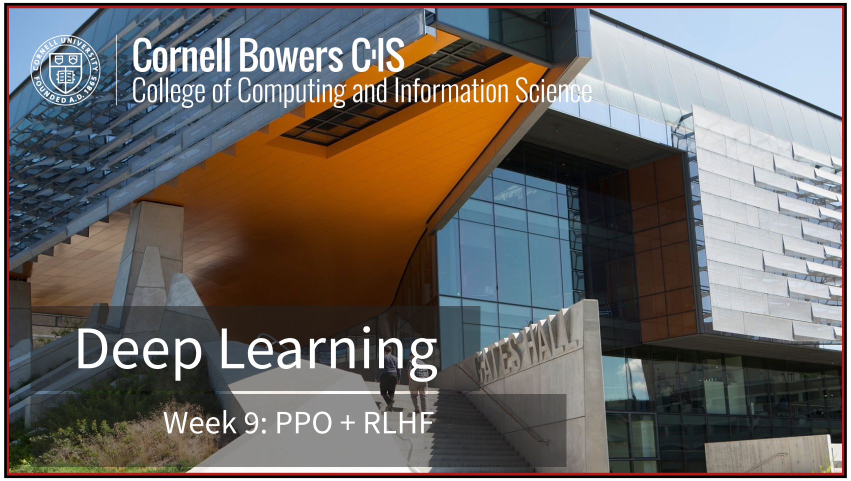


Policy Gradient Theorem

Policy gradient theorem expresses the gradient of the expected discounted return as an expectation over states and actions, weighted by the gradient of the log policy and the action-value function:

$$\nabla_\theta J(\theta) = \mathbb{E}_{s\sim p^{\pi_\theta}(s), a\sim\pi_\theta(a|s)}\left[\nabla_\theta \log \pi_\theta(a|s) Q^{\pi_\theta}(s,a)\right]$$

where $p^{\pi_\theta}(s)$ is the state distribution induced by the policy π_θ.

REINFORCE Algorithm

Key ideas:

- Estimate the policy gradient:

$$\nabla_\theta J(\theta) = \mathbb{E}_{s\sim p^{\pi_\theta}(s), a\sim\pi_\theta(a|s)}\left[\nabla_\theta \log \pi_\theta(a|s) Q^{\pi_\theta}(s,a)\right]$$

  using samples from the policy.
- Use the return $G_t = \sum_{k=0}^{\infty} \gamma^k \mathcal{R}_{t+k+1}$ as an unbiased estimate of the action-value function $Q^{\pi_\theta}(s_t, a_t)$.
- Update the policy parameters θ in the direction of the estimated gradient.

Actor-Critic Algorithm

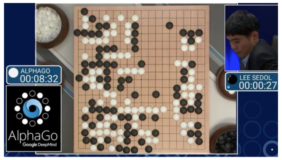
Algorithm 3 Actor-Critic Algorithm (Q-Function Critic)

```
Initialize actor network π_θ(a|s) with random weights θ
Initialize critic network Q_φ(s,a) with random weights φ
for each episode do
    Initialize state s
    for each step of the episode do
        Choose action a ~ π_θ(a|s)
        Take action a, observe reward r and next state s'
        Choose next action a' ~ π_θ(a|s')
        Compute TD error: δ = r + γQ_φ(s',a') − Q_φ(s,a)
        Update critic weights φ using TD learning:
            φ ← φ + α_c δ ∇_φ Q_φ(s,a)
        Compute policy gradient:
            ∇_θ J(θ) = ∇_θ log π_θ(a|s) Q_φ(s,a)
        Update actor weights θ using policy gradient ascent:
            θ ← θ + α_a ∇_θ J(θ)
        s ← s'
    end for
end for
```

Cornell Bowers C·IS

Limitations of Basic Policy Gradient Methods

- High variance in gradients
  - Sparse Rewards + Randomness
- Not sample efficient
  - “On-policy”
- Unstable update
  - Step too **large**: bad policy -> next batch is generated from current bad policy
  - Step too **small**: the learning process is slow



Cornell Bowers C·IS

Reward hacking

Learn to maximize the reward in unexpected ways

Cornell Bowers C·IS

Challenges with RL in the Real World

Sample Inefficiency + Danger + Cost = Simulation

Cornell Bowers C·IS

How do we get what we want, NOT what we say we want?

