CS 329X: Human Centered LLMs

Learning from Human Preferences

Diyi Yang

Announcements

- Get started with hot-take debates
- Sign up for Discussion questions
- Discussion prompt released today
- Project tips

Homework 1

- Deadline: **Oct 14th, 11:59pm PT**. Upload through Canvas.
- **In HW1, you will:** learn about pre-training and fine-tuning; annotate your own preferences; and then personalize LLM to your own preference data!
- **Important:** Take annotation seriously!

Discussion (5%)

- Write responses to the debate prompt for **4** lectures
  - One paragraph, in your own words
  - Debate prompt posted two days before each class

- **Or** do 1 in-class debate + 1 additional reading response
  - Format:
    - Prior Vote (2 mins)
    - A(2 mins) B(2 mins) A(2 mins) B(2 mins) B(1 min) A(1 min)
    - Final vote (2 mins)
    - Winning team will get the 1 additional reading response waived

- Sign up for discussion via this form (on Ed Forum as well)

Outline

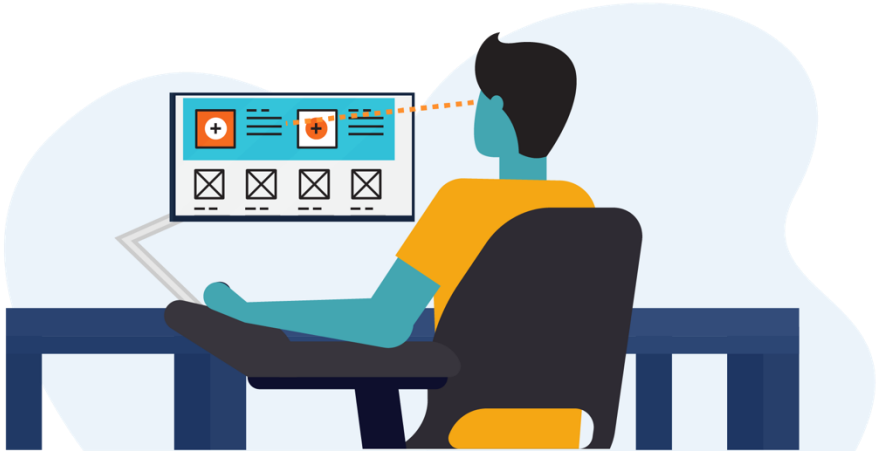
- ❑ **Different type of human feedback** (5 mins)
- ❑ **RLHF** (15 mins)
- ❑ **DPO + many others** (15 mins)
- ❑ **Limitations of human feedback** (5 mins)

Learning Objective: RLHF, DPO

Different Types of Human Feedback

- ⭐ Labeled data points
- ⭐ Edit data points
- ⭐ Change data weights
- ⭐ Binary/scaled user feedback
- ⭐ Natural language feedback
- ⭐ Code language feedback
- ⭐ Define, add, remove feature spaces
- ⭐ Directly change the objective function
- ⭐ Directly change the model parameter
- ⭐ ...

Different Types of Human Feedback



https://eyetechds.com/what-is-eye-tracking/

Eye Tracking

Touch

Talk and See

Incorporating Human Preferences into Learning

Transform **human "preferences"** into **usable model "language"**

- Allow humans to easily provide feedback
- Build models to effectively take the feedback

Figure credit to Chen, Valerie, et al. "Perspectives on Incorporating Expert Feedback into Model Updates." *ArXiv* (2022).

Incorporating Human Feedback into Learning

$$\hat{\theta} = argmax \sum_{(x,y) \in D} L(x, y; \theta)$$

- **Dataset updates:** change the dataset
- **Loss function updates:** add a constraint to the objective
- **Parameter space updates:** change the model parameters

Incorporating Human Feedback into Learning

$$\hat{\theta} = argmax \sum_{(x,y) \in D} L(x, y; \theta)$$

- **Dataset updates:** change the dataset
- **Loss function updates:** add a constraint to the objective
- **Parameter space updates:** change the model parameters

Case Study: Adapting LLMs to Society

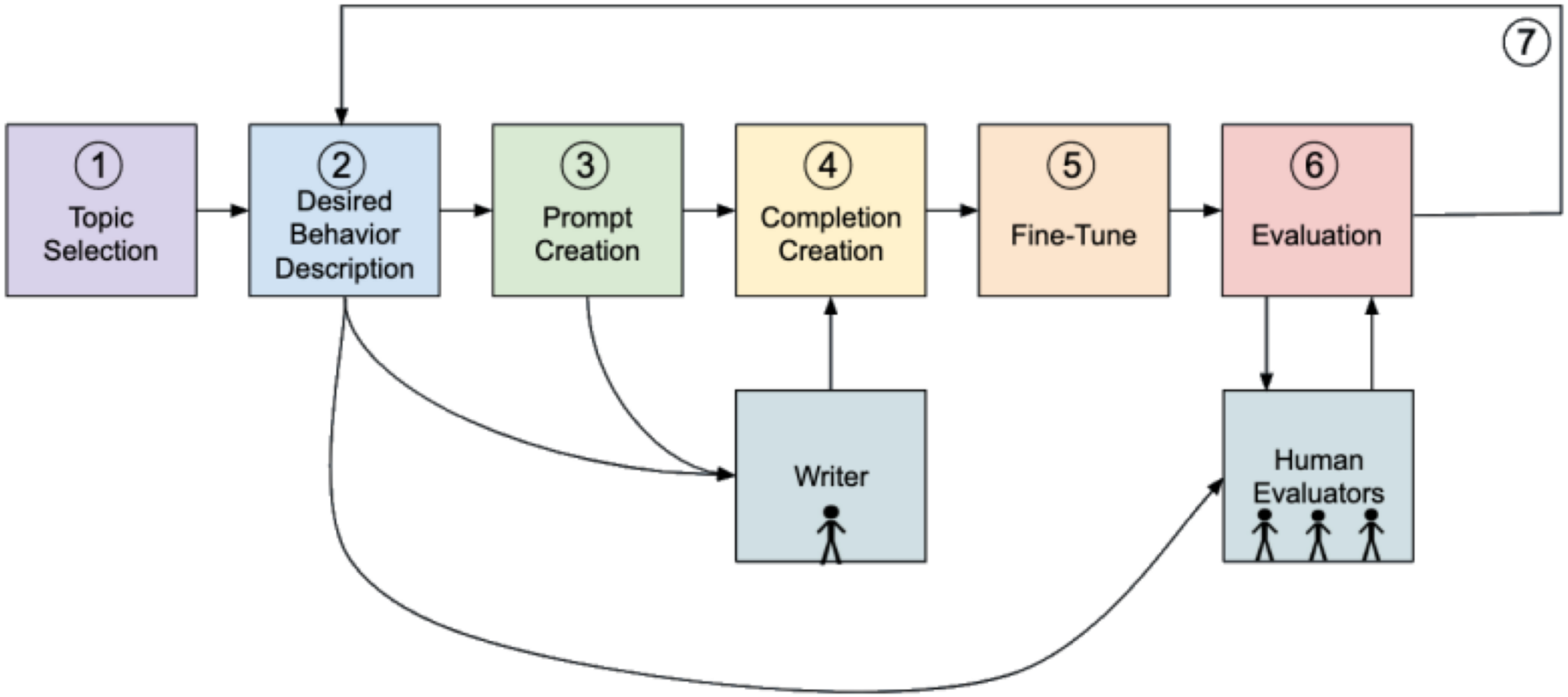
Main idea: adjust the behavior of a pertained language model to be sensitive to predefined norms with value-targeted datasets

