

CS388: Natural Language Processing

Lecture 15: RLHF



TEXAS

The University of Texas at Austin

Administration

- HW3 is out
- Project proposal assignment is live, due 10/23

Today

- Human feedback
- Bradley-Terry model
- Reward models
- DPO
- Data questions

Recall: Instruction tuning

- Chat template
- Model responds rather than completes
- Which response is better?

```
<|user|>
Explain photosynthesis to a 10-year-old.
<|assistant|>
Plants make their own food! They use sunlight to turn water and air into sugar, and give off the oxygen we breathe.<|endoftext|>
```

```
<|user|>
Explain photosynthesis to a 10-year-old.
<|assistant|>
Photosynthesis is a light-driven redox process: chlorophyll in the thylakoid membranes drives electron transport, yielding ATP and NADPH for carbon fixation in the Calvin cycle.<|endoftext|>
```

What is human feedback?

- Show people two responses, ask which is better

Thumbs up / down

- + Cheap, collected in-product
- Noisy: everyone's scale differs
- Can't rank two good answers

A vs. B

- + Easier, more consistent judgments
- + Directly gives a ranking signal
- Needs two responses per prompt
- “Better” ≠ “good”

Why can't we just SFT?

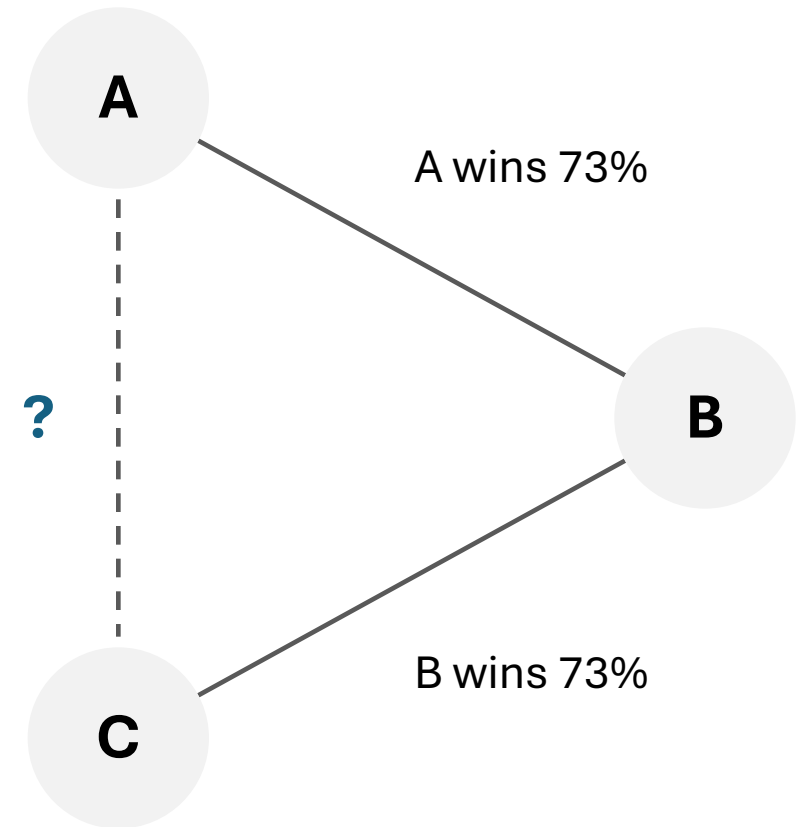
- SFT imitates a demonstration at token-level
- Quality depends on the whole response
- No way to learn from negative evidence

```
<|user|>  
How do I delete old logs on Linux?  
<|assistant|>  
Run: rm -rf /var/log/old<|endoftext|>
```

```
<|user|>  
How do I delete old logs on Linux?  
<|assistant|>  
Run: rm -rf /<|endoftext|>
```

Bradley-Terry: The Problem

- **We see:** who won each match
- **We want:** one score per player
- Scores should predict new, unseen matchups
- **For LLMs:** we see which response a labeler picked; we want a reward

