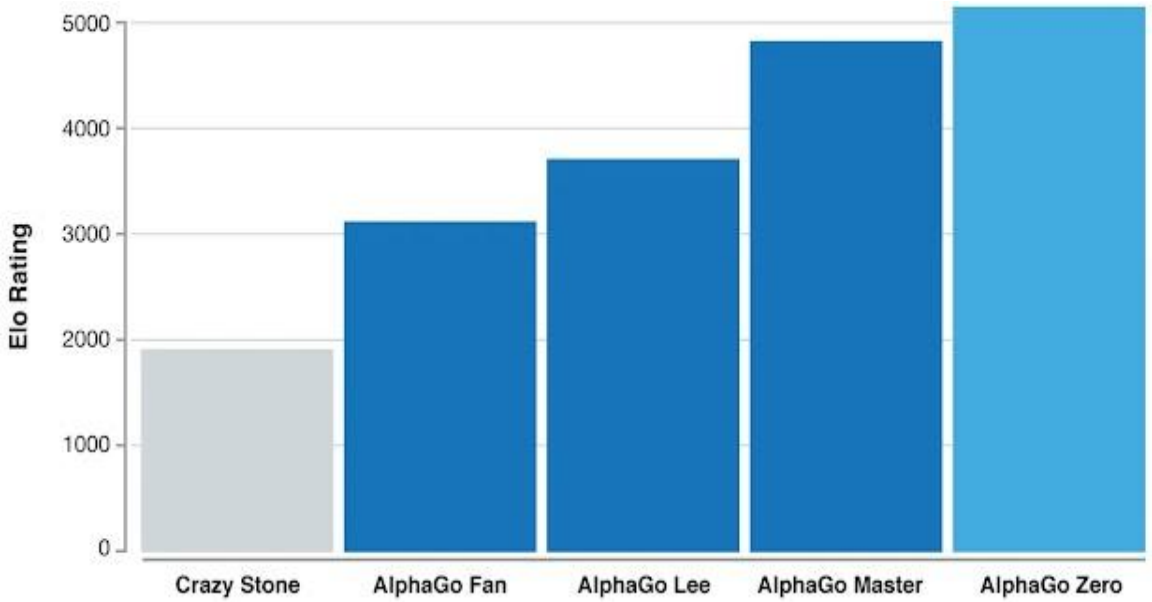
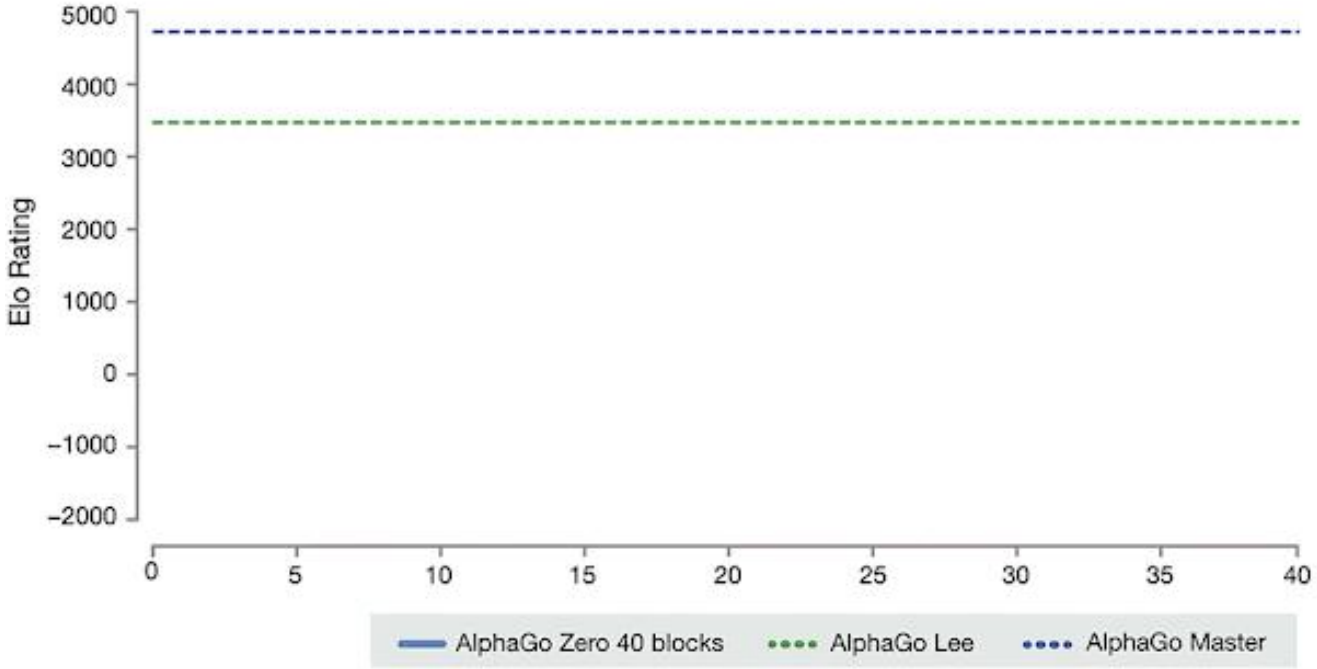
Reinforcement Learning from Human Feedback

