Distortion of AI Alignment:

Does Preference Optimization Optimize for Preferences

Nika Haghtalab
University of California, Berkeley

Based on joint works with

Kunhe Yang
UC Berkeley

Paul Gölz
Cornell

NeurIPS'25:
arxiv.org/pdf/2505.23749

Kasuzato Oko
UC Berkeley

Annie Ulichney
UC Berkeley

Han Bao
ISM Tokyo

ICML'26
Will soon be on arxiv

What is “AI Alignment” or “Post-training” ?

The key to unlocking the power of large pre-trained models for humans use.

→e.g., for safety, instruction following, accuracy, personality, …

Example:

- **Prompt:** Explain the moon landing to a 6 year old in a few sentences.
- **Pre-trained:** Explain the theory of gravity to a 6 year old. Explain the theory of relativity to a 6 year old in a few sentences. Explain the big bang theory to a 6 year old. Explain evolution to a 6 year old ….
- **Instruct:** People went to the moon, and they took pictures of what they saw, and sent them back to the earth so we could all see them.

- **Prompt:** What is 17×24? Show your work.
- **Not math-RL-tuned:** $17 \times 24 = 17 \times 20 + 17 \times 4 = 340 + 68 = 388.$
- **Math-RL-tuned:** $17 \times 24 = 17 \times (20 + 4) = 340 + 68 = 408.$

What is “AI Alignment” or “Post-training” ?

The key to unlocking the power of large pre-trained models for humans use.

→e.g., for safety, instruction following, accuracy, personality, …

Technical perspective:

→Pre-trained model: distribution over good-enough or legible outcomes.

→Post-training moves this distribution outcomes with higher “reward”.

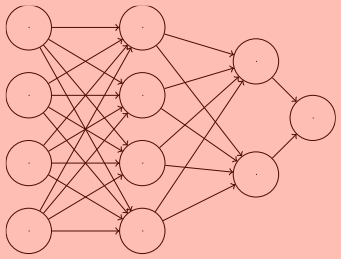
But “reward” …

what is it, where does it come from, how should it be incorporated?

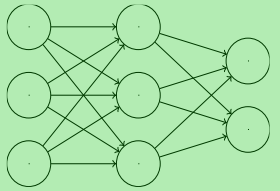
Most important challenges: reward itself is complex, high dimensional, unknown, and signals available to the model are sparse, individually unreliable, expensive to compute.

Post-training Formulation

π_{ref}



Large Pre-trained Model

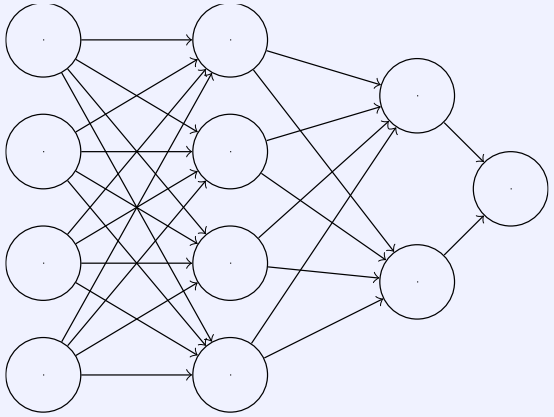


Reward Model

r

π_{ref} and π_{post}:
RL generation policies

For ease, imagine distributions.



Fine-tuned model π_{Post}

Post-training Formulation

Given a pre-trained reference model π_{ref} and a reward model r, do

$$\pi_{post} = argmax_{\pi} \mathbb{E}_{x\sim\pi}[r(x)] - \lambda D_{KL}(\pi \,||\, \pi_{ref})$$

This is done through RL/policy-gradient updates, etc.

Equivalent constrained optimization formulation (for appropriate $\lambda \leftrightarrow \tau$) :

$$\pi_{post} = argmax_{\pi \in \mathrm{KL}(\pi_{ref},\tau)} \mathbb{E}_{x\sim\pi}[r(x)]$$

Reward signals

Rewards can be complex and hard to express.

Examples:

- reward = utility a person derives from generating a particular outcome.
- reward = whether a long mathematical derivation is correct, modular, and easy-to-verify.

Human Feedback

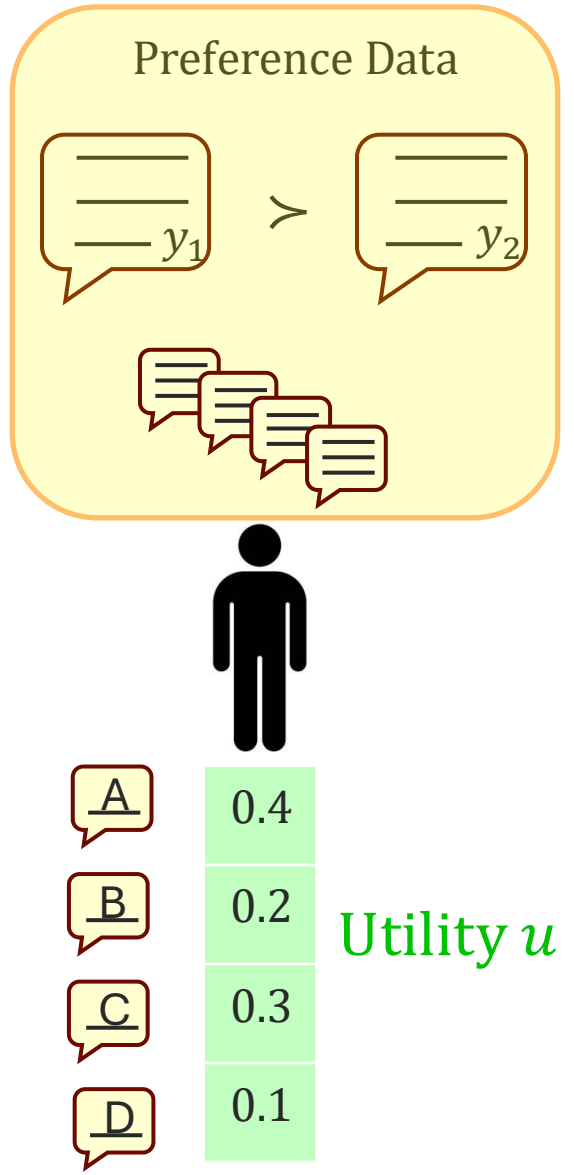
Rewards can be complex and hard to express.

