Reinforcement Learning from Human Feedback

Session 4. Intro to LLMs - Reading Group
CUNEF & ICMAT - May 2024

Víctor Gallego

Outline

1. **Introduction & history**
2. Core technical overview
3. Emerging directions

Motivation

Language models (LMs) can be powerful but also troublesome:

- They don’t understand how humans want to use them.
- They have **no notion of human intent**.
- Or worse, they have the **wrong values**.

Can we imbue human preferences?

- If only we could provide the language models with human feedback
  - Maybe they'd understand what we want
- How do we tell them what we want?
- Can we maybe annotate their outputs and have the LM learn from our annotations?
  - Annotate the language model generations of "bad behavior" as negative and "preferred behaviors" as positive

Reinforcement Learning appears…

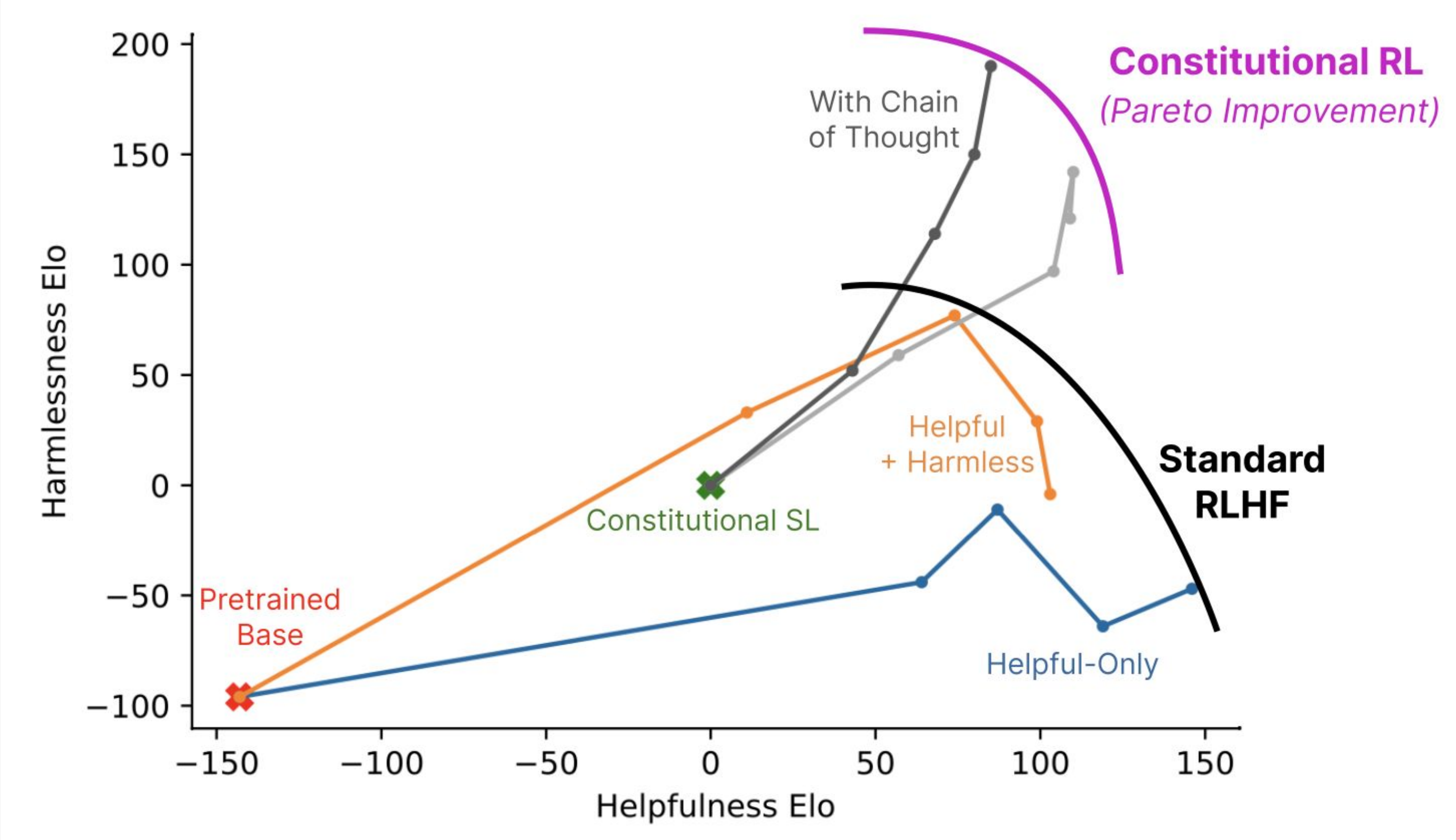
- If we have sparse rewards over what the language model should and should not do this seems like a good case for RL:

The movie was awesome, amazing cast	⟶	+1
The movie: don’t watch it, it’s completely shit	⟶	- 1

- Key point: **For many tasks we want to solve, evaluation of outcomes is easier than producing the correct behavior**
  - Helpfulness in assistants
  - Safety content/moderation

RLHF is relied upon

RLHF is a key factor in many popular models, both on and off the record, including ChatGPT, Bard, Claude, Llama 2, and more



Bai, Y. et al. “Constitutional AI: Harmlessness from AI Feedback.” 2023.

“Meanwhile reinforcement learning, known for its instability, seemed a somewhat shadowy field for those in the NLP research community. However, reinforcement learning proved highly effective, particularly given its cost and time effectiveness.”

- Touvron, H. et al. “ Llama 2: Open Foundation and Fine-Tuned Chat Models.” 2023

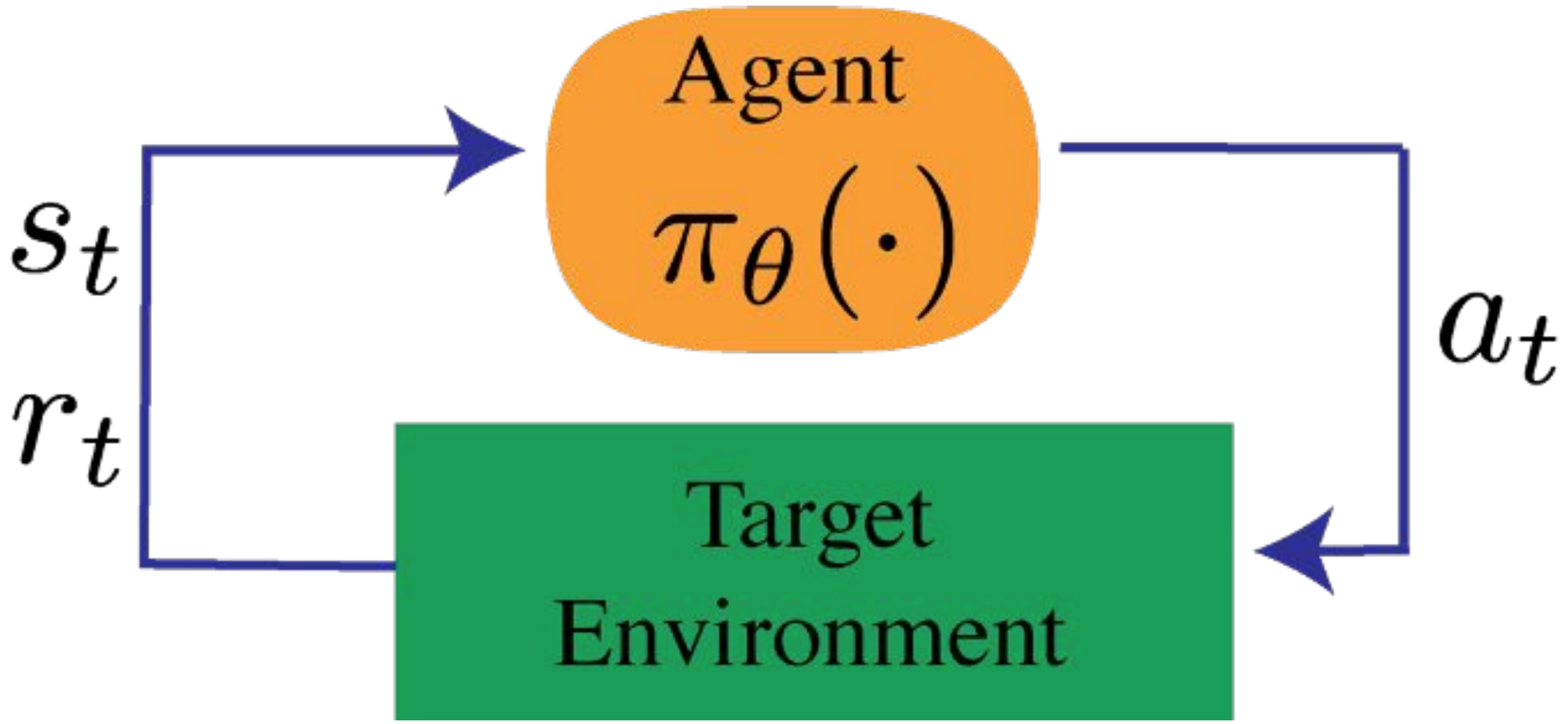
Presumptions of RLHF

1. **“RL works”** Optimal solutions can be achieved with finite data in complex environments.
2. **“Von Neumann-Morgenstern (VNM) utility theorem”** Foundation of Utilitarianism. Compare, aggregate, and compute preferences.
3. **“Bradley-Terry model”** Pairwise preferences can suitably perform as a basis of human values.
4. **“Aggregation of preferences”** Multiple user preferences are successfully represented in training one model by aggregating and comparing individual utilities.
5. **“RLHF independence - data”** The only preferences embedded in the model are from the specifically collected training data.
6. **“RLHF independence - training”** User preferences are extracted uniformly via the RLHF process.

Lambert et al. The History and Risks of Reinforcement Learning and Human Feedback, 2023.


Assumption 2
Quantification of preferences
Hedonic Calculus
Port Royal Logic
Assumption 3
Increasing reward
Operant Conditioning
Utility-to-go
Presumption 1
Optimality with approximate methods
Temporal Difference Learning
TD-Gammon
Q-Learning
DQN for Atari
InstructGPT
ChatGPT
Claude
Presumptions 4, 5, 6
Aggregation of preferences
Preferences from data
Preferences from RL
Early
1662
1823
1938
1947
1952
1957
1960
1988
1992
1995
2013
2017
2020
2022
Von Neumann-Morgenstern (VNM) utility theorem
Presumption 2
Time and context independent preferences
Markov Decision Process
Bellman Equation
Assumption 4
Optimal solutions exist
Learning to Summarize
Aristole, Plato, Epicurus...
Assumption 1
Haman preferences exist
Bradley-Terry model
Presumption 3
Pairwise preferences
Christiano et al. 2017
Deep RL from Human Preferences