Zhenyu Hou
Department of Computer Science & Technology
Tsinghua University

Why Reinforcement Learning

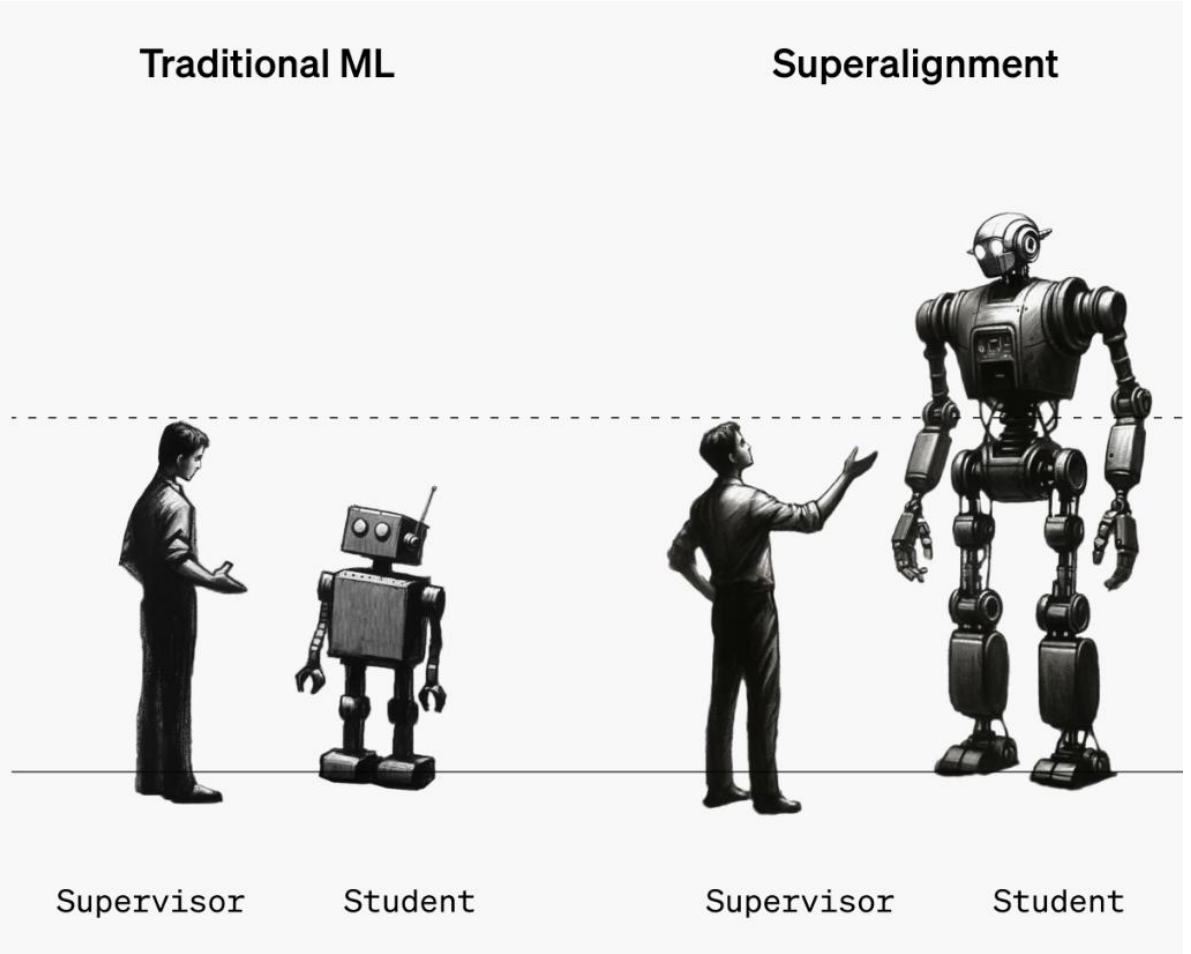




AlphaGo Fan to AlphaGo Master to AlphaGo Zero

Why LLM needs Reinforcement Learning

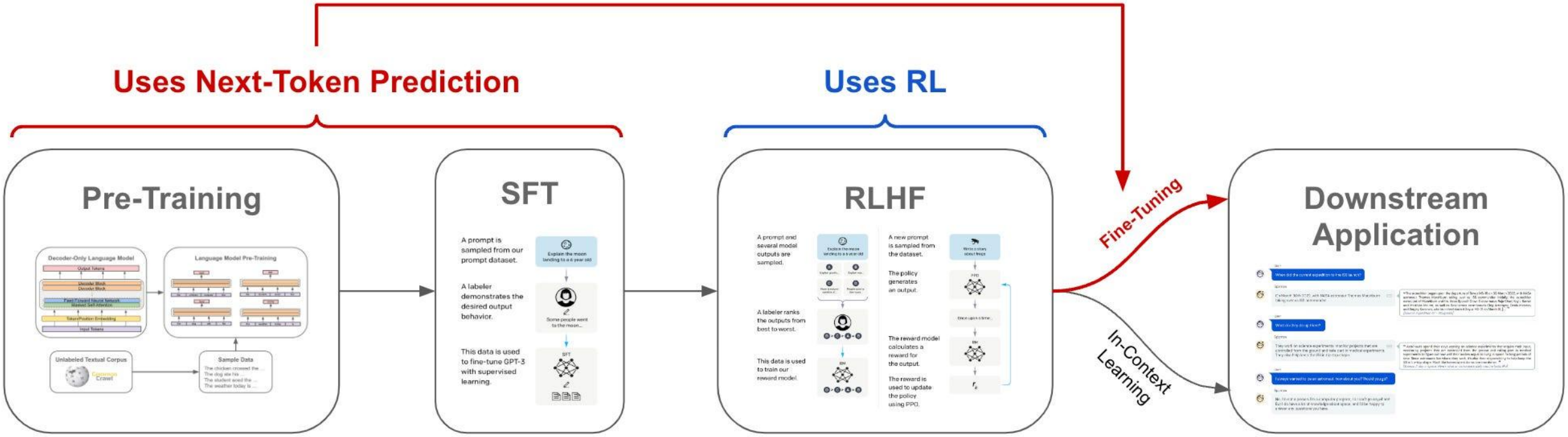
- Supervision signal
- Self-improvement



Why reinforcement learning

- Sequential decision making is everywhere

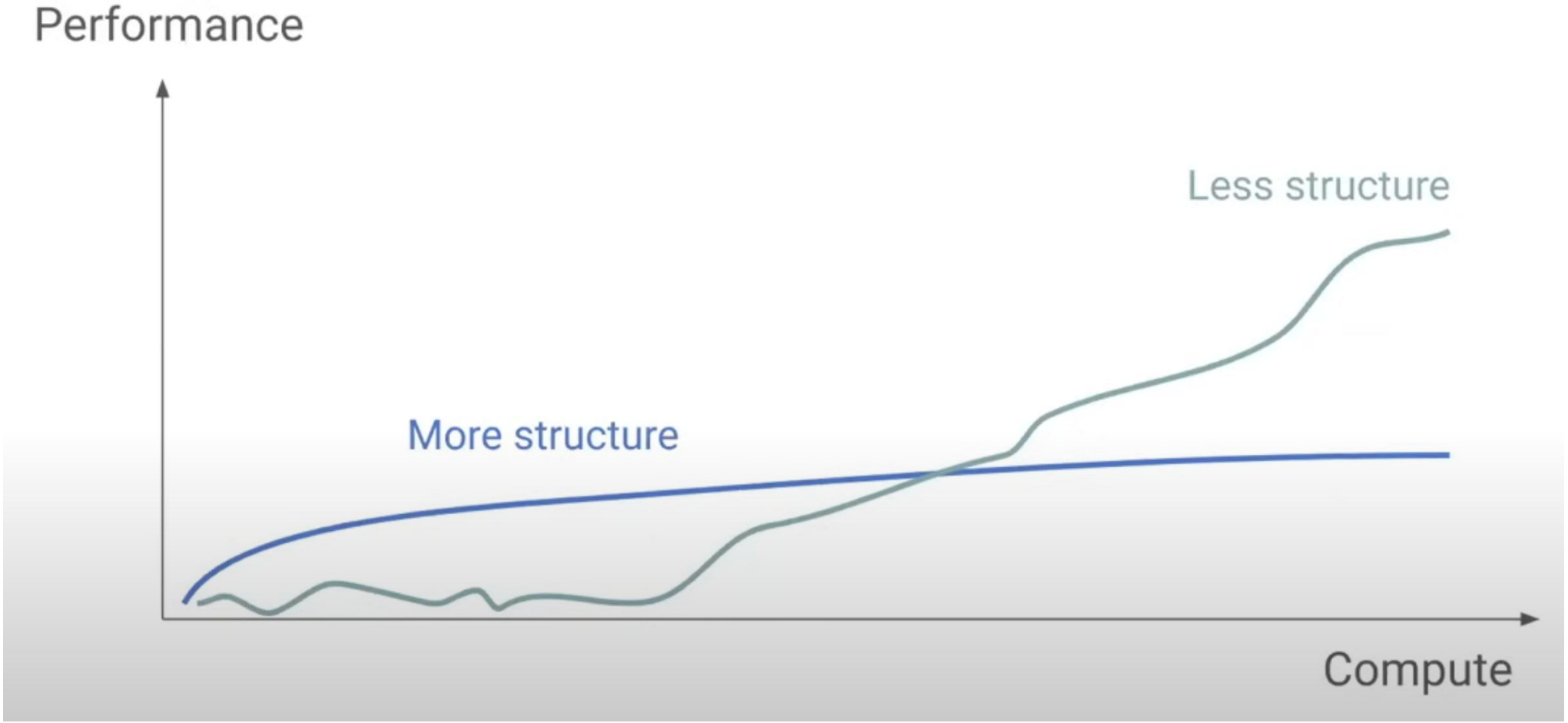
Difference between SFT and RLHF



- Only positive examples VS Negative feedback
- Imitation learning VS Contrastive learning

Ouyang L, Wu J, Jiang X, et al. Training language models to follow instructions with human feedback. 2022.

Difference between SFT and RLHF



From https://www.youtube.com/watch?v=kYWUEV_e2ss

The core of RLHF

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.

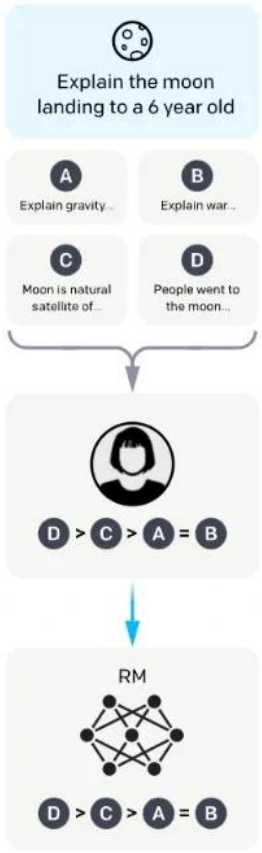
Step 2

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

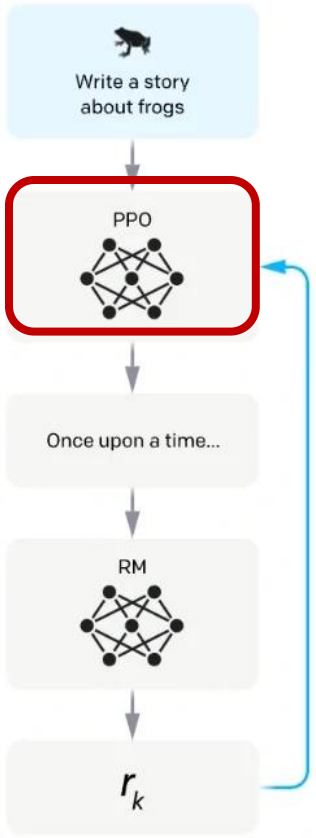
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