With humans in the loop, ordinal preferences are elicited.

Typical Implicit Assumption:

All comparisons are provided according to a single utility function u, Bradley-Terry is common assumption

- $\Pr[y_1 \succ y_2] = \sigma(\beta(u(y_1) - u(y_2)))$
- $\sigma(t) = \frac{1}{1+\exp(-t)}$: logistic sigmoid function



Learning a Reward Model

Given rated pairs $y_1^k \succ y_2^k$ for $k = 1, \ldots$ the first step in using these rewards is to turn them into a reward model.

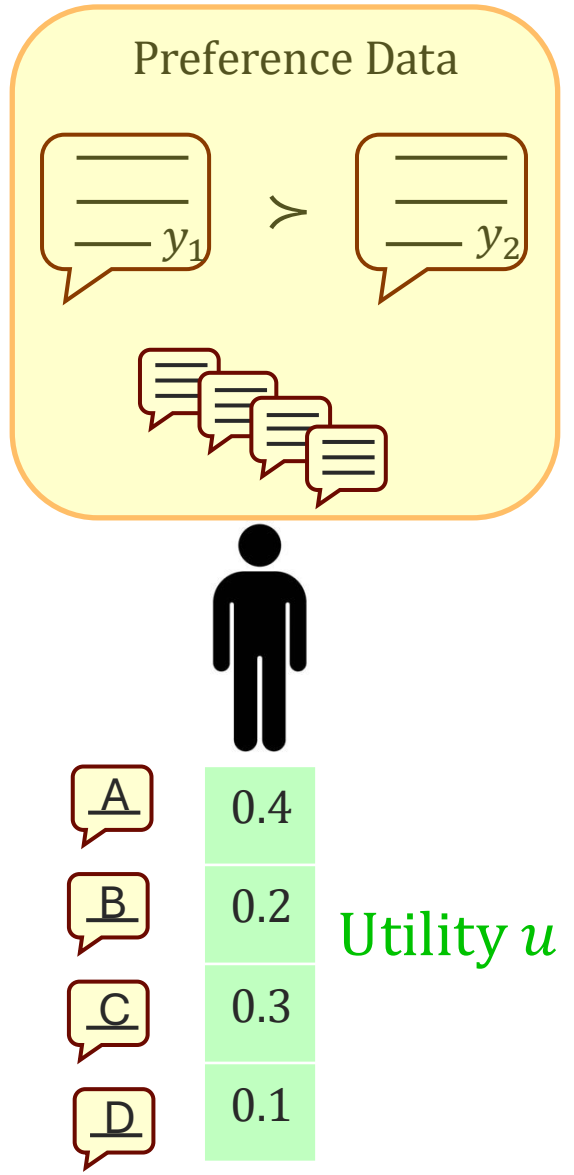
MLE solution, given rated pairs $y_1^k \succ y_2^k$ for $k = 1, \ldots$

$$\hat{r} = \text{argmax}_r \sum_k \log\left(\sigma\left(\beta\left(r\left(y_1^k\right) - r\left(y_2^k\right)\right)\right)\right)$$

With infinite data, MLE chooses the model probabilities that match the observed preference frequencies.

In BT, those probabilities are determined by reward differences.

So, $\hat{r}$ recovers u, up to additive constant.



Learning a Reward Model

Given rated pairs $y_1^k \succ y_2^k$ for $k = 1, \ldots$ the first step in using these rewards is to turn them into a reward model.

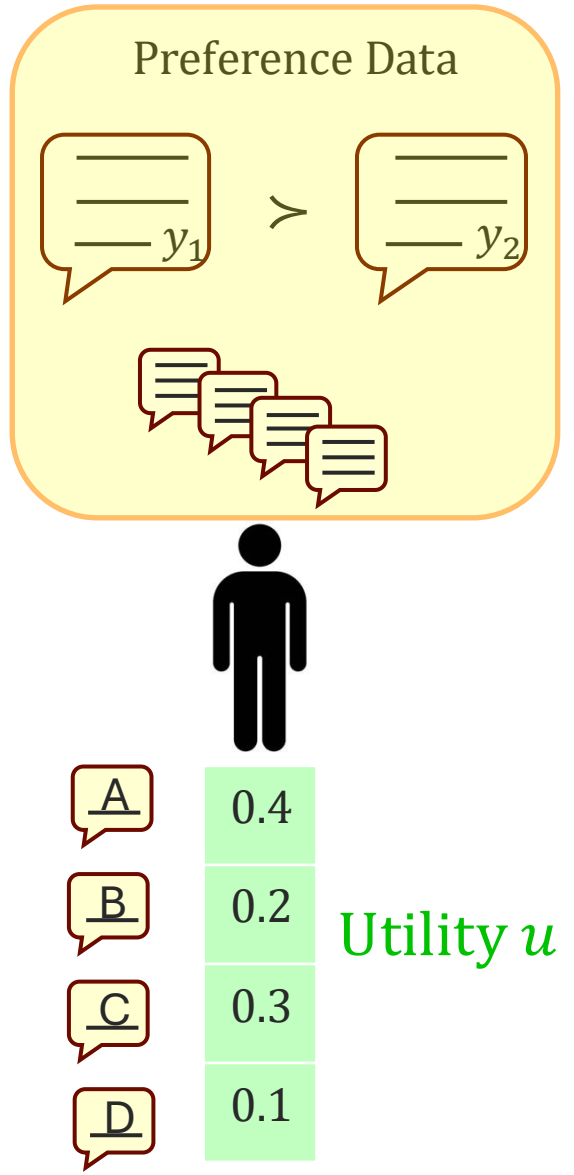
MLE solution, given rated pairs $y_1^k \succ y_2^k$ for $k = 1, \ldots$

$$\hat{r} = \text{argmax}_r \sum_k \log\left(\sigma\left(\beta\left(r\left(y_1^k\right) - r\left(y_2^k\right)\right)\right)\right)$$