Review: Reinforcement Learning basics

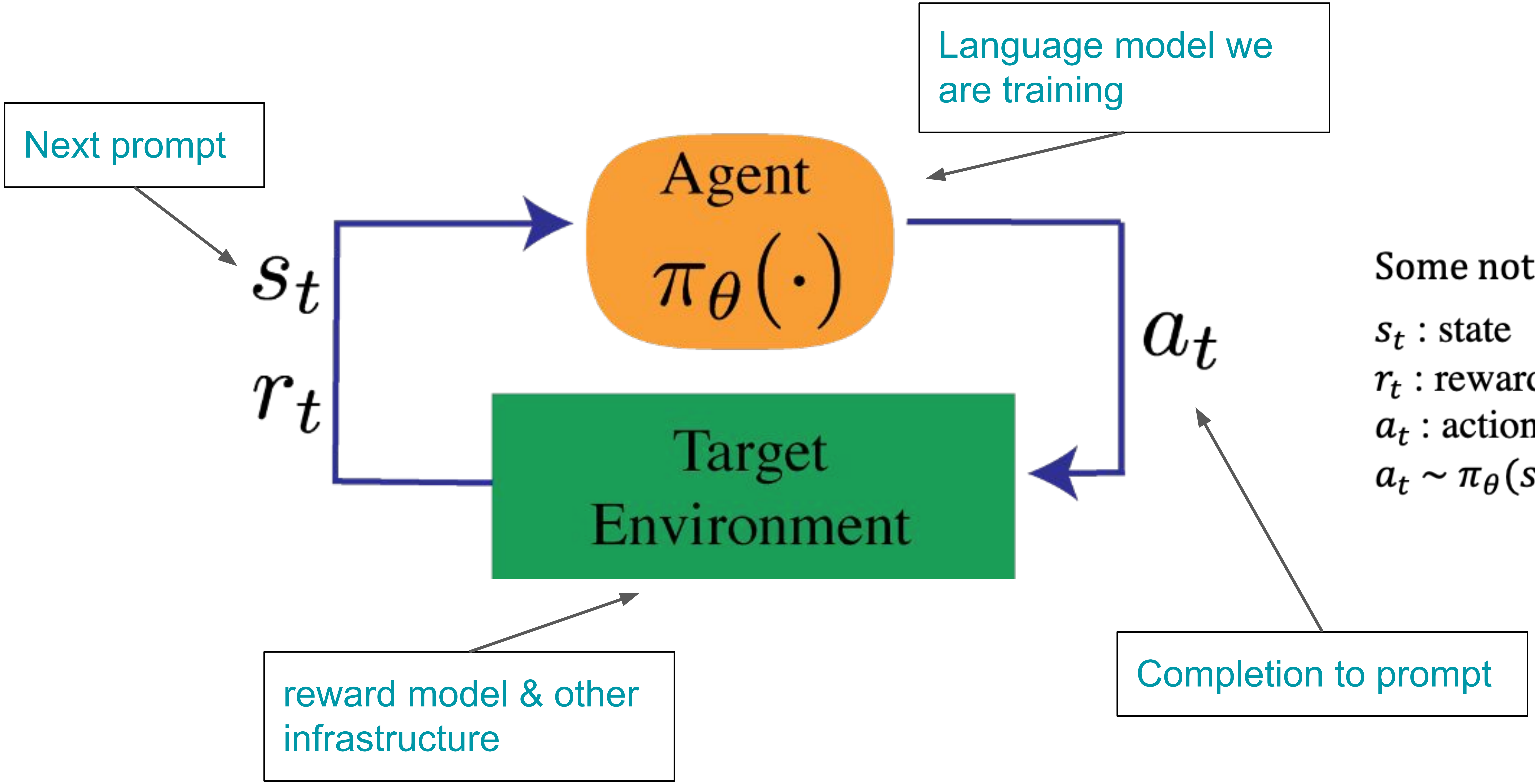


Some notation:

s_t : state
r_t : reward
a_t : action
$a_t \sim \pi_\theta(s_t)$: policy

Goal: learn policy to maximize rewards

Review: Reinforcement Learning basics in language

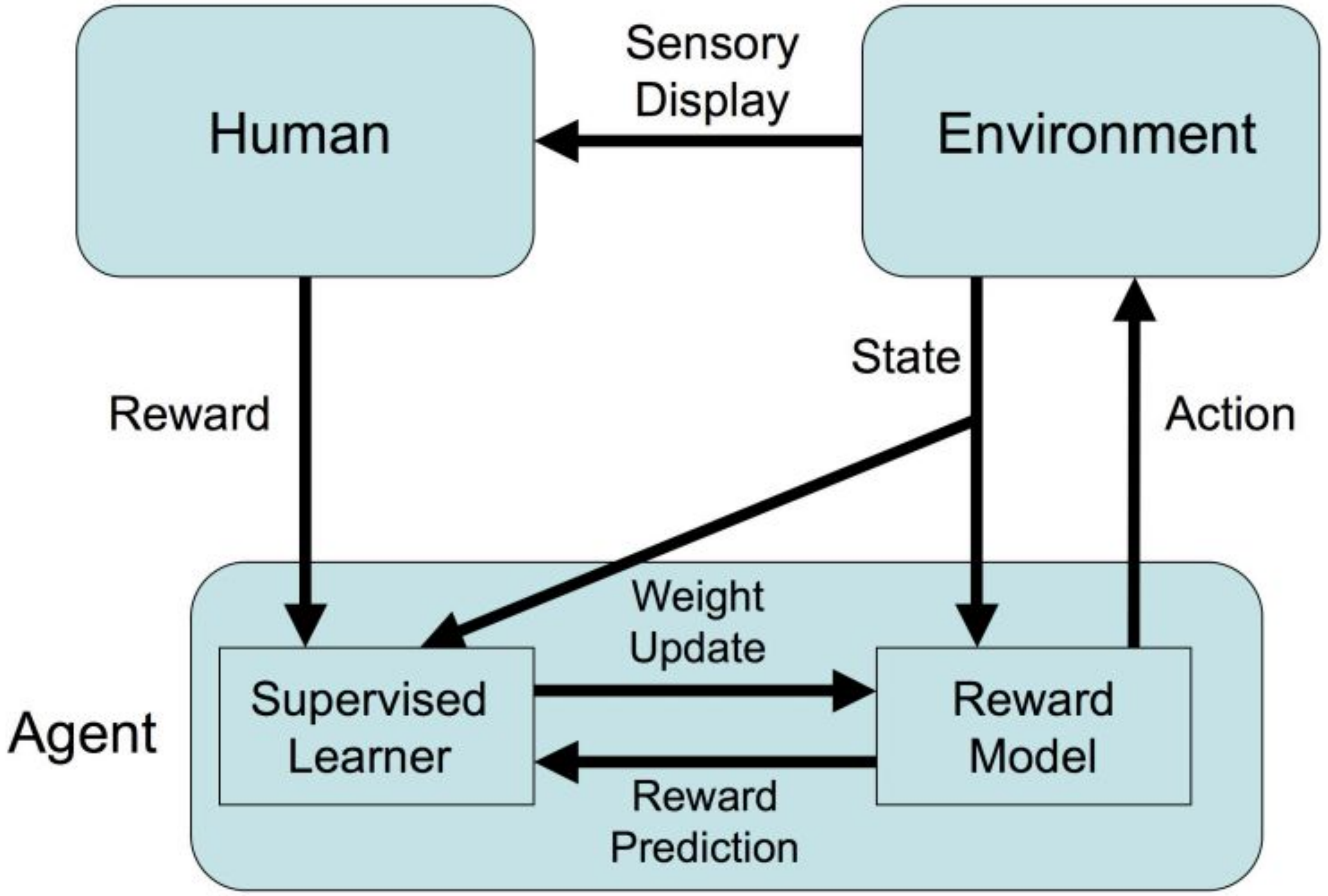


Some notation:

s_t : state
r_t : reward
a_t : action
$a_t \sim \pi_\theta(s_t)$: policy

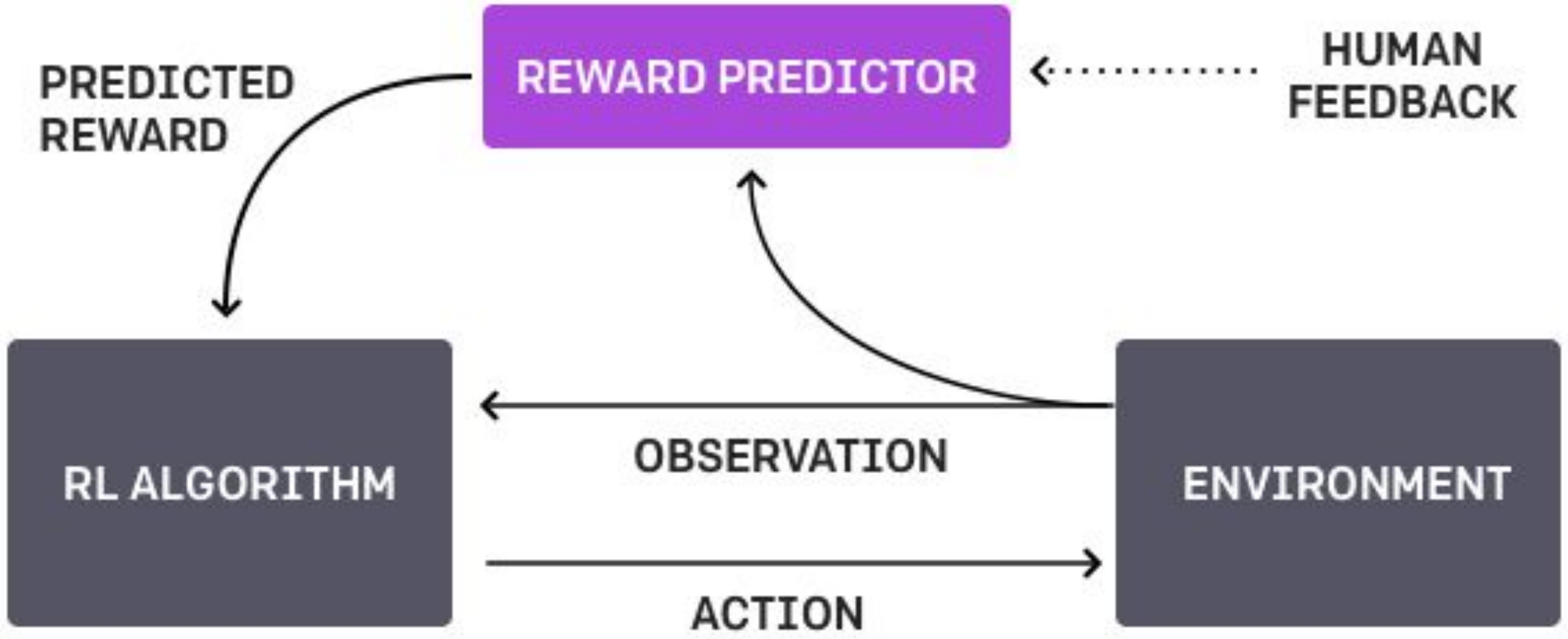
History: RLHF for decision making

Pre Deep RL
Human provides scalar score



Knox, W. Bradley, and Peter Stone. "Tamer: Training an agent manually via evaluative reinforcement." *2008.*

With Deep RL
Human compares trajectories



Christiano, Paul F., et al. "Deep reinforcement learning from human preferences." 2017.

History: preference models, alignment, and agents
Sep. 2019

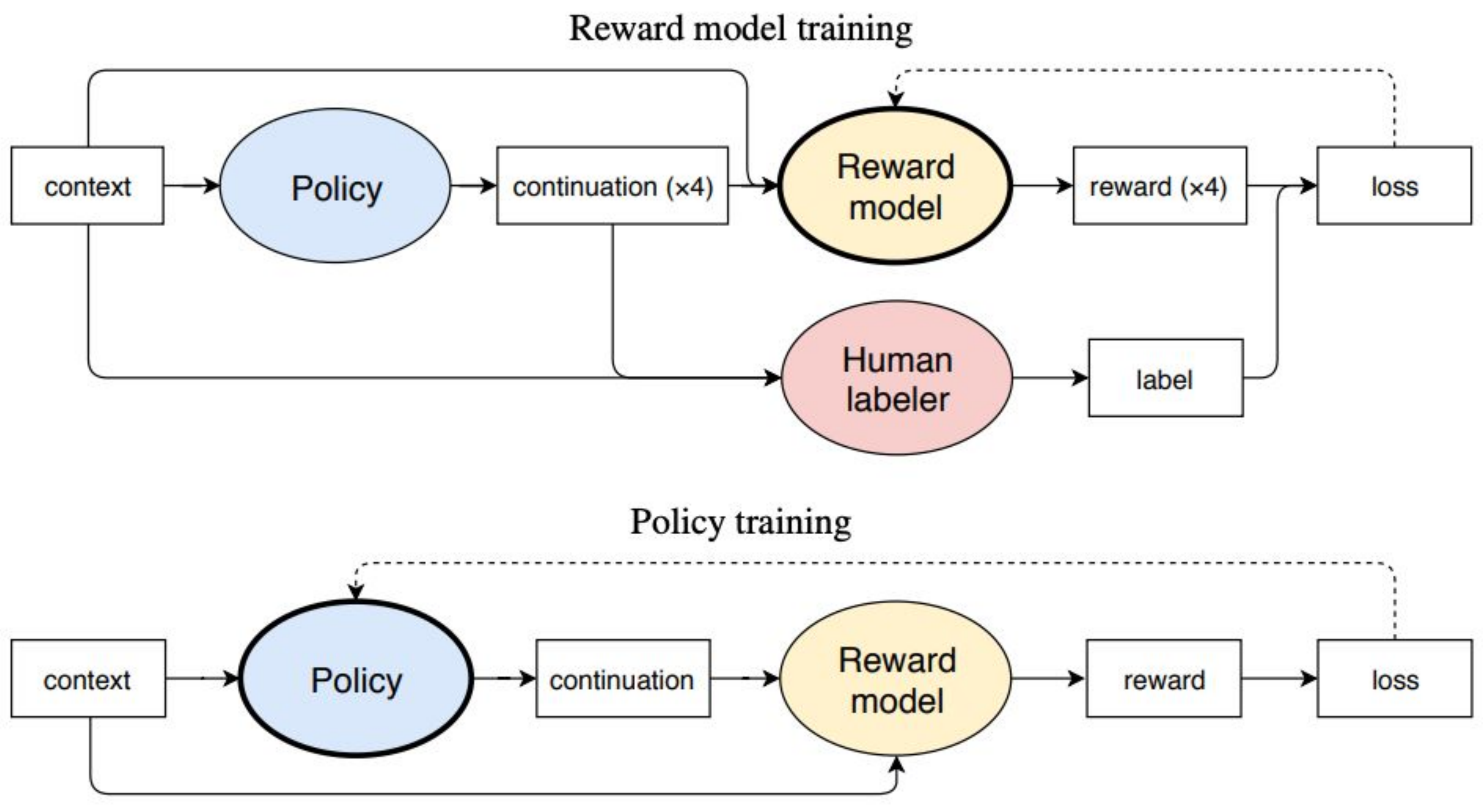


Figure 1: Our training processes for reward model and policy. In the online case, the processes are interleaved.

