

# **DRPO: Doubly robust alignment for LLMs**

**Chengchun Shi**

Associate Professor of Data Science  
London School of Economics and Political Science

# What are large language models

ChatGPT 4o ▾

Share

What are large language models?

Large Language Models (LLMs) are powerful machine learning models trained to understand and generate human-like text. They are a type of **deep neural network**, typically based on the **transformer architecture**, and are trained on vast amounts of text data—books, articles, websites, and more.

**Key Characteristics:**

- **Large scale:** They have **billions (or even trillions)** of parameters. Parameters are the internal weights the model learns during training.
- **Pretrained:** LLMs are initially trained in a general way on large datasets using self-supervised learning. This phase helps them learn grammar, facts, reasoning, and even some world knowledge.

Ask anything

+ Search ...

Understanding Large Language Models Overview

What are large language models?

Large Language Models (LLMs) are advanced artificial intelligence (AI) systems designed to understand, generate, and manipulate human-like text. They are a type of **deep learning model** trained on vast amounts of text data to predict and produce coherent language based on input prompts.

**Key Features of Large Language Models:**

1. **Massive Scale**
  - Trained on billions or even trillions of words from books, articles, code, and web content.
  - Built using **neural networks** (typically **transformers**) with billions or trillions of parameters.
2. **Transformer Architecture**
  - Most LLMs use the **Transformer** model (introduced by Google in 2017), which relies on **self-attention mechanisms** to process words in context.
  - Examples: GPT (OpenAI), PaLM (Google), LLaMA (Meta), Claude (Anthropic).
3. **General-Purpose Capabilities**
  - Can perform a wide range of tasks:

给 DeepSeek 发送消息

深度思考 (R1) 联同搜索

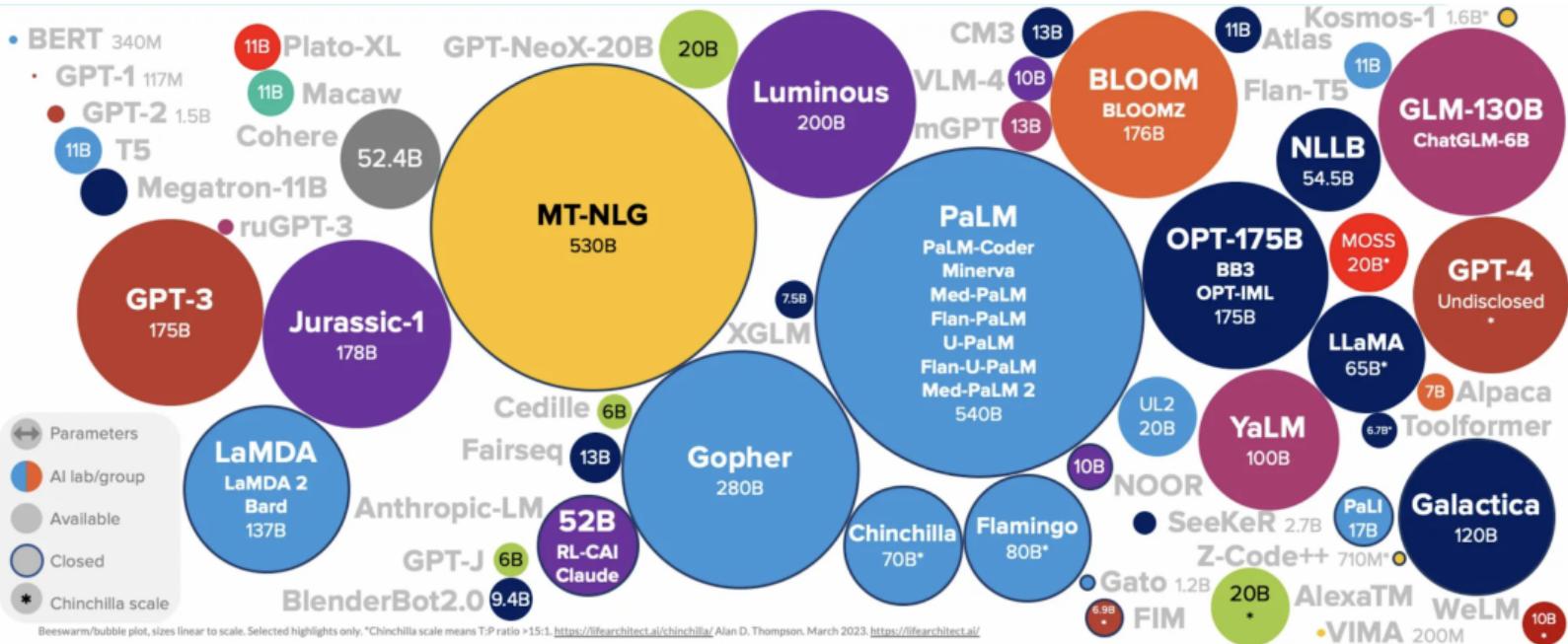
内容由 AI 生成。请仔细阅读



We can ask the LLMs themselves!