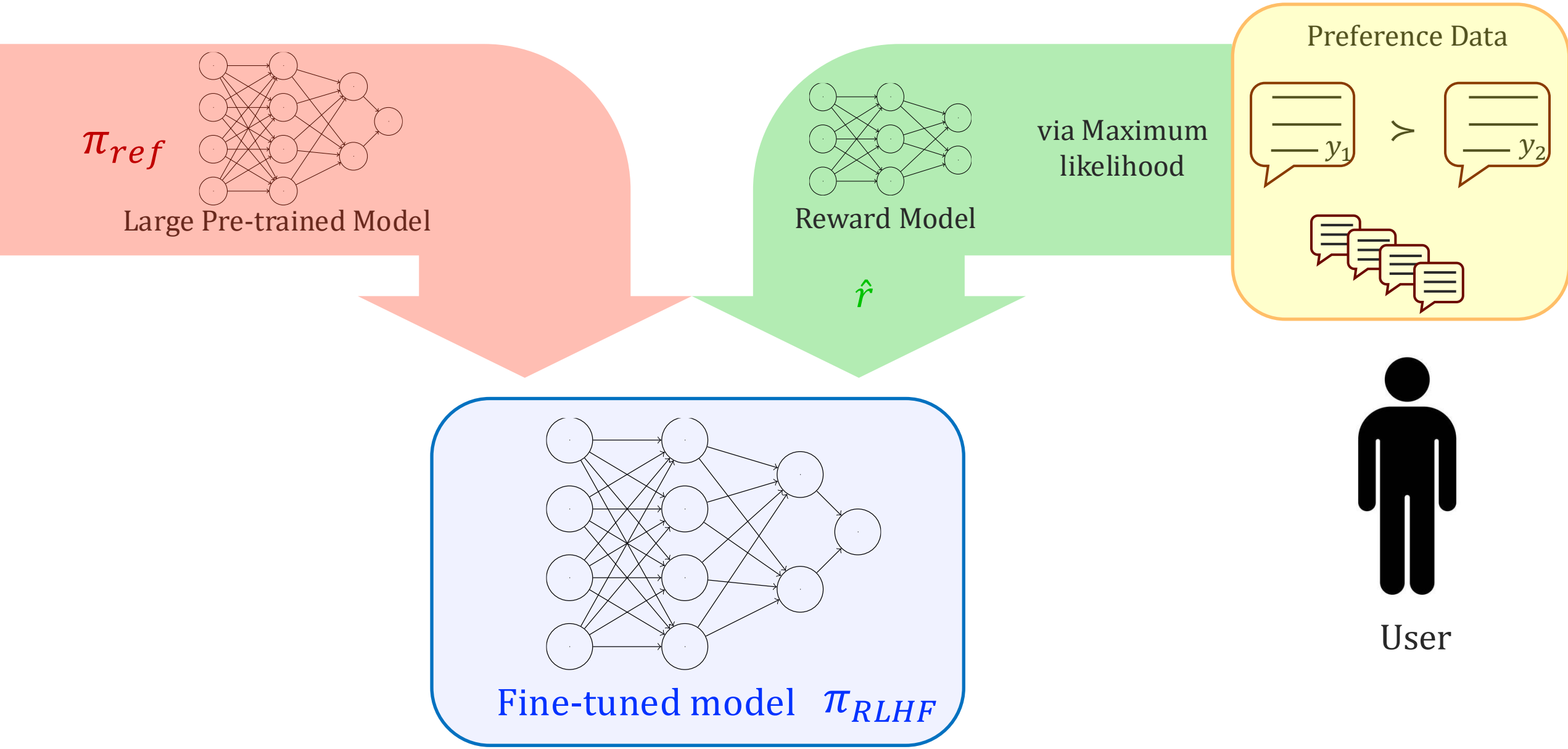
So, $\hat{r}$ recovers u, up to additive constant.

Post-train with $\hat{r}$ as reward model:

$$\pi_{RLHF} = argmax_{\pi \in \text{KL}(\pi_{ref}, \tau)} \mathbb{E}_{x \sim \pi}[\hat{r}(x)]$$



The fine-tuned model, maximizes user's underlying reward function as much as possible, while remaining close to the pre-trained model.



Utilities and subjective opinions

Users derive utility from generated outcome of a model. E.g.,

→How much time I need to still spend editing the generated outcome to make it easily usable or match my style of writing.

→How much I enjoy the responses I'm getting

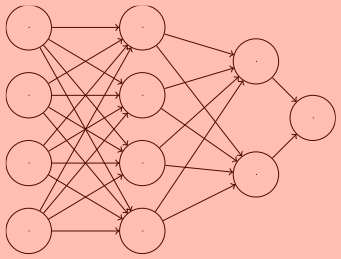
→How the shopping cart my agent made, the report it generated on my behalf, the flight it preselected for me, match my particular needs.