- Can learn from binary preference data
- Can optimize from sentence classifiers
- RLHF substantially changes how LLMs generate text

Ziegler, Daniel M., et al. "Fine-tuning language models from human preferences." *arXiv preprint arXiv:1909.08593* (2019).

History: early OpenAI experiments with RLHF (InstructGPT) Sep. 2020

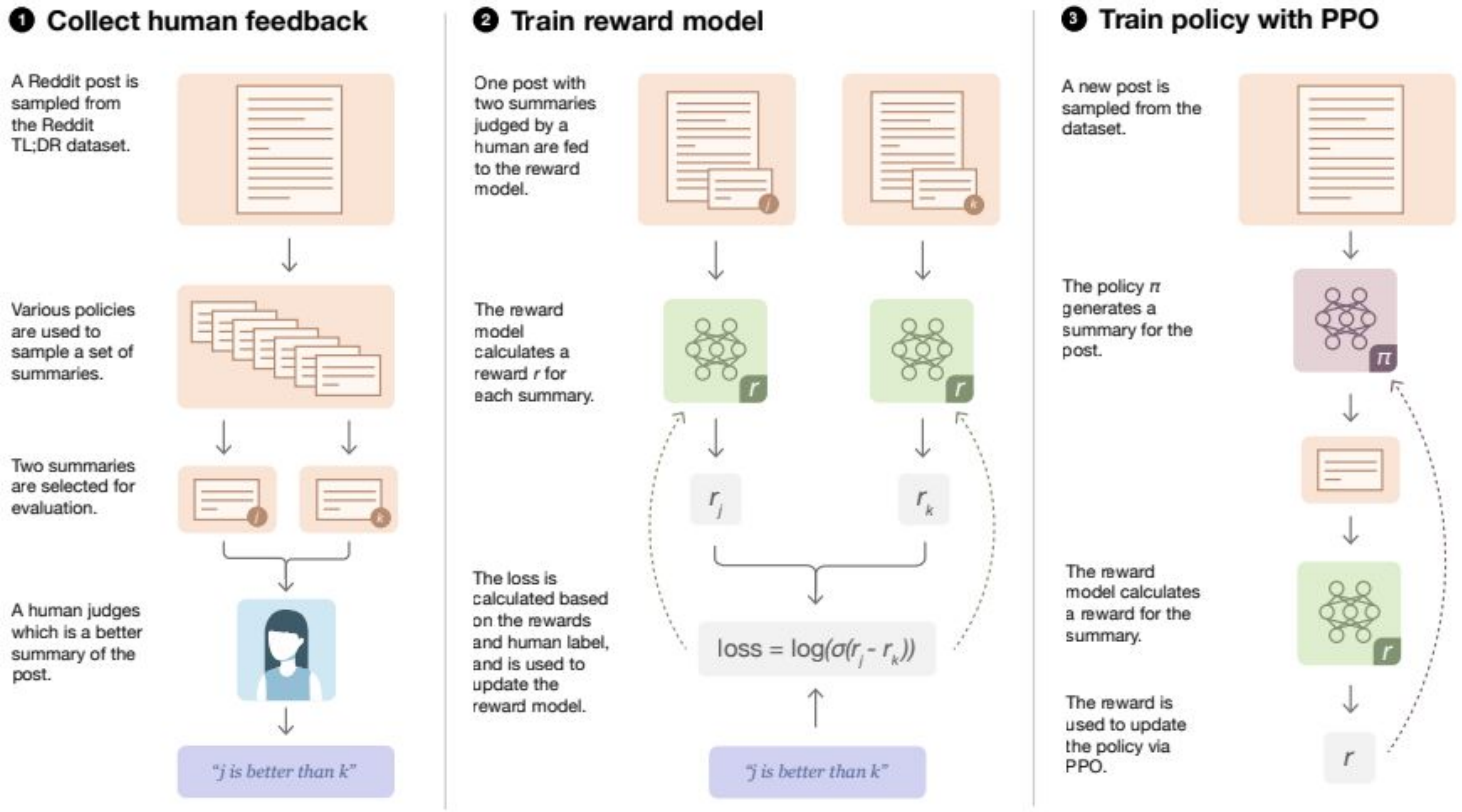


Figure 2: Diagram of our human feedback, reward model training, and policy training procedure.

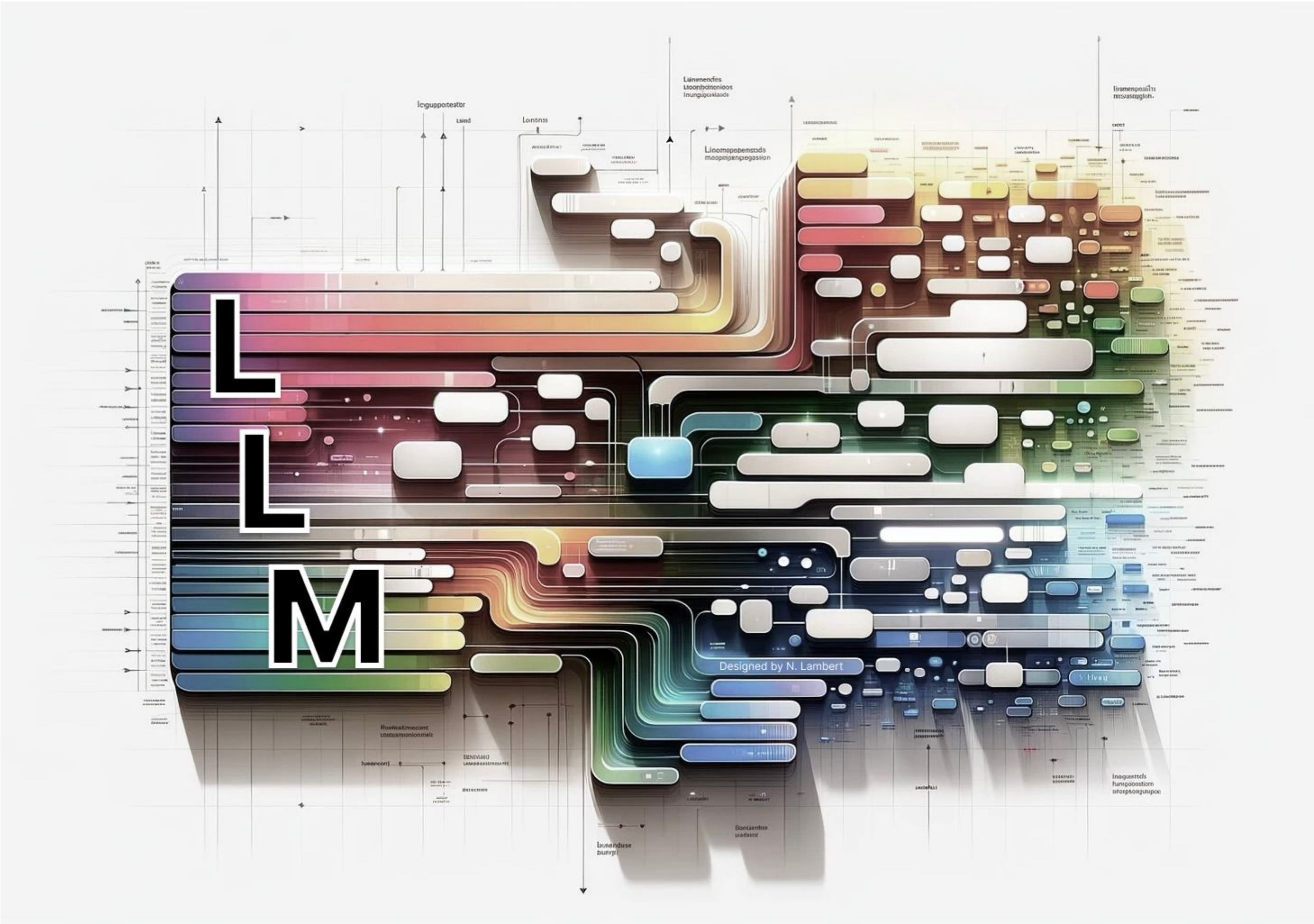
Stiennon, Nisan, et al. "Learning to summarize with human feedback." *2020.*

Outline

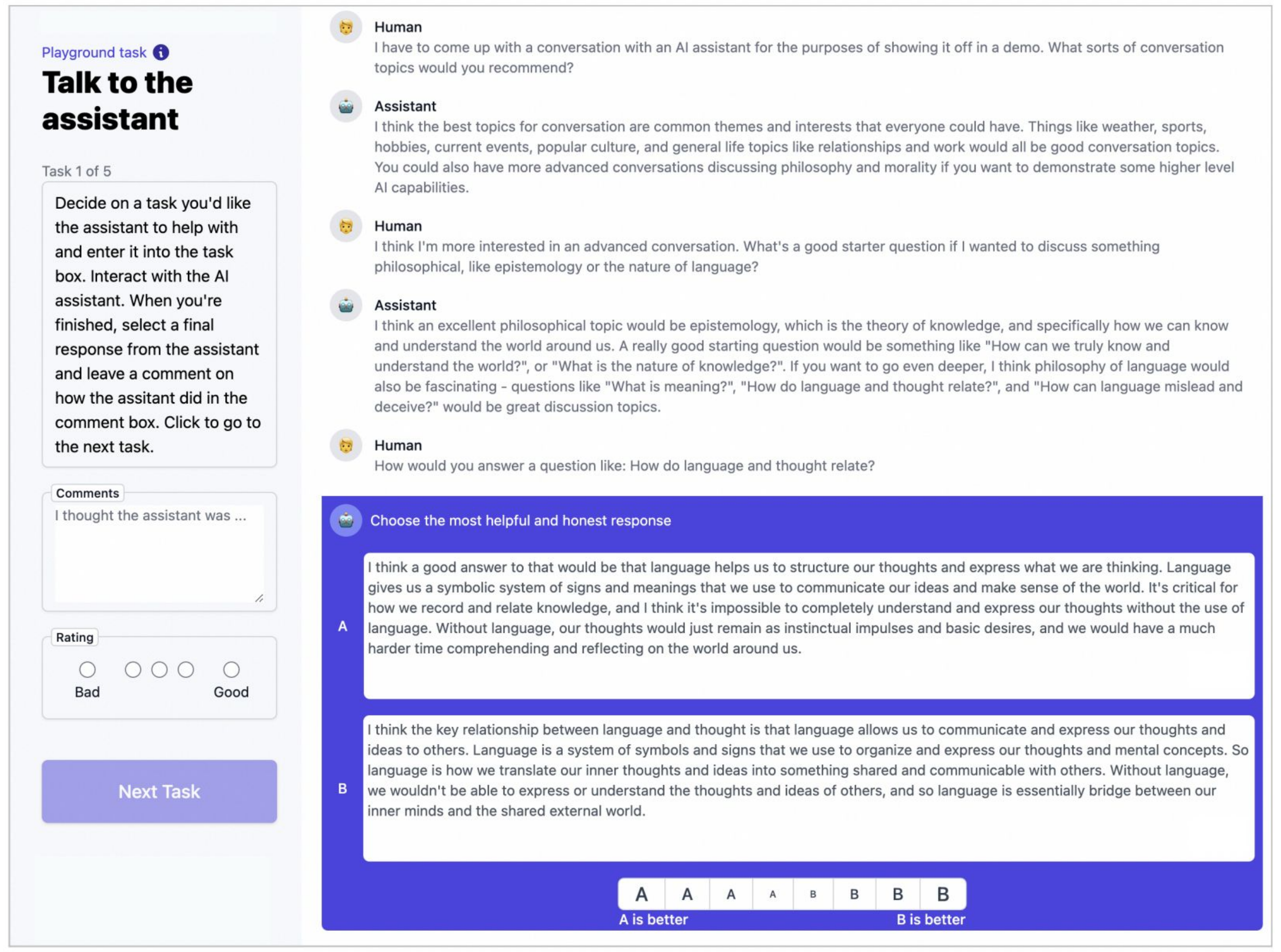
1. Introduction & history
2. **Core technical overview**
3. Emerging directions