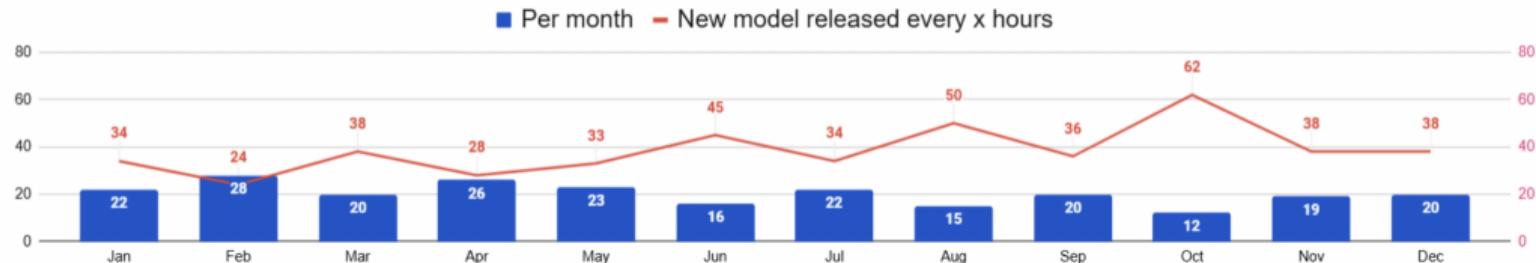
# Language model sizes to Mar/2023



# An era of LLMs

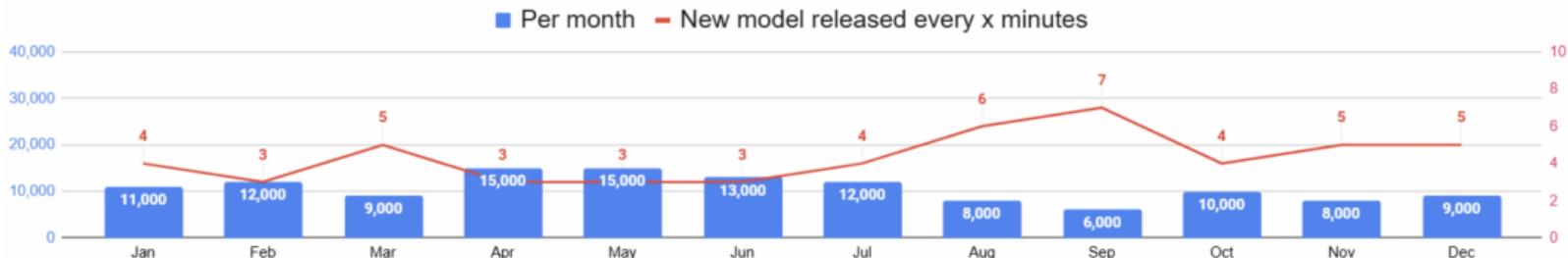
New major models released per month/x hours

LifeArchitect.ai/models (data from LifeArchitect.ai/models-table)



New derivative models released per month/x minutes

LifeArchitect.ai/models (data from Hugging Face)

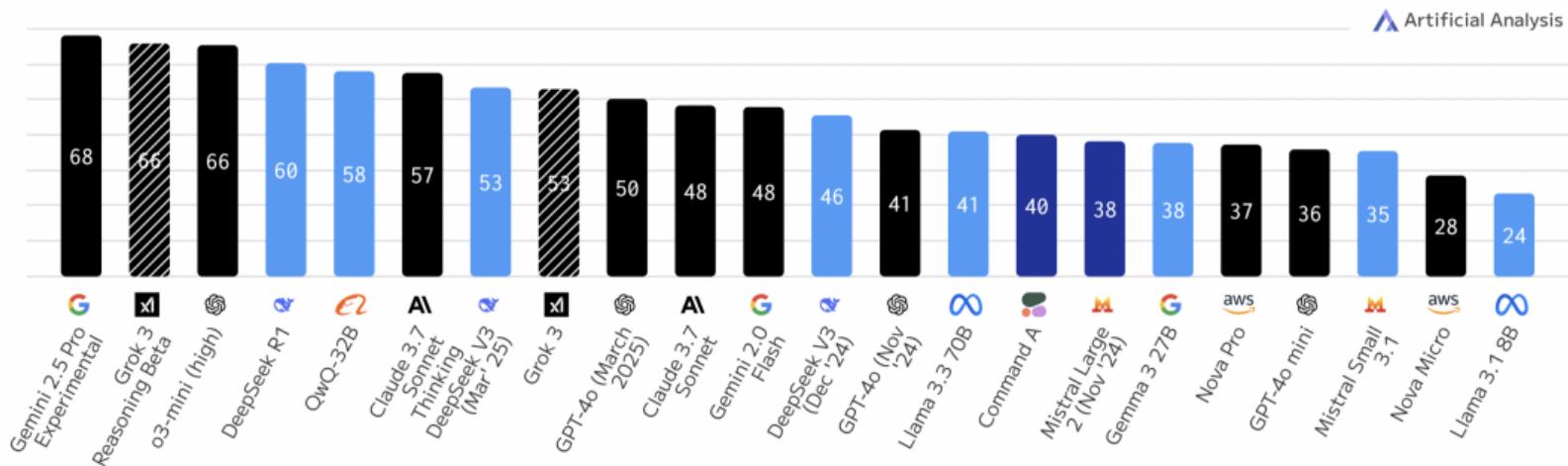


# Artificial analysis intelligence index

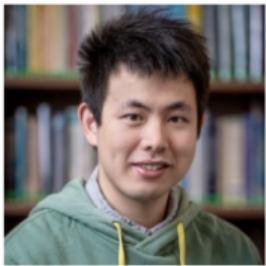
Intelligence Index incorporates 7 evaluations spanning reasoning, knowledge, math & coding