These are subjective utility functions

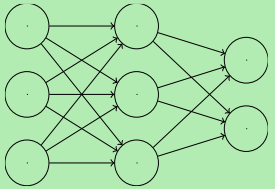
→Heterogenous across users

→The basis for how a human provides feedback.

π_{ref}



Large Pre-trained Model



Reward Model

via Maximum likelihood

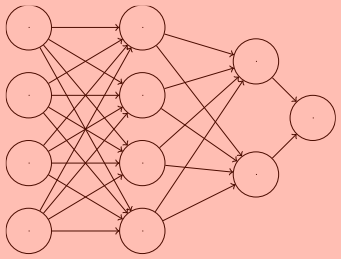
Preference Data

$y_1 \succ y_2$

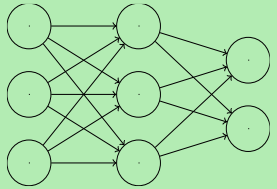
Users

Fine-tuned model π_{RLHF}

π_{ref}



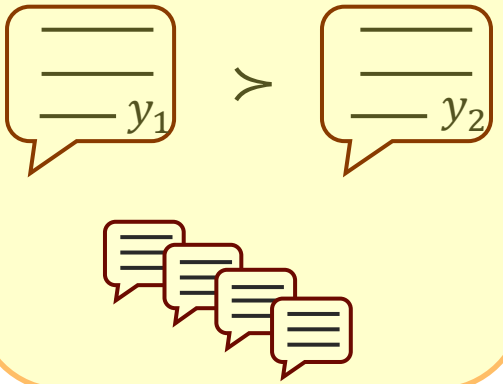
Large Pre-trained Model

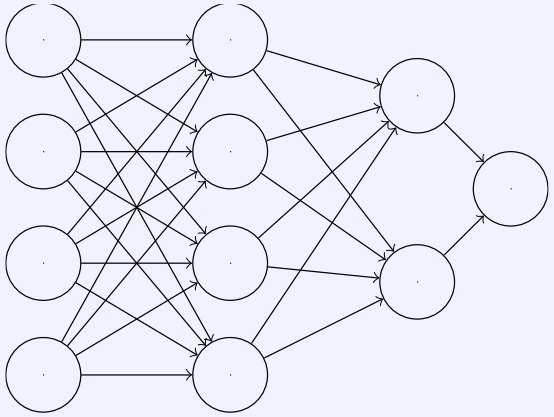


Reward Model

via Maximum likelihood

Preference Data





Fine-tuned model π_{RLHF}

Utilities $u_i s$

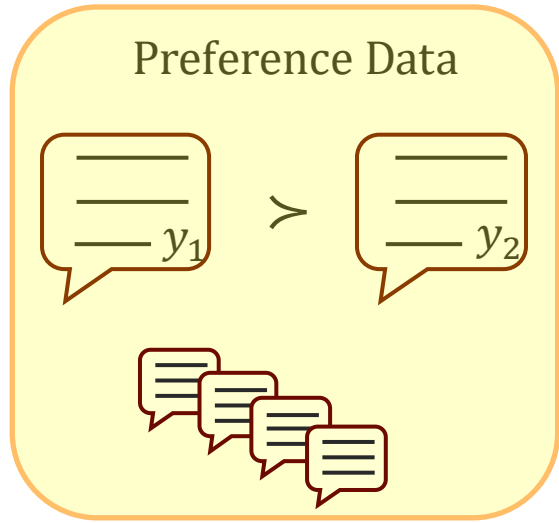
A	0.4	1	0.1	1
B	0.2	0	0	0
C	0.3	0	0	0
D	0.1	0	0.5	0.1

Collection of Heterogenous Preference Data

Comparisons are crowdsourced:

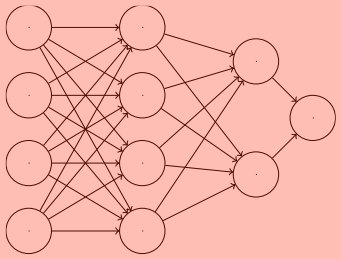
- The set of all comparisons $S = \{(y_1^k \succ_i y_2^k)\}$
- Each comparison for pair of options y_1^k and y_2^k is done by random user $i \sim \mathcal{P}$ who then compares $y_1^k \succ_i y_2^k$ using idiosyncratic Bradley-Terry model wrt u_i.
- This can be extended to d-wise comparisons or full ranks.
- Importantly, like all noise in ML we cannot ask the same question to the same person many times and get fresh random answers to reduce noise and learn their utility.

RLHF is performed on the set S as before.

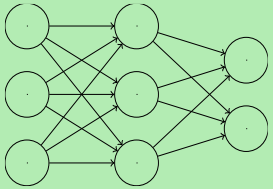


A	0.4	1	0.1	1
B	0.2	0	0	0
C	0.3	0	0	0
D	0.1	0	0.5	0.1

The fine-tuned model, fits a mythical notion of representative reward as much as possible, while remaining close to the pre-trained model.