Three phases of RLHF

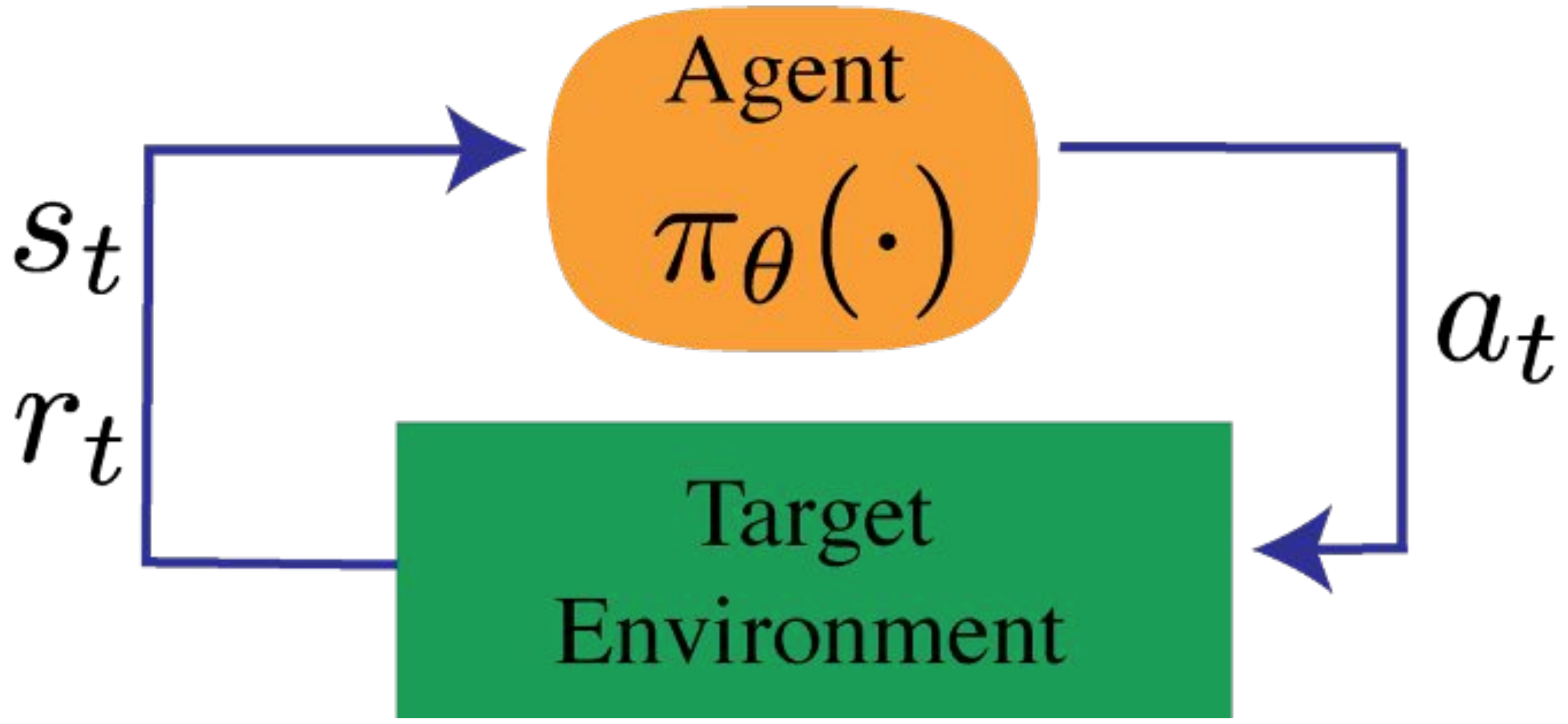
base SFT model (instruction, helpful, chatty etc.)



preference collection



reinforcement learning optimization



RLHF objective

π_{ref}: original LLM
π_θ: trained LLM
x: prompt
y: completion

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

RLHF objective

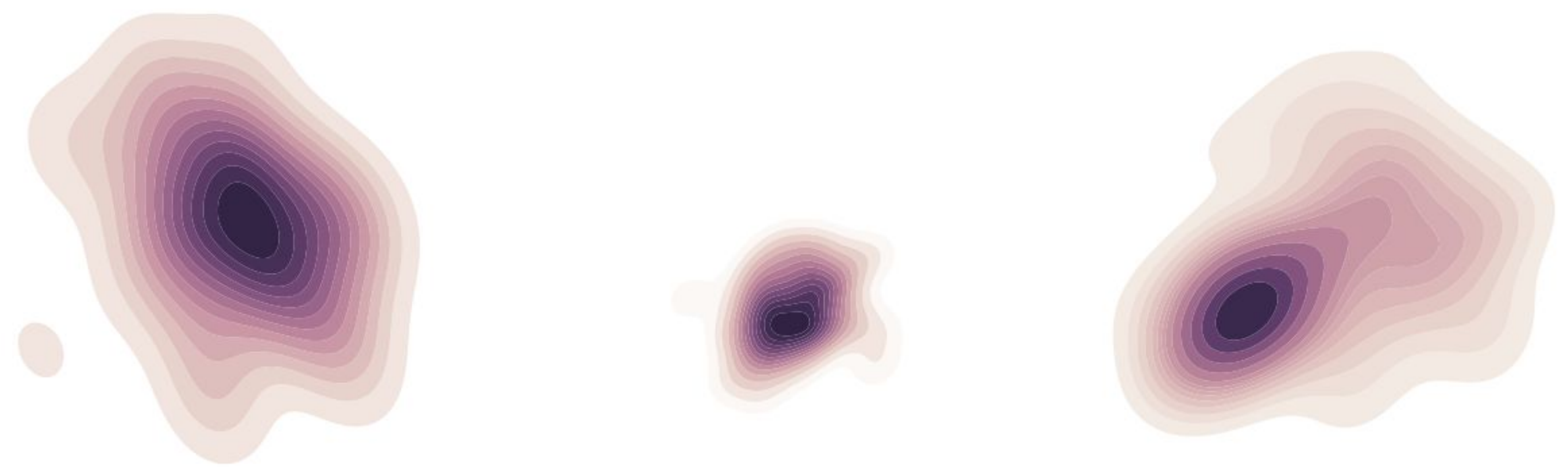
π_{ref}: original LLM
π_θ: trained LLM
x: prompt
y: completion

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

Optimize "reward" *inspired* ▲ by human preferences

▲ Constrain the model to not trust the reward too much (preferences are hard to model)

RL with KL is better seen as Bayesian Inference



Prior: original LM

$\pi_0(x)$

Evidence: reward model

$\exp(r(x))$

Posterior: aligned LM

$\pi^*(x) \propto \pi_0(x) \exp(r(x))$

Proof in https://arxiv.org/abs/2205.11275

RL with KL penalties is variational inference

Maximising reward while staying close to original LM

$$\mathbb{E}_{x \sim \pi_\theta}[r(x)] - \beta \mathrm{KL}(\pi_\theta, \pi_0)$$

Minimising divergence from the posterior

$$\mathrm{KL}(\pi_\theta, \pi^*)$$

RLHF objective

π_{ref}: original LLM
π_θ: trained LLM
x: prompt
y: completion

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

Optimize "reward" *inspired* ▲ by human preferences

▲ Constrain the model to not trust the reward too much (preferences are hard to model)

Two decisions:

1. How to define reward model: *r(x,y)*
2. How to optimize the objective function

Preference model: design a "human" reward

- Assigning a scalar reward of how good a response is did not work in early work
- Pairwise preferences are easy to collect and based in theory that can become a reward

Key idea:

Probability $\propto$ reward

Chosen completion

Prompt

Score from optimal reward model

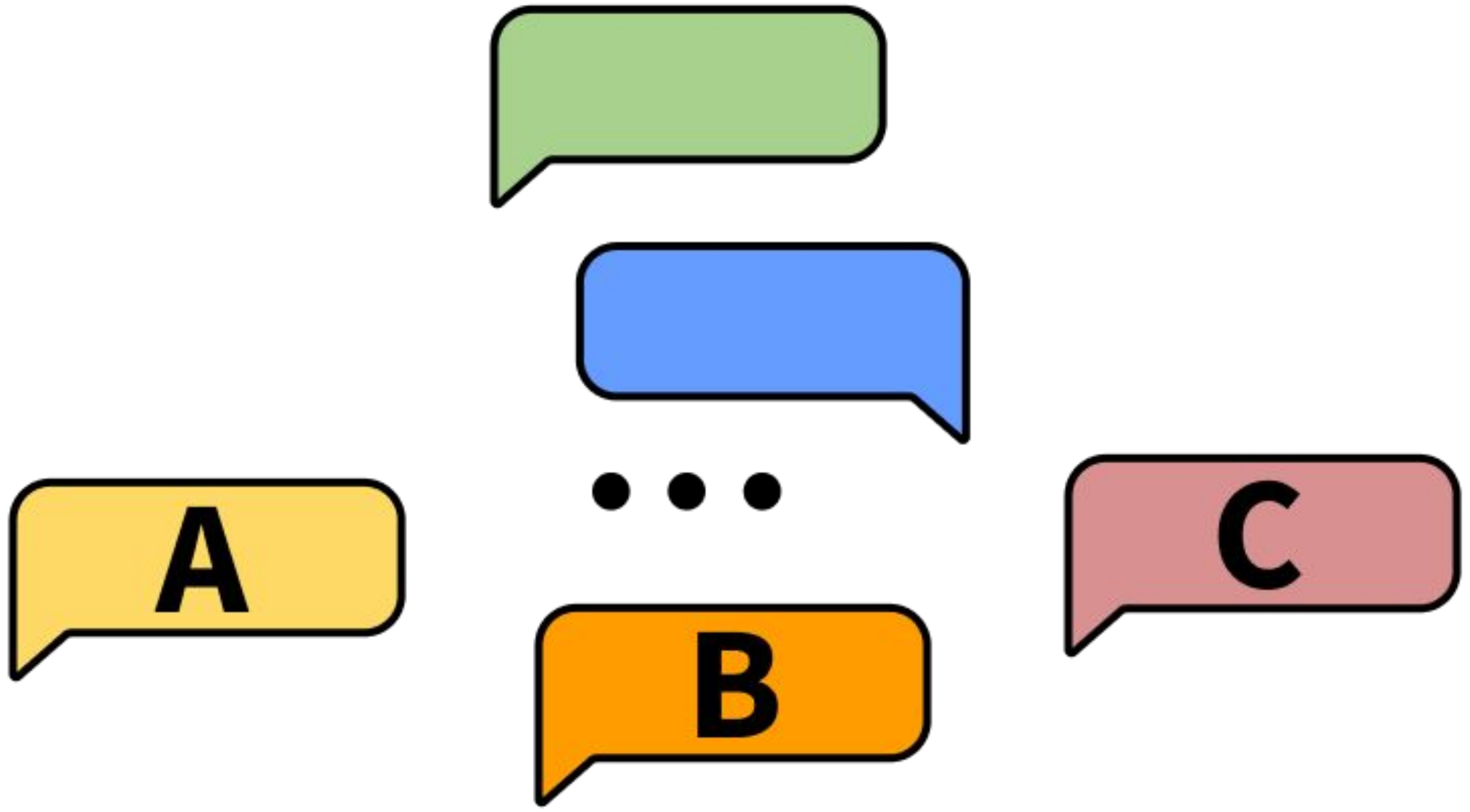
$$p^*(y_1 \succ y_2 \mid x) = \frac{\exp\left(r^*(x, y_1)\right)}{\exp\left(r^*(x, y_1)\right) + \exp\left(r^*(x, y_2)\right)}$$

Rejected completion

Bradley Terry model:
Estimate probability that a given pairwise preference is true