▨ Estimate (independent evaluation forthcoming)

■ Proprietary ■ Open Weights ■ Open Weights (Commercial Use Restricted)



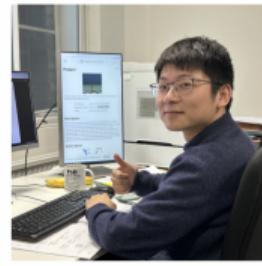
# DRPO: Doubly Robust Alignment for LLMs



Erhan Xu  
LSE



Kai Ye  
LSE



Hongyi Zhou  
THU



Francesco Quinlan  
Oxford

# How to train an LLM

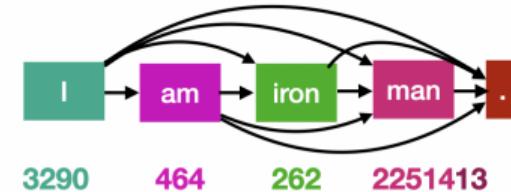
## Note

- $X$ : a sentence or prompt.
- $Y$ : responses.
- $Z$ :  $Z = \mathbb{I}(Y^{(2)} \succ Y^{(1)})$   
represents the resulting human feedback

## Pre-training



autoregressive  
next-token  
prediction



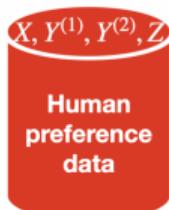
## Post-training



supervised  
fine-tuning

X: What is the capital of UK?

Y: London.



reinforcement learning  
from human feedback

X: What is the capital of UK?

Y<sup>(1)</sup>: France.

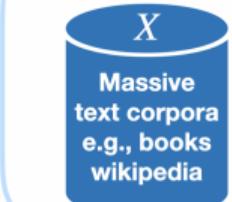
Y<sup>(2)</sup>: London.

# How to train an LLM

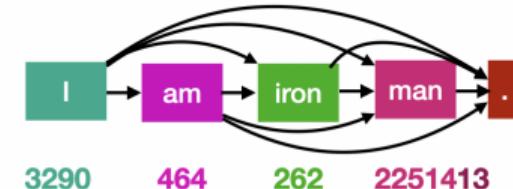
## Note

- $X$ : a sentence or prompt.
- $Y$ : responses.
- $Z: Z = \mathbb{I}(Y^{(2)} \succ Y^{(1)})$   
represents the resulting human feedback

## Pre-training



autoregressive  
next-token  
prediction



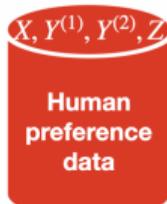
## Post-training



supervised  
fine-tuning

X: What is the capital of UK?

Y: London.



reinforcement learning  
from human feedback

X: What is the capital of UK?

$Y^{(1)}$ : France.

$Y^{(2)}$ : London.



# Reinforcement learning (RL)

---

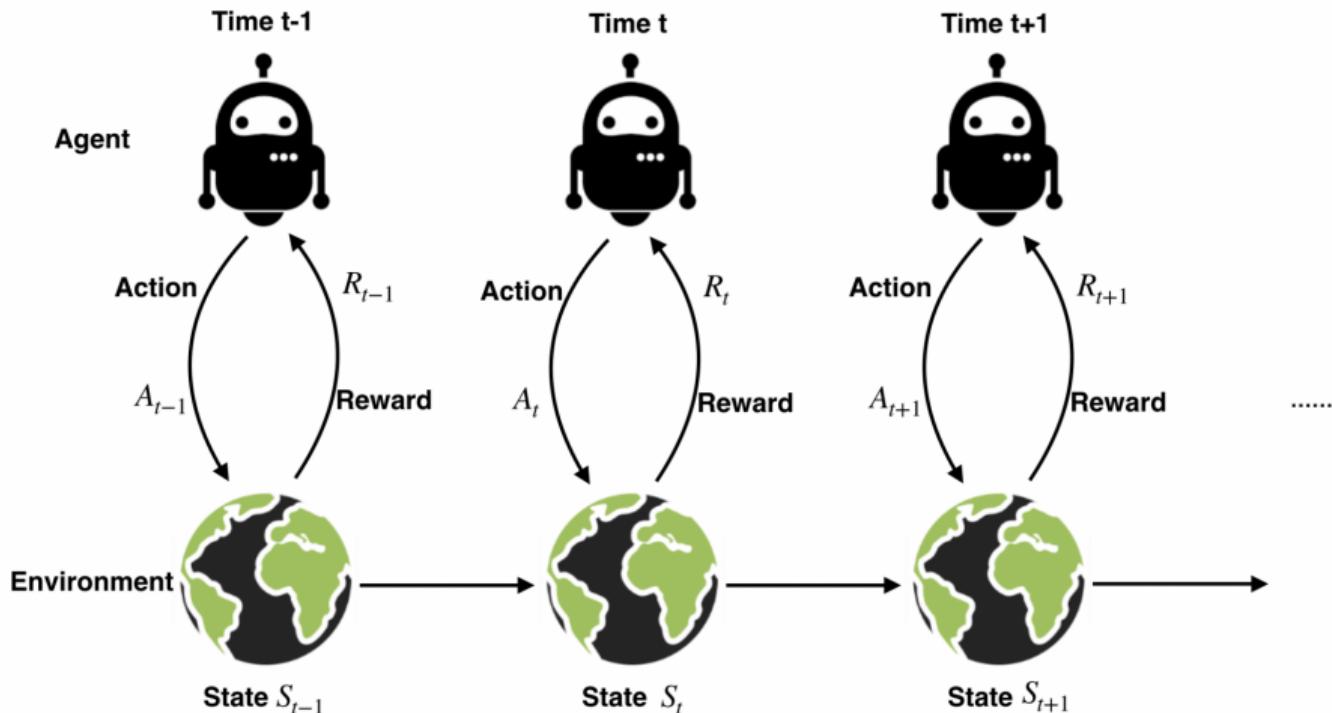
**Andrew Barto and  
Richard Sutton Receive  
A.M. Turing Award**