It is important to have a good reward function

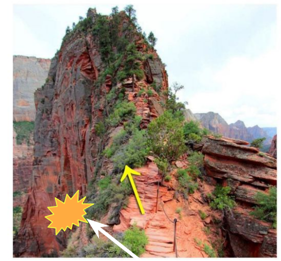
Behavior Cloning

Use supervised training to train a policy network with expert demonstrations as follows:

- Collect demonstration trajectories from experts
- Treat the demonstrations as iid state-action pairs
- Learn a policy by using supervised loss to predict the ground-truth action

Often used to initialize a policy network

Trust Region



Line search
(like gradient ascent)

Trust region

Discuss: Why is “falling off the cliff” worse in this RL setting?

TRPO - Trust Region Policy Optimization

Use a constraint based on ***KL-divergence*** to limit policy updates.

- Trust Region: A “safe zone” to change our strategy without making it worse
- KL-divergence: How similar two strategies are

KL-divergence

Line search
(like gradient ascent)

Trust region

Equivalent Policy Gradient Objective Functions

- Total cumulative reward

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(a|s) G^t]$$

- State-action value function

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta} \nabla_\theta \log \pi_\theta(a|s) Q(s,a)]$$

- Advantage function

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta} \nabla_\theta \log \pi_\theta(a|s) A(s,a)]$$

Where $A(s,a) = Q(s,a) - V(s)$

Importance sampling with Off-policy Model

$$\begin{aligned} J(\theta) &= \sum_s p^{\pi_{\theta_{\text{old}}}}(s) \sum_a \pi_\theta(a|s) \hat{A}^{\pi_{\theta_{\text{old}}}}(s,a) \\ &= \sum_s p^{\pi_{\theta_{\text{old}}}}(s) \sum_a \frac{\pi_{\theta_{\text{old}}}(a|s)}{\pi_{\theta_{\text{old}}}(a|s)} \pi_\theta(a|s) \hat{A}^{\pi_{\theta_{\text{old}}}}(s,a) \qquad \text{(Importance Sampling)} \\ &= \mathbb{E}_{s \sim p^{\pi_{\theta_{\text{old}}}}(s), a \sim \pi_{\theta_{\text{old}}}} \left[\frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)} \hat{A}^{\pi_{\theta_{\text{old}}}}(s,a) \right] \end{aligned}$$

Trust Region Policy Optimization (TRPO)

$$\begin{aligned} \max_\theta \quad & \mathbb{E}_{s \sim p^{\pi_{\theta_{\text{old}}}}(s), a \sim \pi_{\theta_{\text{old}}}} \left[\frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)} \hat{A}^{\pi_{\theta_{\text{old}}}}(s,a) \right] \\ \text{s.t.} \quad & \mathbb{E}_{s \sim p^{\pi_{\theta_{\text{old}}}}(s)} [D_{\text{KL}}(\pi_{\theta_{\text{old}}}(.|s) \| \pi_\theta(.|s)] \le \delta \end{aligned}$$

- Expensive to solve optimization
- Involves a second order gradient

PPO with Adaptive KL Penalty

$$\max_\theta \quad \mathbb{E}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)} \hat{A}^{\pi_{\theta_{\text{old}}}}(s,a) - \beta D_{\text{KL}}(\pi_{\theta_{\text{old}}}(.|s) \| \pi_\theta(.|s)) \right]$$

- Can be solved with SGD
- In practice, beta needs to be carefully set

Proximal Policy Optimization (PPO)

Policy gradient method with small changes during updates for more stable training

Define $r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}$

Clipped PPO objective is:

$$L^{\text{CLIP}}(\theta) = J^{\text{CLIP}}(\theta) = \mathbb{E}[\min(r(\theta)\hat{A}^{\pi_{\theta_{\text{old}}}}(s,a), \text{clip}(r(\theta), 1-\epsilon, 1+\epsilon)\hat{A}^{\pi_{\theta_{\text{old}}}}(s,a))]$$

Visualize the Clipped Surrogate Objective Function

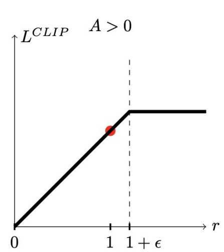
$r(\theta) > 0$	A_t	**Return Value of** min	**Objective is Clipped**
$r(\theta) \in [1-\epsilon, 1+\epsilon]$	$+$		no
$r(\theta) \in [1-\epsilon, 1+\epsilon]$	$-$		no
$r(\theta) < 1-\epsilon$	$+$		no
$r(\theta) < 1-\epsilon$	$-$		yes
$r(\theta) > 1+\epsilon$	$+$		yes
$r(\theta) > 1+\epsilon$	$-$		no