Large Pre-trained Model

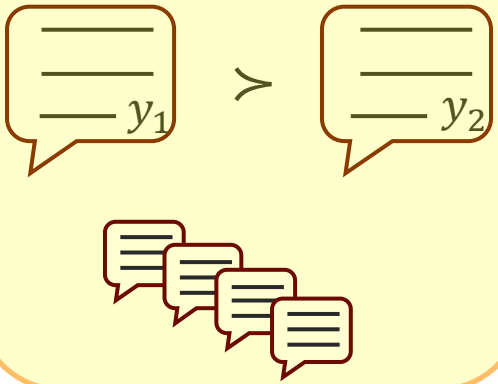


Reward Model

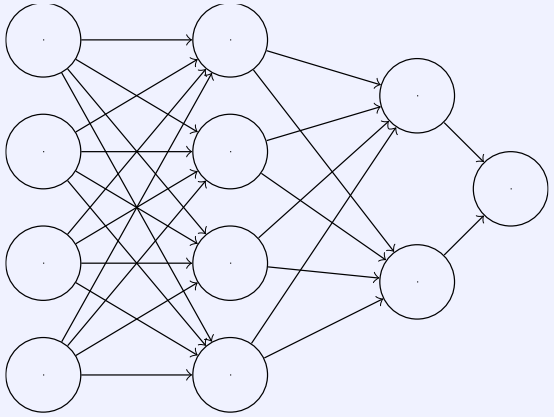
via Maximum likelihood

Fitting to a "mythical" user

Preference Data



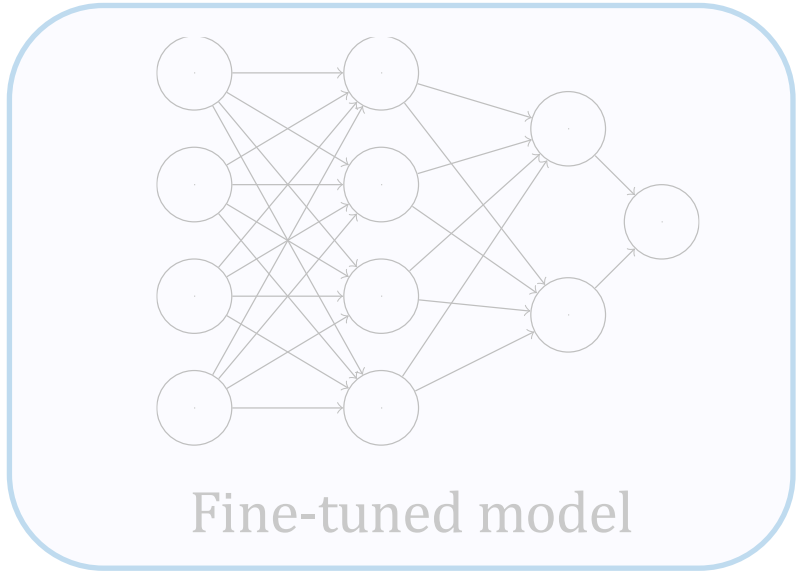
Users

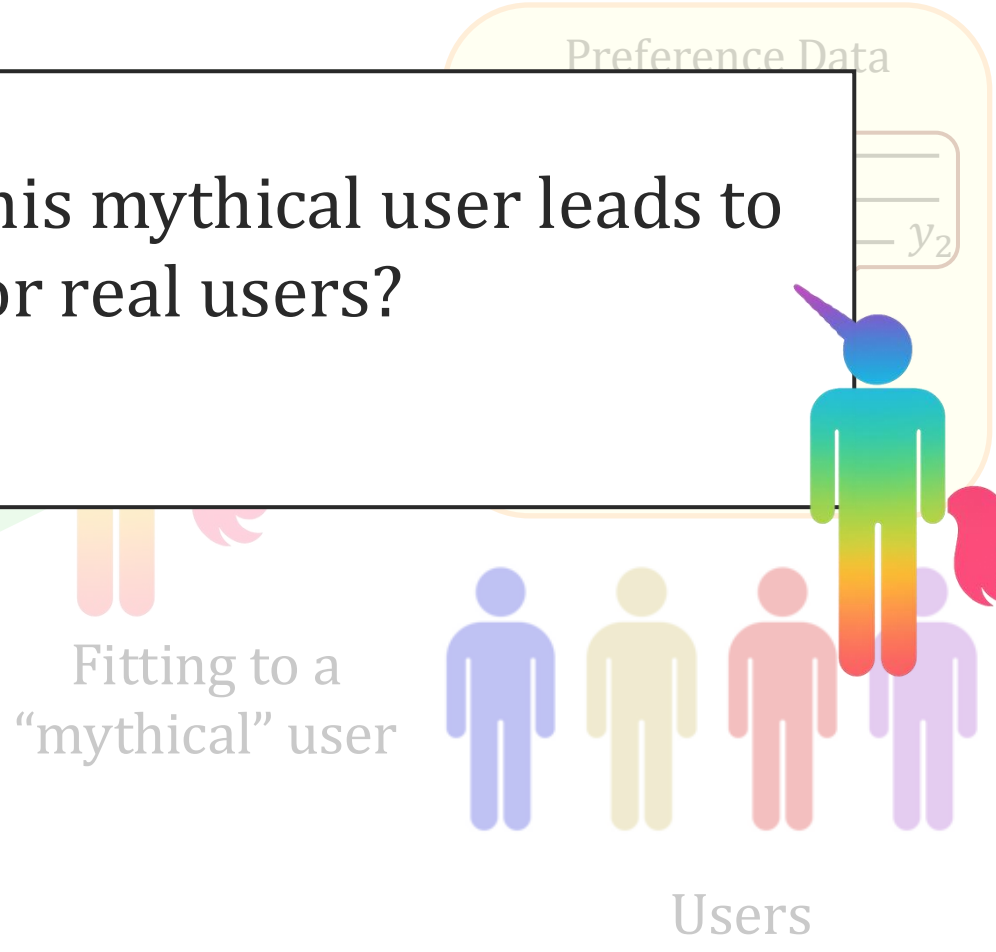


Fine-tuned model

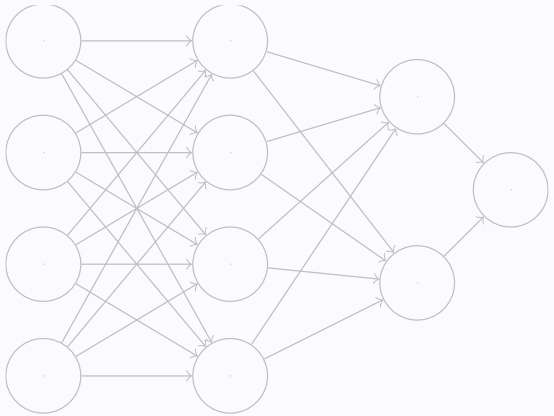
The fine-tuned model, fits a mythical notion of representative reward as much as possible, while remaining close to the pre-trained model.