*The scientists received computing's highest honor for developing the theoretical foundations of reinforcement learning, a key method for many types of AI.*



# Reinforcement learning (Cont'd)



**Objective:** find an optimal policy that maximizes the cumulative reward

# Reinforcement learning from human feedback (RLHF)

---

2017

## Deep Reinforcement Learning from Human Preferences

Paul F Christiano  
OpenAI  
paul@openai.com

Jan Leike  
DeepMind  
leike@google.com

Tom B Brown  
nottombrown@gmail.com

Miljan Martic  
DeepMind  
miljanm@google.com

Shane Legg  
DeepMind  
legg@google.com

Dario Amodei  
OpenAI  
damodei@openai.com

First introduction to deep RLHF

2022

## Training language models to follow instructions with human feedback

Long Ouyang\* Jeff Wu\* Xu Jiang\* Diogo Almeida\* Carroll L. Wainwright\*  
Pamela Mishkin\* Chong Zhang Sandhini Agarwal Katarina Slama Alex Ray

John Schulman Jacob Hilton Fraser Kelton Luke Miller Maddie Simens

Amanda Askell† Peter Welinder Paul Christiano\*†

Jan Leike\* Ryan Lowe\*

OpenAI

First successful application of RLHF to LLM

# Reward learning in RLHF

---



Large language models  
need to align our  
preferences and values.

We can set the reward  
for the AI answers.

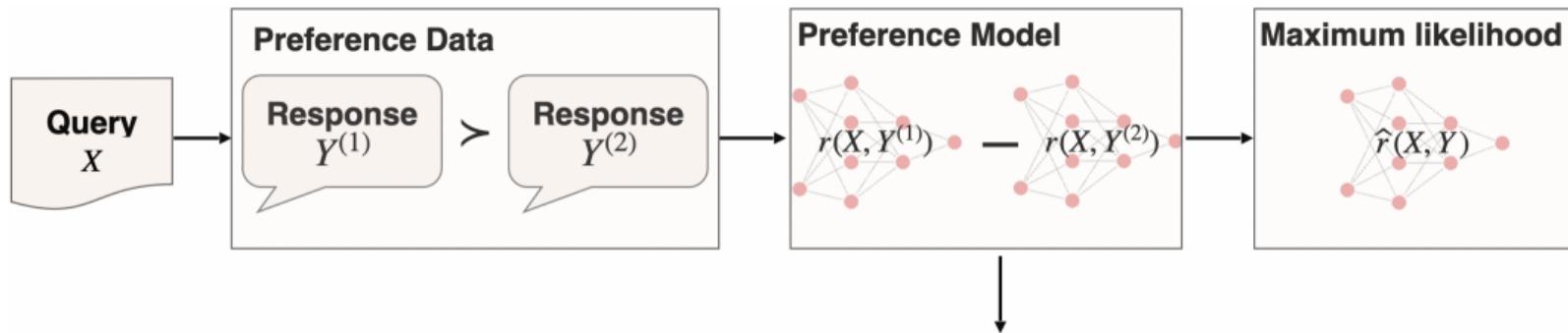
We compare two  
answers and tell which  
one is better. This is  
much easier than give  
the absolute scores. It  
does not require an  
absolute scale and is  
more intuitive for us.

How to get the reward in  
the learning process?



Evaluating complex  
tasks is inherently  
**difficult**, **Scale**  
**inconsistency**  
Models may struggle to  
learn from **sparse**  
**differences** in absolute  
scores.

# Reward learning in RLHF (Cont'd)



**Bradley-Terry (BT) model** (Bradley & Terry, 1952) is most widely adopted to model human preferences:

$$p(Y^{(1)} \succ Y^{(2)} | X) = \sigma(r(X, Y^{(1)}) - r(X, Y^{(2)}))$$

# BT model: an illustrative example

$$p(\text{tea} > \text{coffee}) = \frac{\exp(r(\text{tea}))}{\exp(r(\text{tea})) + \exp(r(\text{coffee}))}$$

Suppose **70%** of people like tea and **30%** of people like coffee. The reward model should satisfy:

$$\frac{1}{1 + \exp(r(\text{coffee}) - r(\text{tea}))} = 0.7 \longrightarrow r(\text{tea}) - r(\text{coffee}) = \log\left(\frac{7}{3}\right) = 0.847$$



# Baseline algorithm I: PPO-based approach

Step 1

Collect demonstration data, and train a supervised policy.

