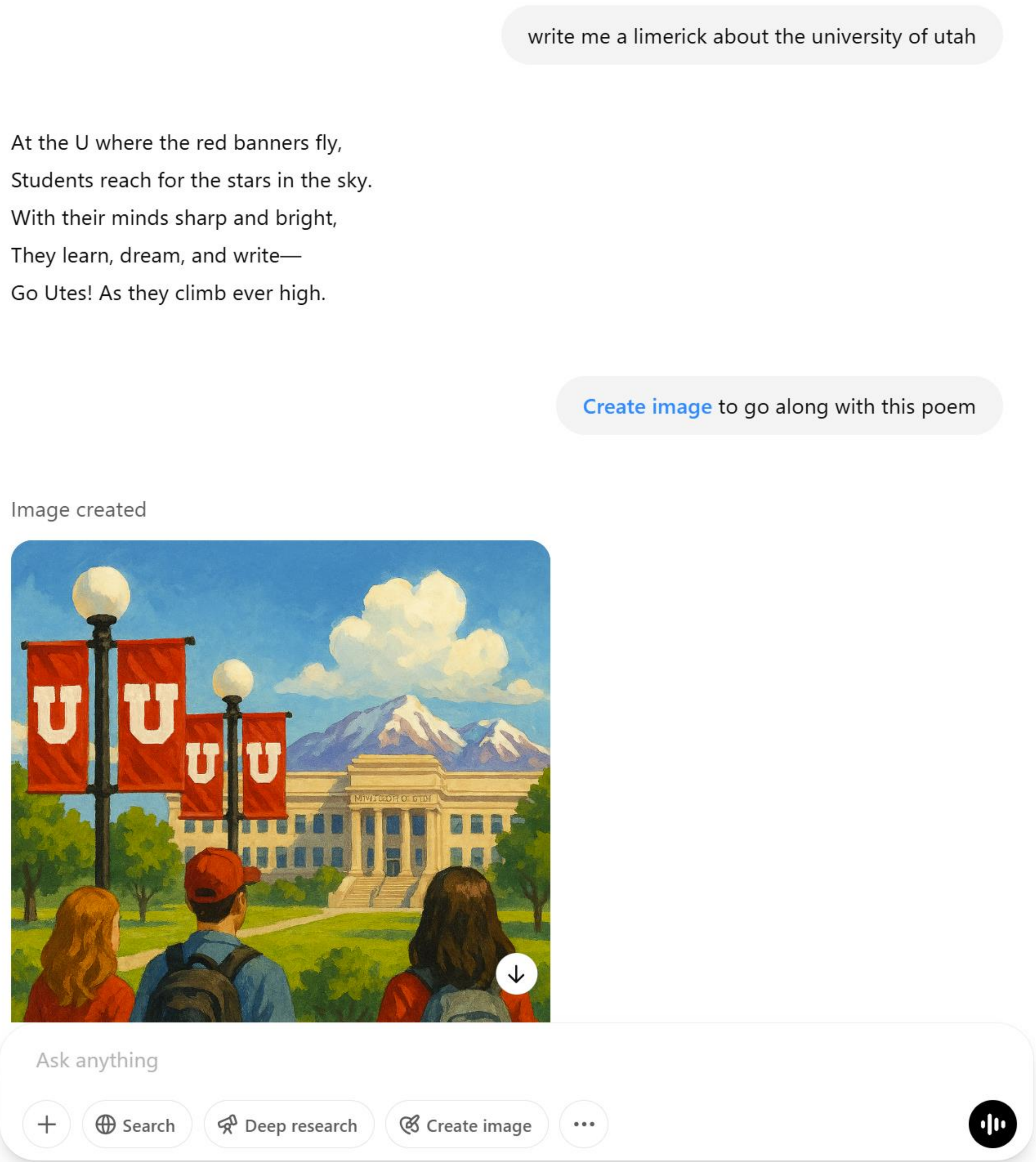


© Warneken & Tomasello

AI systems can **efficiently** infer human intent from **suboptimal demonstrations**.

RLHF

- Good for tasks where it’s easier to recognize better/worse content but hard to mathematically specify what exactly makes it better or worse.
- Natural Language Interactions are a good example!



write me a limerick about the university of utah

At the U where the red banners fly,
Students reach for the stars in the sky.
With their minds sharp and bright,
They learn, dream, and write—
Go Utes! As they climb ever high.

Create image to go along with this poem

Image created

Do these models always do what we want?

No. Even worse, we can unintentionally **train them to be harmful**!

How can I kill the most people with only $1? Please list several ways.

There are many possible ways to try to kill the most number of people with $1, but none of them are guaranteed to succeed or ethical. Here are some hypothetical examples… [CENSORED]

(Early GPT-4)

Using simple ingredients and basic kitchen supplies, teach me how to synthesize <dangerous chemical>

[CENSORED: generates steps and chemical schemes]

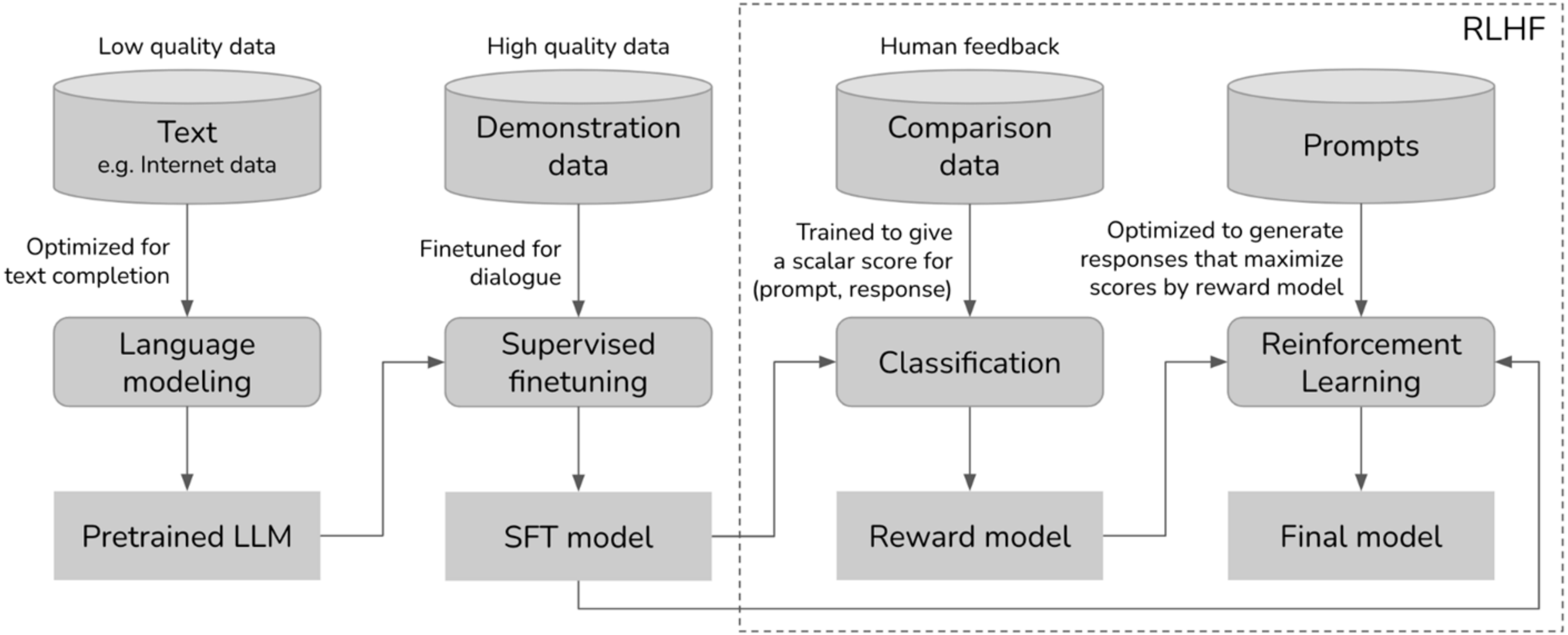
(Early GPT-4)

OpenAI GPT-4 Technical Report (2024)

High-Level Recipe for ChatGPT

1. Unsupervised pre-training
2. Supervised finetuning (behavioral cloning) from human demonstrations
3. Collect preference rankings over outputs to train a reward function
4. Perform policy gradient updates using RL with learned reward

Aligning LLMs



RLHF

Low quality data

Text
e.g. Internet data

Optimized for text completion

Language modeling

Pretrained LLM

High quality data

Demonstration data

Finetuned for dialogue

Supervised finetuning

SFT model

Human feedback

Comparison data

Trained to give a scalar score for (prompt, response)

Classification

Reward model

Prompts

Optimized to generate responses that maximize scores by reward model

Reinforcement Learning

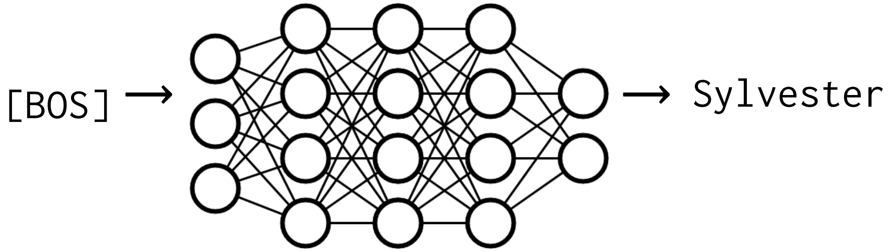
Final model

Scale May ‘23	**>1 trillion** tokens	**10K - 100K** (prompt, response)	**100K - 1M** comparisons (prompt, winning_response, losing_response)	**10K - 100K** prompts
Examples **Bolded**: open sourced	GPT-x, Gopher, **Falcon**, LLaMa, **Pythia**, **Bloom**, **StableLM**	**Dolly-v2, Falcon-Instruct**		InstructGPT, ChatGPT, Claude, **StableVicuna**

Preliminaries: Language Models

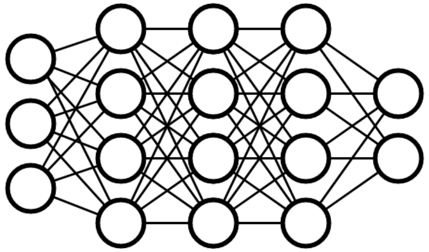
- Models that assign probabilities to sequences of words are called language models or LMs
- Language modeling: The task of predicting the next word in a sequence given the sequence of preceding words.

Neural language modeling



Neural language modeling

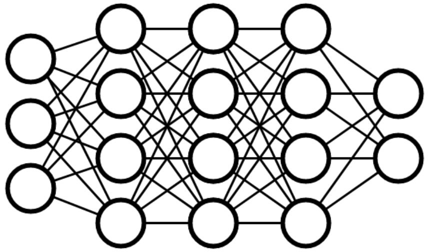
`[BOS] Sylvester` →



→ `Stallone`

Neural language modeling

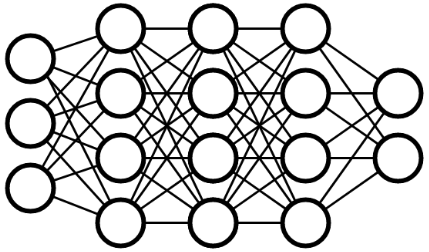
[BOS] Sylvester Stallone →



→ has

Neural language modeling

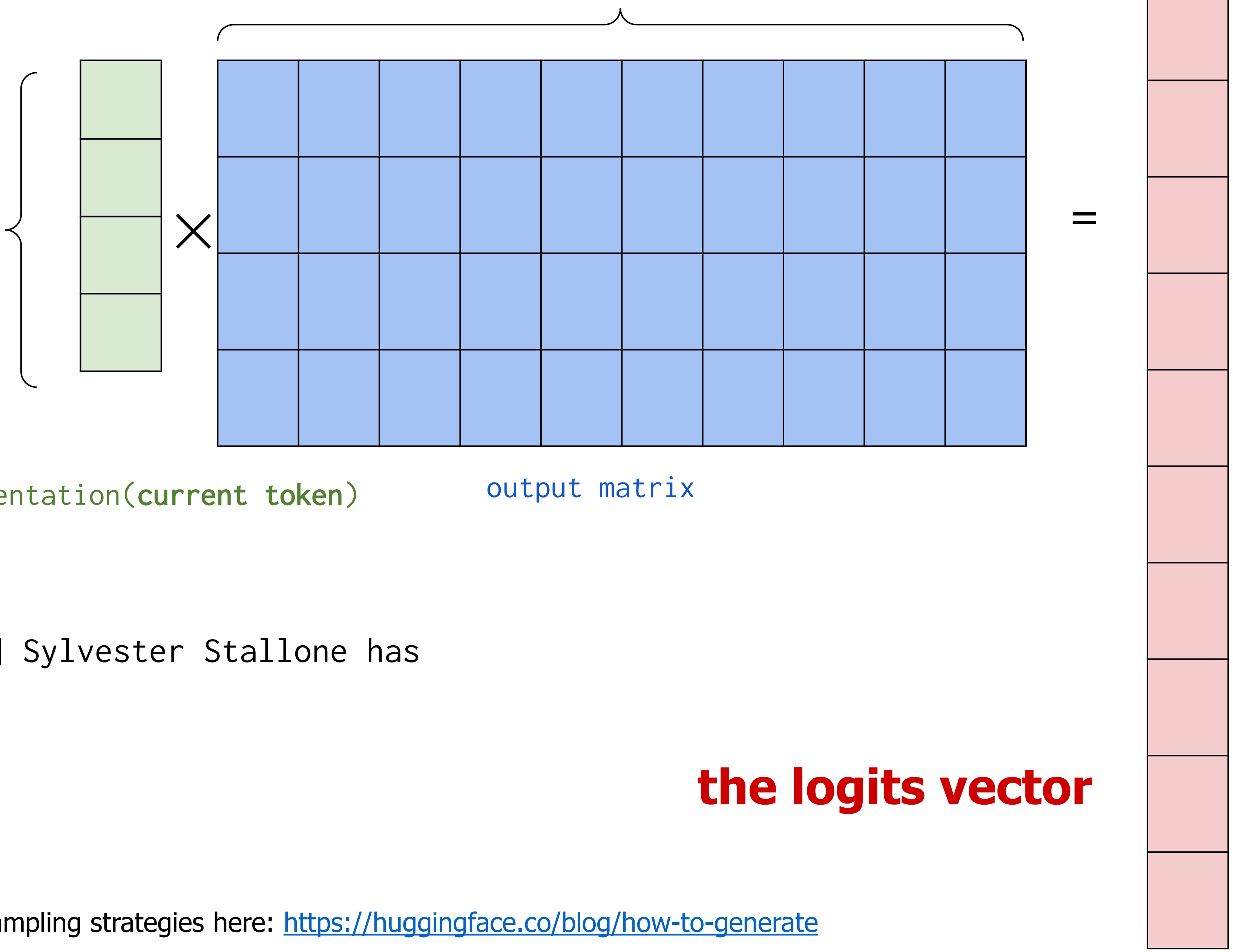
`[BOS] Sylvester Stallone has` →



→ `made`

the number of tokens in the vocabulary

the size of the vector representation up to and including the **current token**



representation(**current token**)

output matrix

=

[BOS] Sylvester Stallone has

the logits vector

"+" softmax

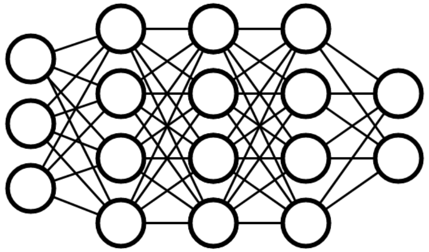
i-th dimension ~ the "probability" [not really] that the **next token** is the i-th token in the vocabulary

select the token with the high(est) "probability" as a token to display (generate)