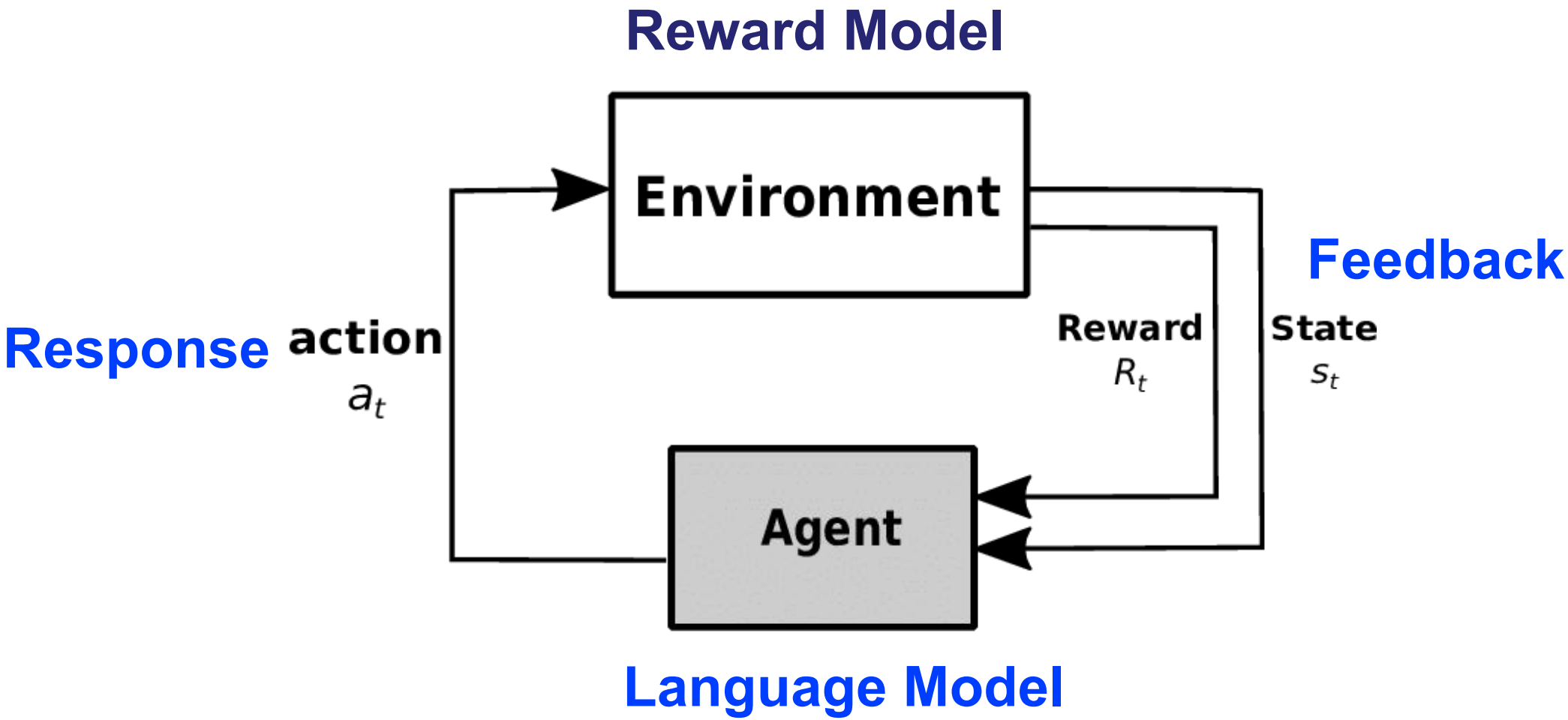
The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



Proximal Policy Optimization

Reinforcement learning from human feedback



Proximal Policy Optimization (PPO)

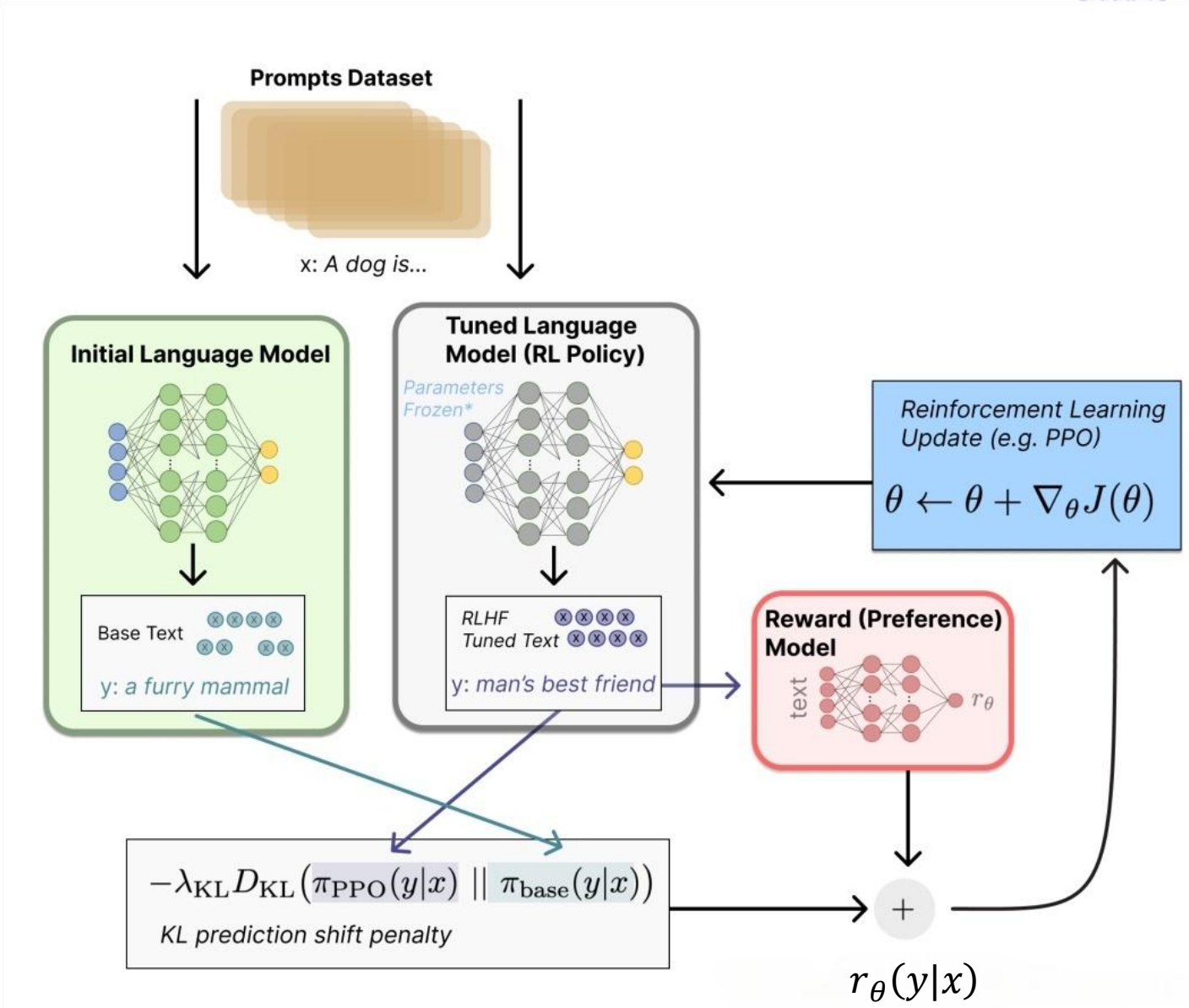
- Introduced by OpenAI in 2017 and is considered as a state-of-the-art policy-based RL algorithm
- Enhances training stability in policy updates by limiting the magnitude to prevent large, disruptive changes
- Balances between exploration and exploitation by maintaining proximity to proven policies

https://jonathan-hui.medium.com/rl-proximal-policy-optimization-ppo-explained-77f014ec3f12

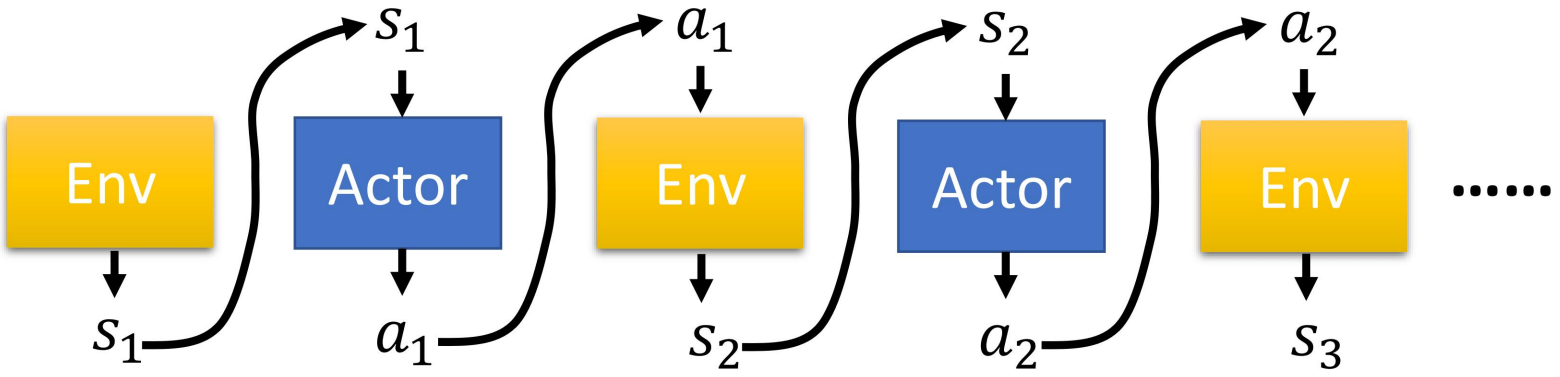
Models in PPO

- Initial Language Model
- Tuned Language Model (RL Policy)
- Reward (Preference) Model

- Reinforcement Learning Update (e.g., PPO)