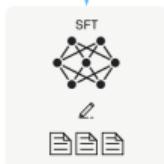
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



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



Step 2

Collect comparison data, and train a reward model.

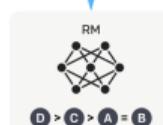
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.



Step 3

Optimize a policy against the reward model using reinforcement learning.

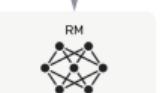
A new prompt is sampled from the dataset.



The policy generates an output.



The reward model calculates a reward for the output.



The reward is used to update the policy using PPO.

$r_k$

– from InstructGPT (Ouyang et al., 2022)

# Baseline algorithm II: DPO-based approach

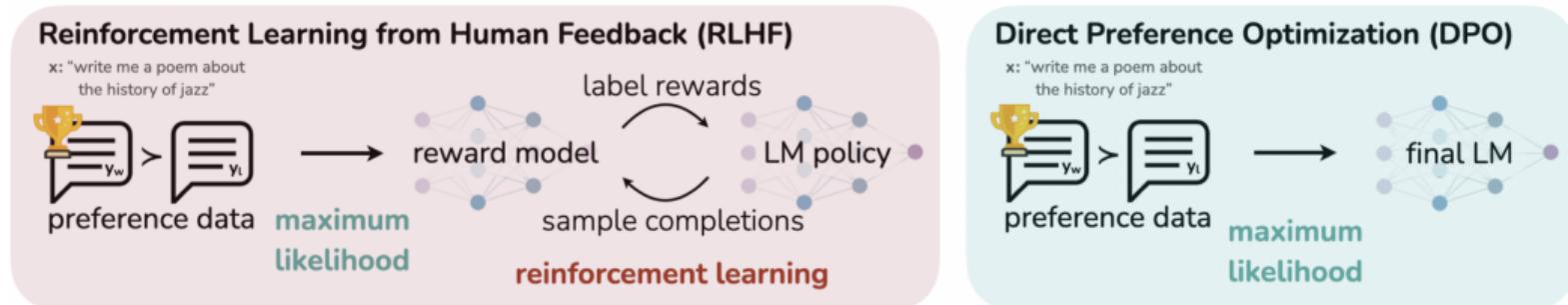


Figure 1: DPO optimizes for human preferences while avoiding reinforcement learning (Rafailov et al., 2023)

Reward function can be derived in closed-form using the optimal policy



$$r(y, x) = \beta \log\left(\frac{\pi^*(y|x)}{\pi_{ref}(y|x)}\right) + C(x)$$

# BT model can be misspecified

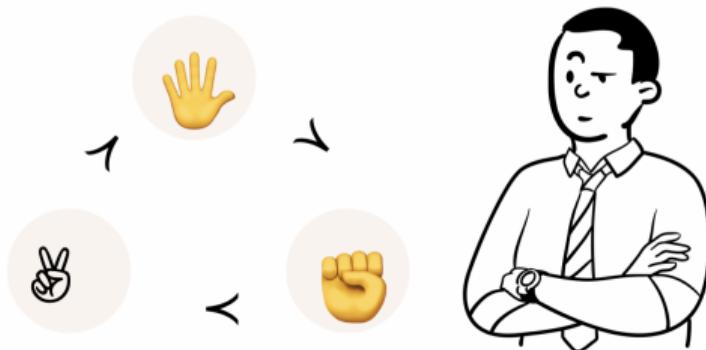
Both **PPO**- and **DPO**-based algorithms rely on **BT model** assumption for human preference modelling, which is likely violated due to **transitivity** ...

## What's the best way to learn a new language?

*Practice speaking daily and immerse yourself in the culture through media and conversation.*

*Use apps like Duolingo and review flashcards.*

*Join a local language group and travel to countries where the language is spoken.*



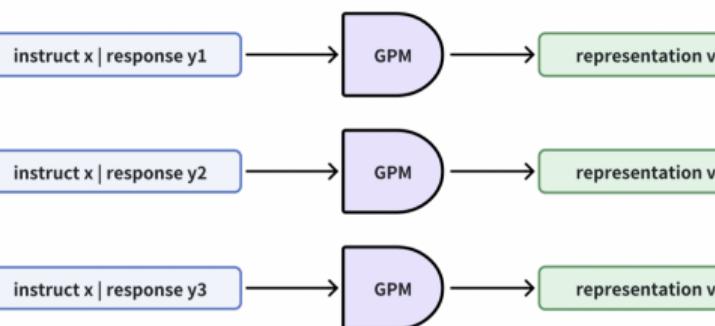
# Even when BT model is correct

---

- **PPO**-based algorithms are highly sensitive to the **reward model**. Misspecifying the reward can
  1. lead to reward hacking (Skalse et al., 2022; Laidlaw et al., 2024)
  2. misguide policy learning (Kaufmann et al., 2023; Zheng et al., 2023; Chen et al., 2024)
- **DPO**-based algorithms are highly sensitive to the **reference policy** (Liu et al., 2024; Gorbatovski et al., 2024; Xu et al., 2024)

# Baseline algorithm III: preference-based approach

General preference modelling (GPM, Zhang et al., 2024)



Preference Score

0	$s(y_1, y_2)$	$s(y_1, y_3)$
$s(y_2, y_1)$	0	$s(y_2, y_3)$
$s(y_3, y_1)$	$s(y_3, y_2)$	0