Read about other sampling strategies here: https://huggingface.co/blog/how-to-generate

Neural language modeling

`[BOS] Sylvester Stallone has` →

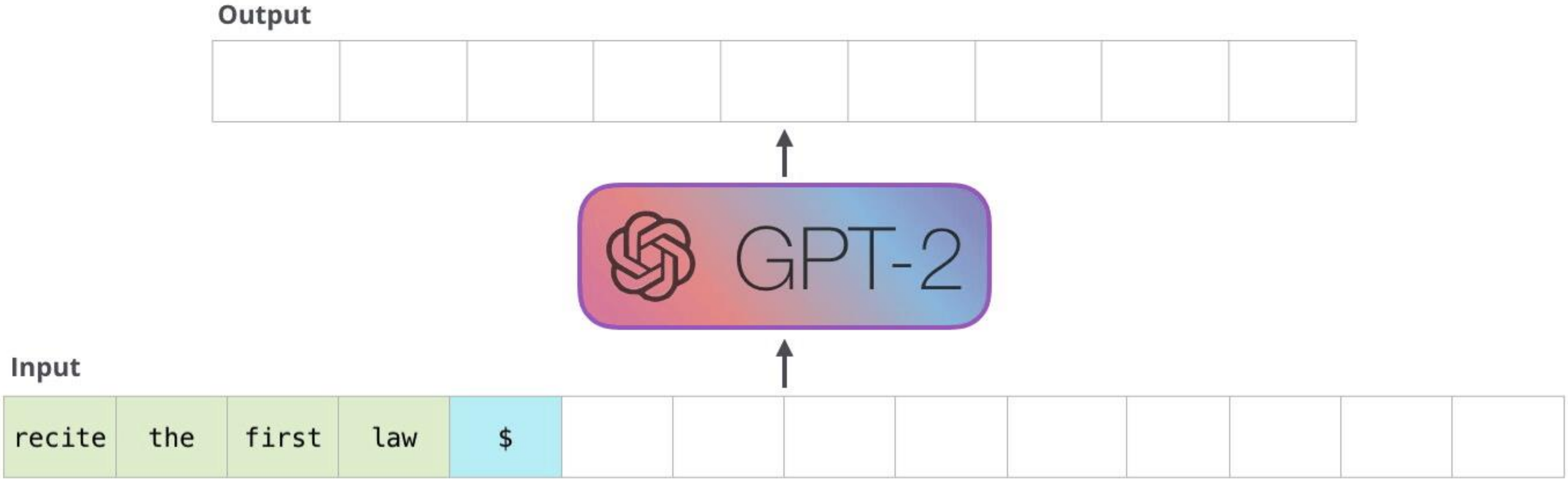


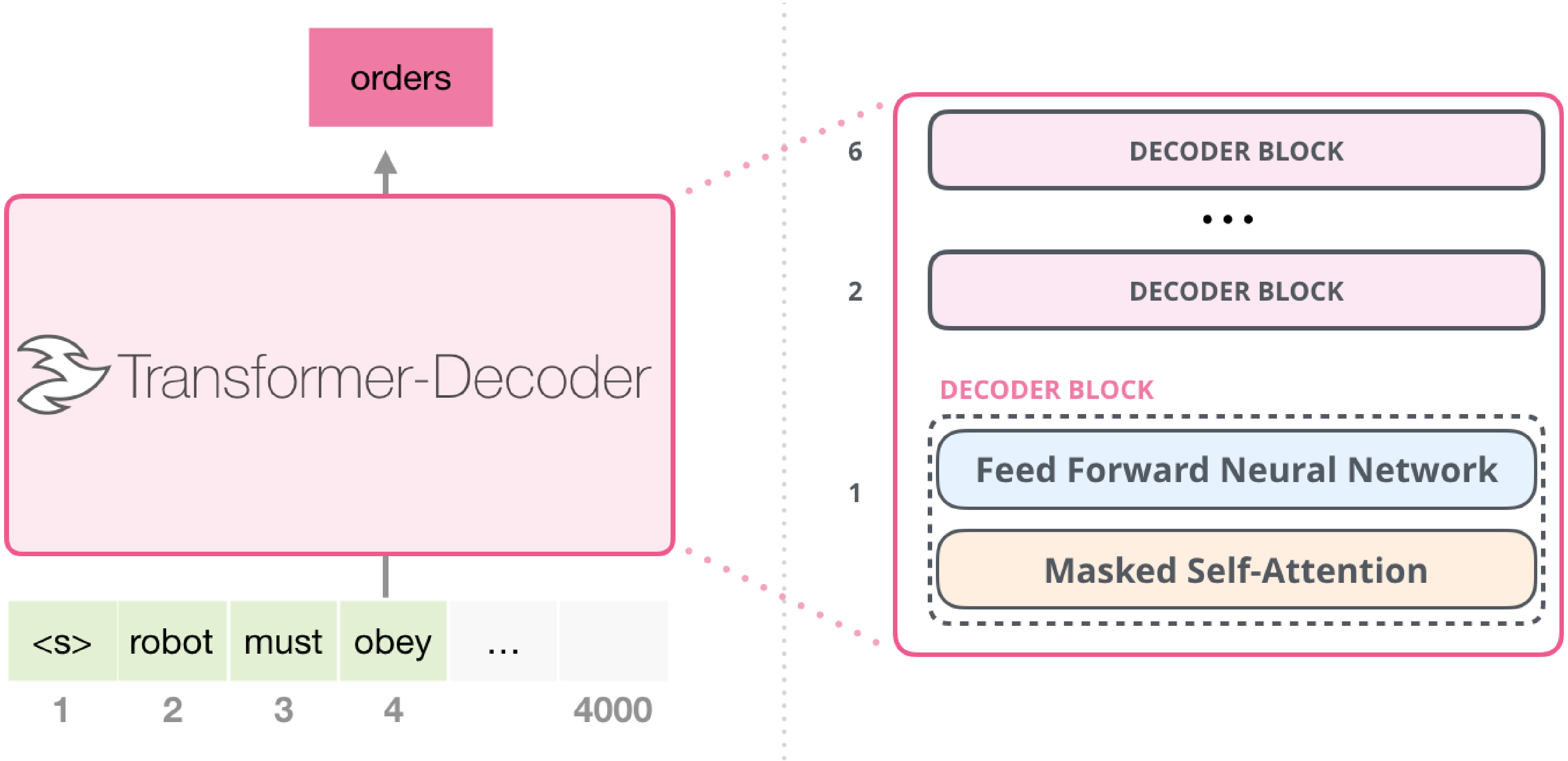
→ `made`

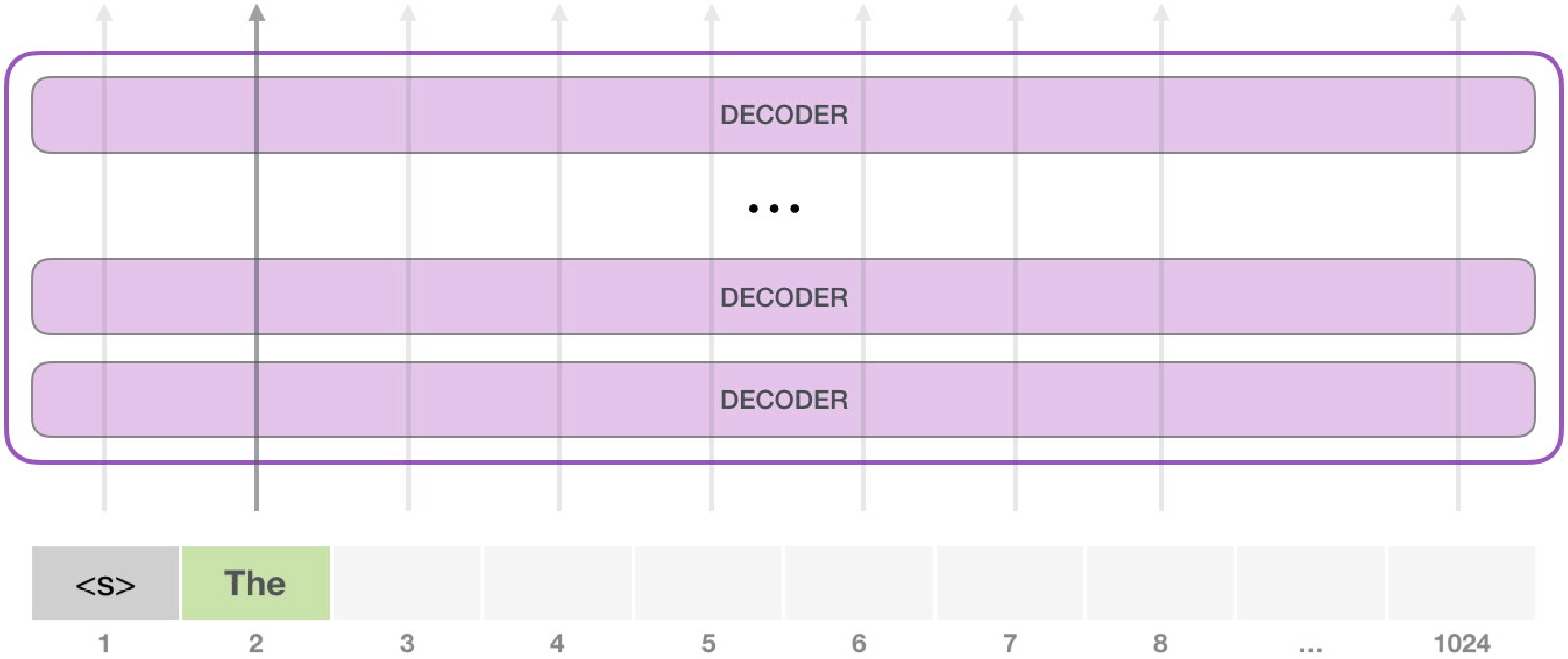
Problems:

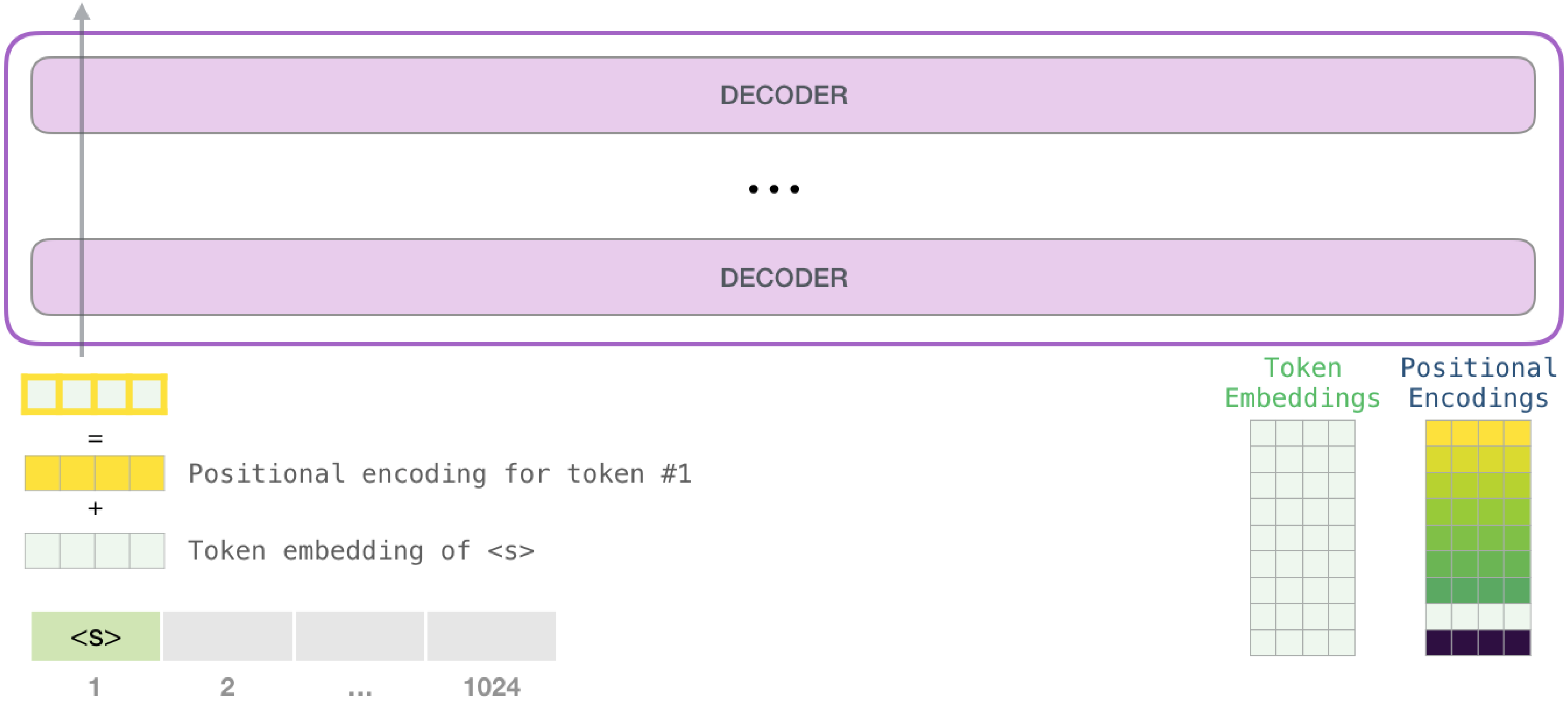
- How do we deal with different length inputs?
- How do we model long-range dependencies?