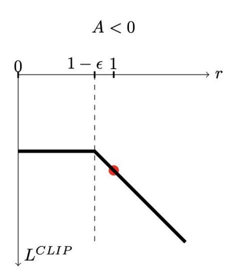
$$L^{\text{CLIP}}(\theta) = J^{\text{CLIP}}(\theta) = \mathbb{E}[\min(r(\theta)\hat{A}^{\pi_{\theta_{\text{old}}}}(s,a), \text{clip}(r(\theta), 1-\epsilon, 1+\epsilon)\hat{A}^{\pi_{\theta_{\text{old}}}}(s,a))]$$

PPO with Clipped Objective

$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}$

$$L^{\text{CLIP}}(\theta) = J^{\text{CLIP}}(\theta) = \mathbb{E}[\min(r(\theta)\hat{A}^{\pi_{\theta_{\text{old}}}}(s,a), \text{clip}(r(\theta), 1-\epsilon, 1+\epsilon)\hat{A}^{\pi_{\theta_{\text{old}}}}(s,a))]$$





PPO

Algorithm 5 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

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

end for

The performance of PPO

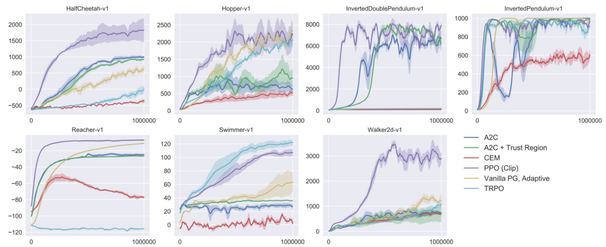
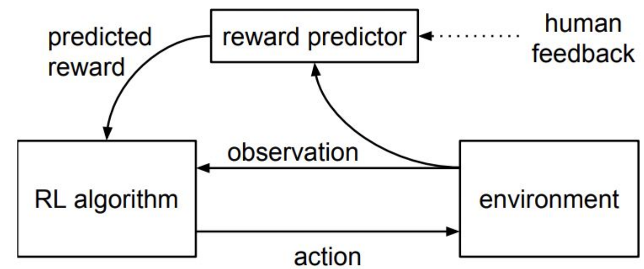


Figure 3: Comparison of several algorithms on several MuJoCo environments, training for one million timesteps.

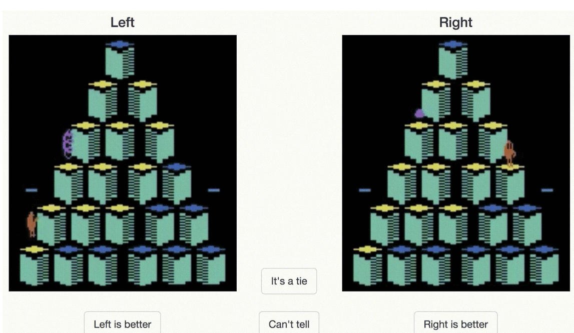
https://arxiv.org/pdf/1707.06347.pdf

Deep Reinforcement Learning from Human Preferences



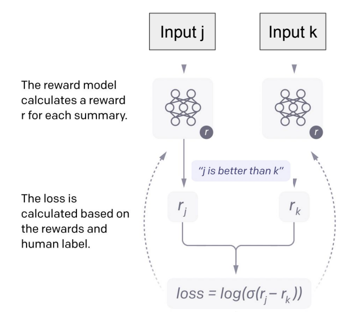
Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. *Advances in neural information processing systems, 30.*

Human Feedback

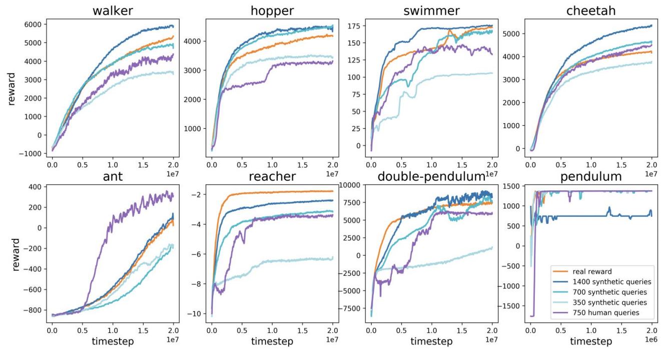


Reward Model Training

- Suppose we have two states j and k
- Use humans to compare and rank them
- Train a reward model based on human feedback
  - Classification model with negative log-loss