Nash learning from human feedback (NLHF, Munos et al., 2023)

$$\max_{\pi} \min_{\nu} \mathbb{E}_{y^{(1)} \sim \pi, y^{(2)} \sim \nu} p(y^{(1)} \succ y^{(2)})$$

Identity preference optimization (IPO, Azar et al., 2023)

$$\max_{\pi} \mathbb{E}_{y^{(1)} \sim \pi, y^{(2)} \sim \pi_{ref}} p(y^{(1)} \succ y^{(2)})$$

# Accurate preference model is vital

---

Many preference-based approaches do **not** require the BT model assumption. However, they still suffer from potential misspecification of **preference model**

*Should I start a pizzeria or sushi restaurant?*

## Preference: pizza vs sushi

- In Italy, 80% vs 20%
- In Japan, 10% vs 90%



– Taken from Weijie's slides

# In summary, all three baseline algorithms suffer from certain model misspecification

---

	Robust to misspecified:	preference model	reward model	reference policy
RLHF	Reward-based	PPO-based DPO-based	X X	X ✓
		IPO	✓	-
	Preference-based	GPM	X	-
		<b>DRPO</b>	✓	✓

Table: Robustness of different algorithms to model misspecification. Our algorithm is denoted by DRPO, short for doubly robust preference optimization.

# Our contribution

---



## Methodology

1. Propose a robust and efficient estimator for preference evaluation
2. Leveraging this estimator, develop a doubly robust preference optimization (DRPO) algorithm for LLM fine-tuning



## Theory

1. Doubly robust
2. Statistically efficient



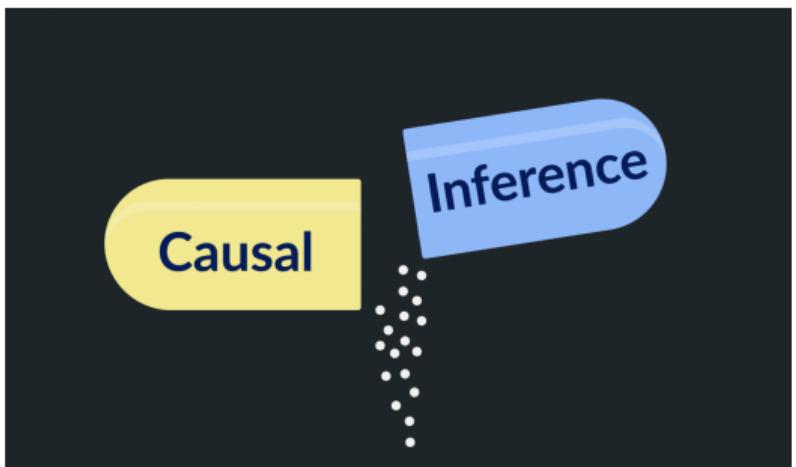
## Application to LLMs

Superior and more robust performance than both PPO- and DPO-based approaches

# Doubly robust (DR) methods

---

Doubly robust methods originate from the **missing data** and **causal inference** literature (see e.g., Robins et al., 1994; Scharfstein et al., 1999)



# Doubly robust methods (Cont'd)

Consider the estimation of **average treatment effect** (ATE) in causal inference. These methods estimate two models:

- A **propensity score** model for treatment assignment mechanism
- Similar to **reference policy** in LLMs
- An **outcome regression** model for patient's outcome given treatment
- Similar to **reward model** in LLMs



- Consistency of the ATE estimator only requires **one** model to be correct
- When **both** are correct, the ATE estimator becomes **semiparametrically efficient**

# Doubly robust methods (Cont'd)

---

These methods were later extensively studied and extended to

- Dynamic treatment regimes (Zhang et al., 2012; 2013)
- Off-policy learning and evaluation (Dudik et al., 2014)
- Causal machine learning (Chernozhukov et al., 2018)
- Conditional independence testing (Shah and Peters, 2020)
- Reinforcement learning (Kallus and Uehara, 2022; Liao et al., 2022)