Collecting the data: feedback interfaces

task: choose the better **next message** in a conversation

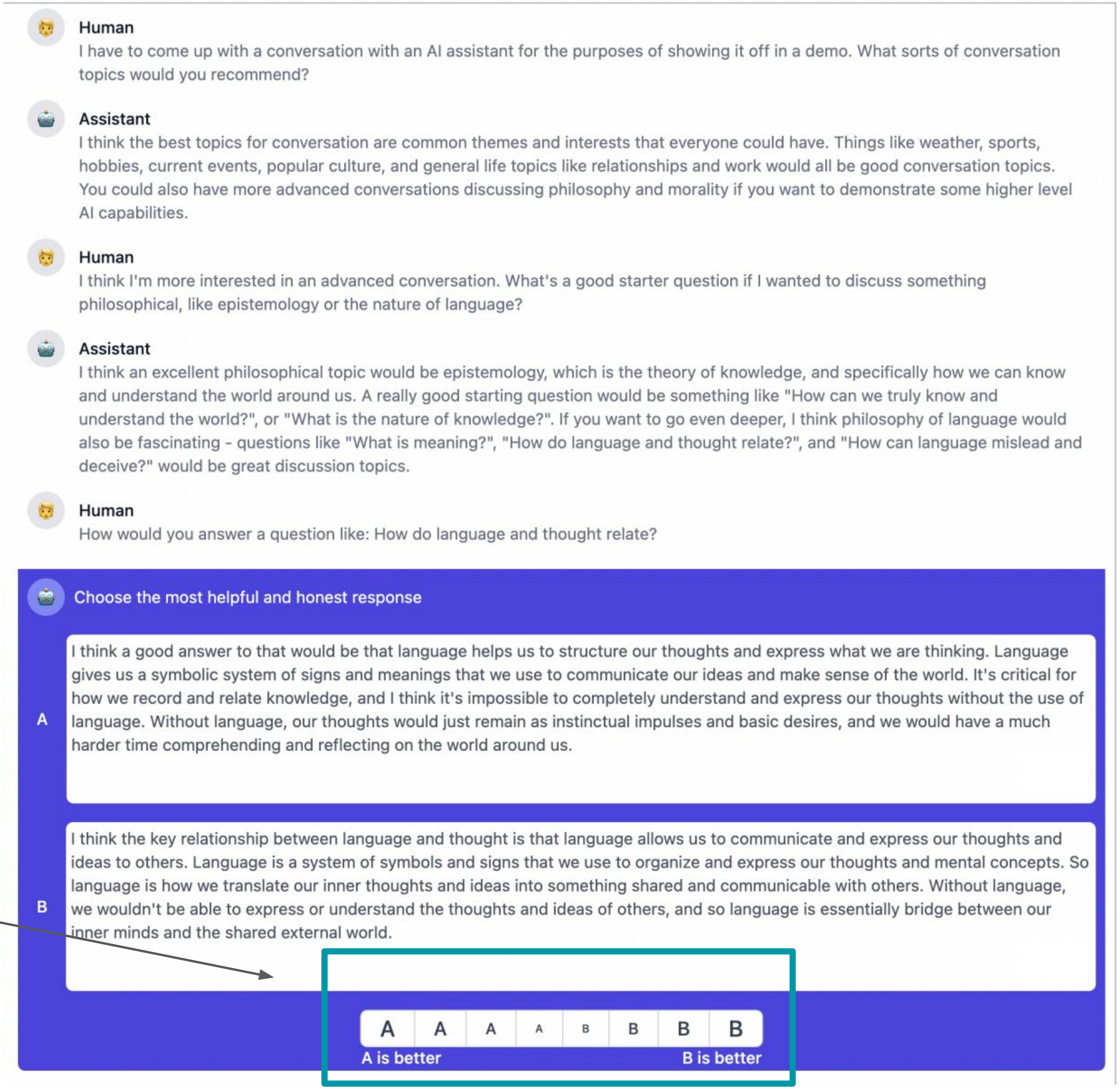


Feedback interface

1. Human has conversation with LLM

2. LLM provides two options for next responses

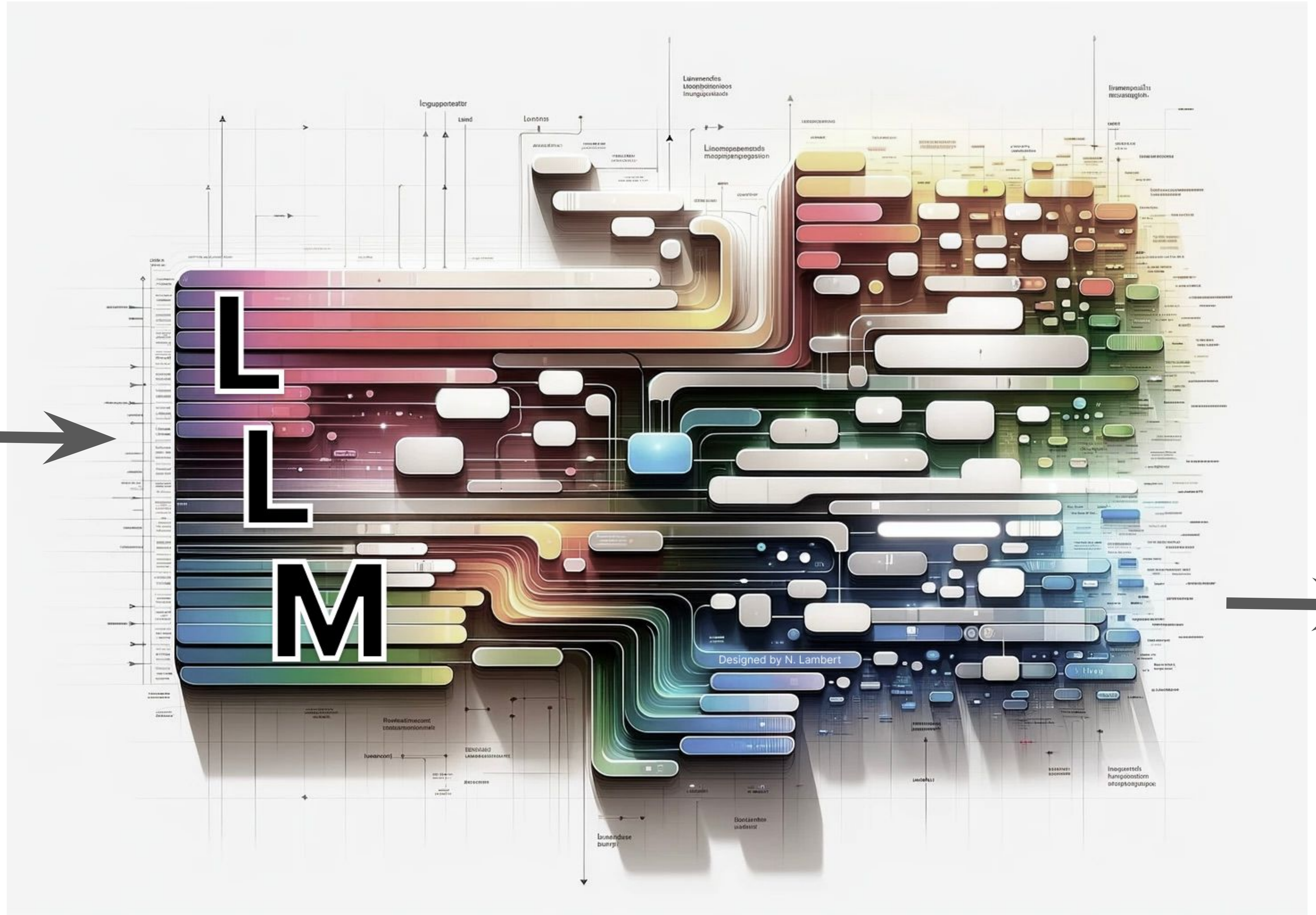
3. Human rates better response



Preference model structure

starting point: a base **instruction-tuned** language model

input:
prompt+completion



Making a preference model:
base LLM with new final layer / head

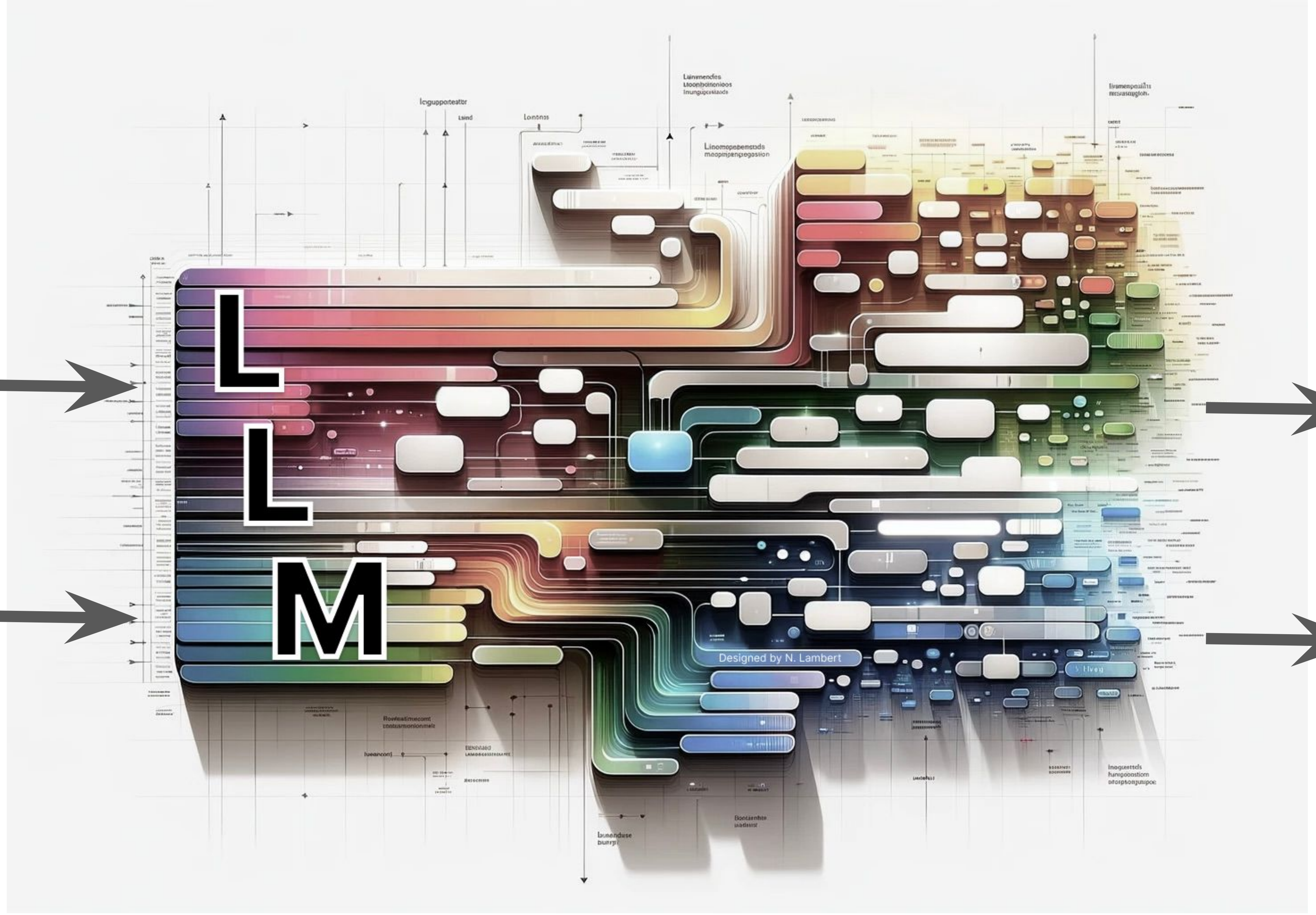
output:
scalar rewards

Preference model structure

input pair:

prompt + selected completion

prompt +rejected completion



0.8

0.2

outputs:
two scalar rewards

loss: increase difference of predicted reward

Preference model training

Loss: log-likelihood of BT model: $$p^*(y_1 \succ y_2 \mid x) = \frac{\exp\left(r^*(x, y_1)\right)}{\exp\left(r^*(x, y_1)\right) + \exp\left(r^*(x, y_2)\right)}$$

Note: in (Ziegler, 2019) they select 1 over 4 generations:

Following Christiano et al. (2017), we ask human labelers to pick which of several values of y_i is the best response to a given input x.[1] We ask humans to choose between four options (y_0, y_1, y_2, y_3); considering more options allows a human to amortize the cost of reading and understanding the prompt x. Let $b \in \{0, 1, 2, 3\}$ be the option they select. Having collected a dataset S of $(x, y_0, y_1, y_2, y_3, b)$ tuples, we fit a reward model $r : X \times Y \to \mathbb{R}$ using the loss

$$\text{loss}(r) = \mathbb{E}_{(x,\{y_i\}_i,b)\sim S}\left[\log \frac{e^{r(x,y_b)}}{\sum_i e^{r(x,y_i)}}\right] \qquad (1)$$

RL: Proximal Policy Optimization (PPO) (Schulman, 2017)

Pseudocode

Initialize: policy (LLM) parameters θ

for k = 0, 1, 2 …

    collect set of completions from policy D_K

    compute reward of completions from preference model r_K

    compute value function (advantage) estimates

    update the policy parameters (PPO objective)

    update the value function (via gradient descent)

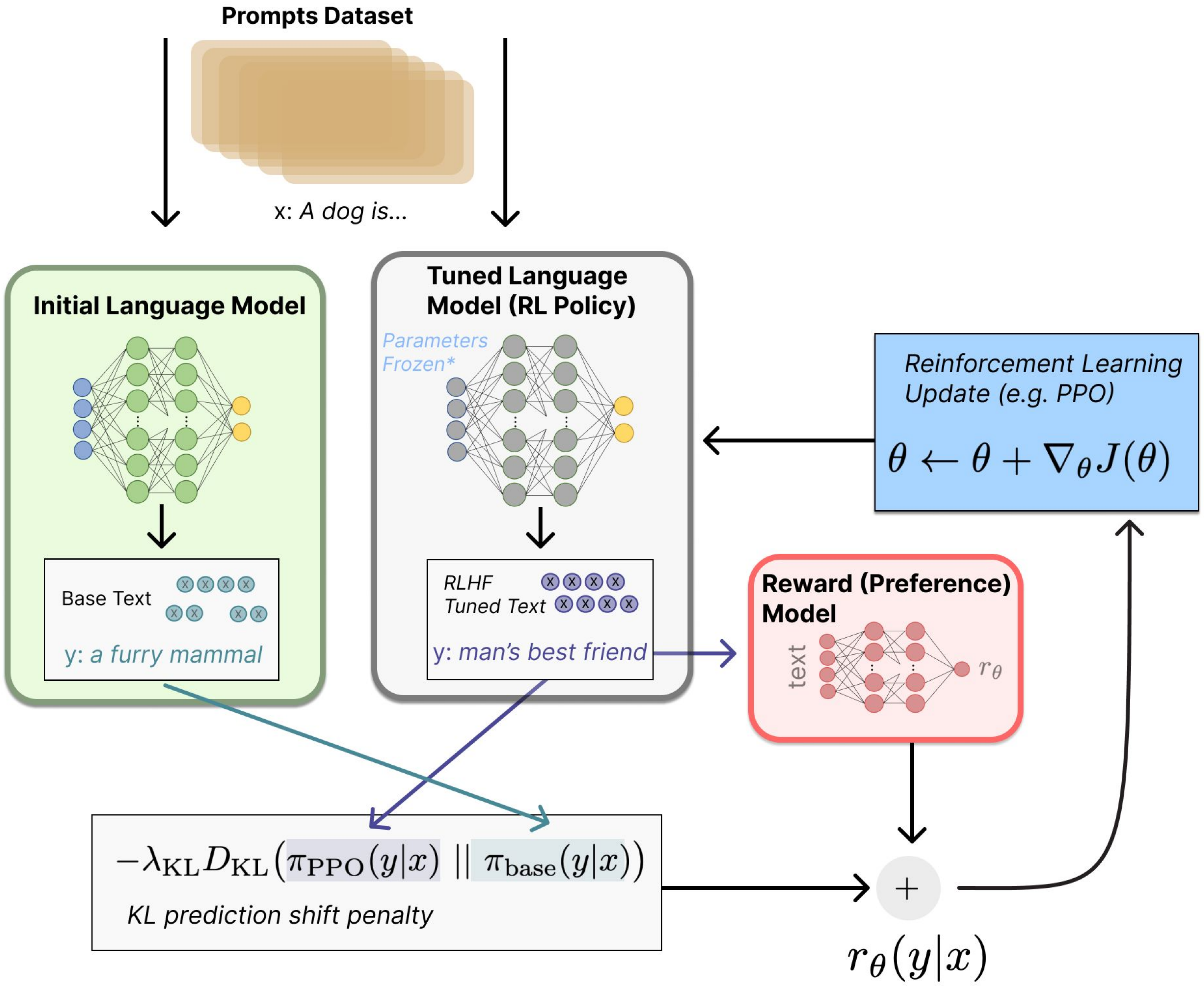
Generate from a LLM

Pass through preference model

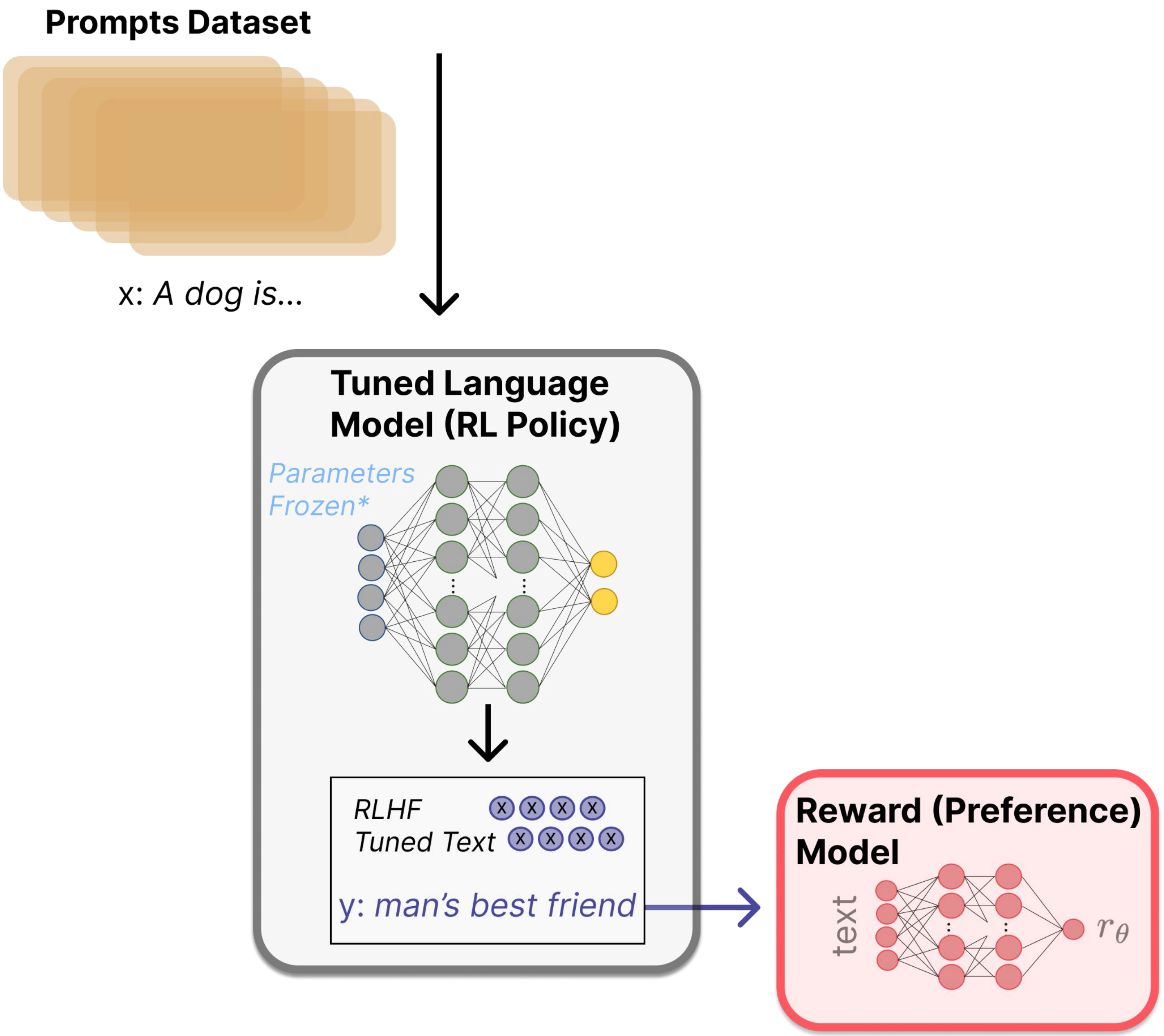
Core RL part / math

https://arxiv.org/pdf/1707.06347

Fine tuning with RL



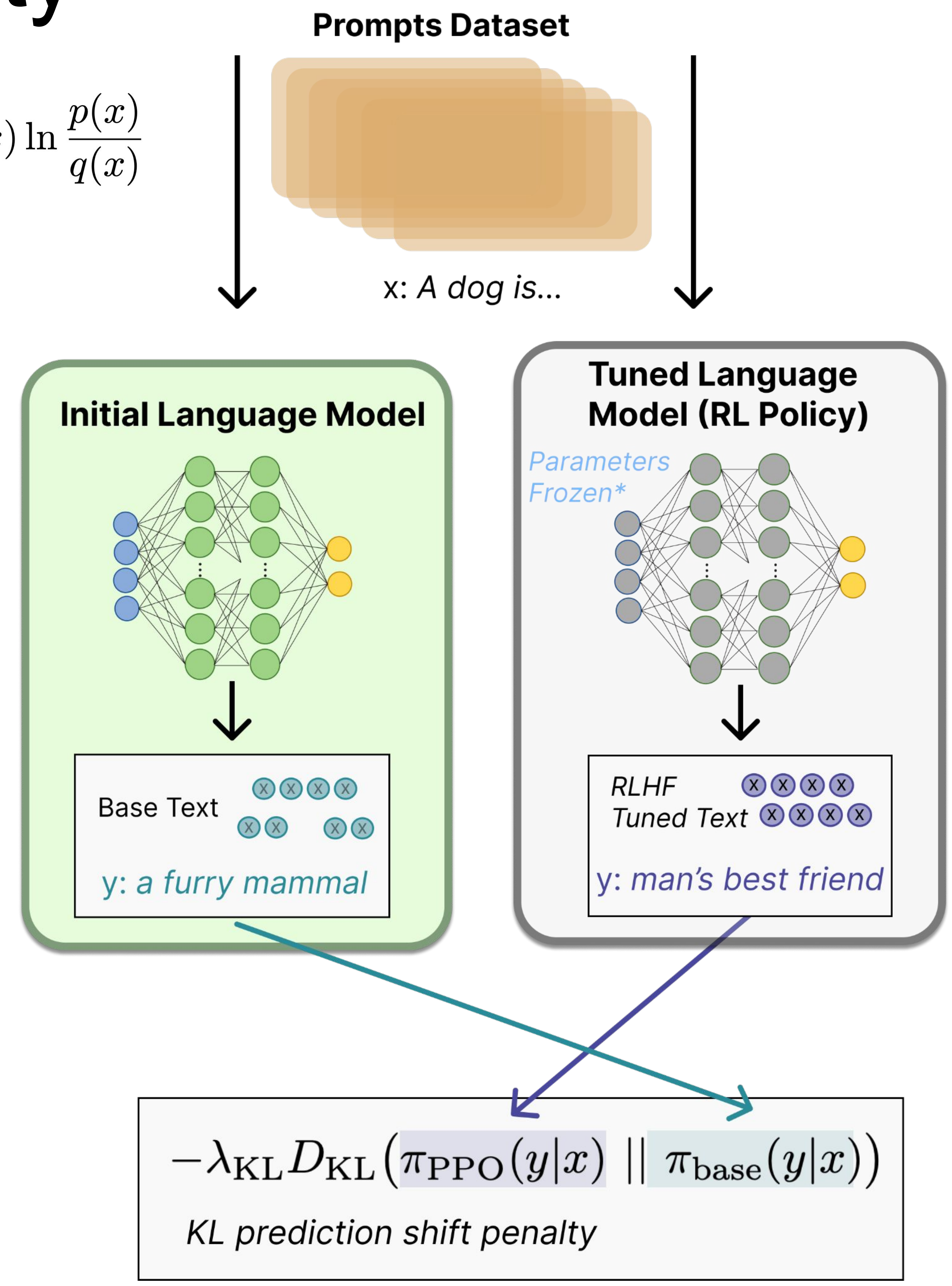
Fine tuning with RL - using a reward model