Actor, Environment, Reward



Trajectory $\tau = \{s_1, a_1, s_2, a_2, \cdots, s_T, a_T\}$

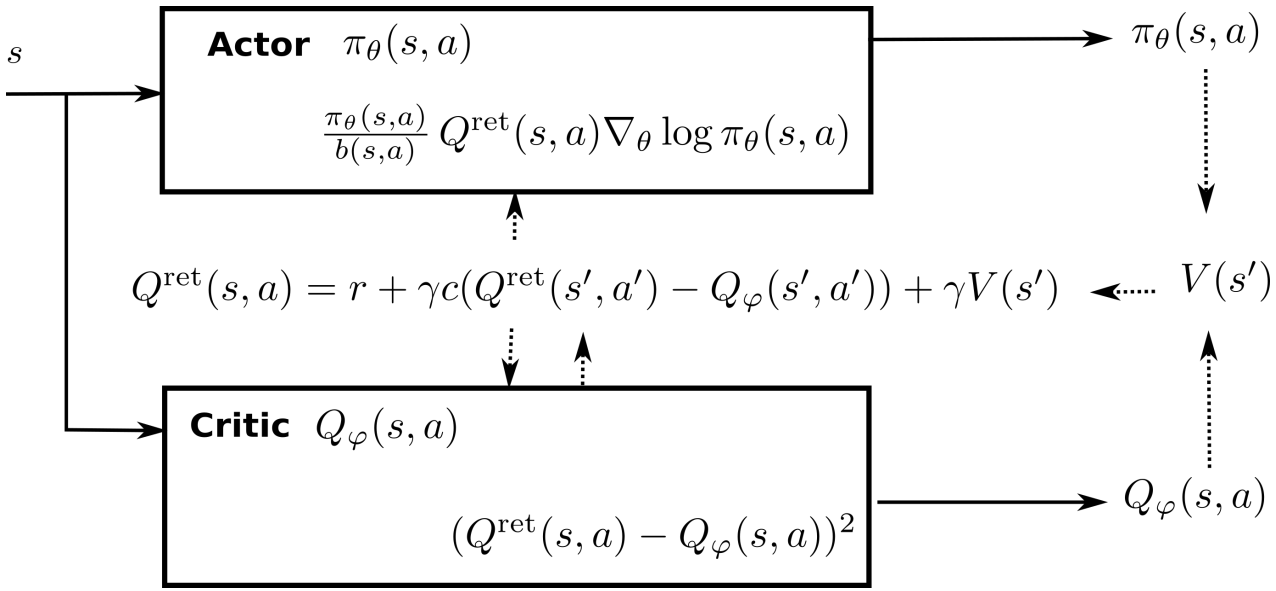
$$p_\theta(\tau)$$

$$= p(s_1)p_\theta(a_1|s_1)p(s_2|s_1,a_1)p_\theta(a_2|s_2)p(s_3|s_2,a_2)\cdots$$

$$= p(s_1)\prod_{t=1}^{T} p_\theta(a_t|s_t)p(s_{t+1}|s_t,a_t)$$

PPO Algorithm

- **Idea**: constrain our policy update with the *clipped surrogate objective function to* constrain the policy change in a small range



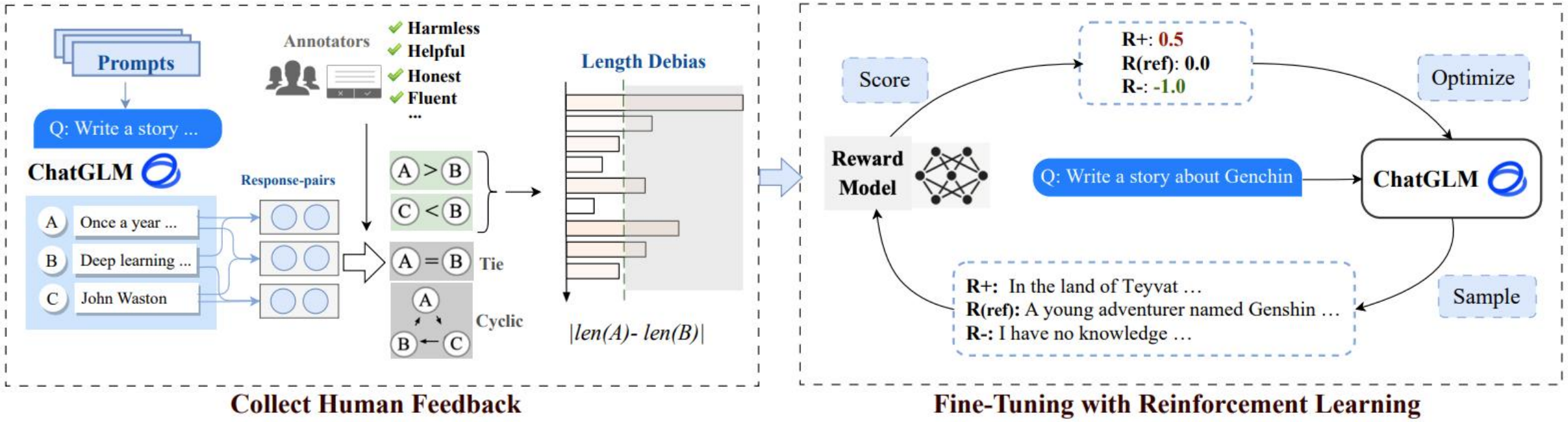
Clipped Surrogate Objective Function:

$$L^{CLIP}(\theta) = \widehat{\mathbb{E}}_t[\min\left(r_t(\theta)\widehat{A}_t,\ clip(r_t(\theta),\ 1-\epsilon,\ 1+\epsilon)\widehat{A}_t\right)]$$

Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms. 2017.

Optimize ChatGLM from Feedback

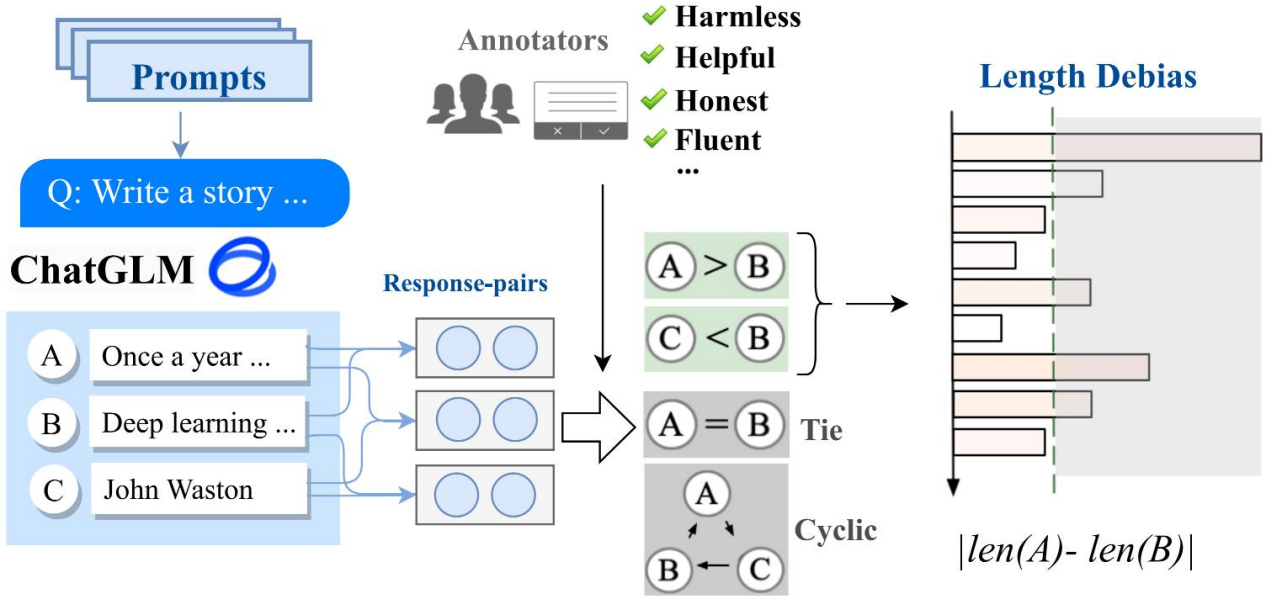
- Train reward model(s) to model human feedback
  - Elaborate data curation to accurately model preference
- On-policy reinforcement learning to optimize ChatGLM from instant feedback in a stable way



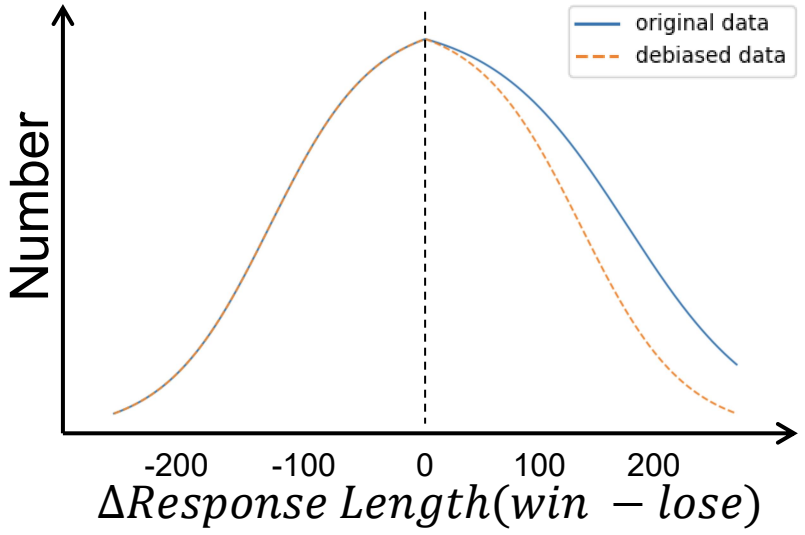
Reward Model as General Preference

Train reward model(s) to model human feedback

- Human preference collection with *elaborate data curation*
  - Resolve conflict preference
  - Reduce various human annotation bias