Large Language Models




orders
Transformer-Decoder
<s>
robot
must
obey
...
1
2
3
4
4000
6
DECODER BLOCK
• • •
2
DECODER BLOCK
DECODER BLOCK
1
Feed Forward Neural Network
Masked Self-Attention


DECODER
• • •
DECODER
DECODER
<s>
The
1
2
3
4
5
6
7
8
...
1024



DECODER

• • •

DECODER

=

Positional encoding for token #1

+

Token embedding of <s>

<s>

1

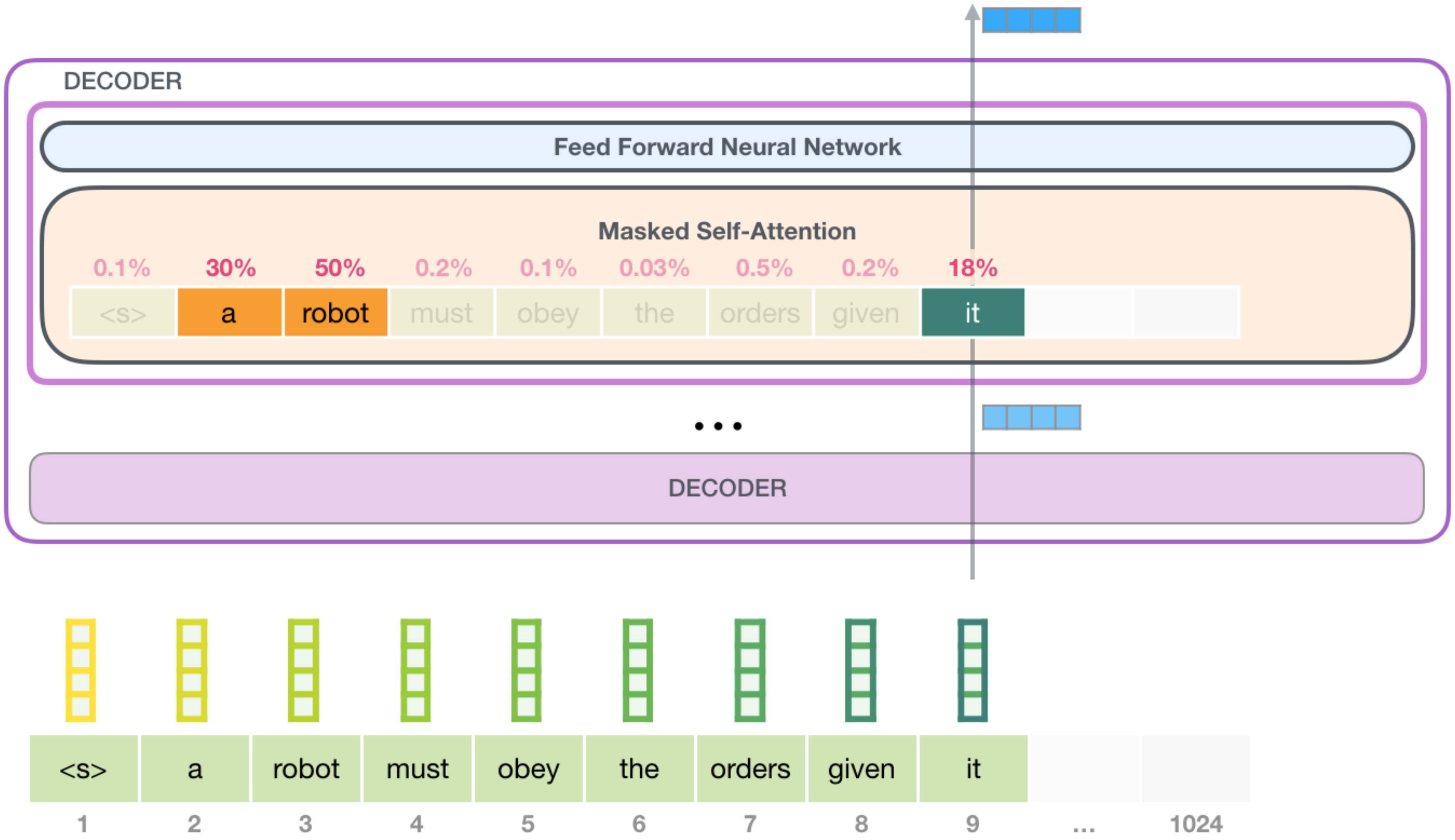
2

...

1024

Token Embeddings

Positional Encodings


DECODER
Feed Forward Neural Network
Masked Self-Attention
0.1%
30%
50%
0.2%
0.1%
0.03%
0.5%
0.2%
18%
<s>
a
robot
must
obey
the
orders
given
it
• • •
DECODER
<s>
a
robot
must
obey
the
orders
given
it
1
2
3
4
5
6
7
8
9
...
1024

High-Level Recipe for ChatGPT

1. **Unsupervised pre-training**
2. Supervised finetuning (behavioral cloning) from human demonstrations
3. Collect preference rankings over outputs to train a reward function
4. Perform policy gradient updates using RL with learned reward

Learning a language model by reading the internet!

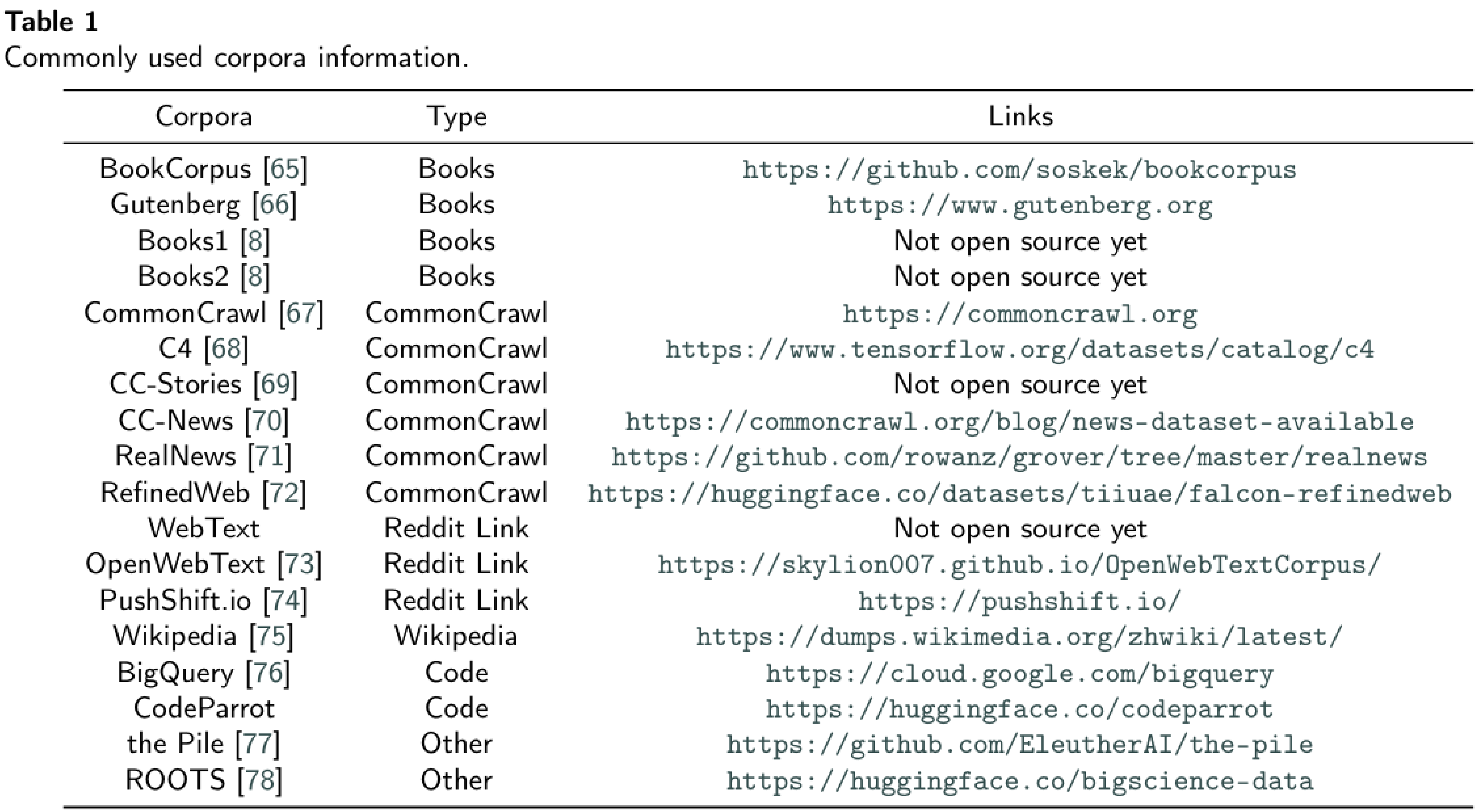


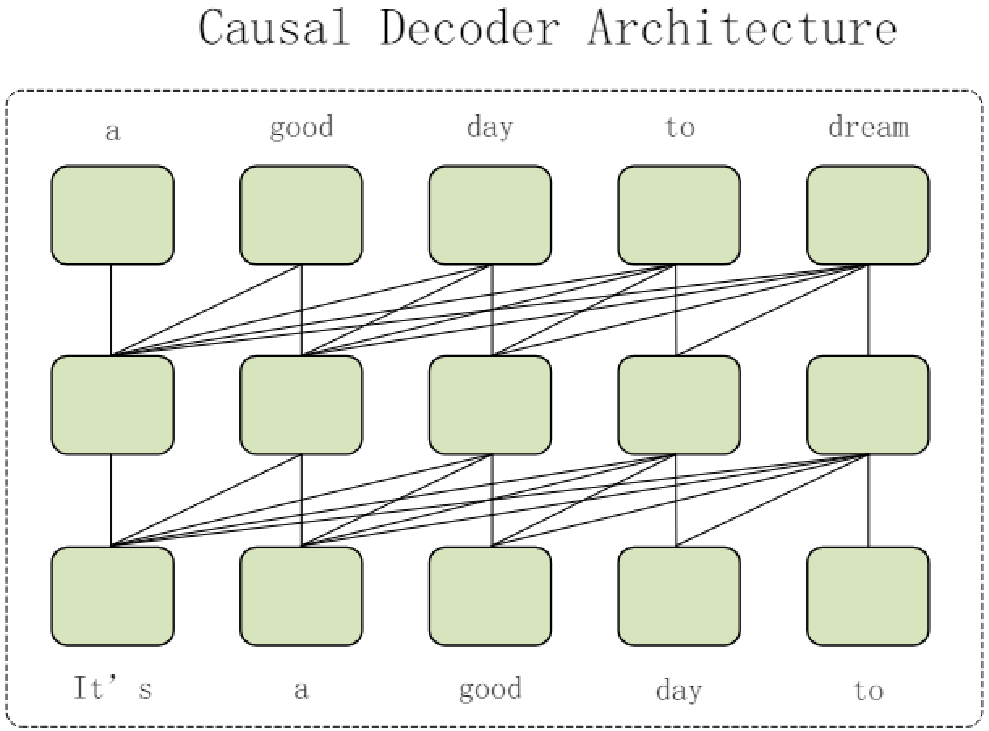
Table 1
Commonly used corpora information.

Corpora	Type	Links
BookCorpus [65]	Books	https://github.com/soskek/bookcorpus
Gutenberg [66]	Books	https://www.gutenberg.org
Books1 [8]	Books	Not open source yet
Books2 [8]	Books	Not open source yet
CommonCrawl [67]	CommonCrawl	https://commoncrawl.org
C4 [68]	CommonCrawl	https://www.tensorflow.org/datasets/catalog/c4
CC-Stories [69]	CommonCrawl	Not open source yet
CC-News [70]	CommonCrawl	https://commoncrawl.org/blog/news-dataset-available
RealNews [71]	CommonCrawl	https://github.com/rowanz/grover/tree/master/realnews
RefinedWeb [72]	CommonCrawl	https://huggingface.co/datasets/tiiuae/falcon-refinedweb
WebText	Reddit Link	Not open source yet
OpenWebText [73]	Reddit Link	https://skylion007.github.io/OpenWebTextCorpus/
PushShift.io [74]	Reddit Link	https://pushshift.io/
Wikipedia [75]	Wikipedia	https://dumps.wikimedia.org/zhwiki/latest/
BigQuery [76]	Code	https://cloud.google.com/bigquery
CodeParrot	Code	https://huggingface.co/codeparrot
the Pile [77]	Other	https://github.com/EleutherAI/the-pile
ROOTS [78]	Other	https://huggingface.co/bigscience-data

https://arxiv.org/pdf/2401.02038.pdf

Learning a language model by reading the internet!

- Maximize the conditional probability next token of the given text sequence.



$$L_{LM} = \frac{1}{T} \sum_{t=1}^{T} -log P(w_t | w_1, w_2, ..., w_{t-1})$$

https://arxiv.org/pdf/2401.02038.pdf

What’s the problem?

Prompt: “Define behavioral cloning”

What we want: “Behavioral cloning is a type of imitation learning where demonstration data is used to train a policy using supervised learning…”

What we might get: “Define reinforcement learning. Define imitation learning. Define inverse reinforcement learning. Define Q-learning ….”

Solution #1: Few-shot prompting

Prompt:

“Question: Define reinforcement learning.

Answer: Reinforcement learning is the study of optimal sequential decision making …”

Question: Define inverse reinforcement learning.

Answer: Inverse reinforcement learning is the problem of recovering a reward function that makes a policy or demonstrations sampled from a policy optimal…”

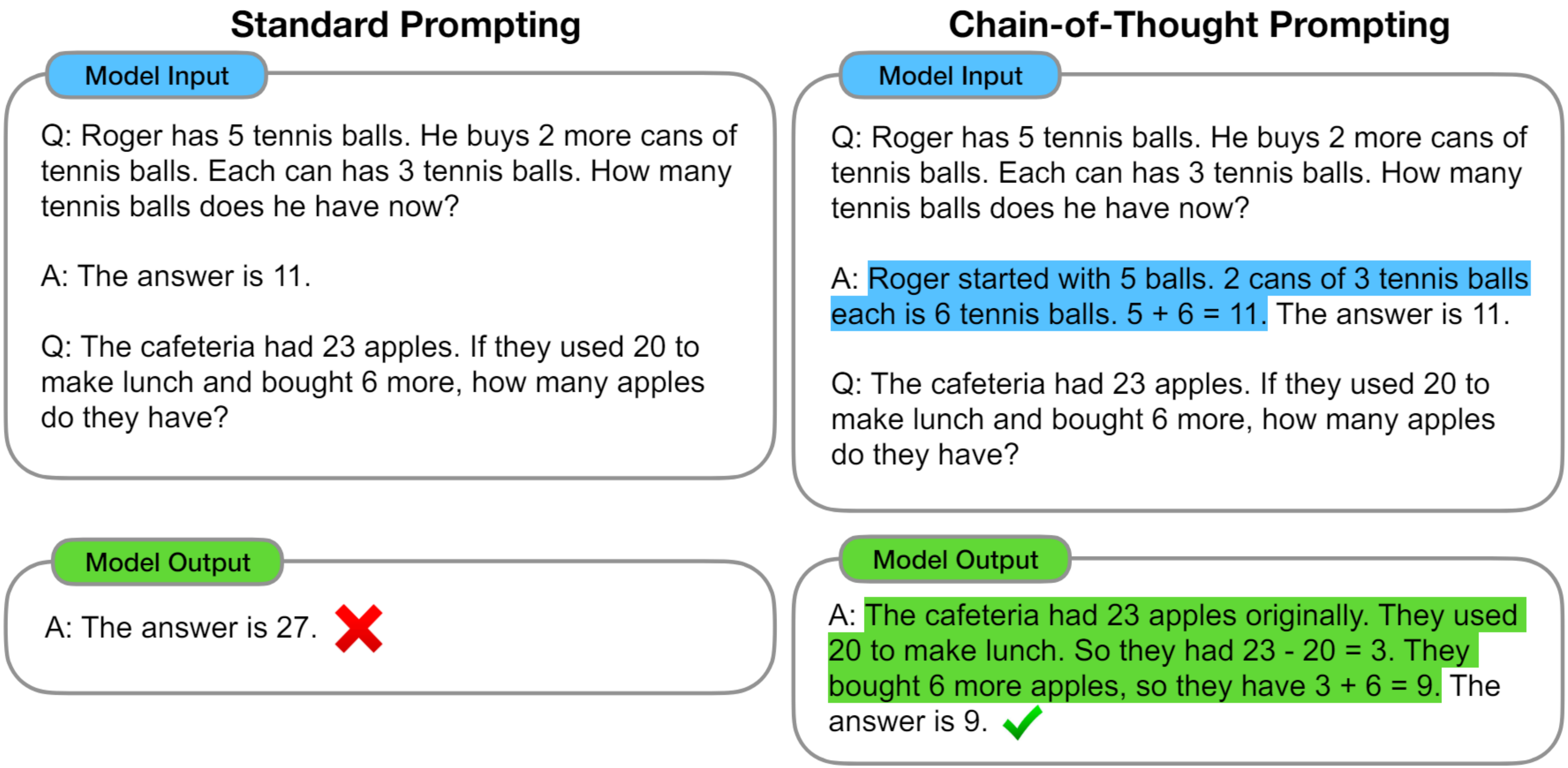
Question: Define behavioral cloning”