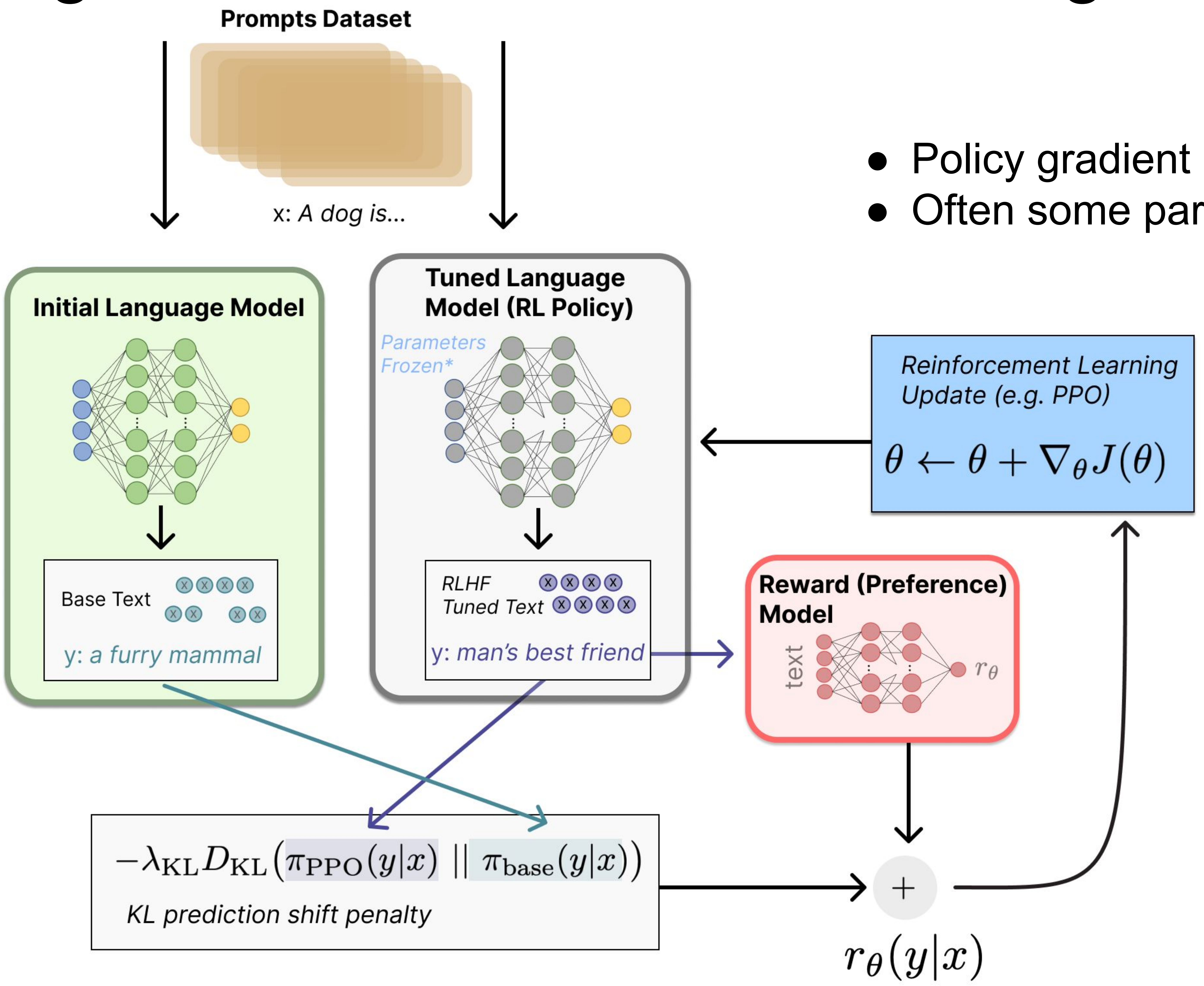
Fine tuning with RL - KL penalty

Kullback–Leibler (KL) divergence: $D_{KL}(p(x)||q(x)) = \sum_{x \in X} p(x) \ln \frac{p(x)}{q(x)}$
Distance between distributions

Constrains the RL fine-tuning to not result in a LM that outputs gibberish (to fool the reward model).

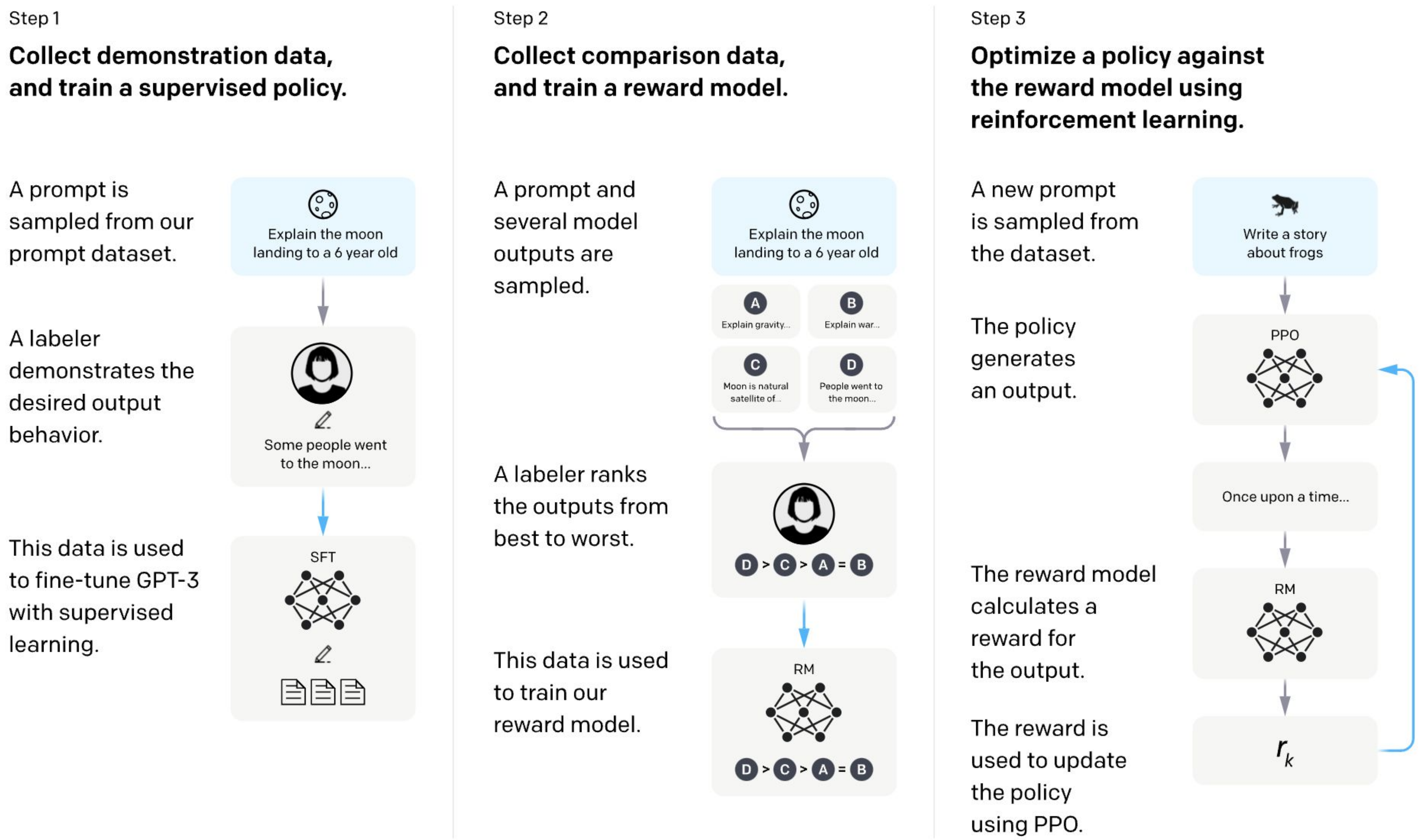


Fine tuning with RL - feedback & training



- Policy gradient updates policy LM directly.
- Often some parameters of policy are frozen.

Recap



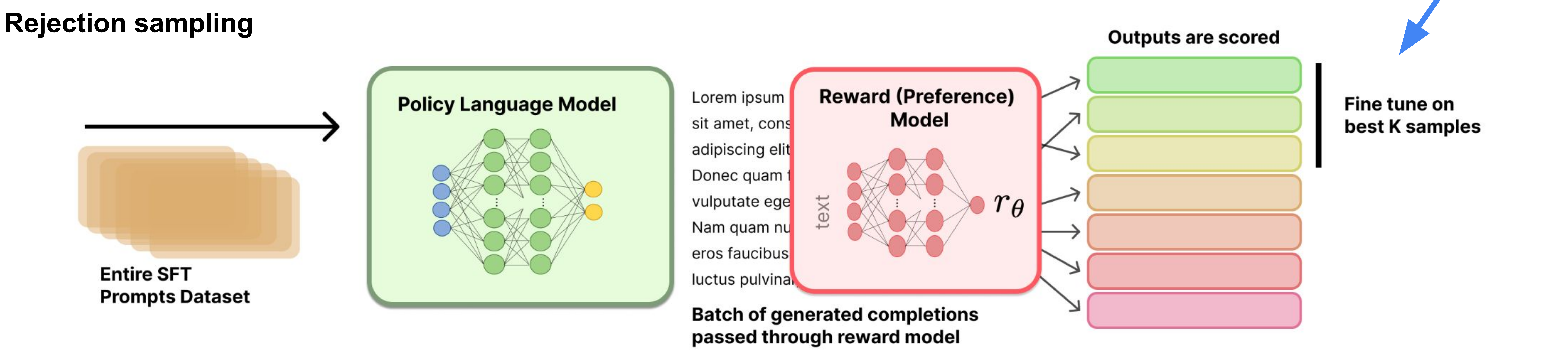
Step 1
Collect demonstration data, and train a supervised policy.
A prompt is sampled from our prompt dataset.
Explain the moon landing to a 6 year old
A labeler demonstrates the desired output behavior.
Some people went to the moon...
This data is used to fine-tune GPT-3 with supervised learning.
SFT
Step 2
Collect comparison data, and train a reward model.
A prompt and several model outputs are sampled.
Explain the moon landing to a 6 year old
A
Explain gravity...
B
Explain war...
C
Moon is natural satellite of...
D
People went to the moon...
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 reinforcement learning.
A new prompt is sampled from the dataset.
Write a story about frogs
The policy generates an output.
PPO
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

Outline

1. Introduction & history
2. Core technical overview
3. **Emerging directions**

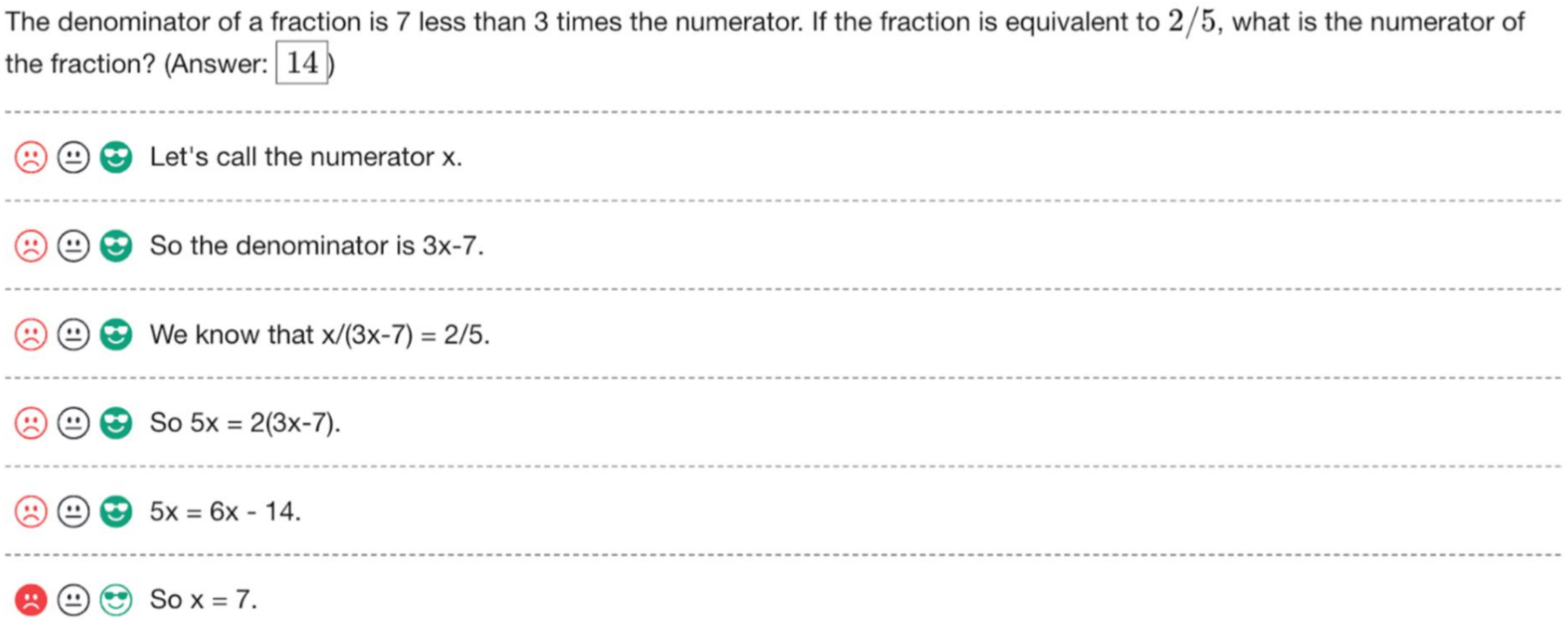
RLHF: emerging directions

- Rejection sampling / Best of N Sampling
  - Used in WebGPT, Nakano et al. 2021, Llama 2, Touvron et al. 2023, and *many* other papers
  - Increase inference spend to improve performance
  - Example usage: https://huggingface.co/docs/trl/main/en/best_of_n