Key steps:

- Choose sensitive topics
- Describe the language model's desired behavior
- Write prompts with value-targeted question-answer pairs

Solaiman, Irene, and Christy Dennison. "Process for adapting language models to society (palms) with values-targeted datasets." Advances in Neural Information Processing Systems 34 (2021): 5861-5873.

Case Study: Adapting LLMs to Society



Solaiman, Irene, and Christy Dennison. "Process for adapting language models to society (palms) with values-targeted datasets." Advances in Neural Information Processing Systems 34 (2021): 5861-5873.

Incorporating Human Feedback into Learning

$$\hat{\theta} = argmax \sum_{(x,y) \in D} L(x, y; \theta)$$

✓**Dataset updates:** change the dataset

- **Loss function updates:** add a constraint to the objective

- **Parameter space updates:** change the model parameters

Loss Function Updates: **Unlikelihood Learning**

Penalize undesirable generations (e.g. not following control, repeating previous context)

$$\mathcal{L}_{ULE}^{t} = \mathcal{L}_{MLE}^{t} + \alpha \mathcal{L}_{UL}^{t}$$

$$\mathcal{L}_{UL}^{t} = -\sum_{y_{neg} \in C} \log(1 - P(y_{neg} \mid \{y^*\}_{<t}))$$

If C is previously seen text, then less repetition and more diversity

Welleck, Sean, et al. "Neural text generation with unlikelihood training." ICLR (2019).

Incorporating Human Feedback into Learning

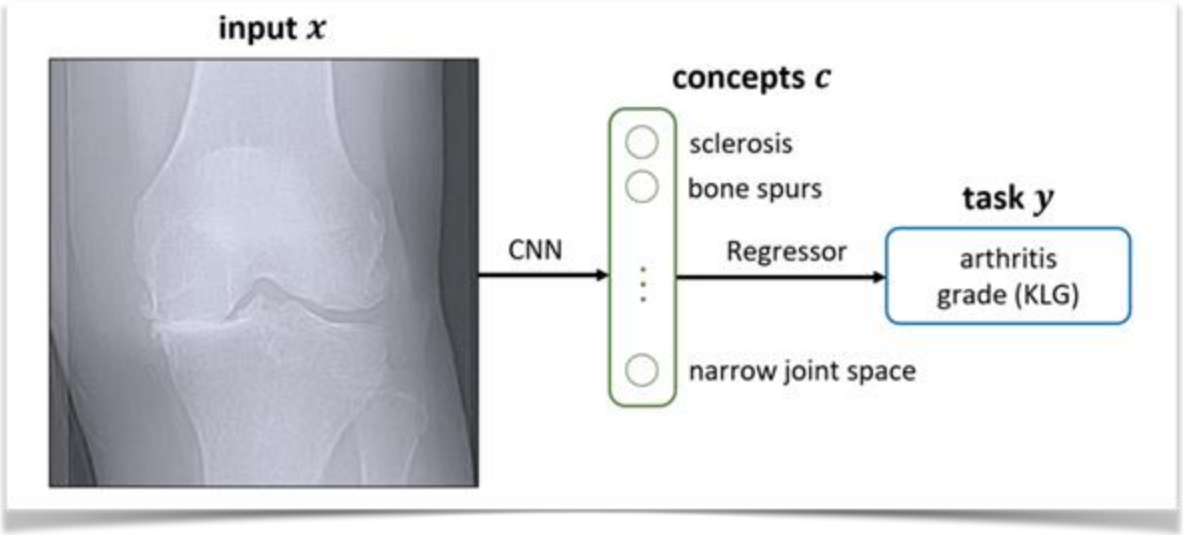
$$\hat{\theta} = argmax \sum_{(x,y) \in D} L(x, y; \theta)$$

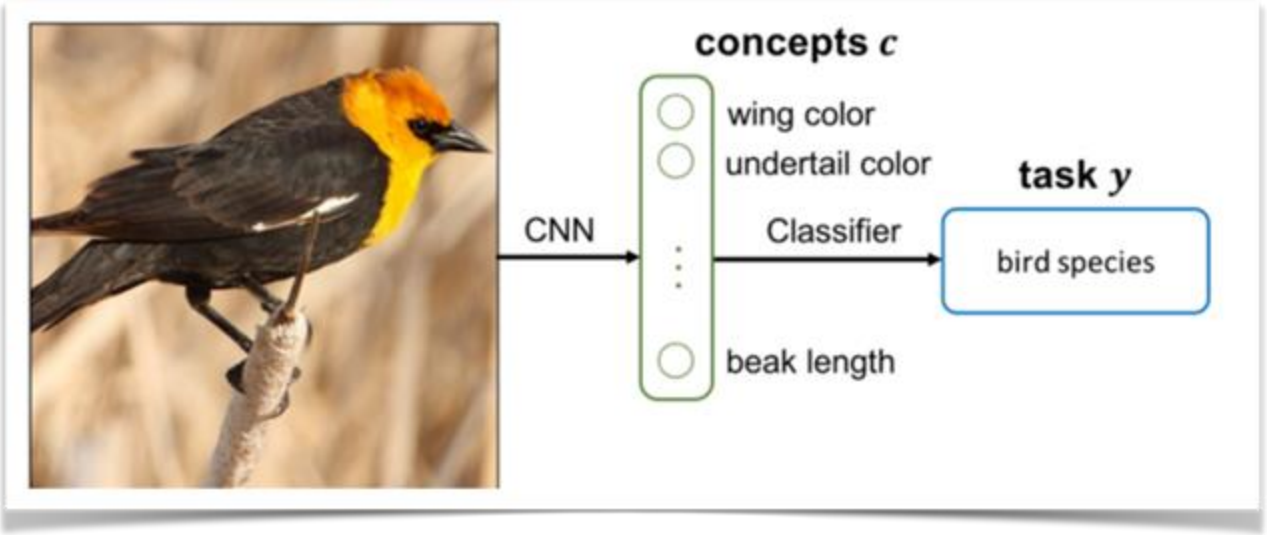
✓**Dataset updates:** change the dataset