Could it be that optimizing a model for this mythical user leads to poor outcomes on average for real users?





Do **ordinal preferences** even contain
enough information to give ensure **high average utility?**



Fine-tuned model

Utilities
$u_i s$

A	0.4	1	0.1	1
B	0.2	0	0	0
C	0.3	0	0	0
D	0.1	0	0.5	0.1

Distortion of AI Alignment

Distortion of AI Alignment from Human Preferences

Can use cardinal preferences

Distortion: $\dfrac{\text{avg. utility of } \textbf{the optimal policy}}{\text{avg. utility of } \textbf{finetuned policy}}$

Both can be regularized/constrained

Uses only ordinal preferences

"Distortion" comes from social choice [Procaccia, Rosenschein '06], two differences:

- Social choice does not have reference policy or regularization, making it simpler.
  - We'll talk about the role of reference policy
- Social choice doesn't use Bradely-Terry modeling

  → Led to some impractical takeaways about distortion of practical methods.

  → We'll give bounds that are non-worst case and more aligned with practice.

Distortion as Fundamental Information Loss

For the purpose of this talk, focus on **idealized setting of RLHF**:

- Infinite number of users $i \sim \mathcal{P}$ can be drawn
- Each user is asked for d-wise comparisons. Take $d = 2$ for intuition.
- Even with infinite samples, we show distortion is lower bounded.

→Distortion quantifies **information loss** of ordinal preferences for aligning AI.

Foreshadowing that this "weak requirement" is not met bcuz of information loss!

Trends in ML: ML methods are typically good "on average"

- Dissatisfied with performance? Then, ask for finer-grained guarantees using more of the same type of data. Scale data sets.
- If RLHF performs poorly even on average, we need a different path forward like different type of data entirely or algorithmic approaches to get most of the current data.

Three Messages to Take Away

1. Distortion is unavoidable

→ Every method will have at least $\approx \frac{\beta}{2}$ distortion

2. RLHF (most common method) can have **exponentially large distortion**!

→ Particularly bad when the reference and sampling distributions don't match.

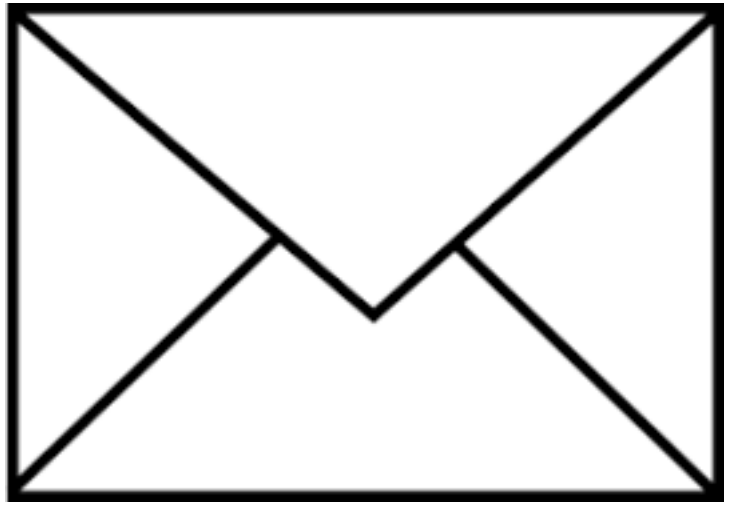
→ There are ways to control RLHF distortion to be $O(\beta)$ when the reference and sampling distributions are close. But it's still $> \beta$.

3. We give the method with minimax optimal distortion of $\frac{\beta}{2}$

→ Called Nash Learning from Human Preferences (NLHF)

For the purpose of this talk, focus on **idealized setting of RLHF** where we have infinite # of users sampled. All statements have finite sample complexity version (see our papers).

Message #1
Distortion is unavoidable



Any finetuning method from ordinal preferences (inc. RLHF, DPO) introduces distortion!

There is an algorithm-independent lower bound of $\approx \frac{\beta}{2}$ on distortion of any method.

Temperature of Bradley-Terry

Unavoidable Distortion: Catastrophic Non-identifiability

Informal Claim: Any method that only accesses ordinal preferences, will have at least $\frac{\beta}{2}$ distortion, even with infinitely many comparison data and BT model.

Possible candidates $a, b_1, b_2, \ldots, b_m$

Candidate a gives significantly more avg utility.

World 1

Two user types:

Minority: $(1, 0, 0, \ldots, 0)$

Majority: $(0, \epsilon, \epsilon, \ldots, \epsilon)$

World 2

A single user type,

$(1, 1, 1, \ldots, 1)$

All candidates equally good

Catastrophic non-identifiability: Both worlds, have $P(x \succ y) = \frac{1}{2}$ for all pairs of candidates, so can't identify candidate with high avg utility.