Human preference collection pipeline

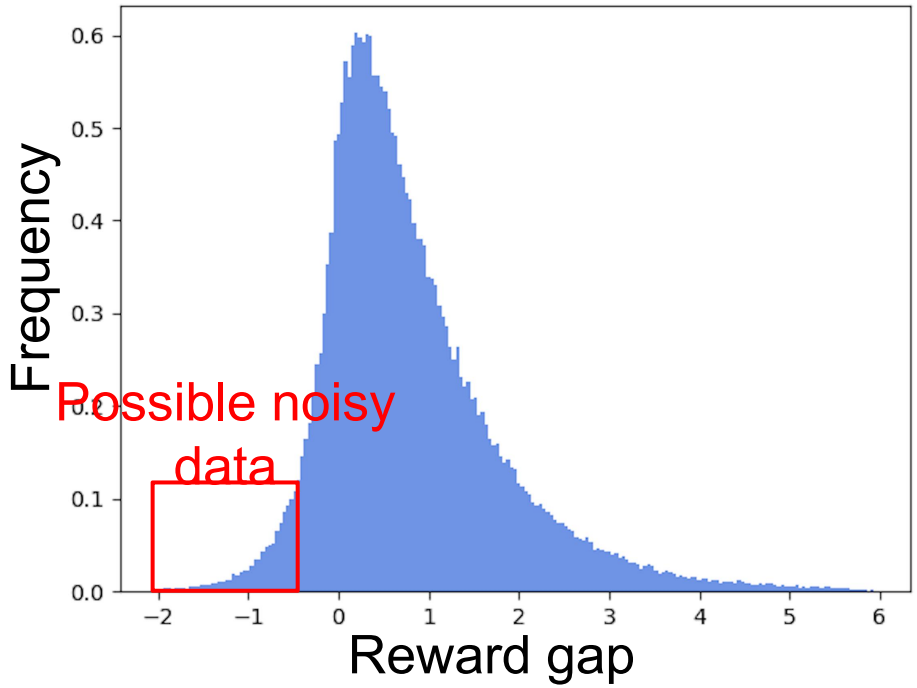


Pearson correlation between ***length*** and ***reward***
~0.3 (original) → **~0.03** (debias)

Reward Model as General Preference

Iterative reward model refinement

- detect and remove/reannotate noisy data
- retrain reward model with margin loss



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

⇩ margin loss

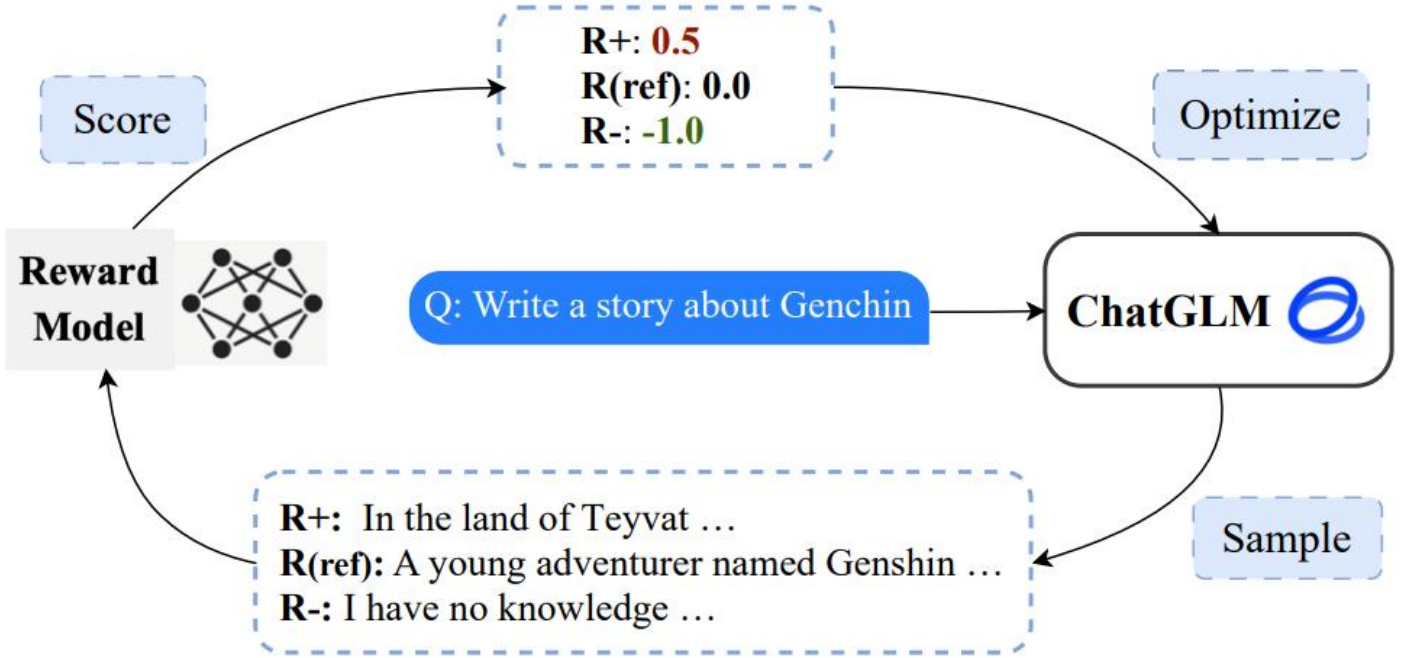
$$\mathcal{L}_{RM} = -E_{(x,y_w,y_l)\sim\mathcal{D}_{RM}}[\log(\sigma(r_{\phi_2}(x,y_w) - r_{\phi_2}(x,y_l) - \text{margin}(x,y_w,y_l)))]$$

ChatGLM with On-Policy Optimization

- Optimize ChatGLM by maximizing the cumulative reward.

$$\text{argmax}_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y_g\sim\pi_\theta}[r_\phi(x, y_g)]$$

- Iterative "*sampling – collecting feedback – optimization*"



ChatGLM with On-Policy Optimization

- Practical lessons
  - Token-level KL can help maintain training stability and reduce reward hacking during PPO

$$r(x, y_g) = r_\phi(x, y_g) - \beta D_{KL}(\pi_\theta(y_g|x)||\pi_0(y_g|x))$$

  - Reward bias reduction: Subtracting a baseline reward from the original reward can help reduce bias and improve stability

$$r(x, y) = (r_\phi(x, y_g) - r_\phi(x, y_{ref})) - \beta D_{KL}\left(\pi_\theta(y_g|x)||\pi_0(y|x)\right)$$

  - Incorporating next-token-prediction loss of SFT data can reduce capability shifting in RLHF training.

$$\mathcal{L} = \mathcal{L}_r - \beta \cdot \sum_{(x,y)\sim\mathcal{D}_S} \log \pi_\theta(y|x)$$

ChatGLM with On-Policy Optimization

- Reward bias reduction
  - Training & Inference of reward model should be both in the pairwise way.
  - Reward distribution varies across different tasks
  - Solution:
    - Add **"*reference reward*"** by pre-sampling a reference response

$$r(x, y) = (r_\phi(x, y_g) - r_\phi(x, y_{ref})) - \beta D_{KL}\left(\pi_\theta(y_g|x)||\pi_0(y|x)\right)$$

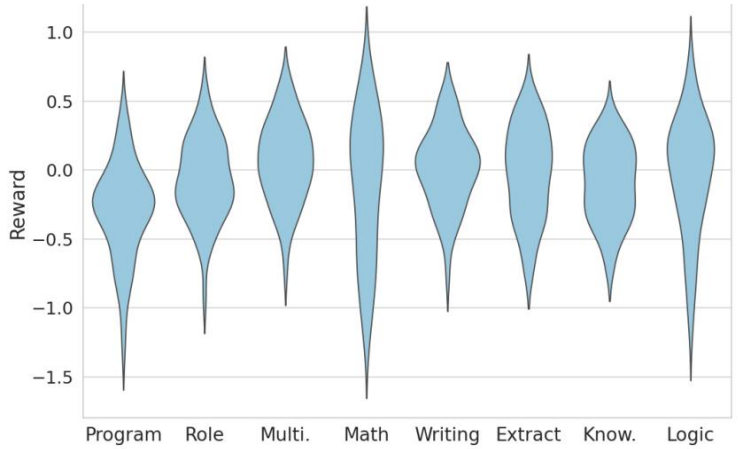
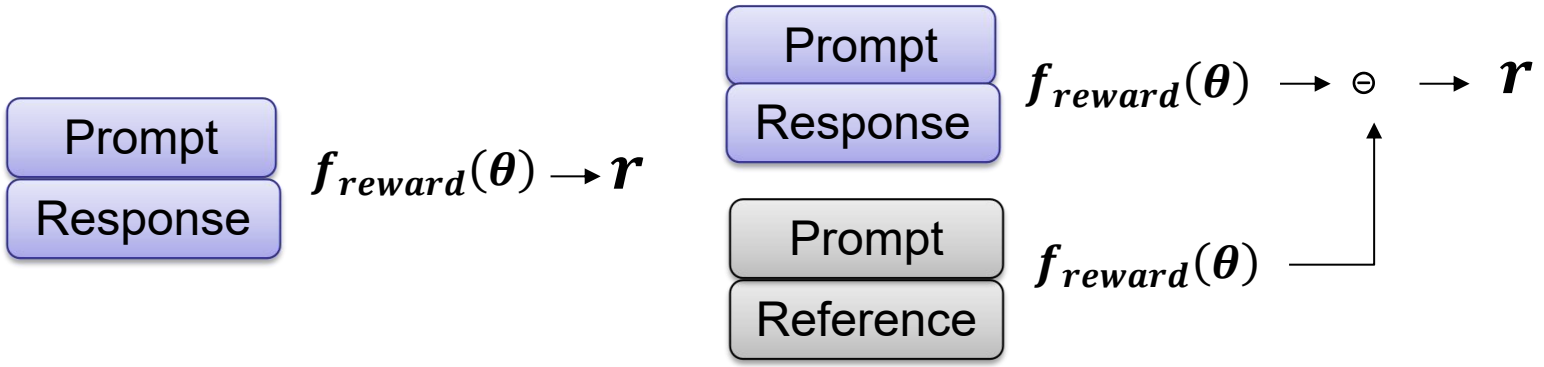


Figure 3: Reward distributions on different tasks.