✓**Loss function updates:** add a constraint to the objective

- **Parameter space updates:** change the model parameters

Parameter Updates: **Concept Bottleneck Model** trains model to explicitly use human-provided concepts

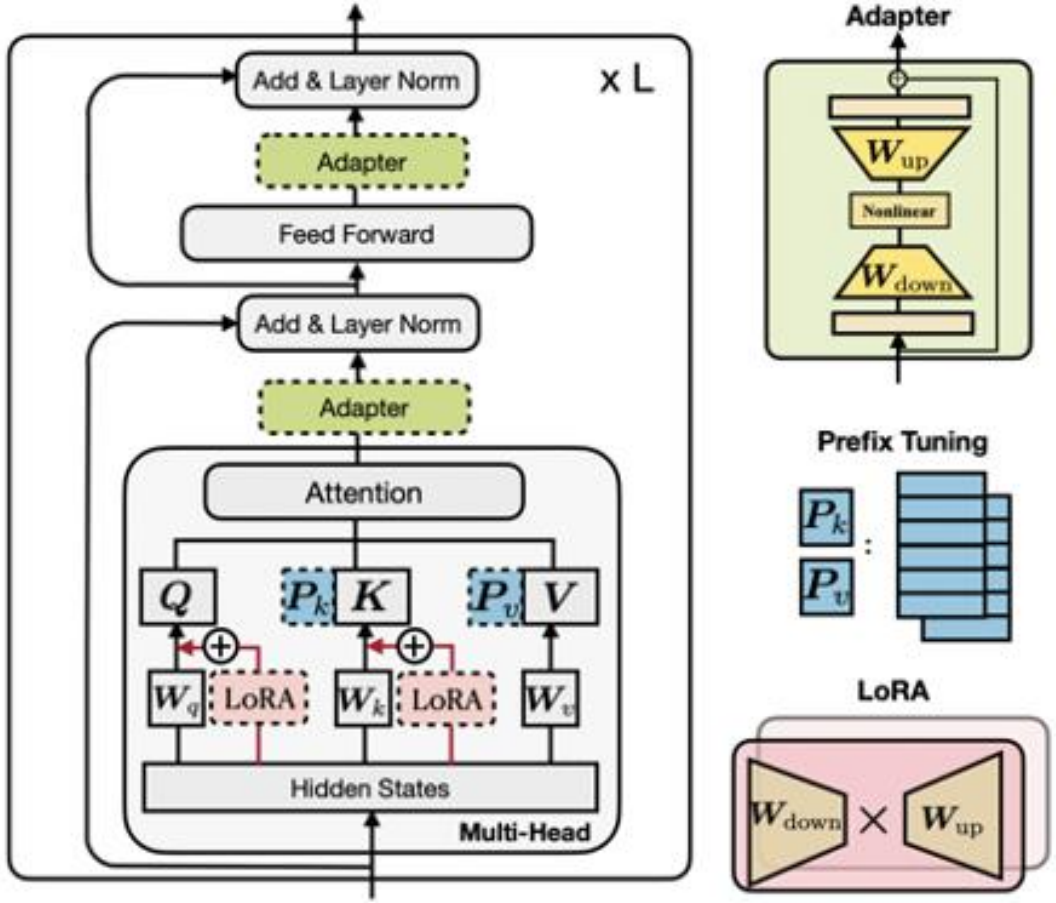




Koh, Pang Wei, et al. "Concept bottleneck models." International Conference on Machine Learning. PMLR, 2020.

Parameter Updates: **Parameter Efficient Fine-tuning**

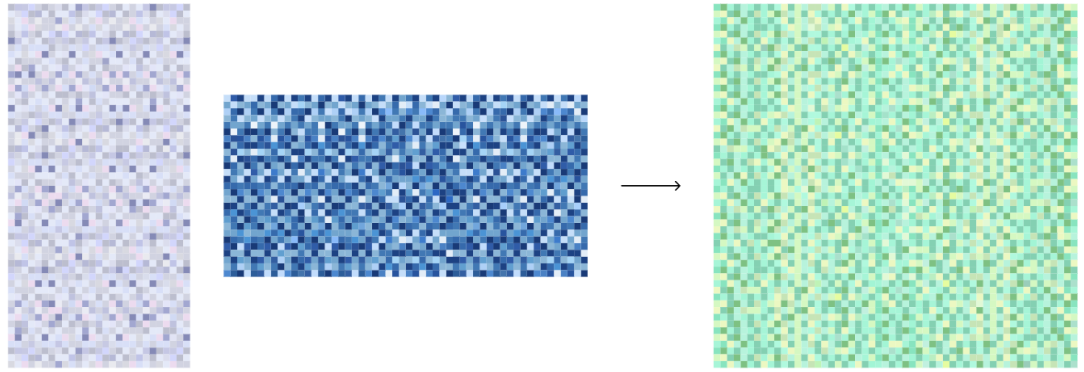
uses small interaction data to steer models towards desired behaviors



LoRA Without Regret

John Schulman in collaboration with others at Thinking Machines

Sep 29, 2025



Incorporating Human Feedback into Learning

$$\hat{\theta} = argmax \sum_{(x,y) \in D} L(x, y; \theta)$$

✓**Dataset updates:** change the dataset

✓**Loss function updates:** add a constraint to the objective

✓**Parameter space updates:** change the model parameters

Outline

- ✓ **Different type of human feedback** (5 mins)
- ❑ **RLHF** (15 mins)

How much water is used?

You're giving feedback on a new version of ChatGPT.

Which response do you prefer? Responses may take a moment to load.

You're giving feedback on a new version of ChatGPT.

Which response do you prefer? Responses may take a moment to load.

Response 1

Got it — feel free to share the next case study or let me know how else you'd like to use these summaries.

I prefer this response

Response 2

Error while searching

I prefer this response

Optimizing for Human Preferences

- Let's say we were training a language model on some task (e.g. summarization).
- For each prompt–output pair (q, o), imagine we had a way to obtain a ***human reward*** of that summary: $R(q, o) \in \mathbb{R}$, higher is better.