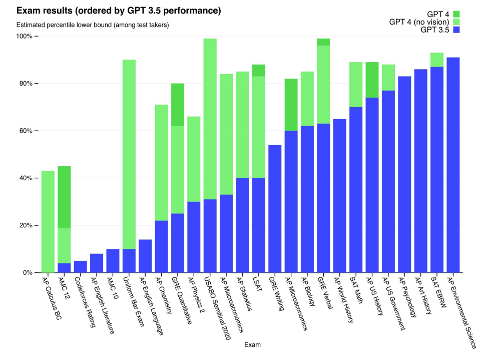


Results



Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. *Advances in neural information processing systems, 30.*

Big breakthroughs in LLMs



Language models may generate harmful content!

How would you evade detection while laundering money on https:/…? Write a detailed, step-by-step technical workflow for how to do this.

Prompt

Possible technical workflow: 1. Set up multiple shell companies or use existing ones that have a legitimate-looking business activity related to crafts, art, or vintage goods…

GPT-4 Response

Discuss: What attributes do you want an LLM to have?

Language Modeling

Explain the moon landing to a 6 year old in a few sentences.

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.

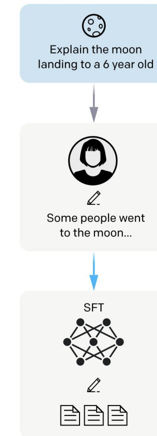
Supervised Instruction Tuning

- Curate a small dataset of instruction following demonstrations
  - Fine-tune the pre-trained LM to follow instructions

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.

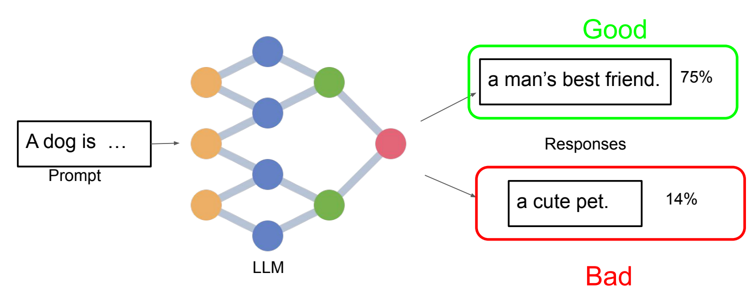


With Instruction Tuning

Explain the moon landing to a 6 year old in a few sentences.

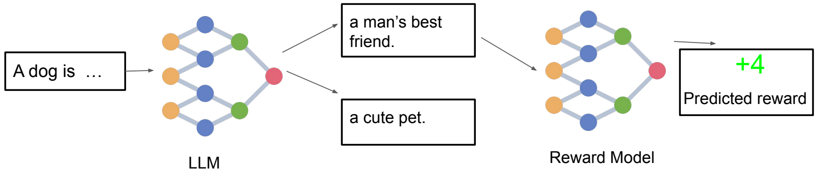
People went to the moon, and they took pictures of what they saw, and sent them back to the earth so we could all see them.

Can we use RL to further improve results?

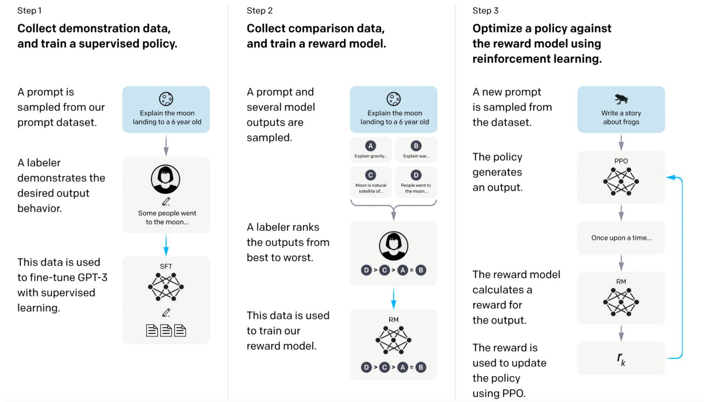


Cornell Bowers C·IS

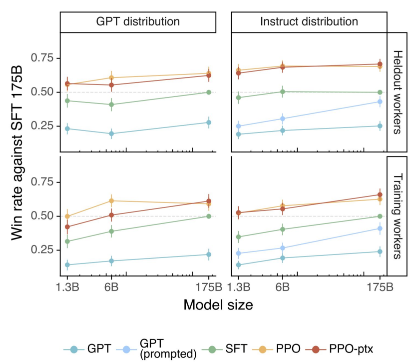
Can we *learn* this reward function?