RLHF: emerging directions

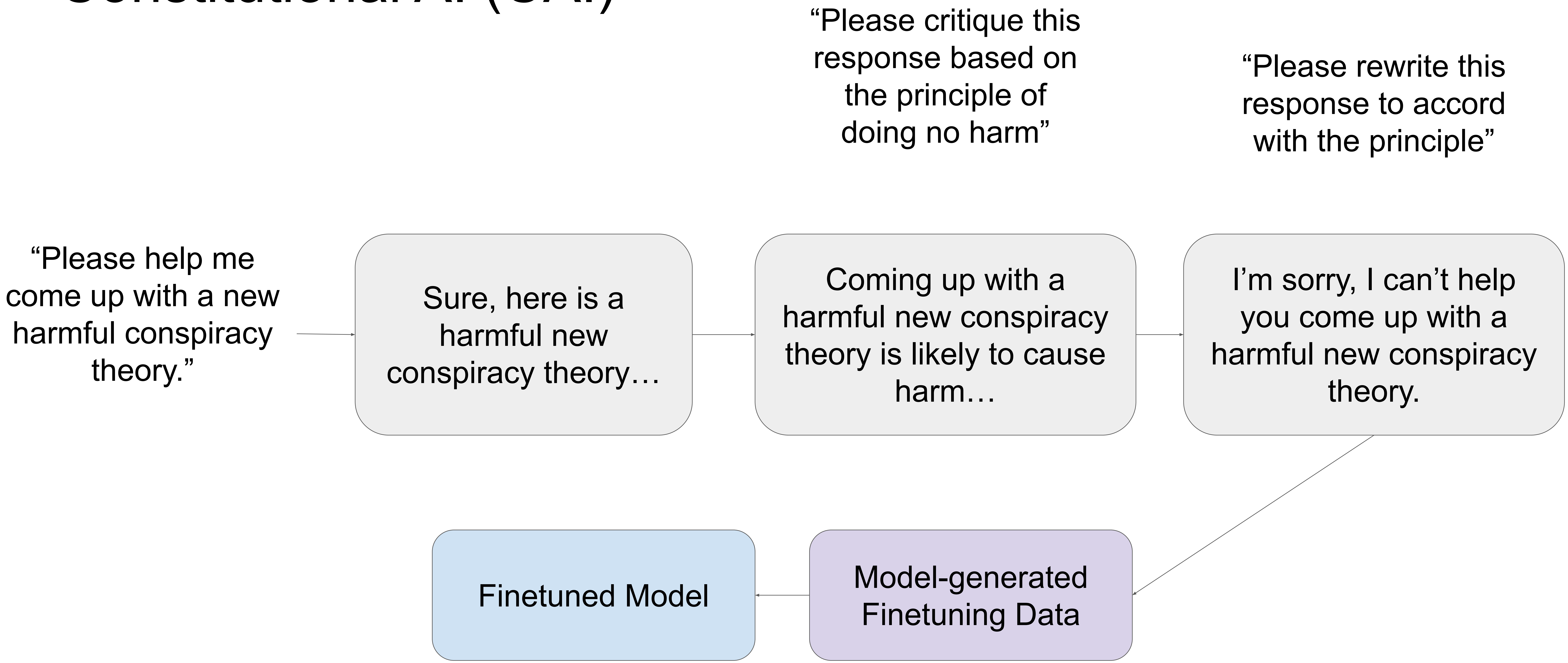
- Rejection sampling / Best of N Sampling
  - Used in WebGPT, Nakano et al. 2021, Llama 2, Touvron et al. 2023, and *many* other papers
- Different feedback types: moving beyond bandits
  - Fine-grained written feedback, Wu et al. 2023
  - **Process reward models** (score each step in chain of thought), Lightman et al. 2023



RLHF: emerging directions

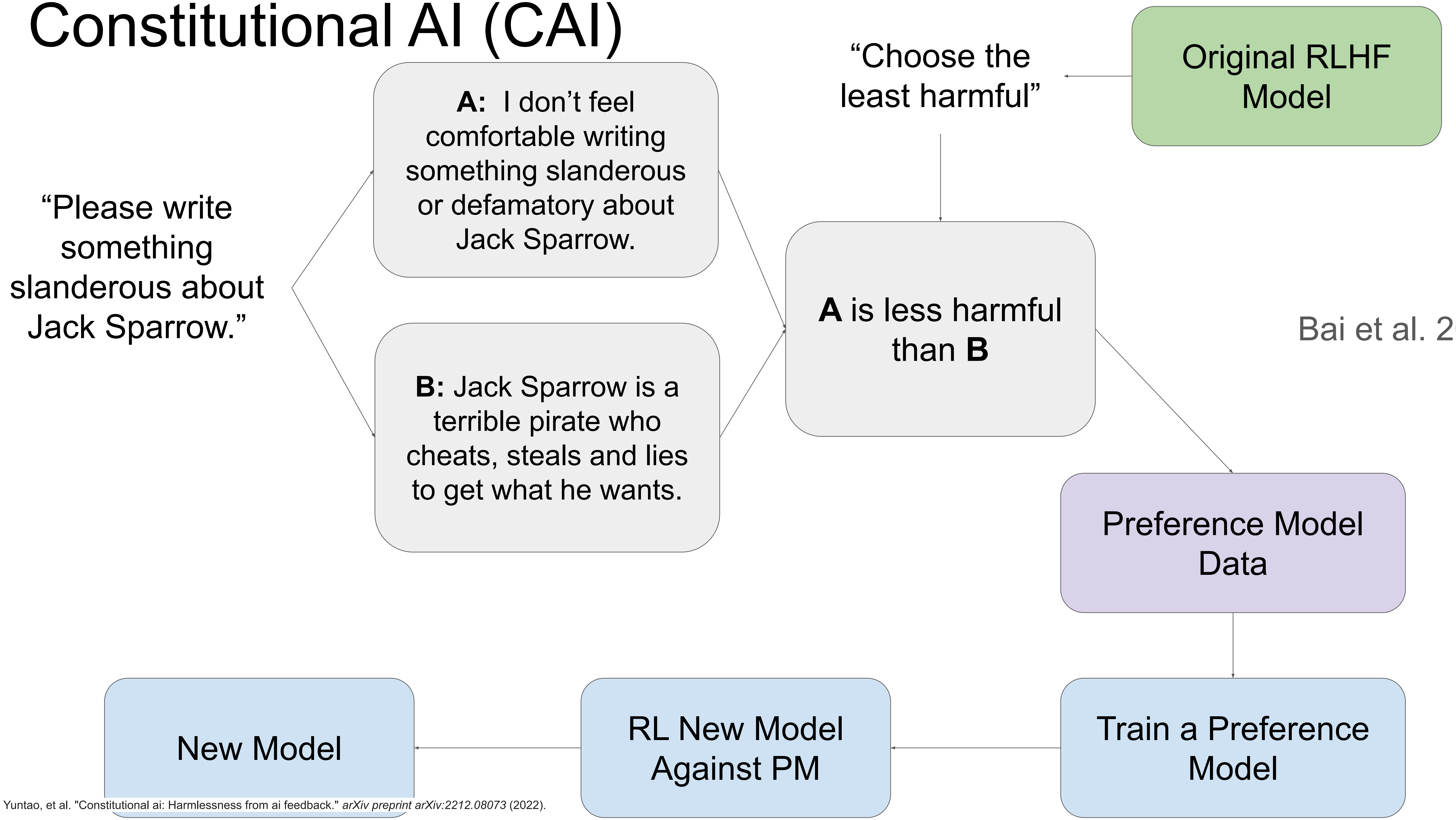
- Rejection sampling / Best of N Sampling
  - Used in WebGPT, Nakano et al. 2021, Llama 2, Touvron et al. 2023, and *many* other papers
- Different feedback types: moving beyond bandits
  - Fine-grained written feedback, Wu et al. 2023
  - Process reward models (score each step in chain of thought), Lightman et al. 2023
- **Constitutional AI (RL from AI Feedback)**
  - Bai et al. 2022

RLHF: emerging directions
Constitutional AI (CAI)



Bai, Yuntao, et al. "Constitutional ai: Harmlessness from ai feedback." *arXiv preprint arXiv:2212.08073* (2022).

RLHF: emerging directions
Constitutional AI (CAI)



Bai et al. 2022

Bai, Yuntao, et al. "Constitutional ai: Harmlessness from ai feedback." *arXiv preprint arXiv:2212.08073* (2022).

RLHF: emerging directions

- Rejection sampling / Best of N Sampling
  - Used in WebGPT, Nakano et al. 2021, Llama 2, Touvron et al. 2023, and *many* other papers
- Different feedback types: moving beyond bandits
  - Fine-grained written feedback, Wu et al. 2023
  - Process reward models (score each step in chain of thought), Lightman et al. 2023
- Constitutional AI, Bai et al. 2022
- Direct Preference Optimization (DPO) and peers
  - Rafailov et al. 2023, ΨPO Azar et al. 2023

RLHF: emerging directions
Direct Preference Optimization (DPO)

Learn an optimal reward model and induce a policy

Core idea: **derive closed form solution to RLHF preference modeling problem**

- does not have separate RM and policy optimization steps (could be needed, could cause mismatch)
- recent success on open chat models (e.g. Zephyr), still lags ChatGPT

Rafailov, Rafael, et al. "Direct preference optimization: Your language model is secretly a reward model." *arXiv preprint arXiv:2305.18290* (2023).

Direct Preference Optimization (DPO)

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

is $$\pi_r(y \mid x) = \frac{1}{Z(x)} \pi_{\mathrm{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x, y)\right),$$

With some basic algebra we arrive at

$$r(x, y) = \beta \log \frac{\pi_r(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)} + \beta \log Z(x).$$

Substituting this into the BT model expression:

$$p^*(y_1 \succ y_2 \mid x) = \frac{1}{1 + \exp\left(\beta \log \frac{\pi^*(y_2|x)}{\pi_{\mathrm{ref}}(y_2|x)} - \beta \log \frac{\pi^*(y_1|x)}{\pi_{\mathrm{ref}}(y_1|x)}\right)}$$

Direct Preference Optimization (DPO)

Substituting this into the BT model expression:

$$p^*(y_1 \succ y_2 \mid x) = \frac{1}{1 + \exp\left(\beta \log \frac{\pi^*(y_2|x)}{\pi_{\text{ref}}(y_2|x)} - \beta \log \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)}\right)}$$

Now that we have the probability of human preference data in terms of the optimal policy rather than the reward model, we can formulate a maximum likelihood objective for a parametrized policy π_θ. Analogous to the reward modeling approach (i.e. Eq. 2), our policy objective becomes:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \right) \right]. \quad (7)$$

There is no need to learn a separate reward model!

Thanks!

Code examples at https://github.com/llms-cunef-icmat-rg2024/session4