`SAN FRANCISCO, California (CNN) -- A magnitude 4.2 earthquake shook the San Francisco ... overturn unstable objects.`	`An earthquake hit San Francisco. There was minor property damage, but no injuries.`	`The Bay Area has good weather but is prone to earthquakes and wildfires.`
	$R(q, o_1) = 8$	$R(q, o_2) = 1.2$
Prompt q	**Output o_1**	**Output o_2**

Optimizing for Human Preferences

- Let's say we were training a language model on some task (e.g. summarization).
- For each prompt–output pair (q, o), imagine we had a way to obtain a ***human reward*** of that summary: $R(q, o) \in \mathbb{R}$, higher is better.

- Now we want to maximize the expected reward of samples from our LM:

$$\mathbb{E}_{q \sim D(q)} \mathbb{E}_{o \sim p_\theta(o|q)} [R(q, o)]$$

Optimizing for human preferences

- How do we actually change our LM parameters θ to maximize this?

$$\mathbb{E}_{q\sim D(q),o\sim p_\theta(o|q)}[R(q,o)]$$

- Let's try doing gradient ascent!

$$\theta_{t+1} := \theta_t + \alpha \nabla_{\theta_t} \mathbb{E}_{q\sim D(q),o\sim p_\theta(o|q)}[R(q,o)]$$

How do we estimate this expectation??

What if our reward function is non-differentiable??

- **Policy gradient** methods in RL (e.g., REINFORCE; [Williams, 1992]) give us tools for estimating and optimizing this objective.

Problem 1: Human-in-the-loop is expensive!

- **Solution:** instead of directly asking humans for preferences, **model their preferences** as a separate (NLP) problem! [Knox and Stone, 2009]

`An earthquake hit San Francisco. There was minor property damage, but no injuries.`

o

$R(q, o_1) = 8.0$

`The Bay Area has good weather but is prone to earthquakes and wildfires.`

o_2

$R(q, o_2) = 1.2$

Train an LM RM to predict human preferences from an annotated dataset, then optimize for RM instead.

Problem 2: Human judgements are noisy and miscalibrated!

- **Solution:** instead of asking for direct ratings, ask for **pairwise comparisons**, which can be more reliable [Phelps et al., 2015; Clark et al., 2018]

```
A 4.2 magnitude
earthquake hit
San Francisco,
resulting in
massive damage.
```

$$o_3$$

$$R(q, o_3) = \ ?$$

$$R(q, o_3) = \quad 4.1? \quad 6.6? \quad 3.2?$$

Problem 2: Human judgements are noisy and miscalibrated!

`An earthquake hit San Francisco. There was minor property damage, but no injuries.`	>	`A 4.2 magnitude earthquake hit San Francisco, resulting in massive damage.`	>	`The Bay Area has good weather but is prone to earthquakes and wildfires.`
o_1		o_3		o_2

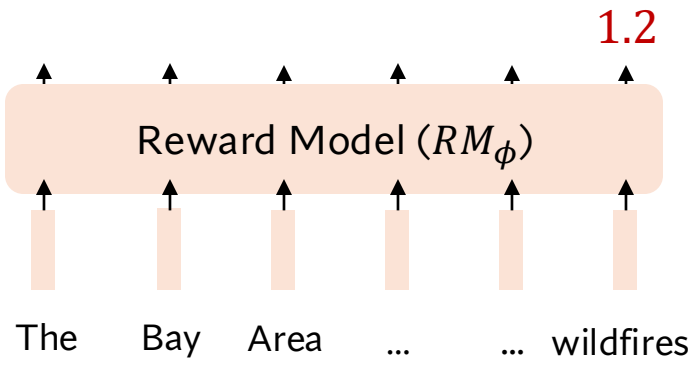
Bradley-Terry [1952] paired comparison model

$$J_{RM}(\phi) = -\mathbb{E}_{(o^w, o^l)\sim D}\left[\log \sigma(RM_\phi(q, o^w) - RM_\phi(q, o^l))\right]$$

"winning" sample

"losing" sample

o^w should score higher than o^l



RLHF: Putting it all together [Christiano et al., 2017; Stiennon et al., 2020]

Finally, we have everything we need:

- A pretrained (possibly instruction-finetuned) LM $p^{PT}(q, o)$
- A reward model $RM_{\phi}(q, o)$ that produces scalar rewards for LM outputs, trained on a dataset of human comparisons
- A method for optimizing LM parameters towards an arbitrary reward function.

RLHF: Putting it all together [Christiano et al., 2017; Stiennon et al., 2020]

- Now to do RLHF:
  - Initialize a copy of the model $p_\theta^{RL}(q,o)$
  - Optimize the following reward with RL:

$$R(q,o) = RM_\phi(q,o) - \beta \underbrace{\log\left(\frac{p_\theta^{RL}(q,o)}{p^{PT}(q,o)}\right)}$$

Pay a price when $p_\theta^{RL}(q,o) > p^{PT}(q,o)$

This is a penalty which prevents us from diverging too far from the pretrained model.
In expectation, it is known as the **Kullback-Leibler** (**KL**) divergence between $p_\theta^{RL}(q,o)$ and $p^{PT}(q,o)$.

Proximal Policy Optimization (PPO)