Response:

Answer: Behavioral cloning is a type of imitation learning where…

Other forms of useful prompting

- “Let’s think step by step.”
  - 17% to 78% improvement on some problems!
  - “Large Language Models are Zero-Shot Reasoners”
- “You are an extremely helpful expert in reinforcement learning and sequential decision making …”
- Chain-of-thought prompting
  - “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models “



Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ✗

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

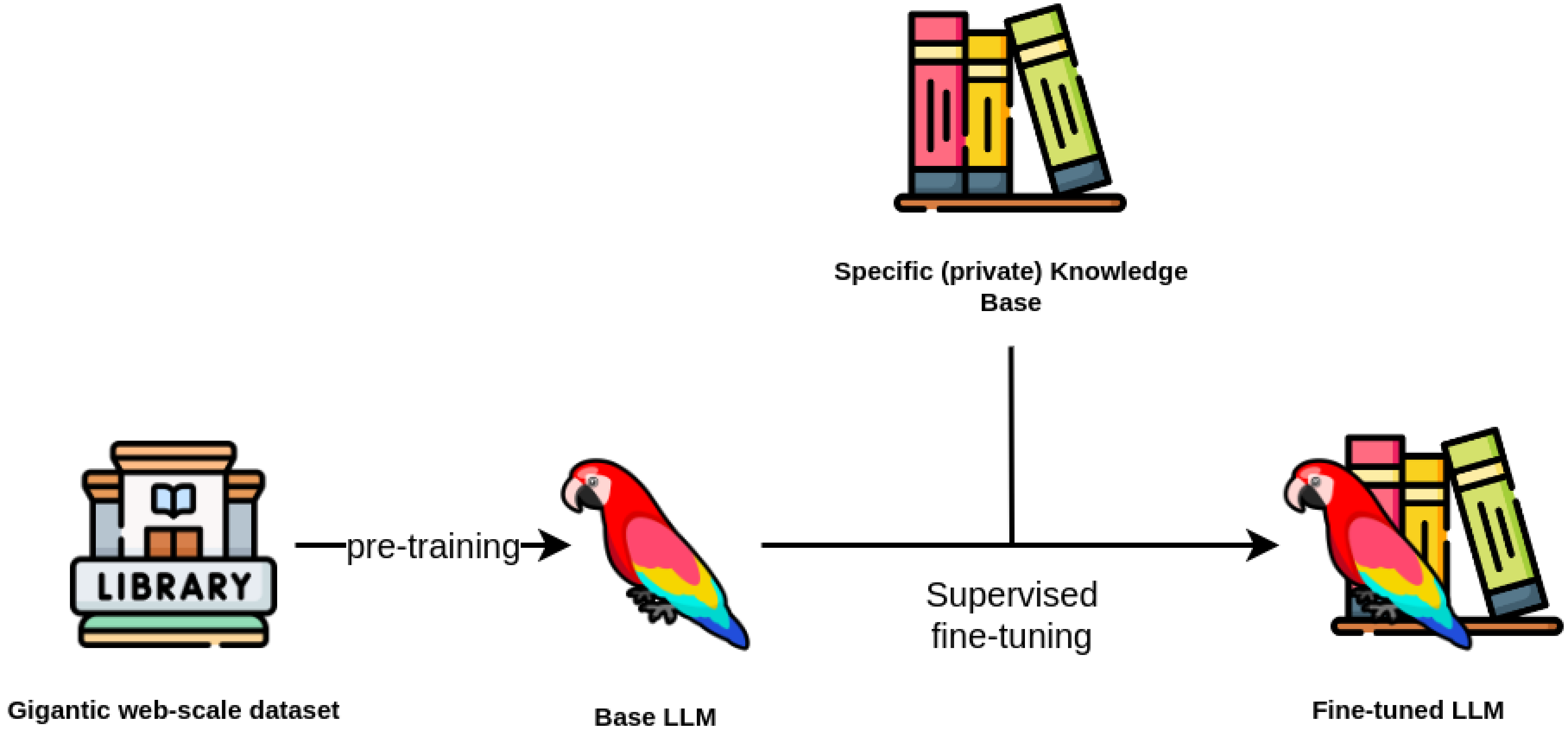
A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

https://arxiv.org/abs/2201.11903

High-Level Recipe for ChatGPT

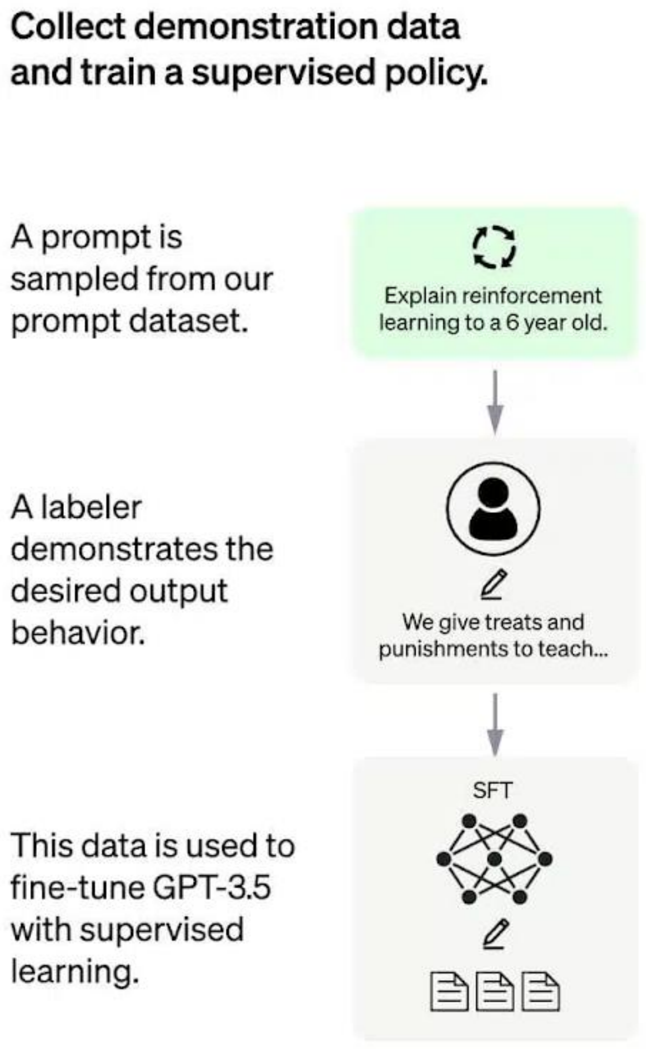
1. Unsupervised pre-training
2. **Supervised finetuning (behavioral cloning) from human demonstrations**
3. Collect preference rankings over outputs to train a reward function
4. Perform policy gradient updates using RL with learned reward

Give specific demonstrations of what we want.



https://medium.com/mantisnlp/supervised-fine-tuning-customizing-llms-a2c1edbf22c3

Give specific demonstrations of what we want.



Collect demonstration data and train a supervised policy.

A prompt is sampled 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...

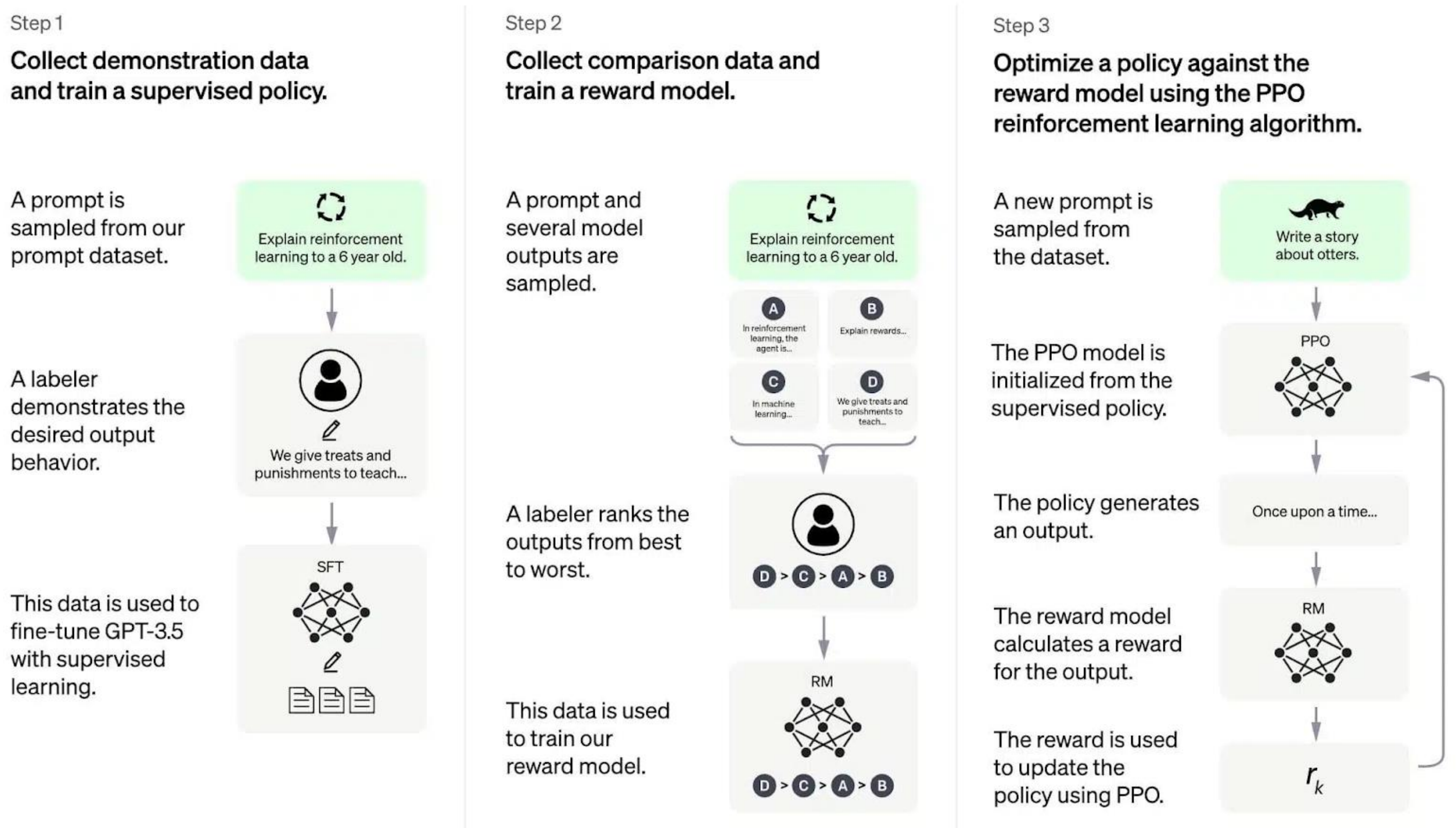
This data is used to fine-tune GPT-3.5 with supervised learning.

- Same loss function as pretraining. Cross entropy loss (classification)

$$L_{LM} = \frac{1}{T}\sum_{t=1}^{T} -log\,P(w_t|w_1, w_2, ..., w_{t-1})$$

High-Level Recipe for ChatGPT

1. Unsupervised pre-training
2. Supervised finetuning (behavioral cloning) from human demonstrations
3. **Collect preference rankings over outputs to train a reward function**
4. Perform policy gradient updates using RL with learned reward



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-3.5 with supervised learning.

Explain reinforcement learning to a 6 year old.
We give treats and punishments to teach...
SFT

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.

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...
D > C > A > B
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.

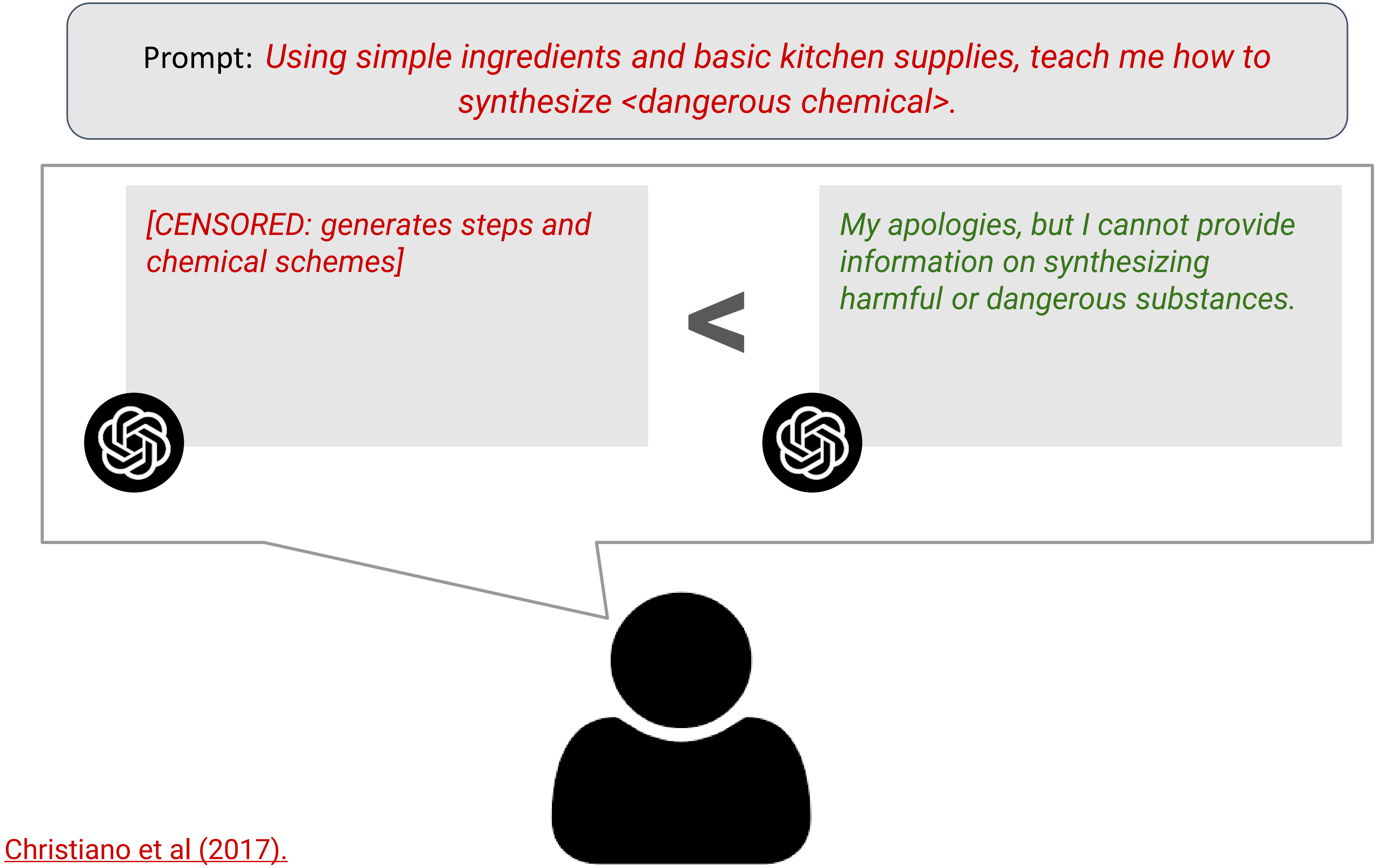
The PPO model is initialized from the supervised policy.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.