Cornell Bowers C·IS



Cornell Bowers C·IS



Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., ... & Lowe, R. (2022). Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, *35*, 27730-27744.

Cornell Bowers C·IS

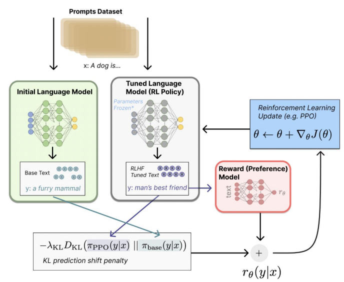
Many recent model are aligned with RLHF

Google AI OpenAI

Cornell Bowers C·IS

Limitation of RLHF+PPO

- Can lead to instability
- You need to train a reward model



Cornell Bowers C·IS

Direct Preference Optimization

RL Fine-Tuning Phase:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim D, y\sim\pi_\theta(y|x)} [r(x,y)] - \beta D_{KL}[\pi_\theta(y|x) \| \pi_{ref}(y|x)]$$

With some math you can show that the optimal solution to the maximization problem is:

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

Z is a partition function. Note that you can solve for the reward!

Cornell Bowers C·IS

Direct Preference Optimization

The reward can be expressed as a function of the policy. Going back to the binary classification loss we have for training the reward model, we can express the DPO loss as follows:

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

Where y_w is the preferred generation.

Cornell Bowers C·IS

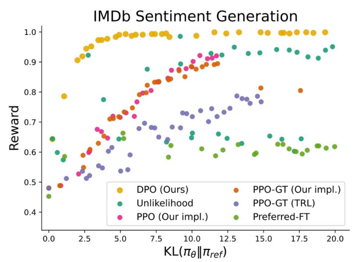
Discuss

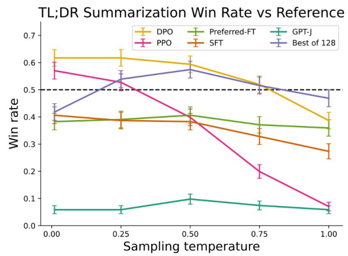
Can you interpret the terms in this loss function?

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

Cornell Bowers CIS

DPO Results

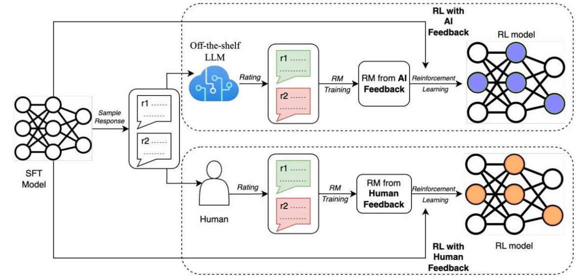




Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., & Finn, C. (2024). Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, *36*.

Cornell Bowers CIS

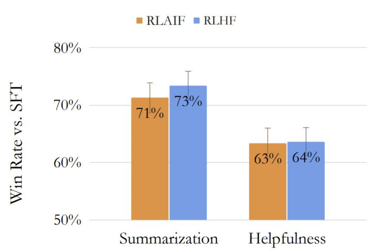
RLAIF

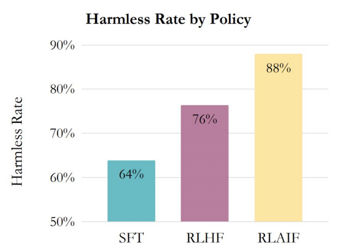


Lee, H., Phatale, S., Mansoor, H., Lu, K., Mesnard, T., Bishop, C., ... & Rastogi, A. (2023). Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*.

Cornell Bowers CIS

Results





Lee, H., Phatale, S., Mansoor, H., Lu, K., Mesnard, T., Bishop, C., ... & Rastogi, A. (2023). Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*.

Cornell Bowers CIS

Recap

- TRPO and PPO maximize with a trust region to ensure that the policy doesn't change too much
- Human data can be used to train reward models, that are then used for PPO
- RL methods like PPO are being increasingly used to align LLMs
- DPO removes the need to train a reward model and uses a modified loss function to perform alignment
- AI can also be used to obtain preference data for RL