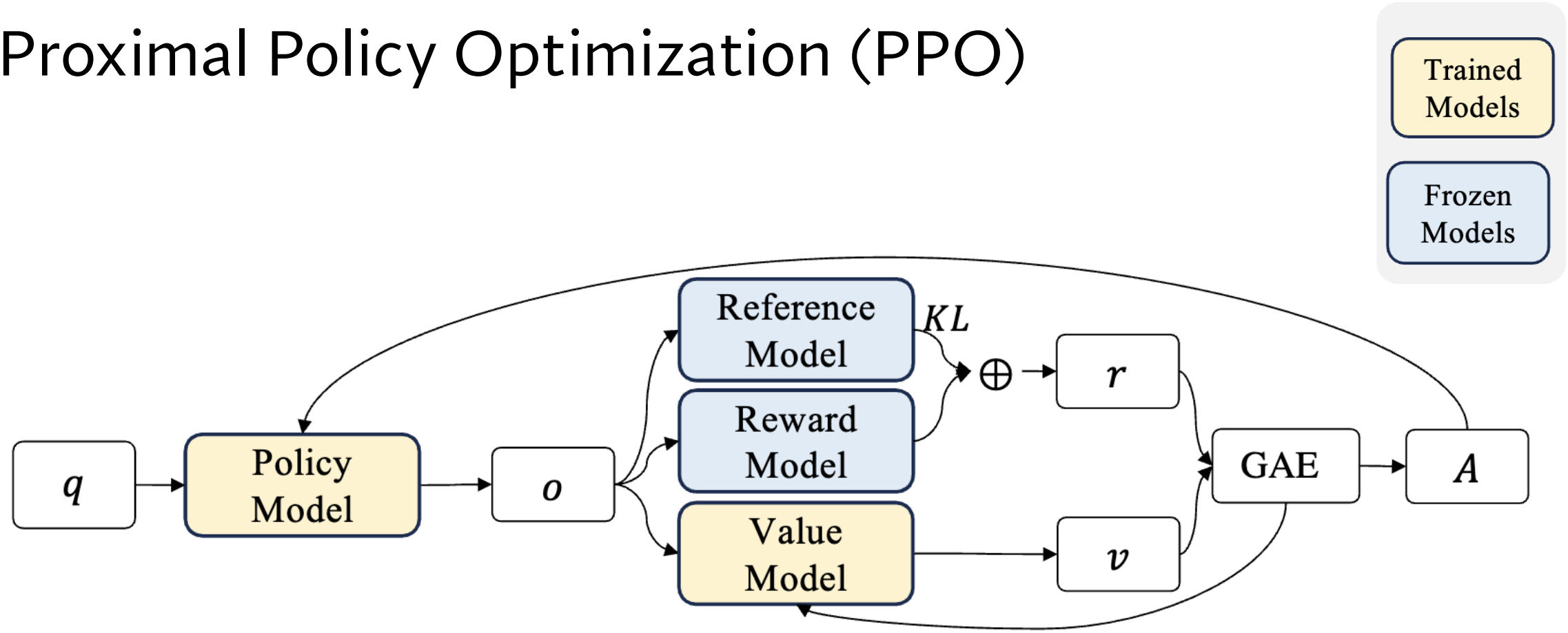
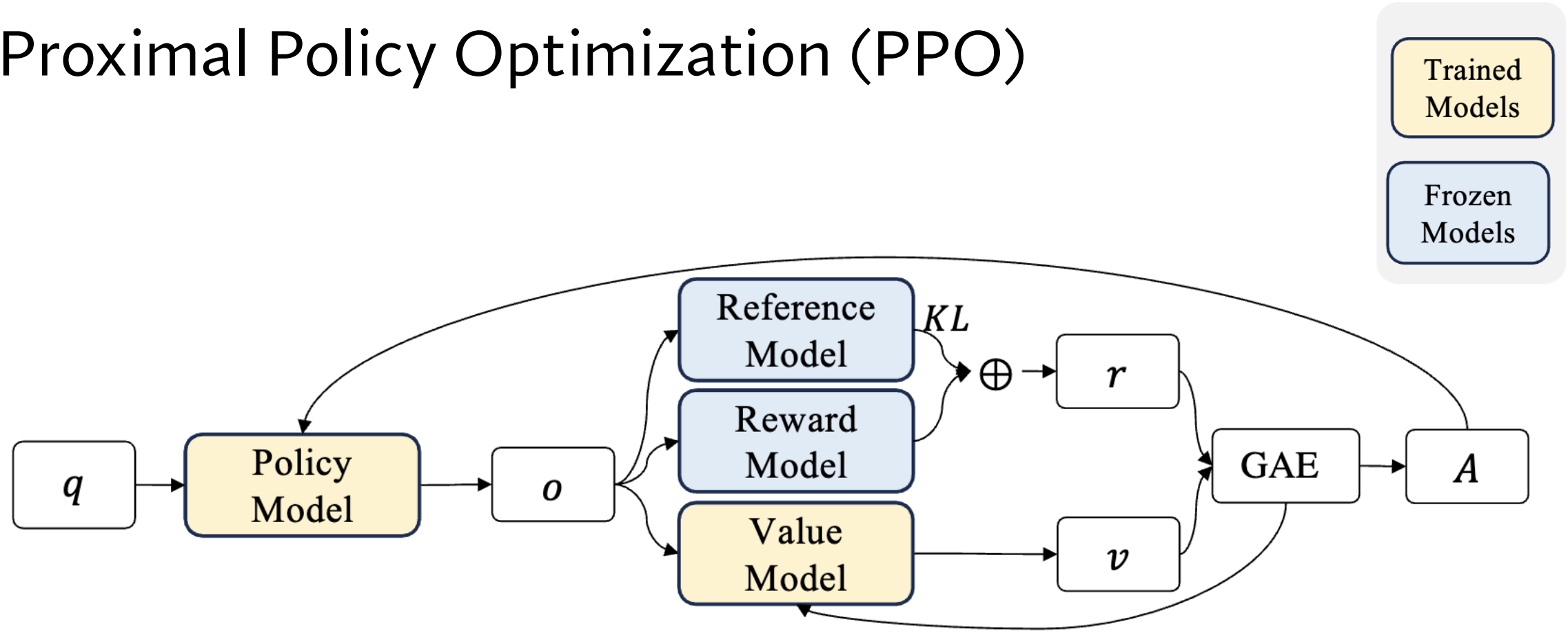


Figure credit to
Shao, et al., "Deepseekmath: Pushing the limits of mathematical reasoning in open language models." arXiv:2402.03300 (2024).

Proximal Policy Optimization (PPO)



Proximal Policy Optimization (PPO): policy update

For each token step t, define the PPO ratio: $r_t(\theta) = \frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q,o_{<t})}$

Clipped Objective: $L^{clip} = E_t[\min(r_t(\theta)\hat{A}_t, clip\ (r_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t)]$

Algorithm PPO with Clipped Objective

Input: initial policy parameters θ_0, clipping threshold ϵ
for $k = 0, 1, 2, \ldots$ **do**
Collect set of partial trajectories $\mathcal{D}_k$ on policy $\pi_k = \pi(\theta_k)$
Estimate advantages $\hat{A}_t^{\pi_k}$ using any advantage estimation algorithm
Compute policy update