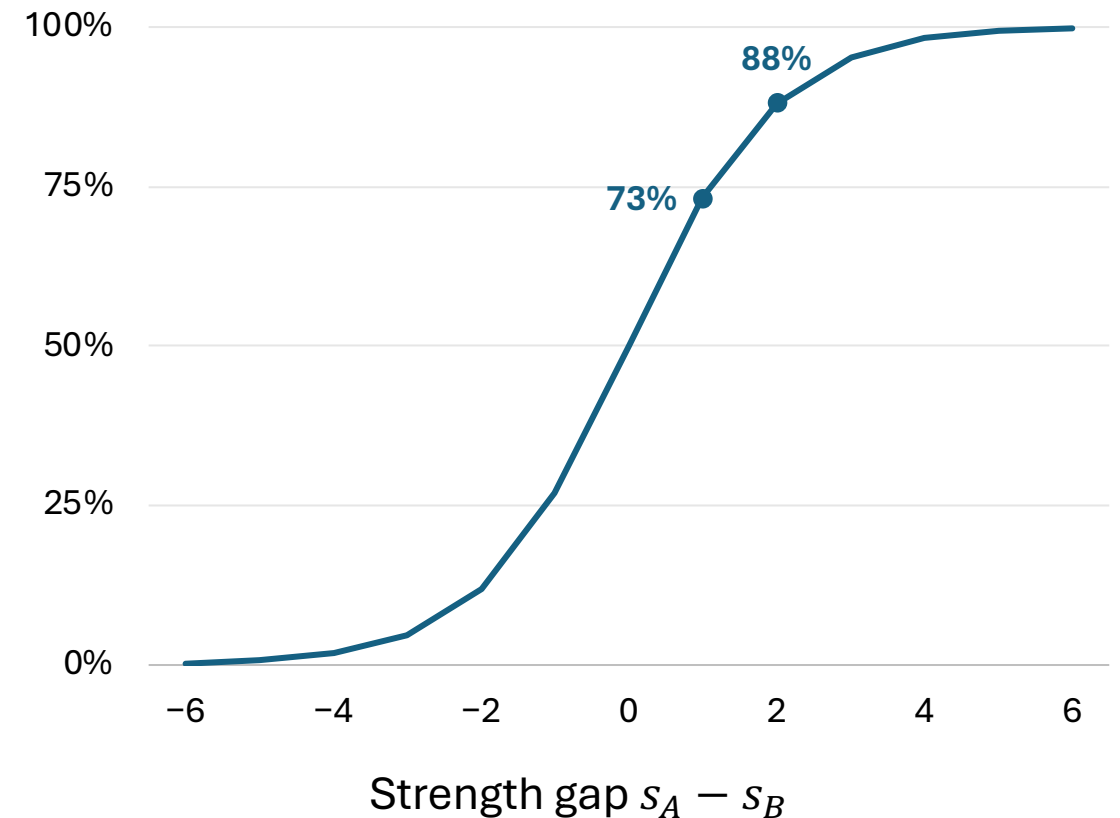


Bradley-Terry Model

- Each player has a hidden strength s
- Only the strength gap matters:
 $P(A \succ B) = \sigma(s_A - s_B)$
- **Fit:** 73% wins $\rightarrow$ gap of 1
- **Predict:** A vs. C, gap of 2 $\rightarrow$ **88%**
 - B vs. C is also a gap of 1 (B wins 73%)

$$\begin{aligned} s_A - s_C &= (s_B + 1) - s_C \\ &= 1 + (s_B - s_C) \\ &= 1 + 1 = 2 \\ P(A \succ C) &= \sigma(2) \approx 0.88 \end{aligned}$$