Non-identifiability with Bradley-Terry

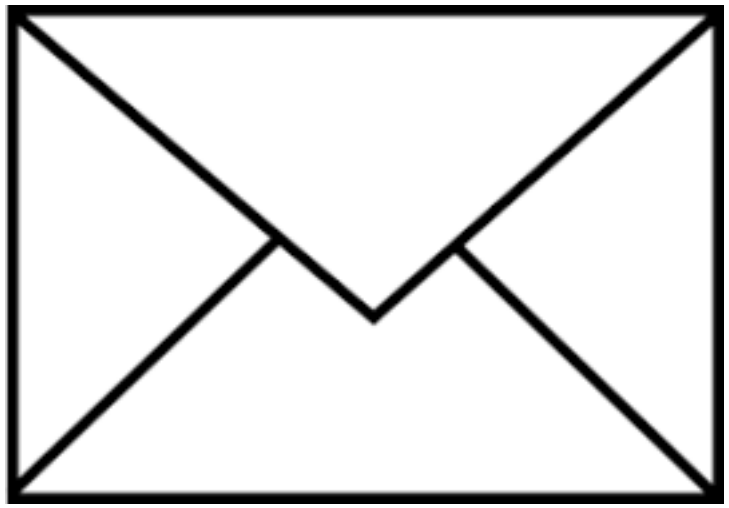
Issues of **non-identifiability** with **Bradley-Terry** long studied.

Prior work:

- Classical: Even homogenous case, can't identify the additive constant.
- Also, it's been known that different distributions over utilities and idiosyncratic Bradley-Terry Noise maps to the same distribution over candidate rankings [Zhao et al '16].

What's interesting and new here: **"Catastrophic"** non-identifiability

- Not just that two distributions over utilities map to the same distribution over rankings.
- But also that the mean utility of the distributions is significantly different.



Message #2

RLHF's distortion significantly larger than needed!

RLHF's distortion can be $\exp(\beta)$ or even unbounded!

As compared to the unavoidable distortion rate of $\approx \beta/2$.

Under idealized setting, some variants of RLHF can have distortion of $> \beta$ but still $O(\beta)$

Why does RLHF do particularly poorly?

Key limitation of RLHF and other reward-based models:

→Sensitivity to the distribution from which comparisons are selected.

For demonstration, we'll use a particularly skewed distribution over comparisons

- MLE attempts to fit a reward to the observed comparisons
- Heterogeneity → Not all pairwise win-rates can be simultaneously fit.
- "Maximum likelihood" sacrifices accuracy on the rarely observed pair
- This gets amplified by the distortion of the reference policy.

The fault is with the MLE, and the KL constraint worsens it!

	A	B	C
Minority ($\delta = \Theta(e^{-\beta})$ fraction)	0	1	0
Majority ($1 - \delta$ fraction)	$1/\beta$	0	1
Average Utility	$\Theta(1/\beta)$	$\Theta(e^{-\beta})$	$\Theta(1)$

π_{ref}: tiny weight on C (e.g., C is harmful)

Sampling distribution: C shows up more often than A and B (e.g., unsafe policy not yet suppressed by sampling distribution)

Optimal constrained π_{opt} should put almost all weight on next best candidate: A

RLHF puts almost all weight on candidate B. Why?

- No homogenous BT model fits perfectly, so MLE is inaccurate on some pairs.

→ Since (A,B) doesn't appear as often, MLE sacrifices accuracy on that

→ MLE reward: $\hat{r}(C) \gg \hat{r}(B) > \hat{r}(A)$

A Finer-Grained Understanding of RLHF

There are two issues:

- The scale of MLE rewards are unreliable
- It combines poorly with KL constraint, especially if off policy method.

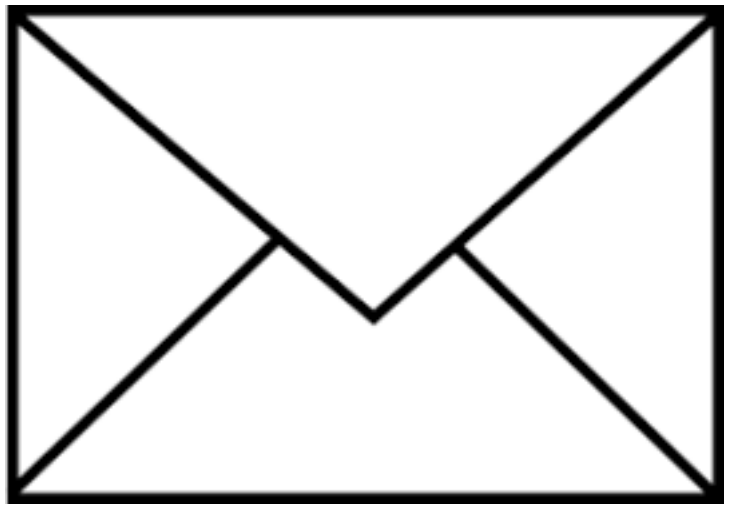
There is a way to mostly fix the first issue as long as the second issue is controlled.

Informal Claim

There is a post-processing of learned MLE rewards such that

$$\text{Distortion of } \widehat{\text{RLHF}} \lesssim O(\beta \times B)$$

Where $B := max_x \log\left(\frac{\pi_{sample}(x)}{\pi_{ref}(x)}\right)$ is the gap between the reference distributions, and where the prompts were sampled from. And this is tight.



Message #3

There is a method that matches $\beta/2$ minmax optimal distortion

Hypothetical No-distortion Setting

Recall: Non-linearity of the sigmoid function in Bradley-Terry model gave rise to some distortion.