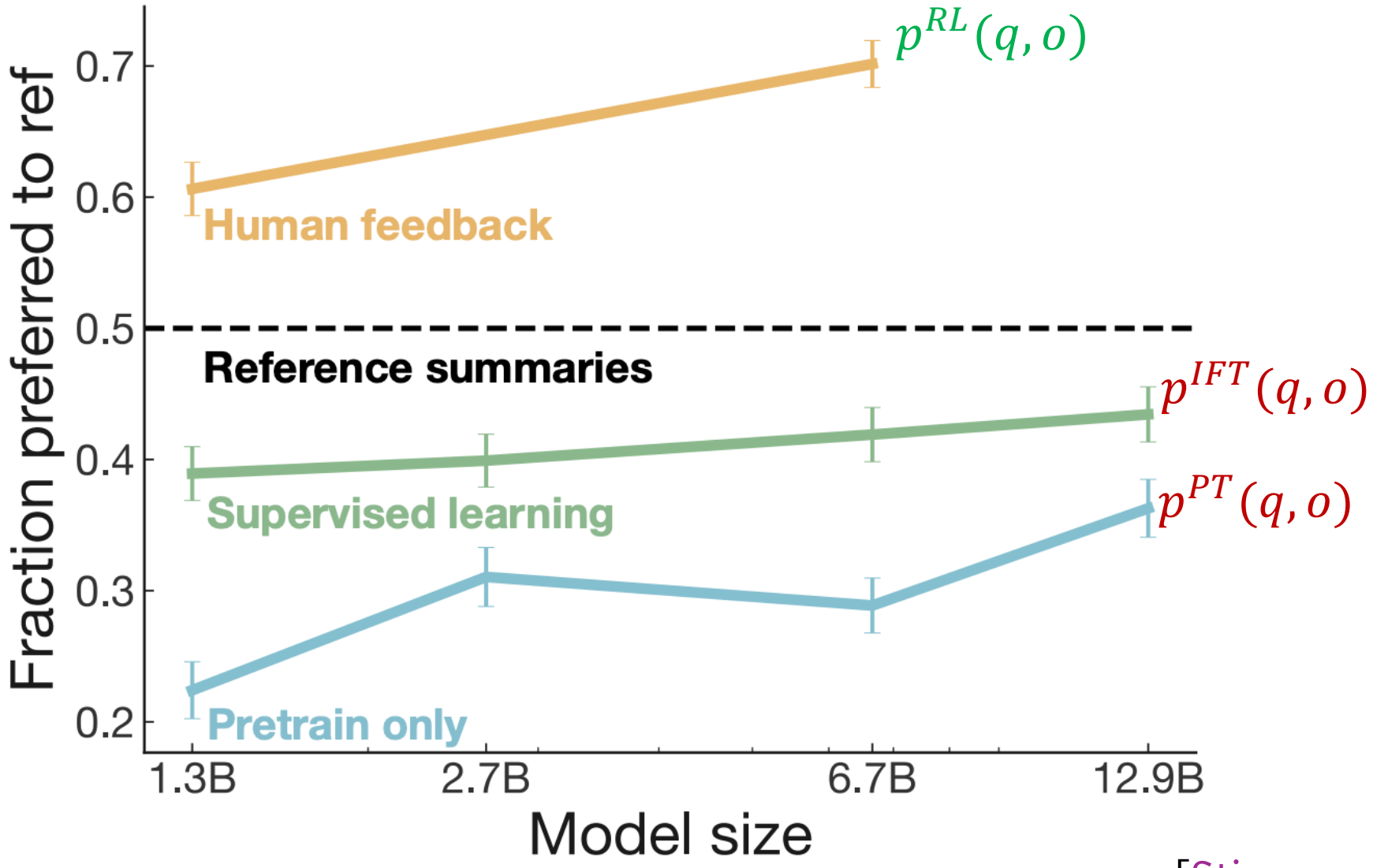
$$\theta_{k+1} = \arg\max_\theta \mathcal{L}_{\theta_k}^{CLIP}(\theta)$$

by taking K steps of minibatch SGD (via Adam), where

$$\mathcal{L}_{\theta_k}^{CLIP}(\theta) = \mathop{\mathrm{E}}_{\tau\sim\pi_k}\left[\sum_{t=0}^{T}\left[\min(r_t(\theta)\hat{A}_t^{\pi_k}, \text{clip}\,(r_t(\theta), 1-\epsilon, 1+\epsilon)\,\hat{A}_t^{\pi_k})\right]\right]$$

For details on algorithm, check out https://web.stanford.edu/class/cs234/slides/lecture6post.pdf

RLHF provides gains over pretraining + finetuning



[Stiennon et al., 2020]

Metaphor Challenge

Outline

- ✓ **Different type of human feedback** (5 mins)
- ✓ **RLHF** (15 mins)
- ❑ **DPO + many others** (15 mins)

Recap the RLHF Objective

$$J(\pi_\theta) = \mathbb{E}_{q\sim D(q), o\sim\pi_\theta(o|q)}[R(q,o)] - \beta D_{KL}(\pi_\theta(o|q) \parallel \pi_{ref}(o|q))$$

q: input

o: model output (response)

$\pi_\theta(o|q)$: policy we're optimizing

$\boldsymbol{R}(q,o)$: reward function based on human feedback

β: KL divergence regularization weight

Optimal Policy Under RLHF [Rafailov+ 2023]

Optimal Policy: closed-form solution from prior work

$$\pi_\theta^*(o|q) \propto \pi_{ref}(o|q) \exp(\frac{R(q,o)}{\beta})$$

Normalized Policy

$$\pi_\theta^*(o|q) = \frac{\pi_{ref}(o|q) \exp(\frac{R(q,o)}{\beta})}{Z(q)} \qquad Z(q) = \sum_{o'} \pi_{ref}(o'|q) \exp(\frac{R(q,o')}{\beta})$$

Log transformation: $R(q,o) = \beta \left(\log \pi_\theta^*(o|q) - \log \pi_{ref}(o|q)\right) + \beta \log Z(q)$

Putting it Together with DPO [Rafailov+ 2023]

Derived DPO reward model:

$$R(\mathrm{q}, o) = \beta(\log \pi_\theta^*(o|q) - \log \pi_{ref}(o|q)) + \beta \log Z(q)$$

The Bradley-Terry model of human preferences

$$L_R(r, D) = -\mathbb{E}_{(q,o_w,o_l)\sim D}[\log \sigma(R(q, o_w) - R(q, o_l))]$$