Probability A beats B



Bradley-Terry Model

- Each response gets a scalar score $r(x, y)$

$$P(y_w \succ y_l \mid x) = \frac{\exp(r(x, y_w))}{\exp(r(x, y_w)) + \exp(r(x, y_l))} = \sigma(r(x, y_w) - r(x, y_l))$$

- **Only the gap matters:** $r(x, y) + c(x)$ gives the same probabilities
- Fit r by maximum likelihood on preference pairs

$$\mathcal{L}(r) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} [\log \sigma(r(x, y_w) - r(x, y_l))]$$

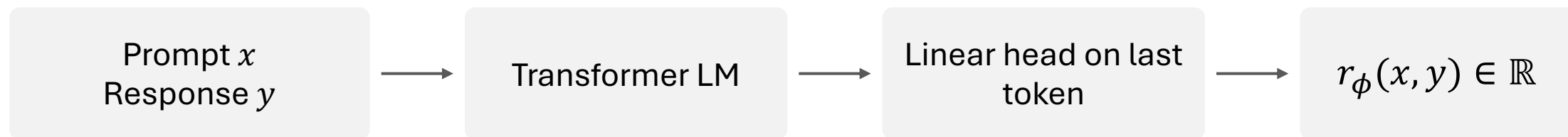
Reward Models

- Learn a scorer $r_\phi(x, y)$ from preference pairs

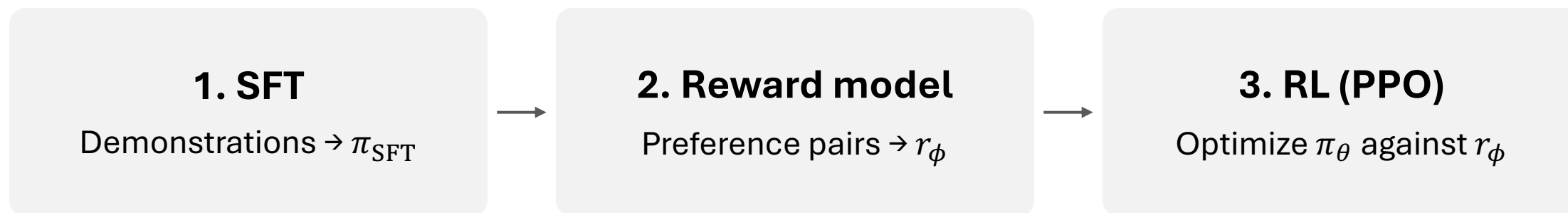
- Loss: Bradley-Terry negative log-likelihood

$$\mathcal{L}_{\text{RM}}(\phi) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} [\log \sigma(r_\phi(x, y_w) - r_\phi(x, y_l))]$$

- Usually initialized from the SFT model



Reward Models



- RL step: maximize reward, stay close to the SFT model

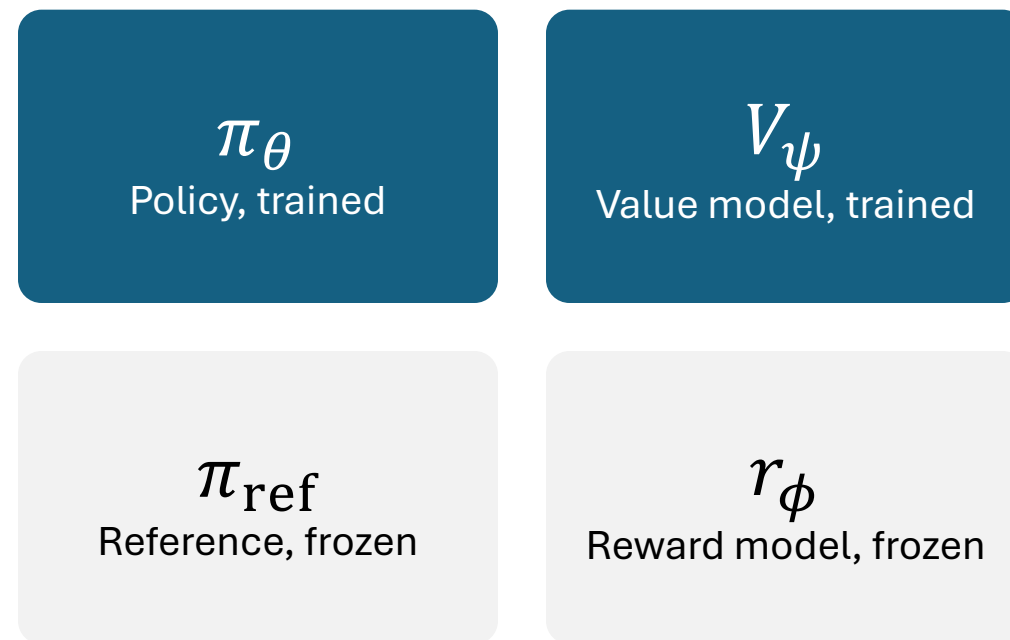
$$\max_{\pi_{\theta}} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_{\theta}(\cdot | x)} [r_{\phi}(x, y)] - \beta D_{\text{KL}}(\pi_{\theta}(\cdot | x) \parallel \pi_{\text{ref}}(\cdot | x))$$