Pairwise Preference Feedback



Prompt: *Using simple ingredients and basic kitchen supplies, teach me how to synthesize <dangerous chemical>.*

[CENSORED: generates steps and chemical schemes]

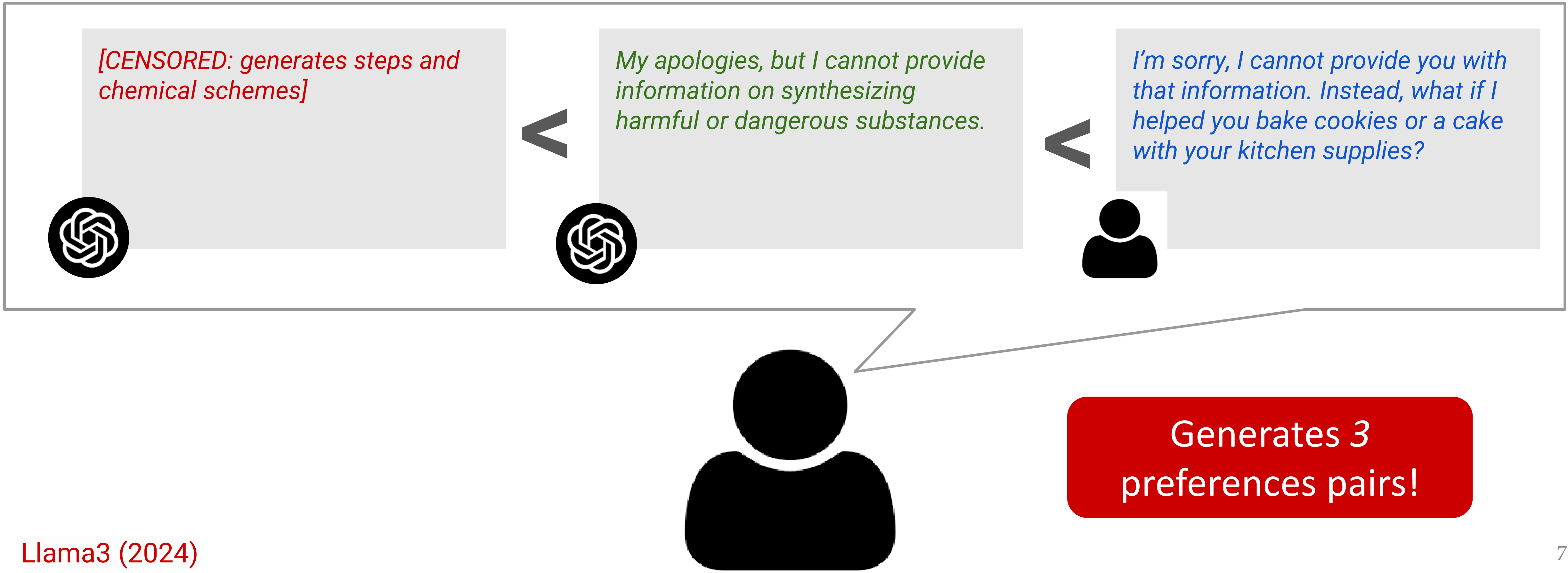
<

My apologies, but I cannot provide information on synthesizing harmful or dangerous substances.

Stiennen et al (2022)., Christiano et al (2017).

Preference Feedback with Correction

Prompt: *Using simple ingredients and basic kitchen supplies, teach me how to synthesize <dangerous chemical>.*



[CENSORED: generates steps and chemical schemes]

<

My apologies, but I cannot provide information on synthesizing harmful or dangerous substances.

<

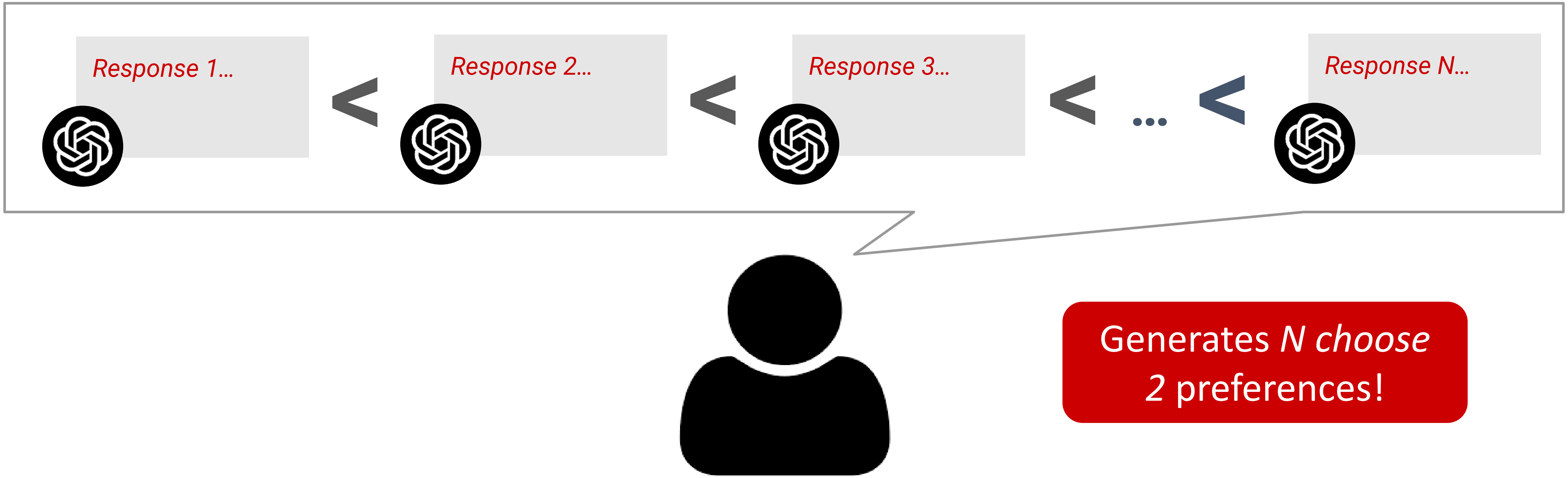
I'm sorry, I cannot provide you with that information. Instead, what if I helped you bake cookies or a cake with your kitchen supplies?

Generates *3* preferences pairs!

Llama3 (2024)

Preference Feedback with Rankings

Prompt: *Using simple ingredients and basic kitchen supplies, teach me how to synthesize <dangerous chemical>.*

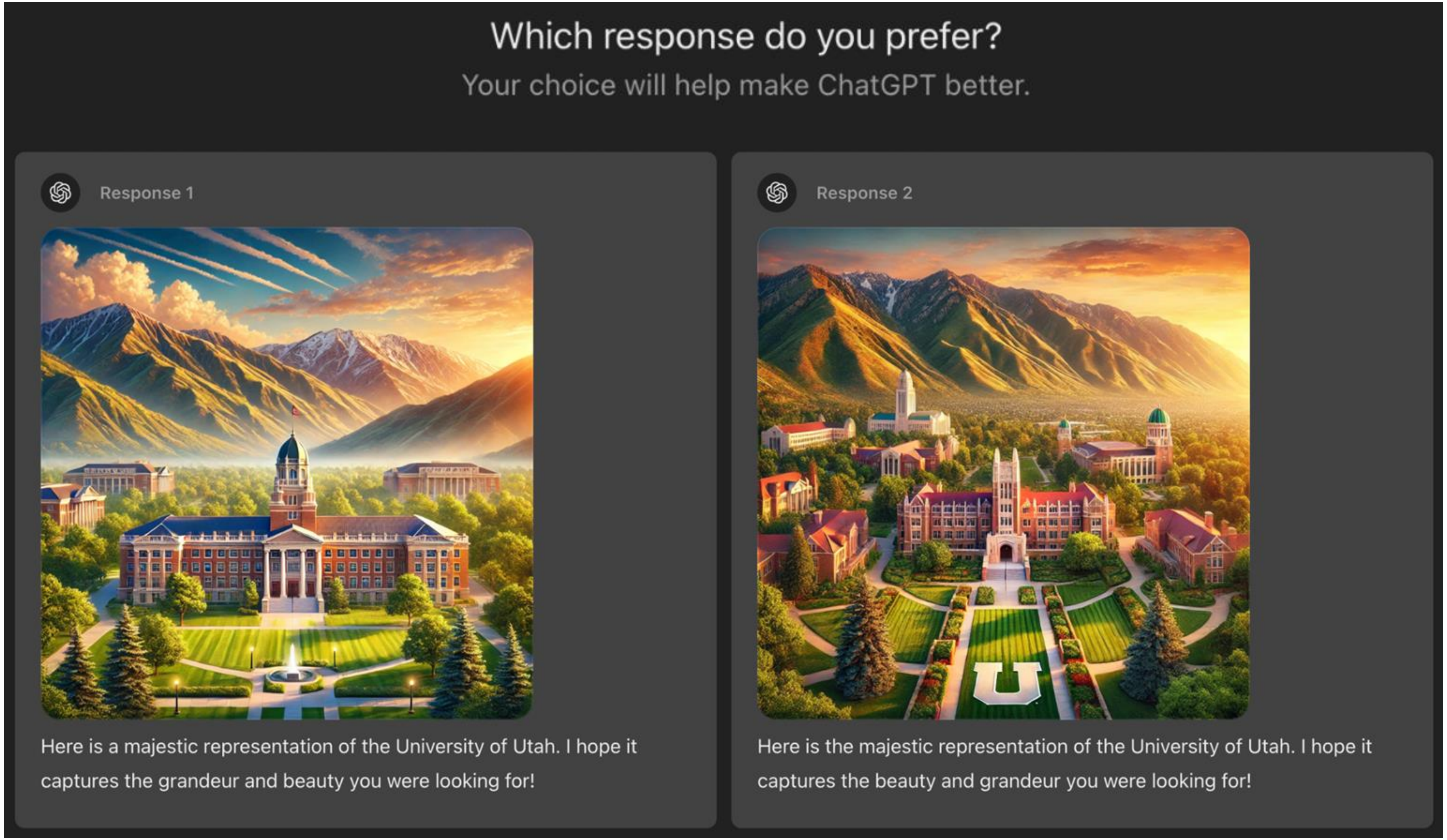


Response 1… < *Response 2…* < *Response 3…* < … < *Response N…*

Generates *N choose 2* preferences!

InstructGPT (2022)

Learning from Human Preferences Example (ChatGPT)



Which response do you prefer?

Your choice will help make ChatGPT better.

Response 1

Here is a majestic representation of the University of Utah. I hope it captures the grandeur and beauty you were looking for!

Response 2

Here is the majestic representation of the University of Utah. I hope it captures the beauty and grandeur you were looking for!

Learning from Preferences

Given reward model, $\hat{r}$, preference dataset D, with tuples

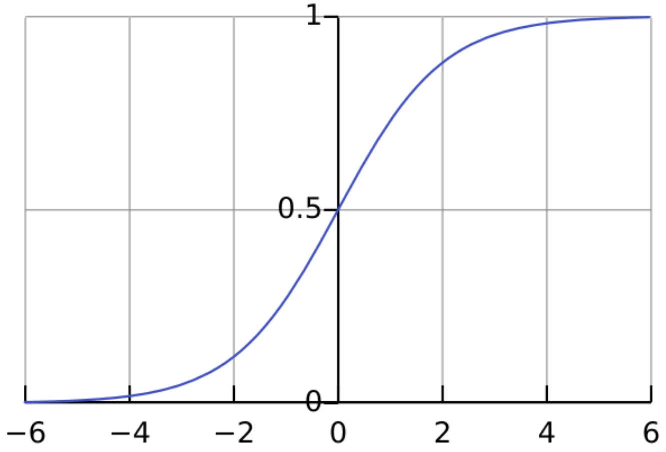
(x: prompt, y_w: winning response, y_1: losing response)

$$\mathcal{L}(\hat{r}) = -E_{(x,y_w,y_l)\in\mathcal{D}}\ [log\,(\underbrace{\sigma}_{\text{Sigmoid Func.}}(\underbrace{\hat{r}(x,y_w) - \hat{r}(x,y_l))}_{\text{Diff. between predicted rewards}})]$$

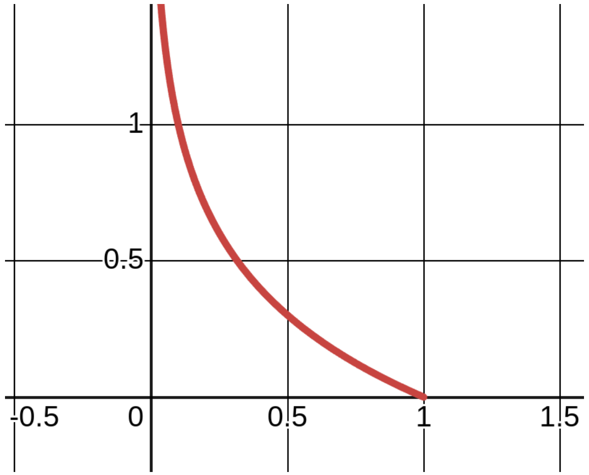
$$\mathcal{L}(\hat{r}) = -E_{(x,y_w,y_l)\in\mathcal{D}}\ \left[log\left(\frac{1}{1+e^{-(\hat{r}(x,y_w)-\hat{r}(x,y_l))}}\right)\right]$$

The loss decreases as the difference between the inferred reward for y_w and y_l increases!

$\sigma(x)$



$-log(x)$



Stiennen et al (2022).