A hypothetical world:

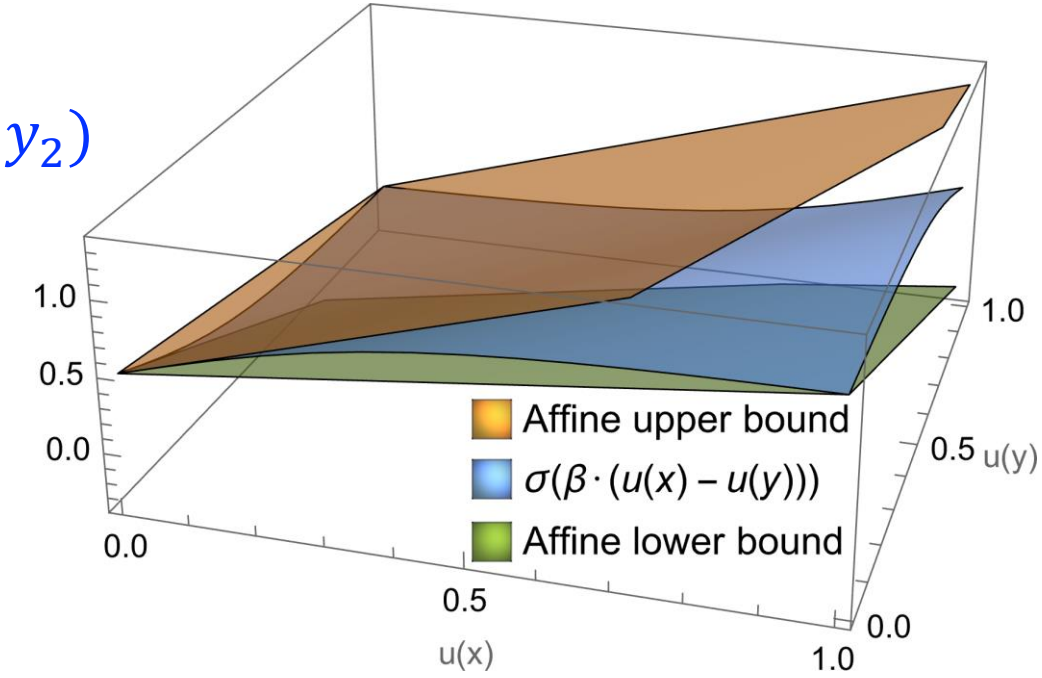
- If instead of Bradley-Terry, each user i voted candidate y as their most favorite, with probability $\propto u_i(y)$.
- **# of votes for y** is an unbiased estimator $\mathbb{E}_{i\sim\mathcal{P}}[u_i(y)]$.
- After sufficiently many users sampled, we can find y that **optimizes average utility.**

Linear Approximations of Bradley-Terry

Bradley-Terry is non-linear. In expectation it doesn't reveal total utility, rather, "preference intensity"

Lemma: approximate linearization

$$l_\beta \cdot \text{AvgUtil}(y_1) - L \cdot \text{AvgUtil}(y_2) \leq \underbrace{\mathbb{E}_{i\sim\mathcal{P}}[\Pr[y_1 \succ_i y_2]]}_{\text{Will concentrate to the expected win rate } p(y_1 \succ y_2)} - \frac{1}{2} \leq L \cdot \text{AvgUtil}(y_1) - l_\beta \cdot \text{AvgUtil}(y_2)$$



Aggregation via Maximal Lotteries

This method is inspired by "maximal lotteries" from social choice literature [Fishburn 1984]! Recently adopted in the post-training under the name **Nash Learning from Human Preferences** [Munos et al. 2023] and variants [Swamy et al., 2024, Wu et al., 2024].

It is a reward-free method:

- Estimates the win-rates: $\mathrm{M}_{a,b} = p(a \succ b) - p(b \succ a)$ over the all users.
- Outputs MinMax/Nash eq of the game with utilities $\mathrm{M}_{a,b}$, regularized or constrained.

$$\pi_{\mathrm{NLHF}} = \underset{\pi_1 \in \mathrm{KL}(\pi_{ref},\tau)}{\operatorname{argmax}} \underset{\pi_2 \in \mathrm{KL}(\pi_{ref},\tau)}{\min} \mathbb{E}_{(x,y)\sim\pi_1,\pi_2}[\mathrm{M}_{x,y}]$$

Distortion of Maximal Lotteries

Because this method is a Nash equilibrium of a symmetric 2 player zero-sum game, for the ***maximal lotteries*** policy we have

$$0 = \min_{\pi_2 \in \mathrm{KL}(\pi_{ref}, \tau)} \mathbb{E}_{(x,y)\sim \pi_{\mathrm{NLHF}}, \pi_2}[M_{ab}]$$

$$p(x \succ y) - p(y \succ x) = 2(p(x \succ y) - 0.5)$$

Instantiate π_2 as π^*: **the optimal constrained policy**, maximizing average utility.

$$0 \leq \mathbb{E}_{(x,y)\sim \pi_{\mathrm{NLHF}}, \pi^*}[p(x \succ y) - 0.5] \leq \beta \left(L_\beta . AvgUtil(\pi_{NLHF}) - \ell_\beta . AvgUtil(\pi^*) \right)$$

$$distortion: \frac{AvgUtil(\pi^*)}{AvgUtil(\pi_{NLHF})} \leq \frac{L_\beta}{\ell_\beta} \approx \frac{\beta}{2}$$

A Word on Optimization

RL from Human Feedback. vs. NL from from Human Feedback

$$\pi_{RLHF} = \underset{\pi_1 \in \mathrm{KL}(\pi_{ref},\tau)}{\operatorname{argmax}} \underset{\pi_2 \in \mathrm{KL}(\pi_{ref},\tau)}{\min} \mathbb{E}_{(x,y)\sim\pi_1,\pi_2}[p(x \succ y) - 0.5]$$

$$\text{Learn an } \hat{r} \text{ then } \pi_{RLHF} = argmax_{\pi \in \mathrm{KL}(\pi_{ref},\tau)} \mathbb{E}_{x\sim\pi}[\hat{r}(x)]$$

NLHF's performance guarantees hold, no matter what's the reference distribution or the regularization approach.

NLHF requires Minmax optimization. It can also be implemented by policy-gradient methods.

→ Can be implemented with a set of parameters via self-play

→ Either through learning $p(x \succ y)$ or Direct optimization methods

→ In practice, not significant difference in the efficiency of the implementation.

[Monus et al'24, Rosset et al '24]

The Story is about
Rewards and Optimization Interplay

Beyond RLHF