Log Z term cancels as the loss only measures differences in rewards

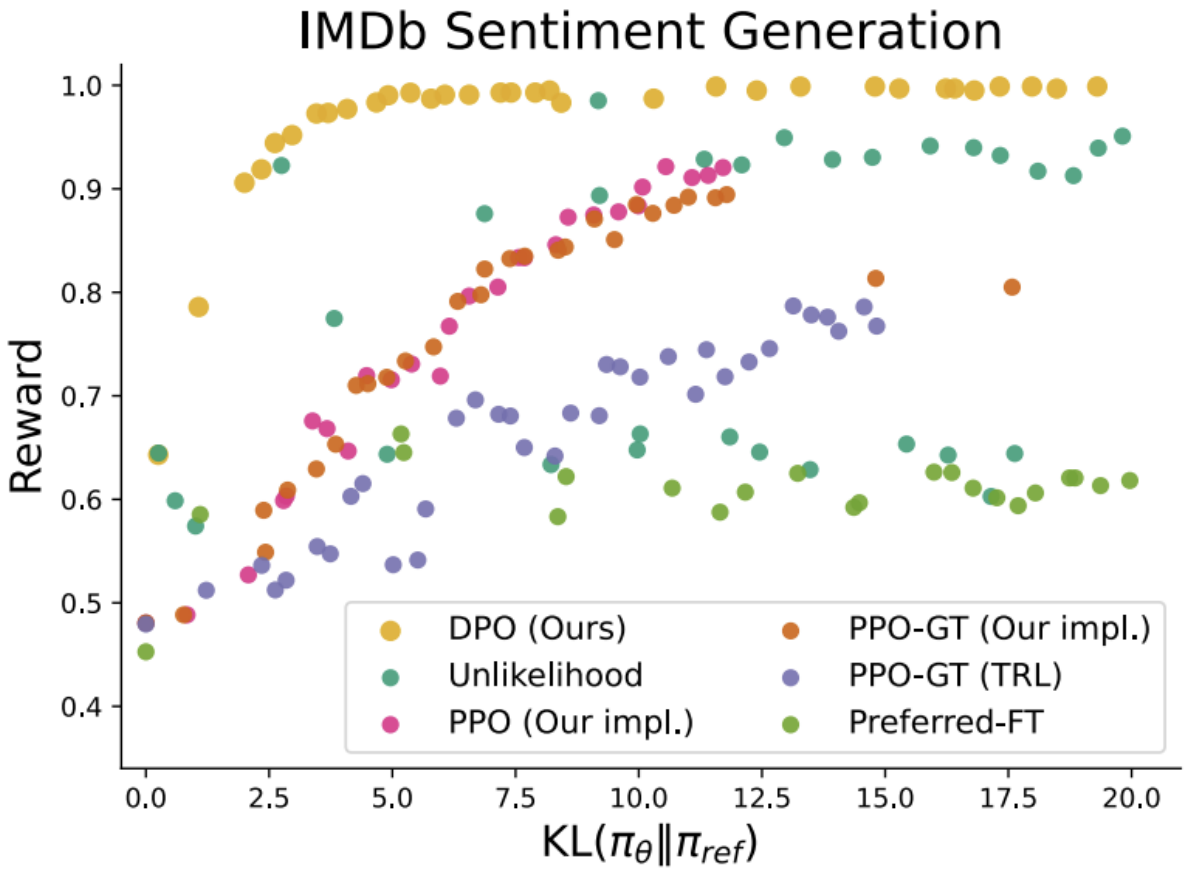
Final loss function for DPO:

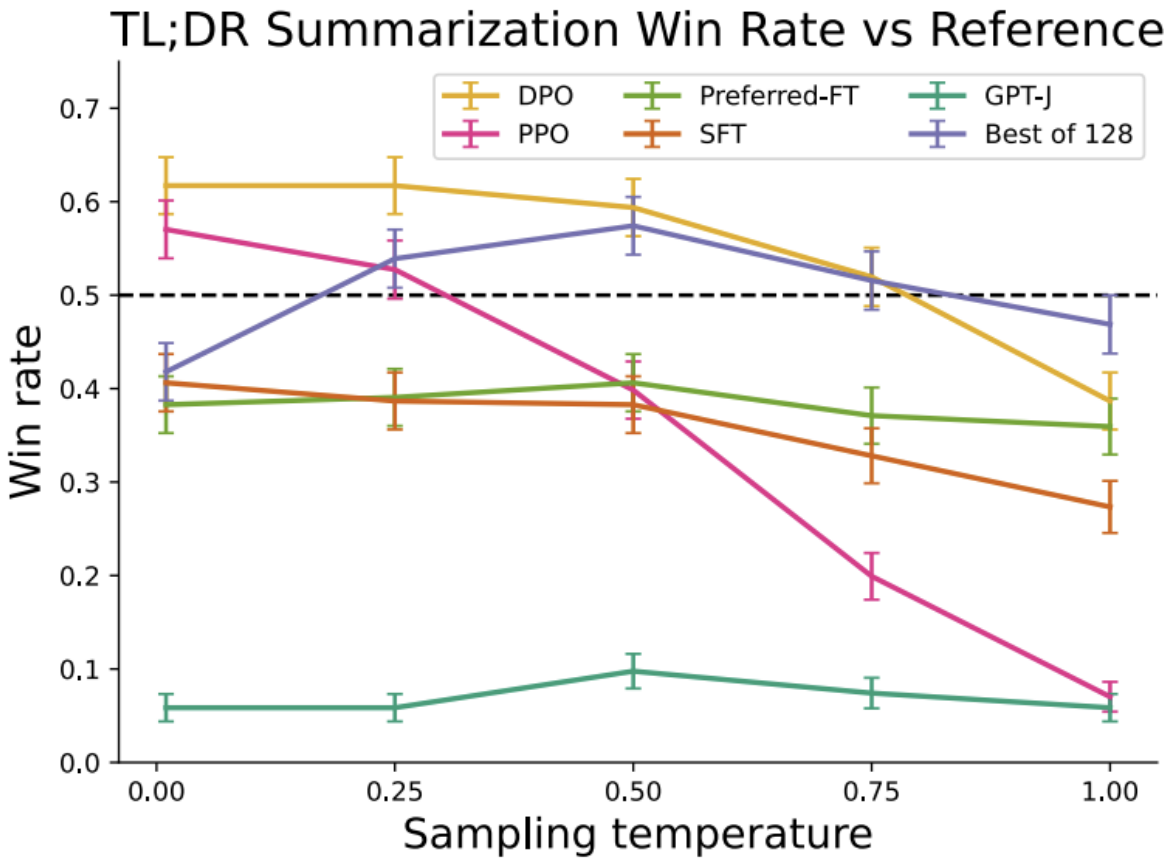
$$L_{DPO}(\pi_\theta, \pi_{ref}) = -\mathbb{E}_{(q,o_w,o_l)\sim D}\left[\log \sigma(\beta \log \frac{\pi_\theta(o_w|q)}{\pi_{ref}(o_w|q)} - \beta \log \frac{\pi_\theta(o_l|q)}{\pi_{ref}(o_l|q)})\right]$$