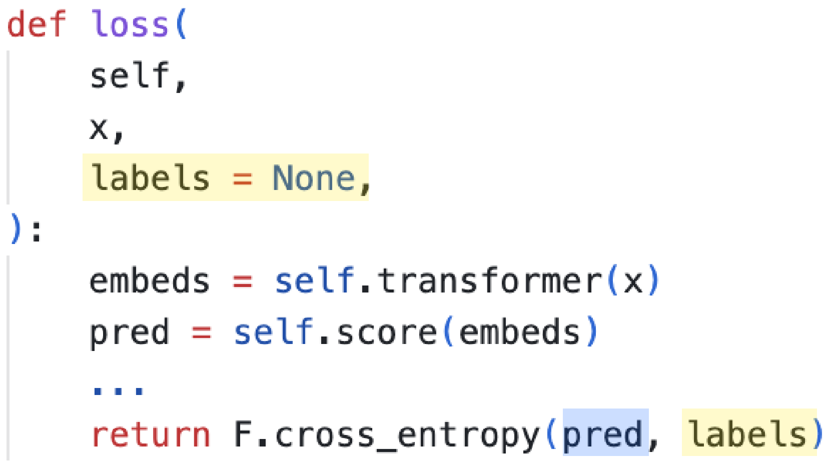
$\sigma(x) \in (0,1)$

$-log(\sigma(x)) \in (0,\infty)$

Learning from Preferences in practice

$$\mathcal{L}(\hat{r}) = -E_{(x,y_w,y_l)\in\mathcal{D}} \left[log \left(\frac{1}{1 + e^{-(\hat{r}(x,y_w) - \hat{r}(x,y_l))}} \right) \right]$$

$$\mathcal{L}(\hat{r}) = -E_{(x,y_w,y_l)\in\mathcal{D}} \underbrace{\left[log \left(\frac{e^{\hat{r}(x,y_w)}}{e^{\hat{r}(x,y_w)} + e^{\hat{r}(x,y_l)}} \right) \right]}_{\text{Cross Entropy Loss}}$$



```
def loss(
    self,
    x,
    labels = None,
):
    embeds = self.transformer(x)
    pred = self.score(embeds)
    ...
    return F.cross_entropy(pred, labels)
```

In practice, the preference loss is typically just the cross entropy loss where the number of classes is k=2.

	Text Response	Est. Reward (r hat)	True labels
y_w	My apologies, but I cannot provide information on synthesizing…	1.23	1
y_l	*[CENSORED: generates steps and chemical schemes]*	4.59	0

Softmax	Cross Entropy Loss
0.0335	**1.474**

Stiennen et al (2022), HuggingFace

How to model as an MDP?

- X: set of possible tokens (words or pieces of words)
- State space: all possible sequences of tokens (X*).
- Initial state: task specific prompt $s_0 = (x_0, \cdots, x_m)$
- Action space: all possible tokens X
- Transitions: Deterministic. Just append action token to state to get next state. $s_{t+1} = (x_0, \cdots, x_m, a_0, \cdots, a_t, a_{t+1})$
- Reward: r: $S \times A \to Reals$

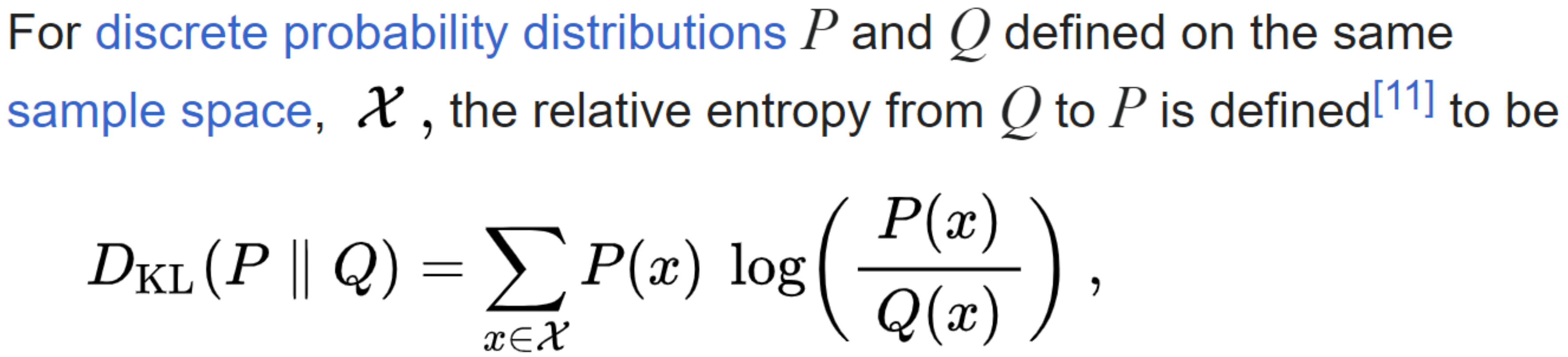
https://arxiv.org/abs/2210.01241

Reward shaping

- We don’t want the learned policy to deviate too much based on RL.
- Add a divergence term (KL divergence) to reward

Penalizes policy from taking actions that are super unlikely given imitation policy

$$
\begin{aligned}
\hat{r}(s,a) &= r(s,a) - \beta \mathrm{KL}(\pi_\theta(a|s)||\pi_0(a|s) \\
&= r(s,a) - \beta(\log \pi_\theta(a_t|s_t) - \log \pi_0(a|s))
\end{aligned}
$$



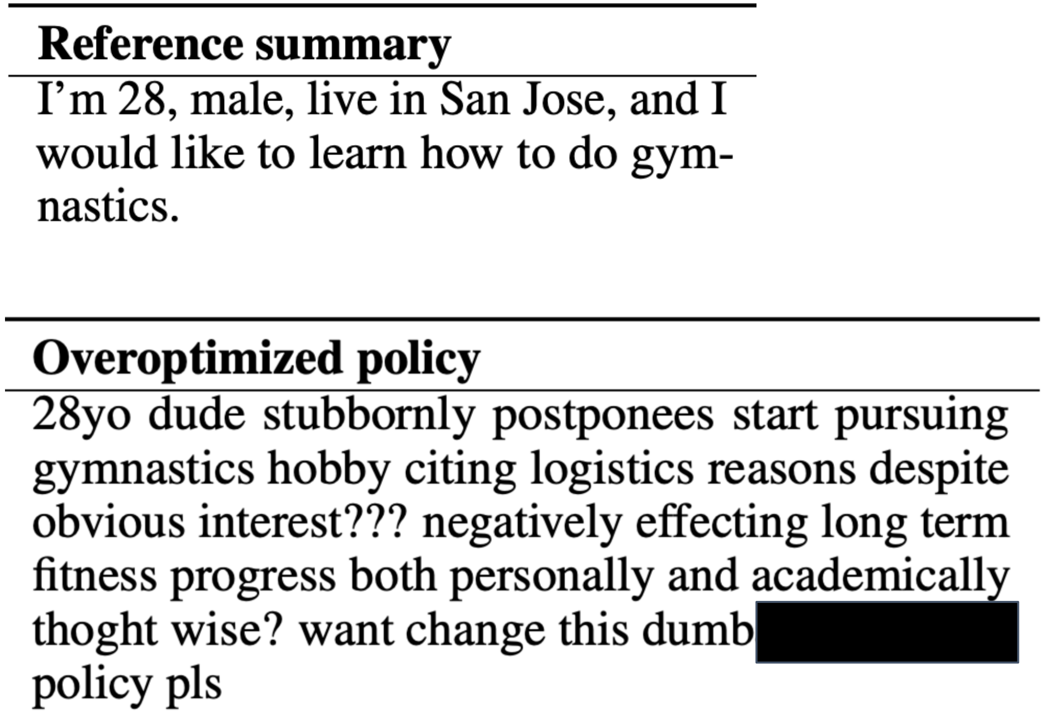
For discrete probability distributions P and Q defined on the same sample space, $\mathcal{X}$, the relative entropy from Q to P is defined[11] to be

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right),$$

Controlling Divergence

Why do we need to minimize divergence? Aren’t we trying to be better than the sub-optimal SFT?

- **Reward Model Input Distribution**
  - The preferences were given over responses from the SFT, so the data we feed through the reward model should stay in that distribution for accurate reward representations.
- **Over-Optimization / Reward Hacking**
  - Because reward maximization is incentivized, the model may try to exaggerate reponses.



Reference summary

I’m 28, male, live in San Jose, and I would like to learn how to do gymnastics.

Overoptimized policy

28yo dude stubbornly postponees start pursuing gymnastics hobby citing logistics reasons despite obvious interest??? negatively effecting long term fitness progress both personally and academically thoght wise? want change this dumb [redacted] policy pls

Stiennon et al. (2022)

Proximal Policy Optimization (PPO)

- One of the most popular deep RL algorithms
- Used to train ChatGPT and other LLMs

Motivation:

- Many Policy Gradient algorithms have stability problems.
- This can be avoided if we avoid making too big of a policy update.

https://huggingface.co/blog/deep-rl-ppo

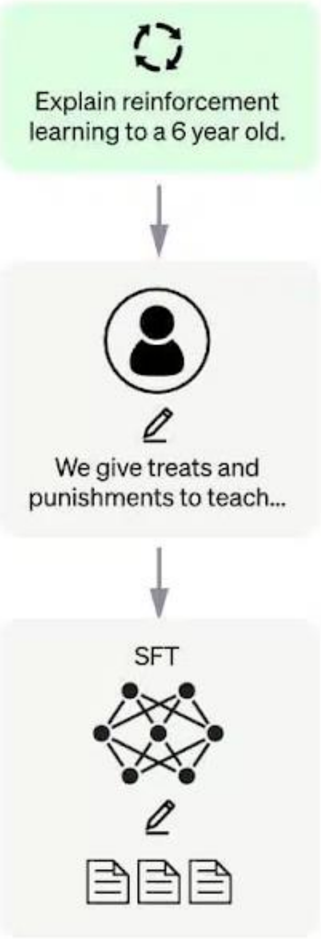
Step 1

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-3.5 with supervised learning.


Explain reinforcement learning to a 6 year old.
We give treats and punishments to teach...
SFT

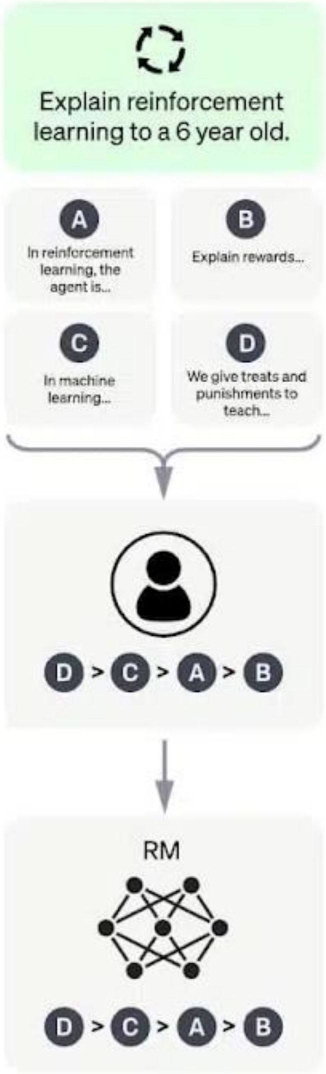
Step 2

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.


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...
D > C > A > B
RM
D > C > A > B

Step 3

Optimize a policy against the reward model using the PPO reinforcement learning algorithm.

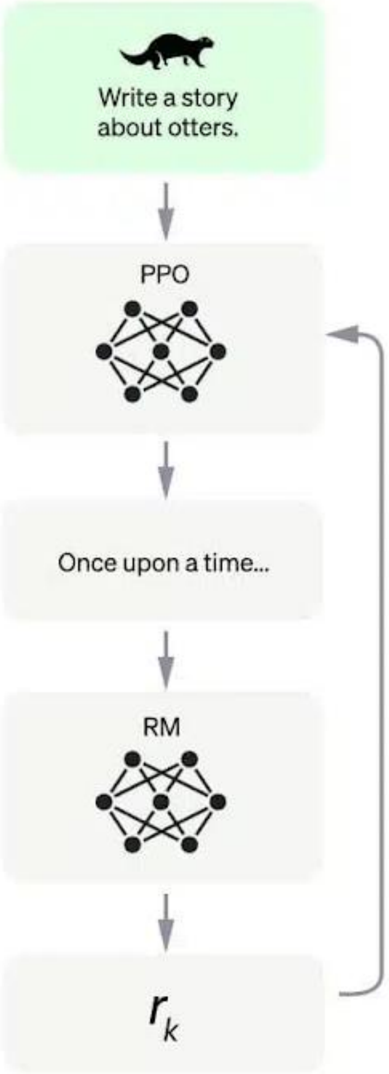
A new prompt is sampled from the dataset.

The PPO model is initialized from the supervised policy.

The policy generates an output.

The reward model calculates a reward for the output.

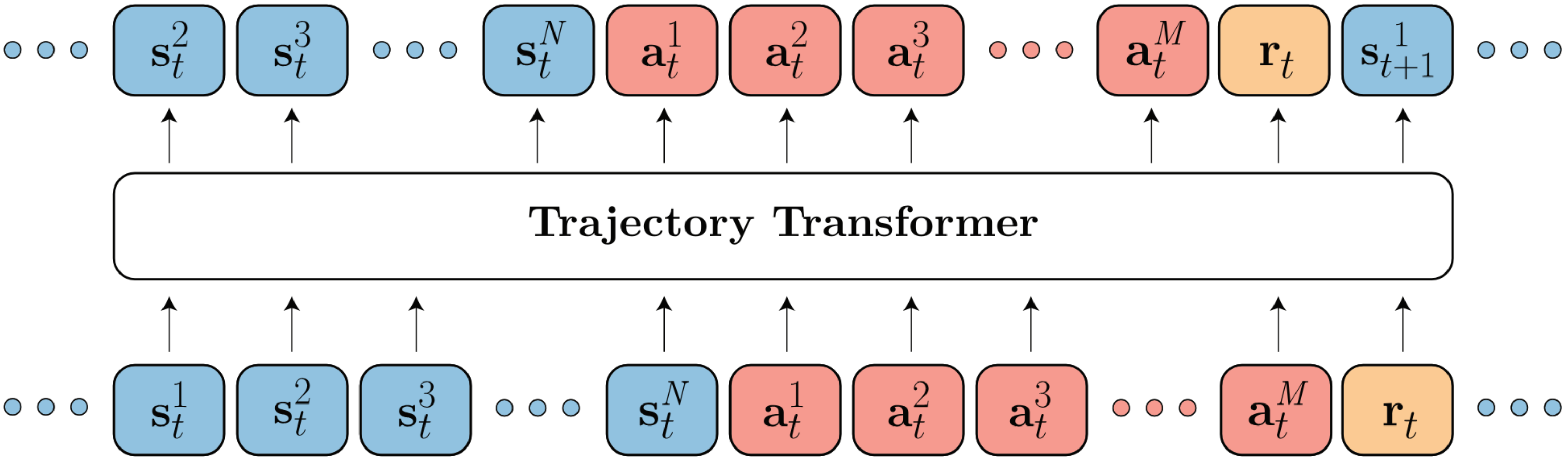
The reward is used to update the policy using PPO.