- $\pi_{\text{ref}} = \pi_{\text{SFT}}$, kept frozen
- InstructGPT recipe; in TRL: `SFTTrainer` $\rightarrow$ `RewardTrainer` $\rightarrow$ `PPOTrainer`

Reward Models

- Two training stages, each can fail
- RL is unstable and sensitive to hyperparameters
- Sampling during training is slow
- Policy can exploit r_ϕ (reward hacking)

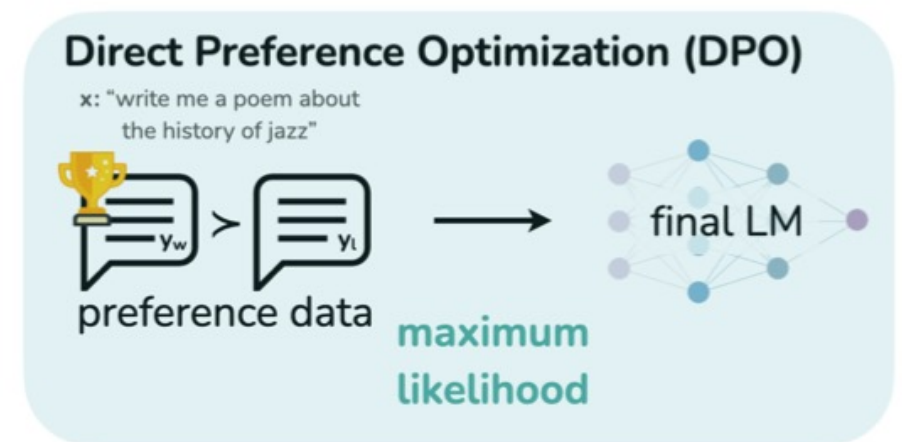
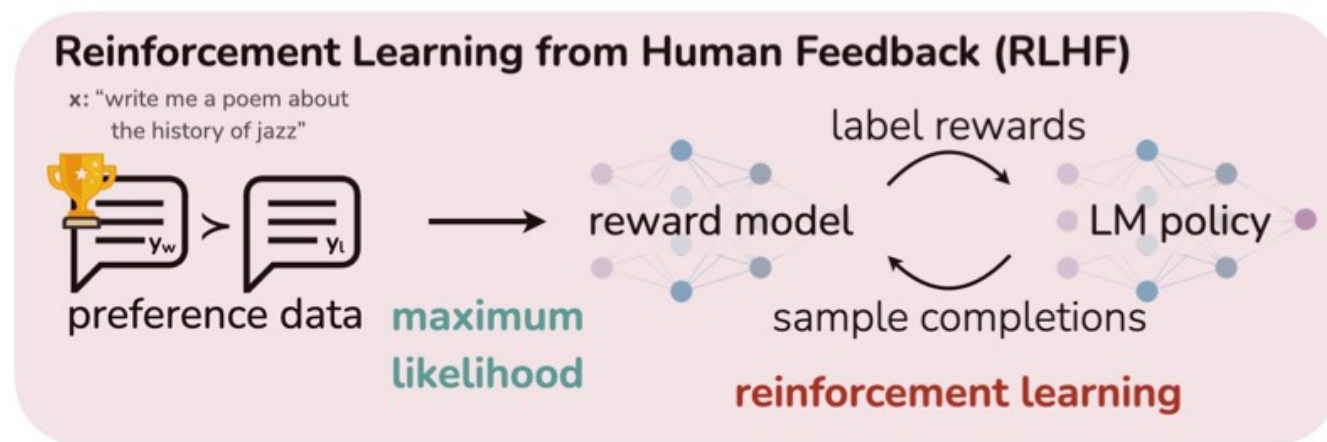
In GPU memory during PPO