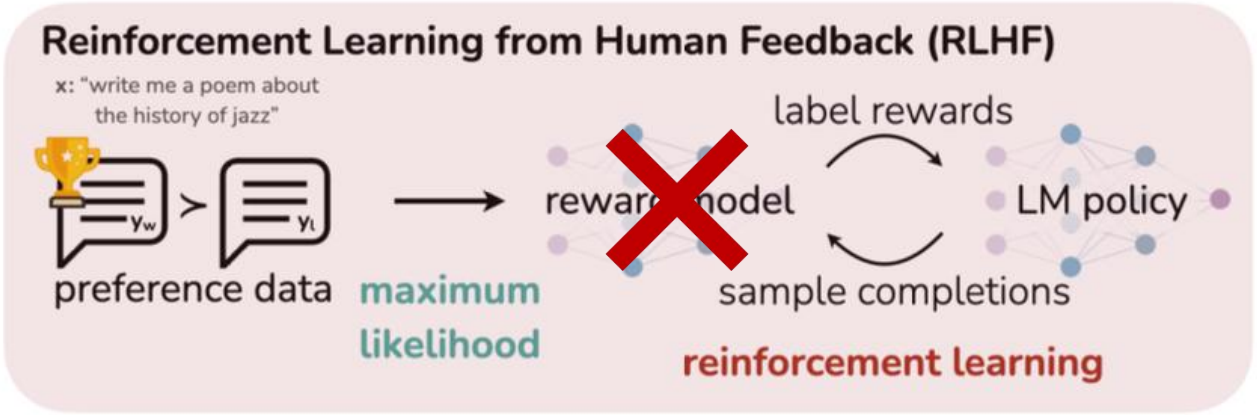


Issues with RLHF

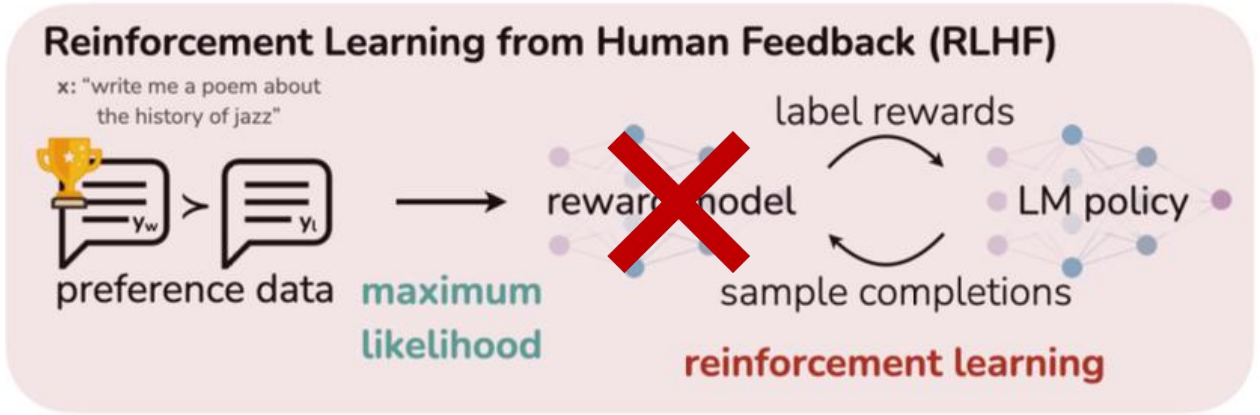
- Reward Model
  - High demands for accuracy and generalizability, otherwise easy to be hacked.
  - However, the consistency rate of annotation data in the industry is generally only around 70%.
- Reinforcement Learning
  - The rewards are too sparse (sentence score only obtained at the last step)
  - There are many PPO hyperparameters, resulting in very unstable effects.

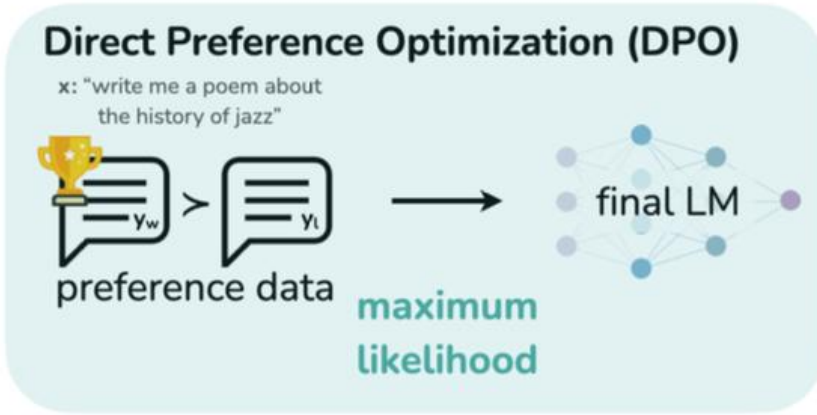
Direct Preference Optimization

Do we need the reward model?



No reward model training





Direct Preference Optimization (DPO)

- DPO, a simple RL-free algorithm for training language models from preferences
- Conversion of the RLHF optimization problem:

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

Policy to be optimized **Reference policy**

Optimal policy shouldn't be too different from reference policy

Optimal solution to KL-constrained maximization objective

Reward model

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

Where $Z(x) = \sum_y \pi_{ref}(y|x) \exp\left(\frac{1}{\beta} r(x,y)\right)$ is the partition function

Rafailov R, Sharma A, Mitchell E, et al. Direct preference optimization: Your language model is secretly a reward model. 2024.

DPO Derivation

- Reward function in terms of its corresponding **optimal policy** π_r, the **reference policy** π_{ref}, and the unknown partition function $Z(\cdot)$:

$$r(x, y) = \beta \log \frac{\pi_r(y|x)}{\pi_{\text{ref}}(y|x)} + \beta \log Z(x)$$

- Human preference probability (using Bradley-Terry model) in terms of **optimal policy** π^* and **reference policy** π_{ref} :

$$p^*(y_1 \succ y_2|x) = \frac{1}{1 + exp\left(\beta \, log \frac{\pi^*(y_2|x)}{\pi_{\text{ref}}(y_2|x)} - \beta \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)}\right)}$$

Preferred response

DPO

- Maximum likelihood objective for a parameterized policy πθ:

$$L_{DPO}(\pi_\theta; \pi_{\text{ref}}) = -E_{(x, y_w, y_l)\sim\mathbb{D}}\left[\log\sigma\left(\beta\, log\, \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right)\right]$$

Maximize the probability of y_w **Minimize the probability of y_l**

- What does the DPO update do?