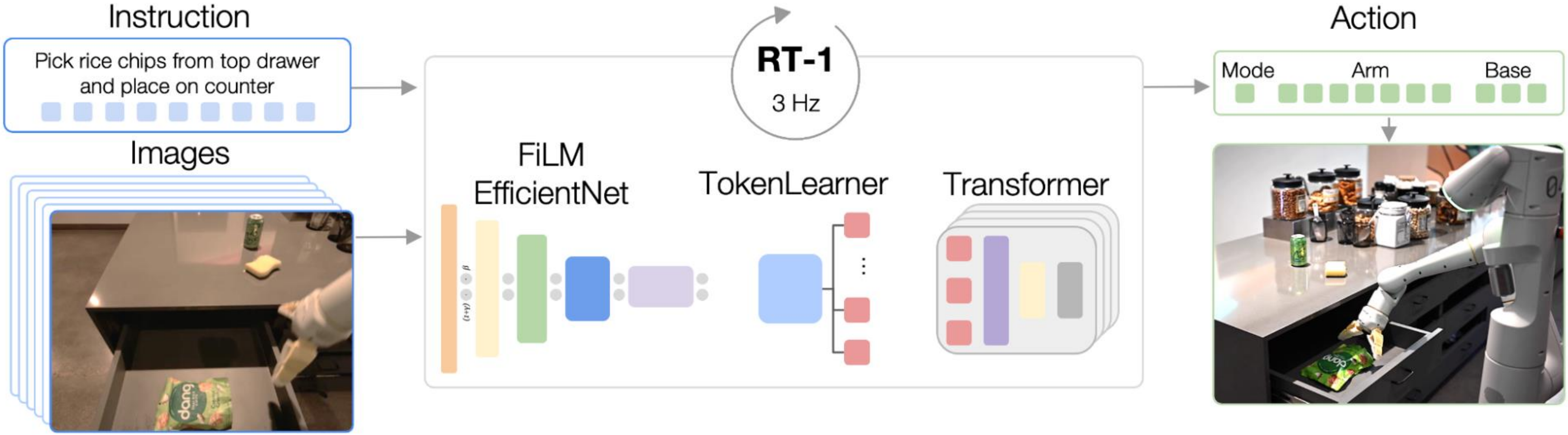
Voila!

Sequential decision making

- Offline RL, Behavioral Cloning, goal conditioned RL, etc…



Robotics



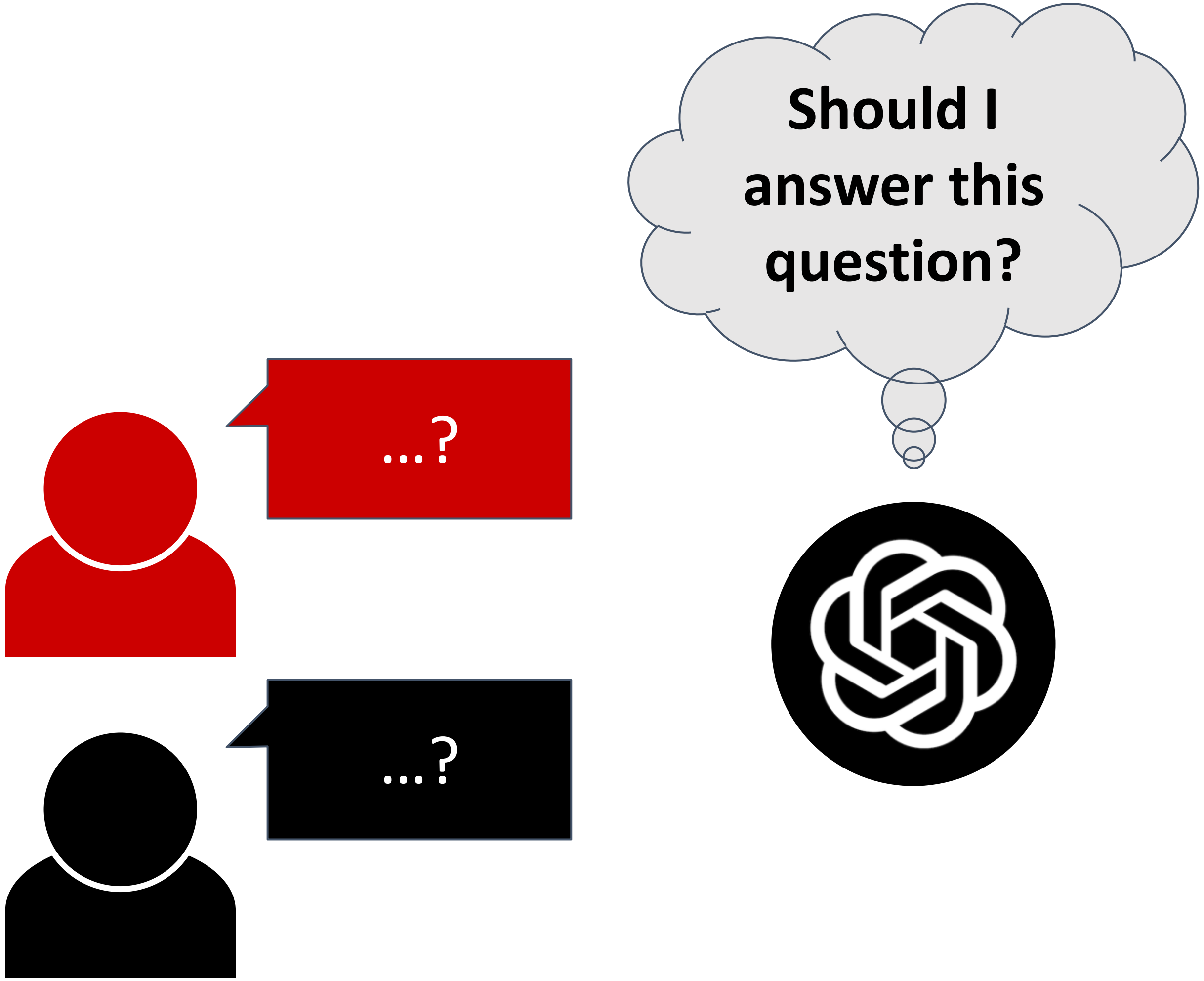
https://research.google/blog/rt-1-robotics-transformer-for-real-world-control-at-scale/

Helpful vs. Harmless

- RLHF attempts to train models that **carefully walk the line between helpful and harmless.**

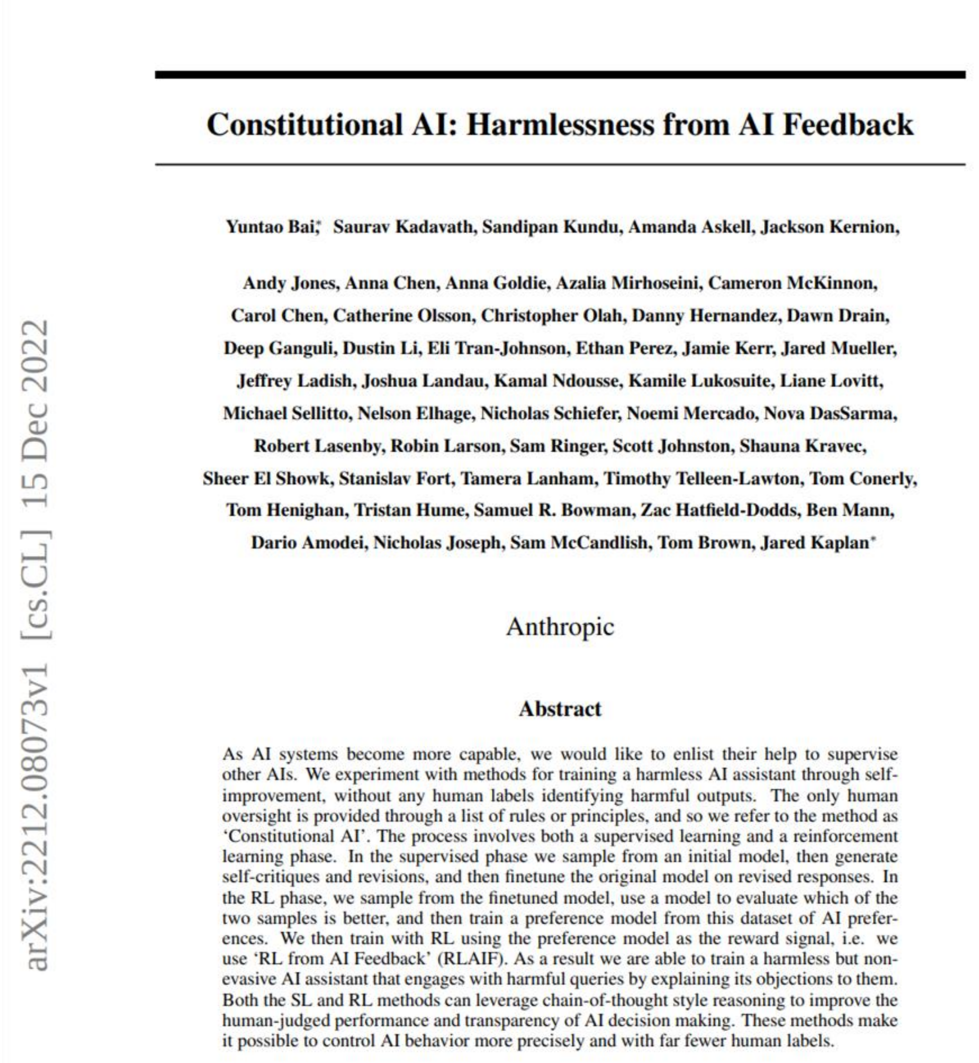
- Over-Optimization and reward misidentification can result in being **too harmless and not helpful.**

- Still largely an open problem for how to balance this!



Bai et al. (2022)

Constitutional AI



Constitutional AI: Harmlessness from AI Feedback

Yuntao Bai*, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion,

Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon,
Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain,
Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller,
Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt,
Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma,
Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec,
Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly,
Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann,
Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, Jared Kaplan*

Anthropic

Abstract

As AI systems become more capable, we would like to enlist their help to supervise other AIs. We experiment with methods for training a harmless AI assistant through self-improvement, without any human labels identifying harmful outputs. The only human oversight is provided through a list of rules or principles, and so we refer to the method as 'Constitutional AI'. The process involves both a supervised learning and a reinforcement learning phase. In the supervised phase we sample from an initial model, then generate self-critiques and revisions, and then finetune the original model on revised responses. In the RL phase, we sample from the finetuned model, use a model to evaluate which of the two samples is better, and then train a preference model from this dataset of AI preferences. We then train with RL using the preference model as the reward signal, i.e. we use 'RL from AI Feedback' (RLAIF). As a result we are able to train a harmless but non-evasive AI assistant that engages with harmful queries by explaining its objections to them. Both the SL and RL methods can leverage chain-of-thought style reasoning to improve the human-judged performance and transparency of AI decision making. These methods make it possible to control AI behavior more precisely and with far fewer human labels.

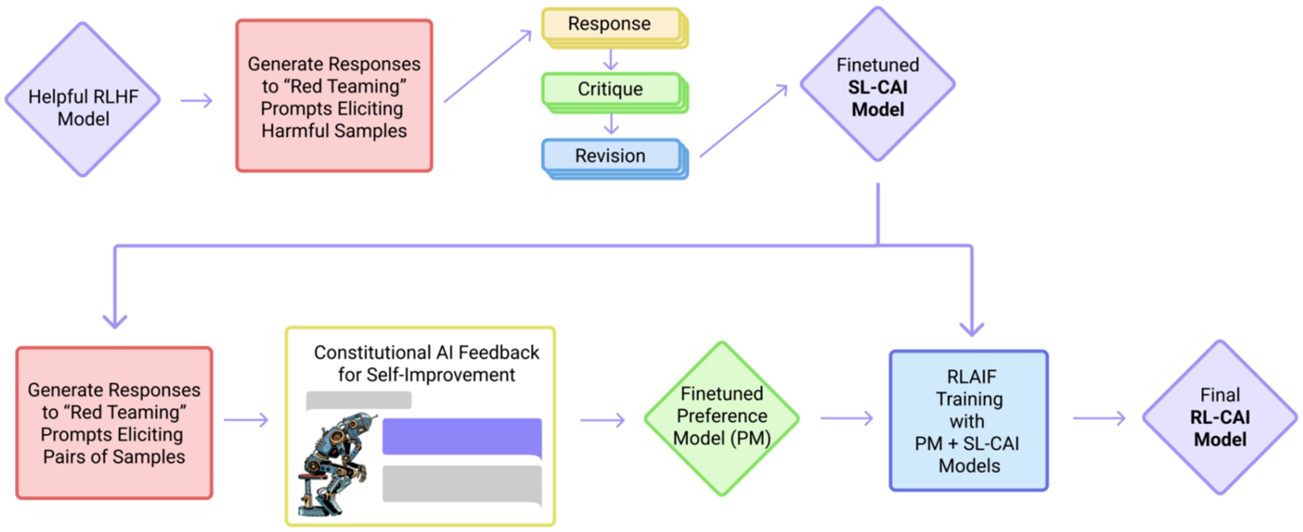

Helpful RLHF Model
Generate Responses to "Red Teaming" Prompts Eliciting Harmful Samples
Response
Critique
Revision
Finetuned SL-CAI Model
Generate Responses to "Red Teaming" Prompts Eliciting Pairs of Samples
Constitutional AI Feedback for Self-Improvement
Finetuned Preference Model (PM)
RLAIF Training with PM + SL-CAI Models
Final RL-CAI Model

Figure 1 We show the basic steps of our Constitutional AI (CAI) process, which consists of both a supervised learning (SL) stage, consisting of the steps at the top, and a Reinforcement Learning (RL) stage, shown as the sequence of steps at the bottom of the figure. Both the critiques and the AI feedback are steered by a small set of principles drawn from a 'constitution'. The supervised stage significantly improves the initial model, and gives some control over the initial behavior at the start of the RL phase, addressing potential exploration problems. The RL stage significantly improves performance and reliability.

Bai et al. (2022)

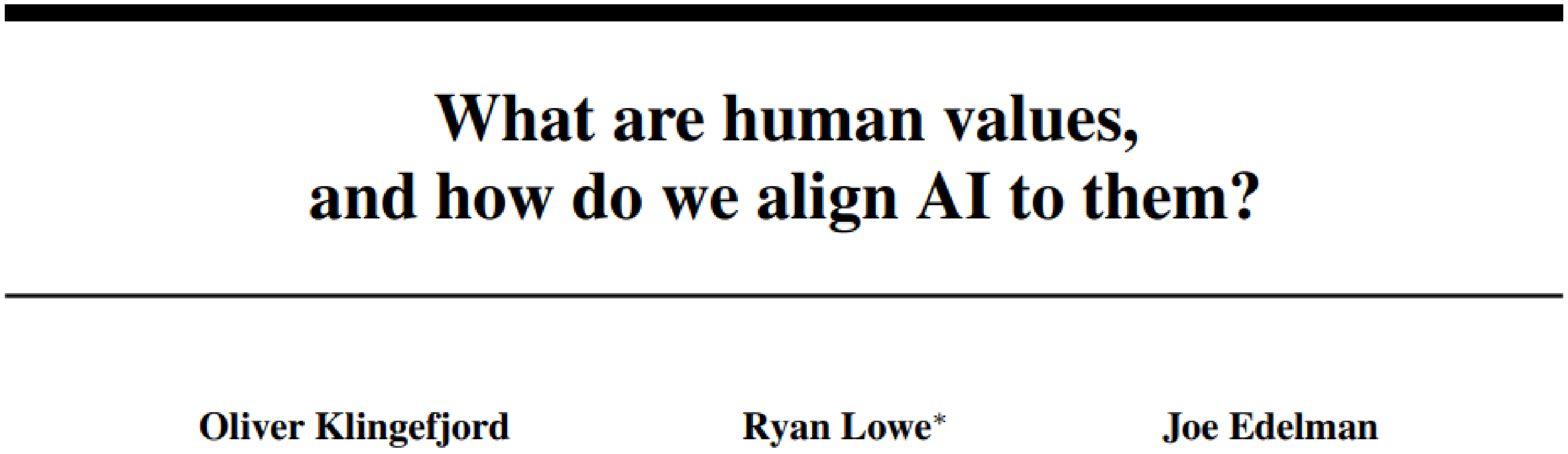
Belief Distribution and Human Noise

If the reward model was trained from human input, *is the resulting reward representation biased/skewed?*

- Disagreement between human preferences occurs within datasets.
  - Ziegler et al: **60% label agreement.**
  - Stiennon et al: **72% label agreement.**
- How should we account for difference in preferences and opinions?