# When DR methods meet LLMs

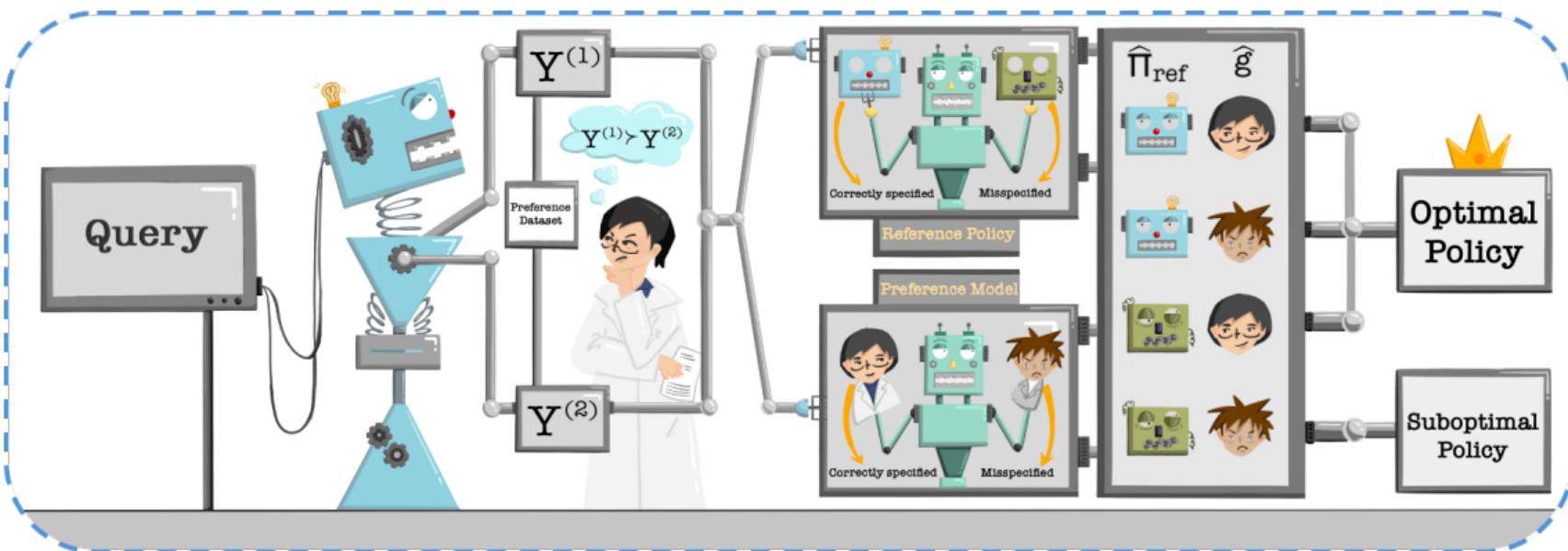


Figure: a summary of our methodology.  $\hat{\pi}_{ref}$  denotes the estimated reference policy and  $\hat{g}$  denotes the estimated preference model.

# When DR methods meet LLMs (Cont'd)

---

- **Preference evaluation:** for any target policy  $\pi$ , evaluate its **total preference**

$$p(\pi) = \mathbb{E}_{y^{(1)} \sim \pi, y^{(2)} \sim \pi_{ref}} p(y^{(1)} \succ y^{(2)})$$

We estimate two models from the data:

1. a preference model
2. a reference policy

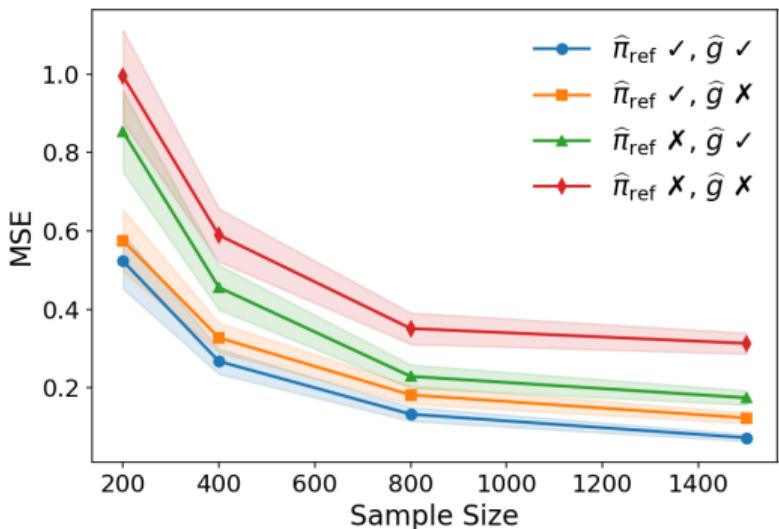
and develop a **doubly robust** and **semiparametrically efficient** estimator  $\hat{p}(\pi)$

- **Preference optimization:**

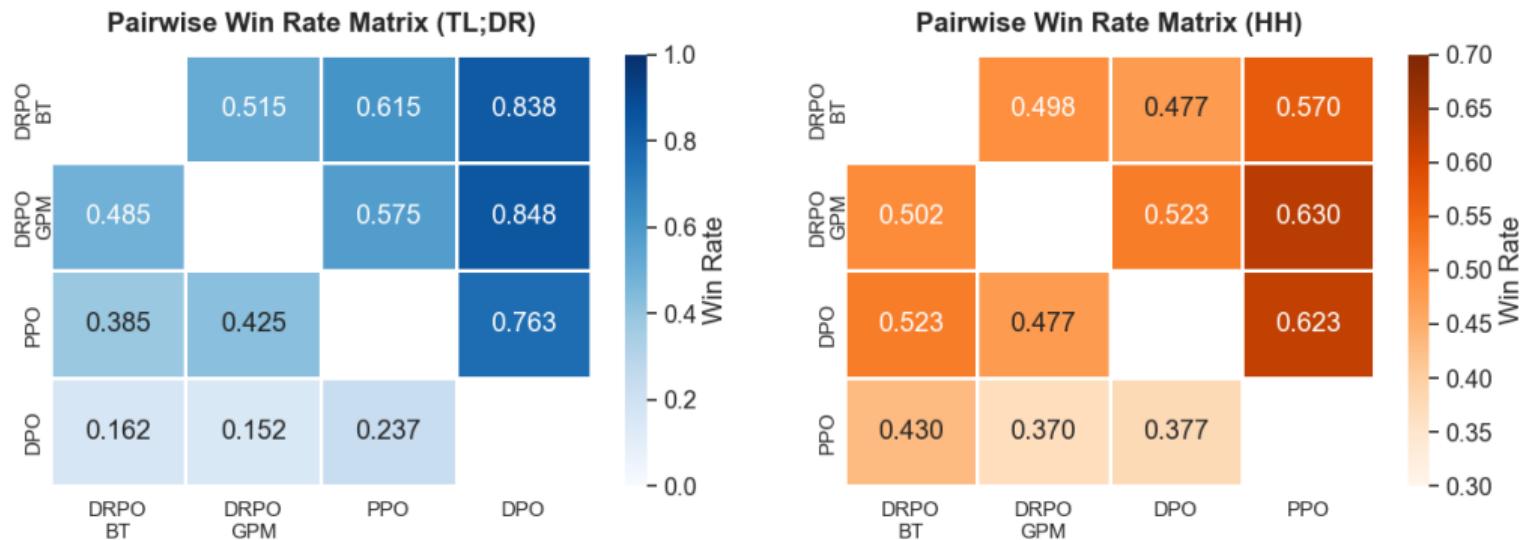
$$\hat{\pi} = \arg \max_{\pi} \hat{p}(\pi) - \beta \text{KL}(\pi, \hat{\pi}_{ref})$$