Higher weight when reward estimate is wrong

Increase likelihood for preferred response

Decrease likelihood for dispreferred response

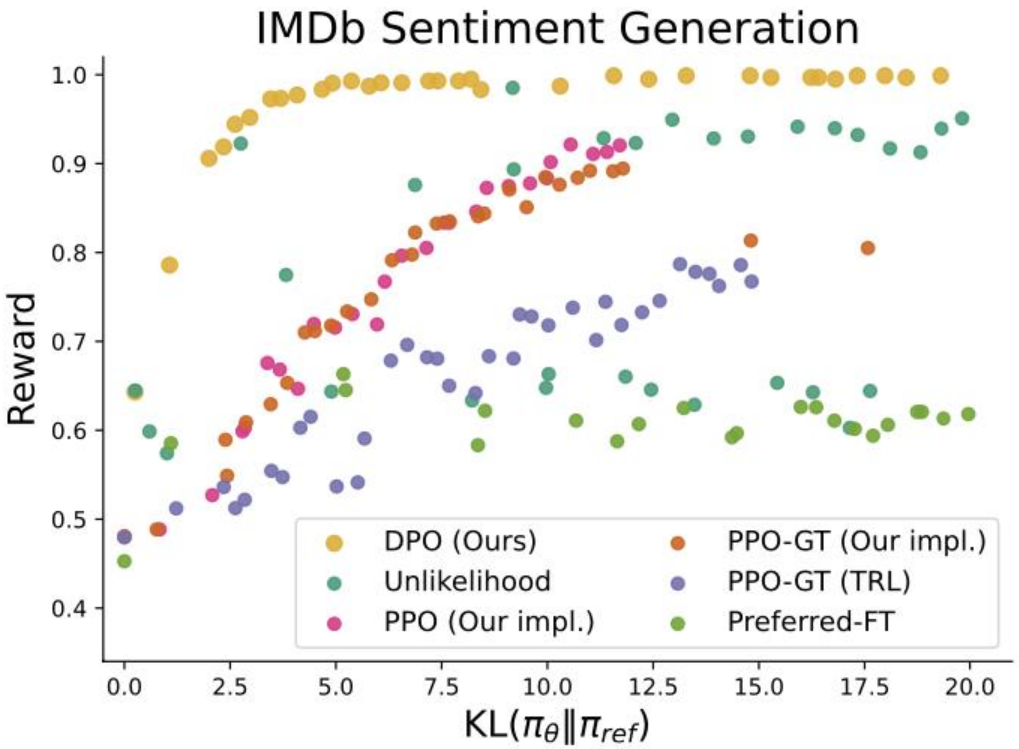
$$\nabla_\theta L_{\text{DPO}}(\pi_\theta; \pi_{ref}) = -\beta E_{(x, y_w, y_l)\sim\mathbb{D}}\left[\sigma\left(\widehat{r_\theta}(x, y_l) - \widehat{r_\theta}(x, y_w)\right)\left[\nabla_\theta \log \pi(y_w|x) - \nabla_\theta \log \pi(y_l|x)\right]\right]$$

$$\widehat{r_\theta}(x, y) = \beta\, log\, \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$$

Instance where DPO outperforms PPO

- Task : Controlled Sentiment Generation
- **Goal**: For a given movie review x, generate y with positive sentiment

- DPO's reward/KL tradeoff strictly dominates PPO： DPO is more stable and performs better in generating positive sentiment from movie reviews than PPO. For example, given a movie review with unclear sentiment, DPO can more effectively adjust the output to exhibit positive sentiment.



Reinforcement Learning in the development of LLM

Subtopics of RL

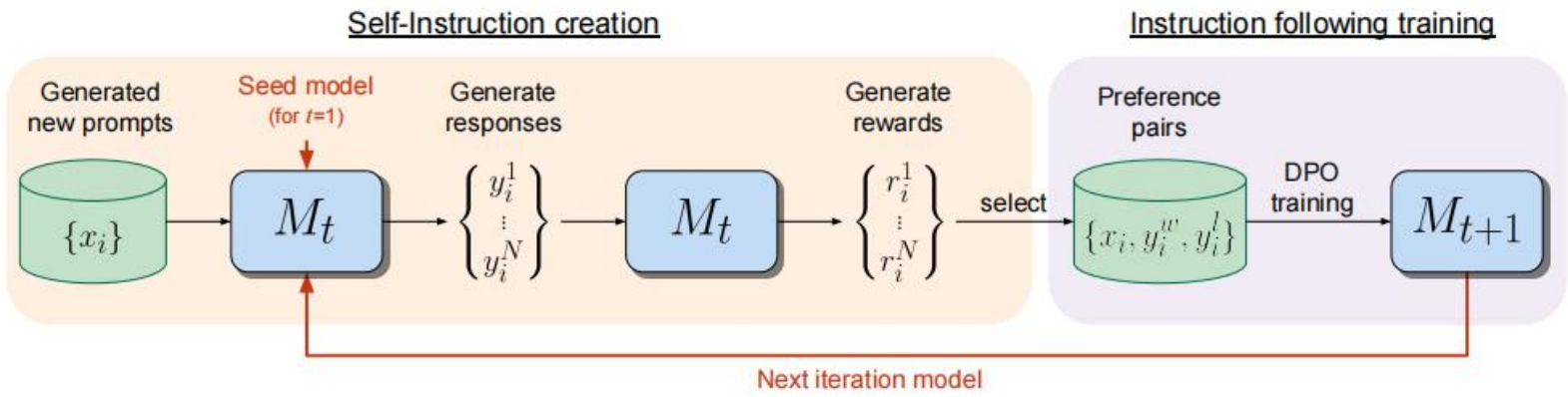
- The objective of RL was formulated as

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

- This objective encompassed two primary goals:
  - maximizing the rewards of responses
  - minimizing the deviation of the aligned policy model $\pi_\theta(y|x)$ from the initial reference (SFT) model $\pi_{ref}(y|x)$.
- The discussion on RL was divided into some subtopics:
  - On-policy RL vs. Off-policy RL
  - On-policy RL (PPO)
  - Off-policy RL
    - Reference-Based RL vs. Reference-Free RL (SimPO)
    - Instance-wise (KTO)
    - List-wise methods (RRHF)

Iterative/Online DPO

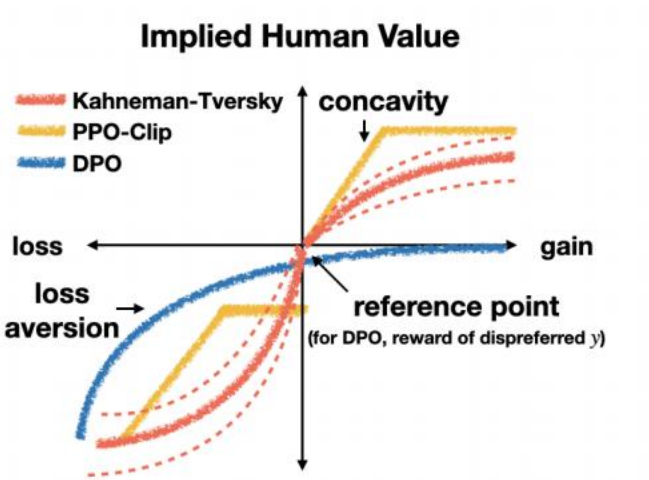
- Identified Issues
  - Current approaches commonly train reward models from human preferences, which may then be bottlenecked by human performance level
  - these separate frozen reward models cannot then learn to improve during LLM training
- Innovations: the language model itself is used via LLM-as-a-Judge prompting to provide its own rewards during training

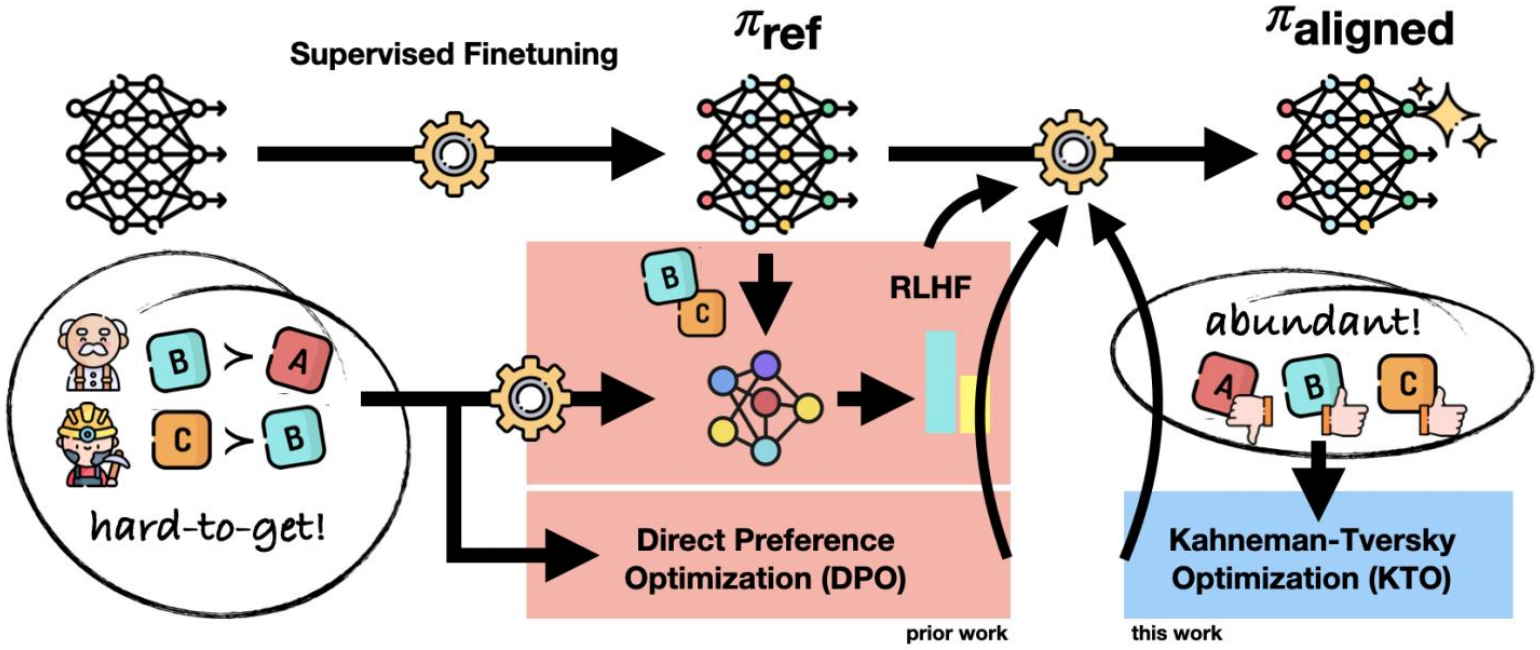


Yuan W, Pang R Y, Cho K, et al. Self-rewarding language models. 2024.