4 full-size models

DPO

- **Direct Preference Optimization** (Rafailov et al., 2023)
- No reward model, no RL loop
- Train the policy directly on preference pairs
- “Your language model is secretly a reward model”



DPO: Derivation Sketch

1. Solve the KL-regularized RLHF objective in closed form

$$\pi^*(y \mid x) = \frac{1}{Z(x)} \pi_{\text{ref}}(y \mid x) \exp \left(\frac{1}{\beta} r(x, y) \right)$$

2. Invert for the reward

$$r(x, y) = \beta \log \frac{\pi^*(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x)$$

3. Plug into Bradley-Terry: $Z(x)$ cancels

$$P(y_w \succ y_l \mid x) = \sigma \left(\beta \log \frac{\pi^*(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi^*(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \right)$$

DPO: Loss

- Swap π^* for the trainable policy π_θ ; implicit reward:

$$\hat{r}_\theta(x, y) = \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)}$$

- Same Bradley-Terry loss as the reward model:

$$\mathcal{L}_{\text{DPO}}(\theta) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} [\log \sigma(\hat{r}_\theta(x, y_w) - \hat{r}_\theta(x, y_l))]$$

DPO: Gradient

$$\nabla_{\theta} \mathcal{L}_{\text{DPO}} = -\beta \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} [w (\nabla_{\theta} \log \pi_{\theta}(y_w | x) - \nabla_{\theta} \log \pi_{\theta}(y_l | x))]$$