# Application to IMDb dataset

- **Task:** produce positive movie reviews
- **Objective:** evaluate total preference of a DPO-trained policy over a SFT-based reference policy
- **Ground truth:** 0.681

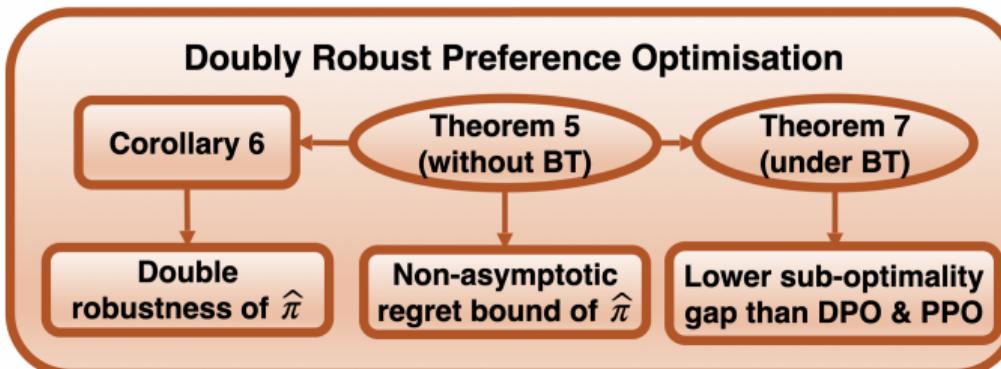
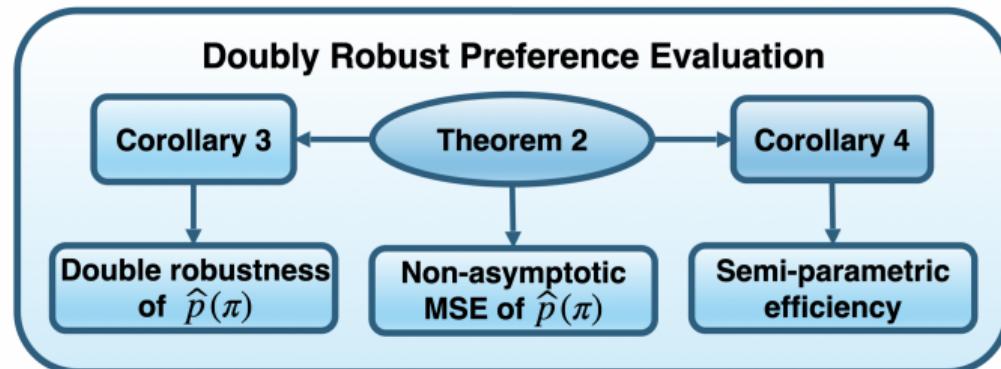


# Applications to TL;DR and HH datasets



**Figure: Pairwise win rate** matrices between different methods across two datasets. **Left:** TL;DR dataset. **Right:** HH dataset. Each entry indicates how often the row method outperforms the column method; higher values denote better performance.

# A summary of our theory



# More details

---

- Preference evaluation

- Double robustness of  $\hat{p}(\pi)$ : MSE of  $\hat{p}(\pi)$  decays to zero when either reference policy or preference model (not necessarily both) is correct
- Semiparametric efficiency: When both models are “approximately” correct,  $\hat{p}(\pi)$  achieves the efficiency bound (the smallest-possible MSE one can hope for  $p(\pi)$ )

- Preference optimization

- Double robustness of  $\hat{\pi}$ : Regret of  $\hat{\pi}$  decays to zero when either reference policy or preference model (not necessarily both) is correct

- Sub-optimality gaps:

- PPO:  $O(n^{-1/2} + \|\hat{r} - r\|)$
- DRPO:  $O(n^{-1/2} + \|\hat{r} - r\| \|\hat{\pi}_{ref} - \pi_{ref}\|)$
- DPO:  $O(n^{-1/2} + \|\hat{\pi}_{ref} - \pi_{ref}\|)$

# Thank You!

---



Papers can be found on my personal website

[callmespring.github.io](https://callmespring.github.io)