Reward for winning sample

Reward for losing sample

DPO Outperforms and Works Well at Scale

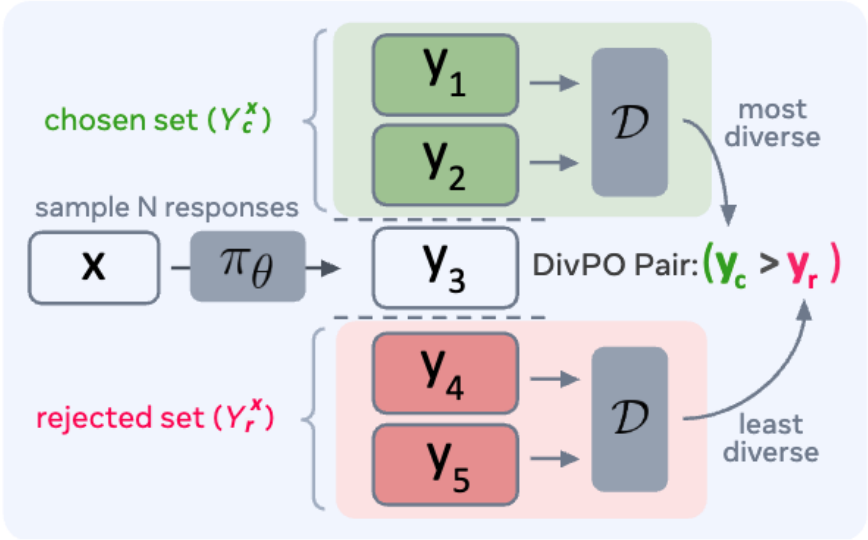




Diverse Preference Optimization

Define diversity criterion:

- Model probability
- Word frequency
- LLM-as-a-diversity judge



Most diverse responses $y_c = \arg\max_{y_i \in Y_c^x} D(y_i, Y_c^x)$

Least diverse responses $y_r = \arg max_{y_i \in Y_r^x} D(y_i, Y_r^x)$

$$L_{DivPO}(\pi_\theta, \pi_{ref}) = -\mathbb{E}_{(x,y_c,y_r)\sim D}\left[\log \sigma(\beta \log \frac{\pi_\theta(y_c|x)}{\pi_{ref}(y_c|x)} - \beta \log \frac{\pi_\theta(y_r|x)}{\pi_{ref}(y_r|x)})\right]$$

Lanchantin, Jack, Angelica Chen, Shehzaad Dhuliawala, Ping Yu, Jason Weston, Sainbayar Sukhbaatar, and Ilia Kulikov. "Diverse preference optimization." arXiv preprint arXiv:2501.18101 (2025).

Persona Generation Task Results

	Diversity ↑				Quality ↑	
Method	**First name**	**City**	**Occupation**	**Avg**	**ArmoRM**	**Valid JSON %**
Llama 3.1-8B-Instruct	30.45%	13.82%	27.93%	24.07%	0.141	45.51%
GPT-4o	2.55%	0.74%	0.38%	1.22%	0.140	100.00%
SFT	22.18%	9.14%	20.98%	17.43%	**0.142**	**99.58%**
DPO	22.95%	10.44%	27.92%	20.44%	**0.142**	99.25%
DivPO, $\mathcal{D}$=Freq	**49.68%**	27.87%	57.47%	45.01%	0.139	98.73%
DivPO, $\mathcal{D}$=Prob	48.89%	**29.86%**	**58.44%**	**45.73%**	0.139	97.32%
Online SFT	24.92%	6.92%	19.46%	17.10%	0.139	99.54%
Online DPO	11.61%	3.19%	10.82%	8.54%	0.139	**99.99%**
Online DivPO, $\mathcal{D}$=Freq	52.04%	**45.35%**	**65.03%**	**54.14%**	**0.141**	99.80%
Online DivPO, $\mathcal{D}$=Prob	**53.85%**	29.21%	55.77%	46.28%	0.134	98.26%

DivPO increases diversity in generated personas while maintaining quality.

Outline

- ✓ **Different type of human feedback** (5 mins)
- ✓ **RLHF** (15 mins)
- ✓ **DPO + many others** (15 mins)
- ❑ **Limitations of human feedback** (10 mins)

Let’s talk about project ☺

Pick A Question That You're Excited About

- Broadly relevant to HCI + NLP
  - Why is your project a good fit to "human-centered LLM"
  - Could you formulate a research question to deeply explore it?
  - What type of data might be available for you to use?
  - Which software or tools could you use to work on it?
  - How do you evaluate the outcome of your project?

What Could Be A Final Project?

- ⭐ Develop new methodologies to leverage human feedback/preferences
- ⭐ Fairness, bias, or ethical issues around existing LLMs/VLMs
- ⭐ Improve existing LLM pipelines
- ⭐ Building interactive systems to allow humans to interact with LLMs
- ⭐ Simulating personas via LLMs
- ⭐ Understanding culture, values, belief in/of LLMs
- ⭐ Modeling sycophancy, overreliance, companionship, etc
- ⭐ LLMs for social good (e.g., accessibility, misinformation, persuasion, etc)
- ⭐ Position papers or a critic (talk to us first)

Resources to Check Out

- Top course projects sometimes end up into actual paper submissions to either full conferences or workshop venues.

- Checking out workshop papers published in:
  - HCI+NLP @ NAACL 2022
  - HCI+NLP @ EACL 2021
  - Human Evaluation of Generative Models @ NeurIPS 2022
  - In2Writing @ CHI 2023
  - InterNLP @ NeurIPS 2022

Project Team

- Recommend 2~3 people
- Divide the work between team members clearly
- Reach out to us if anything is needed

Key Considerations

- Availability of data
  - Be careful in deciding whether to collect and annotate your own data
  - Huggingface datasets
- ML framework
  - Huggingface, sklearn, keras, pytorch, Tensorflow
- Availability of computation
  - GCP, Google Colab
- Availability of evaluation
  - Evaluation metrics, auto vs. human eval

Recommendations for Successful Projects

- Start early and work on it every week rather than rushing at the end
- Get your data first!
- Have a clear, well-defined research question (novel/creative ones ++)
- **Results should teach us something**
- Visualize results well
- Come to office hours and talk to us!

Common Issues

- Data not available or hard to get access to
- No code written for model/data processing
- Team starts late
- Results/Conclusion don’t say much besides that it didn’t work
- Even if results are negative or unexpected, analyze them

Project Ideas (Optional)