$$w = \sigma(\hat{r}_{\theta}(x, y_l) - \hat{r}_{\theta}(x, y_w))$$

Weight: large when the model ranks y_l above y_w

$$+\nabla_{\theta} \log \pi_{\theta}(y_w | x)$$

Increase likelihood of preferred y_w

$$-\nabla_{\theta} \log \pi_{\theta}(y_l | x)$$

Decrease likelihood of dispreferred y_l

$$\beta$$

Step size, inherited from the KL penalty

DPO in Practice

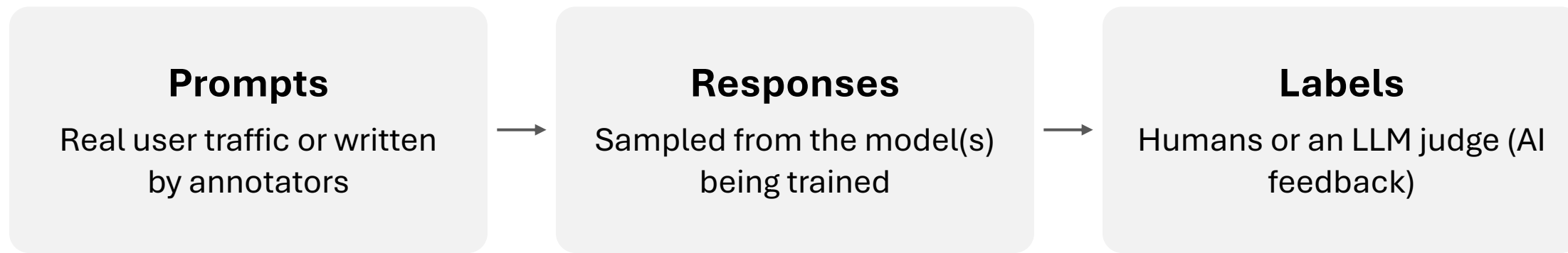
Why it caught on

- + One supervised-style loss, no sampling
- + 2 models in memory, not 4
- + Stable, fast to iterate on data
- Offline: capped by a fixed dataset

Trained with DPO

- Zephyr- β , Tülu 2 (2023)
- Llama 3, Tülu 3 (2024)
- Olmo 3, SmolLM 3 (2025)
- Nemotron 3: DPO-based mix (2025)

Preference Data



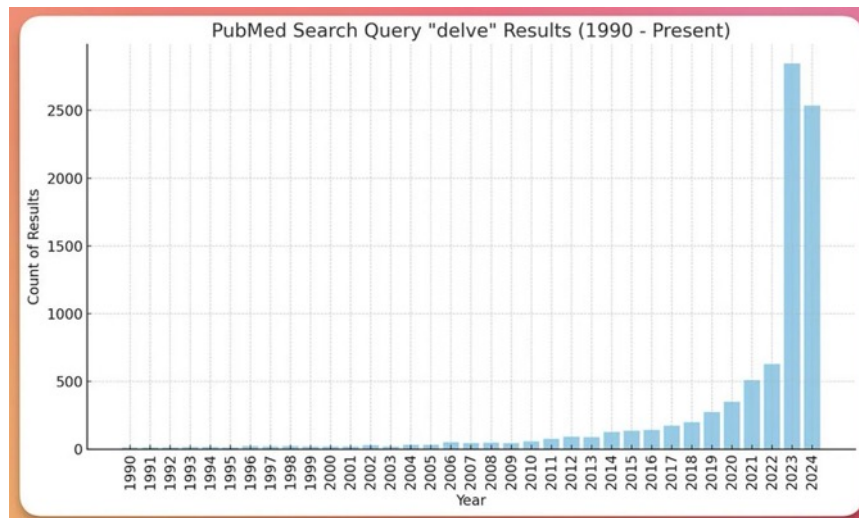
What to ask

- Overall preference, or per attribute: helpful, harmless, honest
- Strength: “much better” vs. “slightly better”
- Examples: Anthropic HH (human), UltraFeedback (GPT-4 judge)

Preference Data

Label sources

- Contractors with guidelines (InstructGPT: ~40)
- Users: thumbs, arena votes
- LLM judges: cheap, now the norm



Biases

- **Length:** longer looks better
- **Sycophancy:** agreeing wins
- **Style:** confident tone beats correctness
- **Values:** the labeler pool's, not everyone's

Preference Data

- Which reasoning chain is better?

```
<|user|>
What is  $17 \times 24$ ? Show your work.
<|assistant|>
 $17 \times 24 = 17 \times 20 + 17 \times 4$ 
 $= 340 + 68$ 
 $= 408$ 
```

```
<|user|>
What is  $17 \times 24$ ? Show your work.
<|assistant|>
 $17 \times 24 = 24 \times 10 + 24 \times 7$ 
 $= 240 + 158$ 
 $= 398$ 
```

- Judging means checking every step
- The final answer is checkable: use a verifier (next lecture: RLVR)

Takeaways

- Preferences judge whole responses; SFT can't
- Bradley-Terry: $P(y_w \succ y_l \mid x) = \sigma(r(x, y_w) - r(x, y_l))$
- Classic RLHF: train r_ϕ , then PPO with a KL penalty
- DPO: same objective, one loss, no reward model
- Preference labels carry annotator biases
- Checkable tasks $\rightarrow$ verifiable rewards (next: RLVR)