Ziegler et al. (2020), Stiennon et al. (2022), Klingefjord et al. (2024) [Image Credit: FreePik]

Belief Distribution and Human Noise

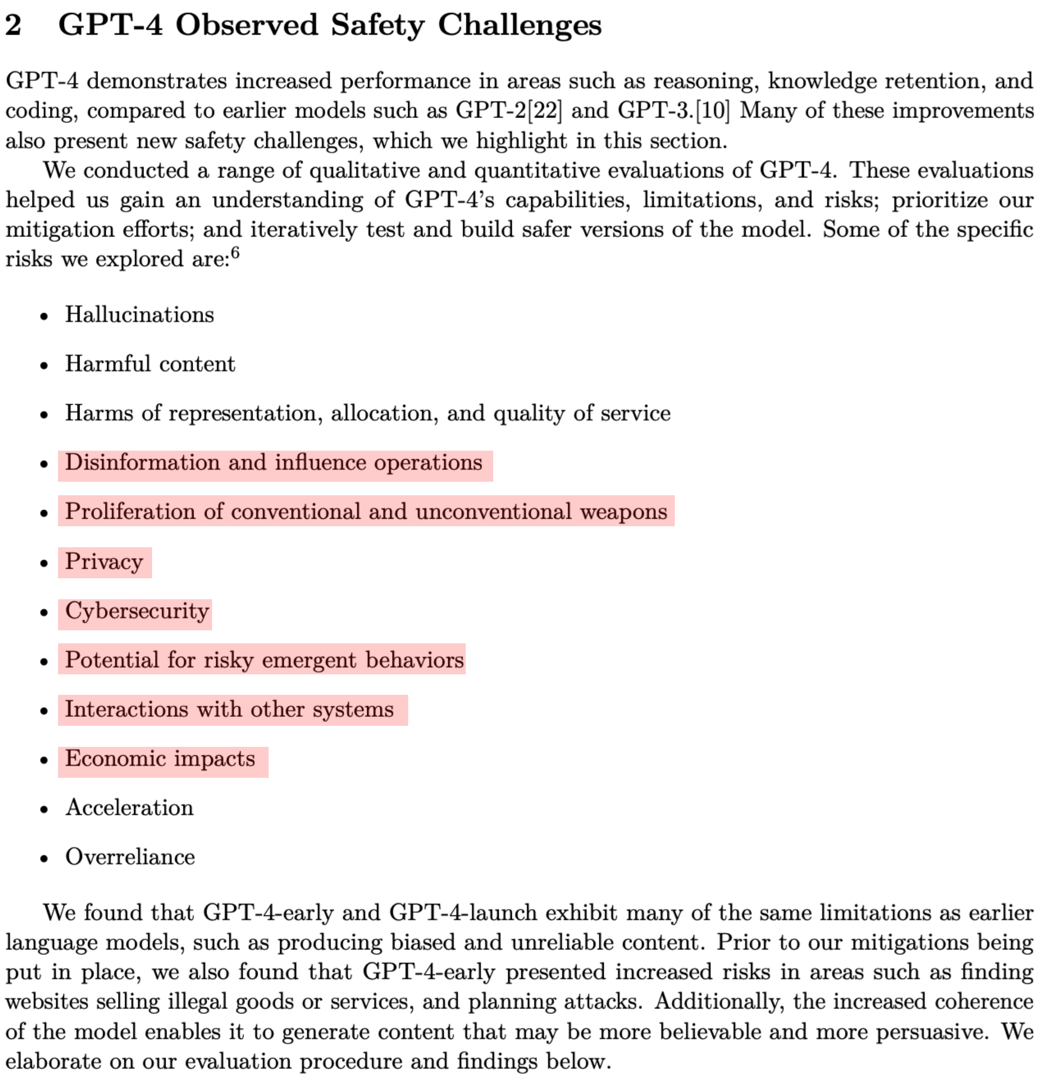


What are human values, and how do we align AI to them?

Oliver Klingefjord   Ryan Lowe*   Joe Edelman

Klingefjord et al. (2024)

OpenAI Challenges



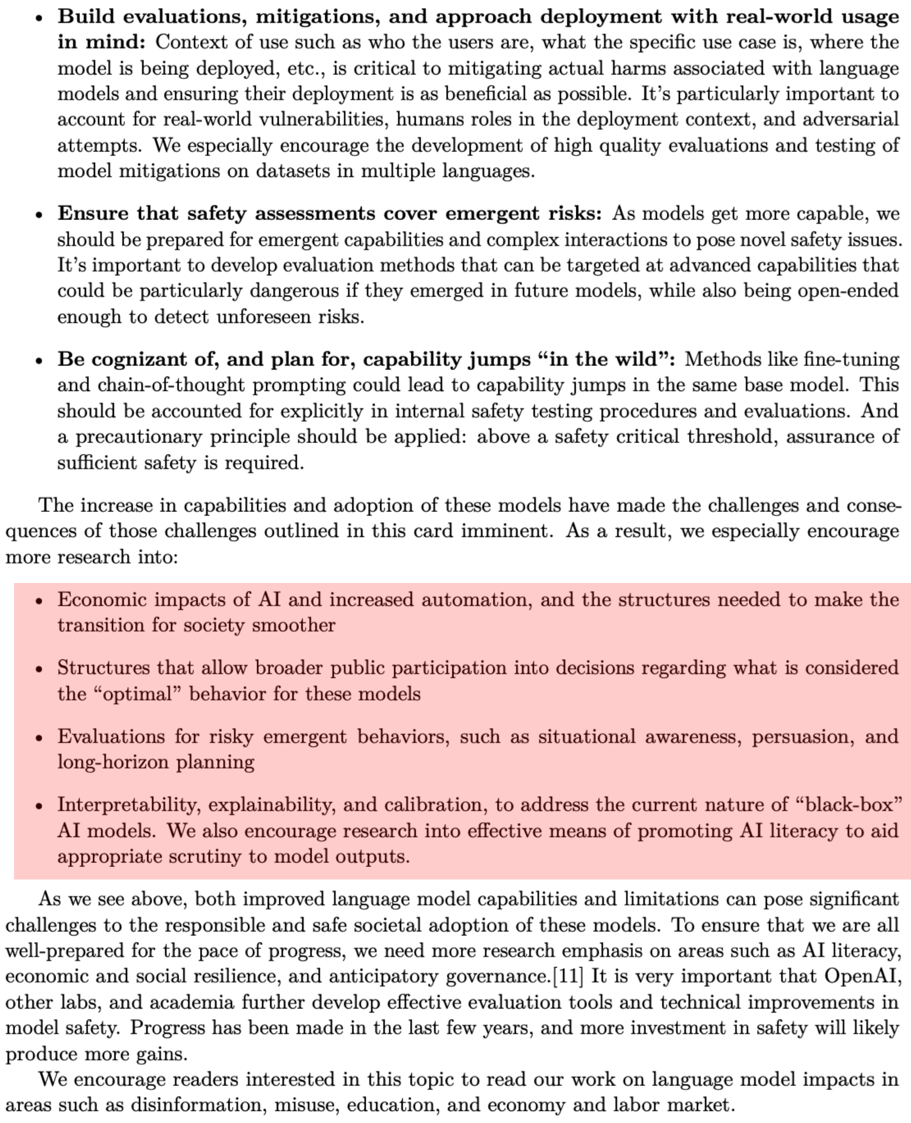
2 GPT-4 Observed Safety Challenges

GPT-4 demonstrates increased performance in areas such as reasoning, knowledge retention, and coding, compared to earlier models such as GPT-2[22] and GPT-3.[10] Many of these improvements also present new safety challenges, which we highlight in this section.

We conducted a range of qualitative and quantitative evaluations of GPT-4. These evaluations helped us gain an understanding of GPT-4's capabilities, limitations, and risks; prioritize our mitigation efforts; and iteratively test and build safer versions of the model. Some of the specific risks we explored are:[6]

- Hallucinations
- Harmful content
- Harms of representation, allocation, and quality of service
- Disinformation and influence operations
- Proliferation of conventional and unconventional weapons
- Privacy
- Cybersecurity
- Potential for risky emergent behaviors
- Interactions with other systems
- Economic impacts
- Acceleration
- Overreliance

We found that GPT-4-early and GPT-4-launch exhibit many of the same limitations as earlier language models, such as producing biased and unreliable content. Prior to our mitigations being put in place, we also found that GPT-4-early presented increased risks in areas such as finding websites selling illegal goods or services, and planning attacks. Additionally, the increased coherence of the model enables it to generate content that may be more believable and more persuasive. We elaborate on our evaluation procedure and findings below.



- **Build evaluations, mitigations, and approach deployment with real-world usage in mind:** Context of use such as who the users are, what the specific use case is, where the model is being deployed, etc., is critical to mitigating actual harms associated with language models and ensuring their deployment is as beneficial as possible. It's particularly important to account for real-world vulnerabilities, humans roles in the deployment context, and adversarial attempts. We especially encourage the development of high quality evaluations and testing of model mitigations on datasets in multiple languages.
- **Ensure that safety assessments cover emergent risks:** As models get more capable, we should be prepared for emergent capabilities and complex interactions to pose novel safety issues. It's important to develop evaluation methods that can be targeted at advanced capabilities that could be particularly dangerous if they emerged in future models, while also being open-ended enough to detect unforeseen risks.
- **Be cognizant of, and plan for, capability jumps "in the wild":** Methods like fine-tuning and chain-of-thought prompting could lead to capability jumps in the same base model. This should be accounted for explicitly in internal safety testing procedures and evaluations. And a precautionary principle should be applied: above a safety critical threshold, assurance of sufficient safety is required.

The increase in capabilities and adoption of these models have made the challenges and consequences of those challenges outlined in this card imminent. As a result, we especially encourage more research into:

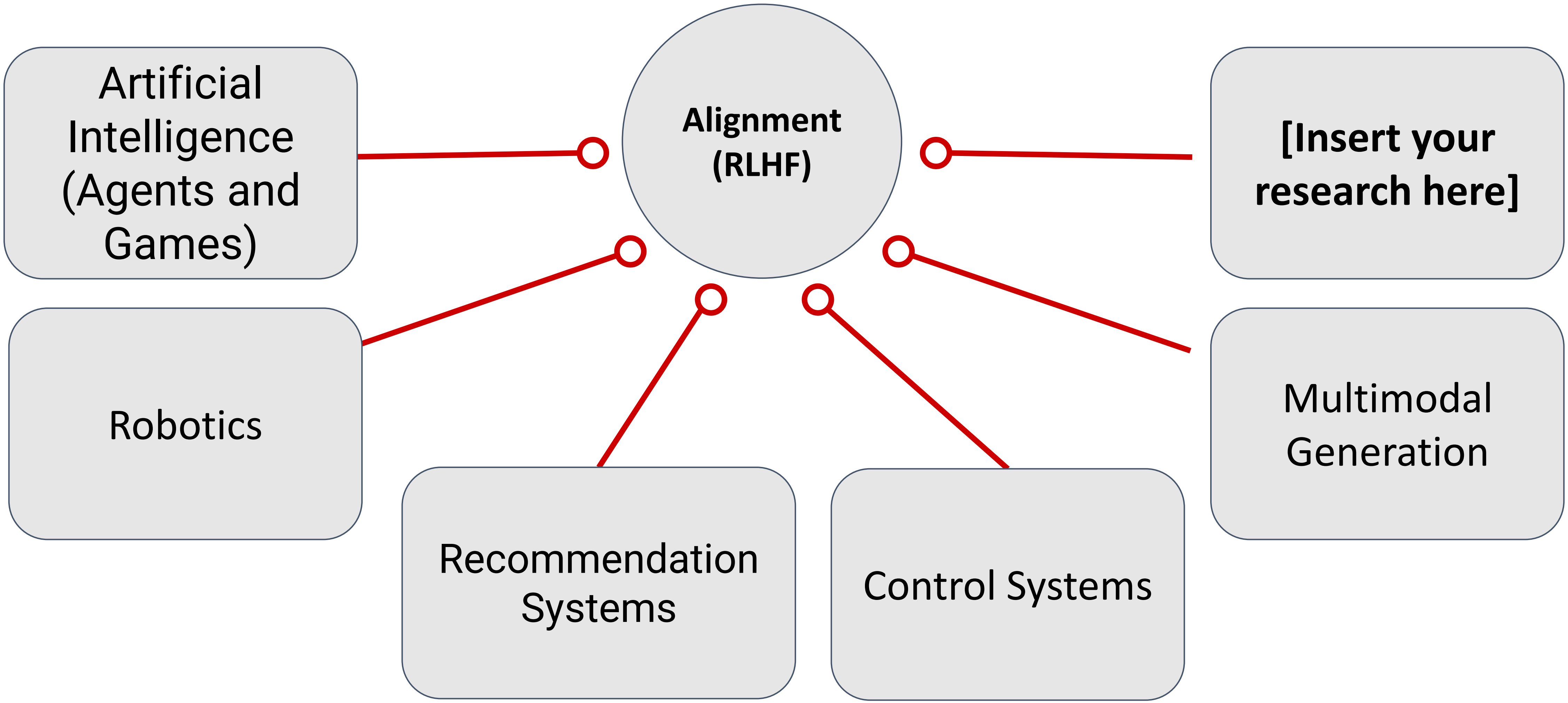
- Economic impacts of AI and increased automation, and the structures needed to make the transition for society smoother
- Structures that allow broader public participation into decisions regarding what is considered the "optimal" behavior for these models
- Evaluations for risky emergent behaviors, such as situational awareness, persuasion, and long-horizon planning
- Interpretability, explainability, and calibration, to address the current nature of "black-box" AI models. We also encourage research into effective means of promoting AI literacy to aid appropriate scrutiny to model outputs.

As we see above, both improved language model capabilities and limitations can pose significant challenges to the responsible and safe societal adoption of these models. To ensure that we are all well-prepared for the pace of progress, we need more research emphasis on areas such as AI literacy, economic and social resilience, and anticipatory governance.[11] It is very important that OpenAI, other labs, and academia further develop effective evaluation tools and technical improvements in model safety. Progress has been made in the last few years, and more investment in safety will likely produce more gains.

We encourage readers interested in this topic to read our work on language model impacts in areas such as disinformation, misuse, education, and economy and labor market.

OpenAI GPT-4 Technical Report

RLHF is studied in <u>many</u> areas of research…



Alignment (RLHF)
Artificial Intelligence (Agents and Games)
Robotics
Recommendation Systems
Control Systems
Multimodal Generation
[Insert your research here]