Kahneman-Tversky Optimization (KTO)

- Above methods do not fully consider human perception biases as described in prospect theory (Humans exhibit biases in perceiving random variables)
- KTO only using binary signals for learning

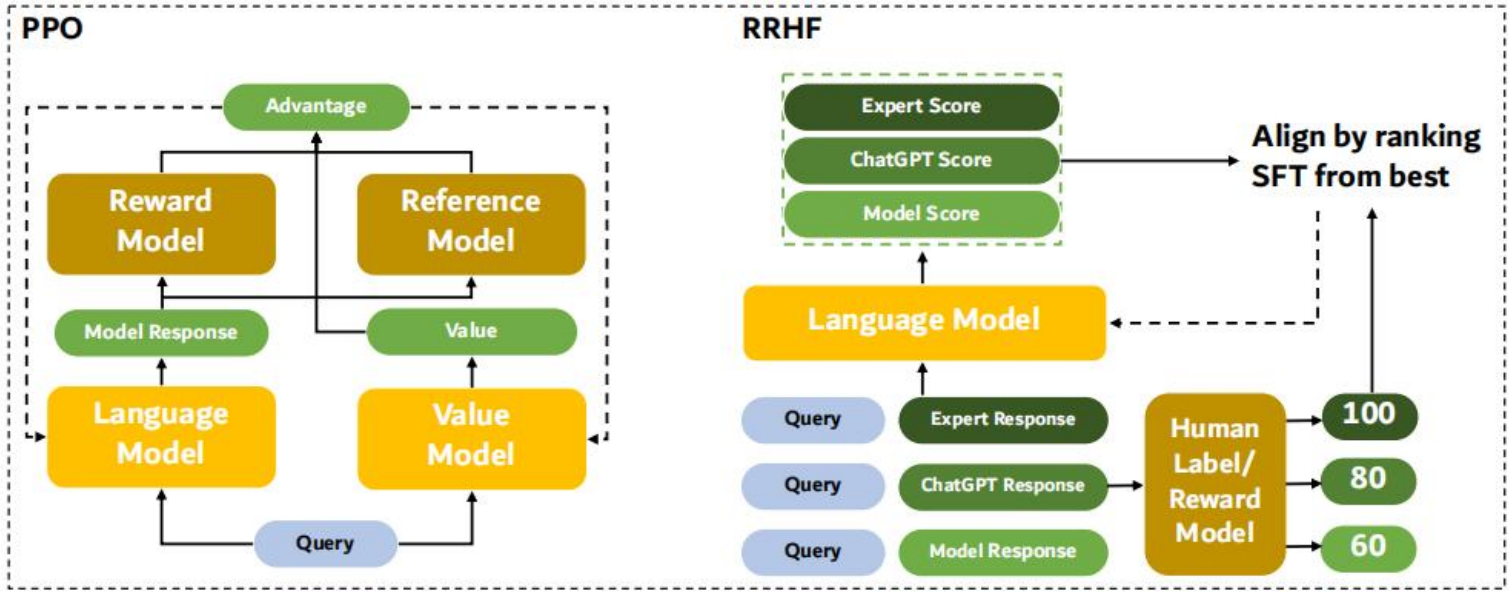




Ethayarajh K, Xu W, Muennighoff N, et al. Kto: Model alignment as prospect theoretic optimization. 2024.

Rank Responses to Align Language Models with Human Feedback without tears (RRHF)

- PPO and DPO focused on paired preferences, while research on RLHF has collected listwise preferences to accelerate the data collection process, subsequently converting them into paired preferences
- RRHF directly use listwise datasets for preference optimization



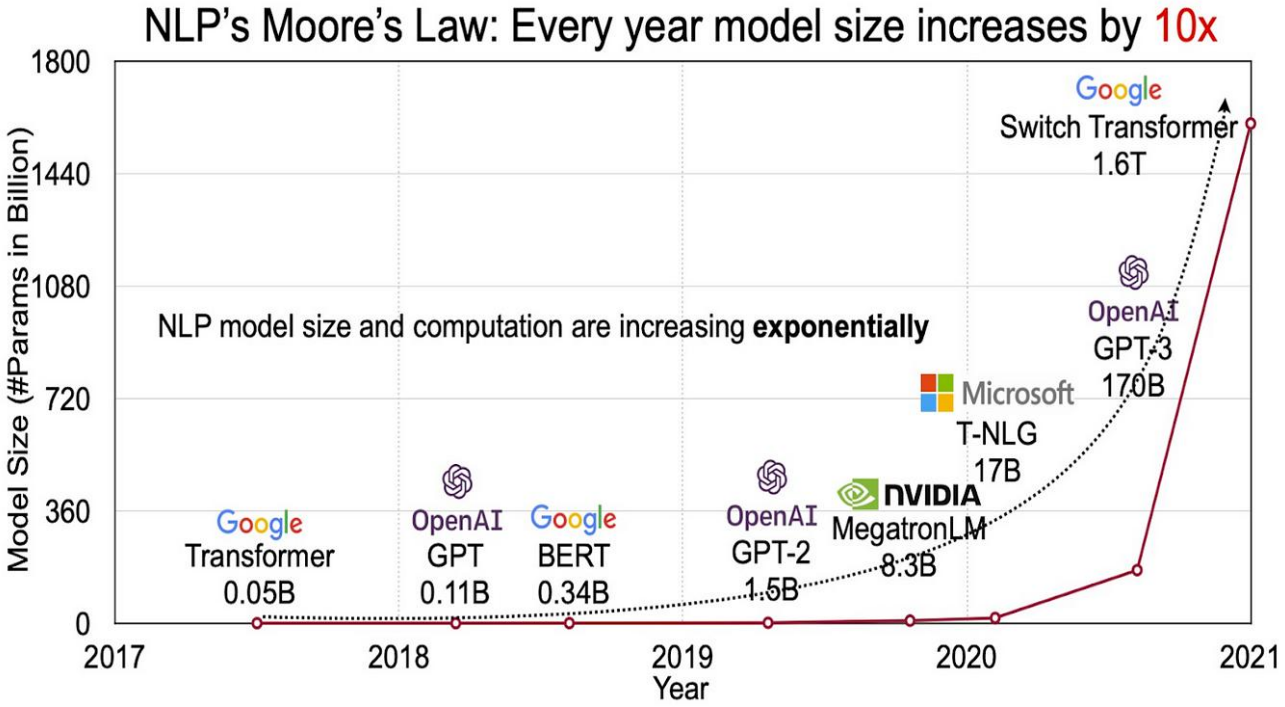
Yuan Z, Yuan H, Tan C, et al. Rrhf: Rank responses to align language models with human feedback without tears. 2023.

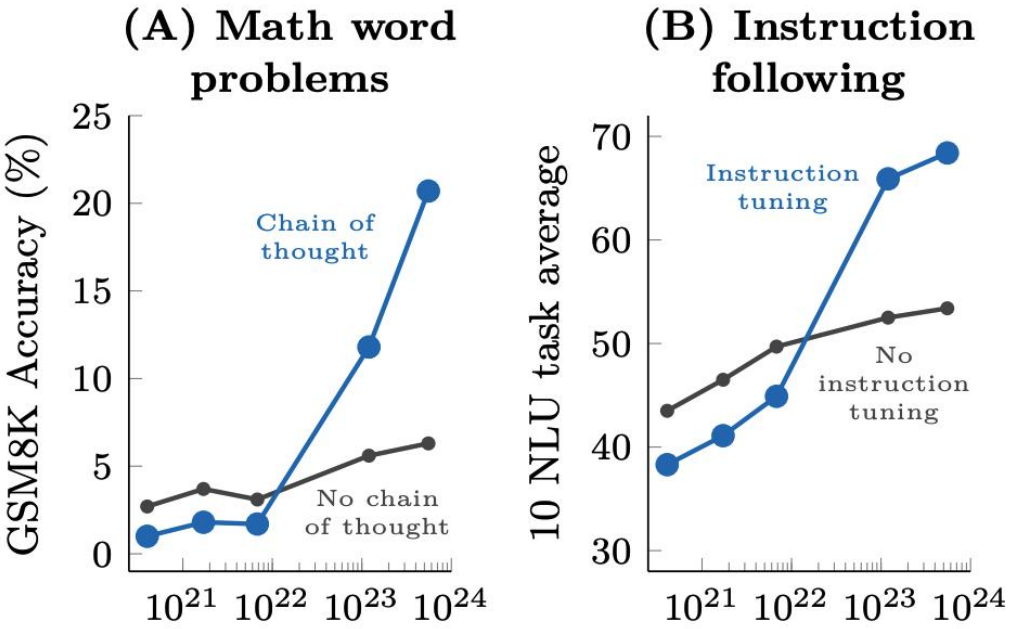
Summery

- LLMs are trained with objectives that don't align with following human provided instructions.
- InstructGPT utilizes RLHF by curating a dataset of human ratings on prompt outputs.Through higher quality data and state-of-the-art PPO method, InstructGPT achieves much better alignment with 100x fewer parameters than GPT-3.
- DPO:An RL-free algorithm used for aligning preference data. It provides much more stable results when compared with RLHF.

Scaling of Large Language Models

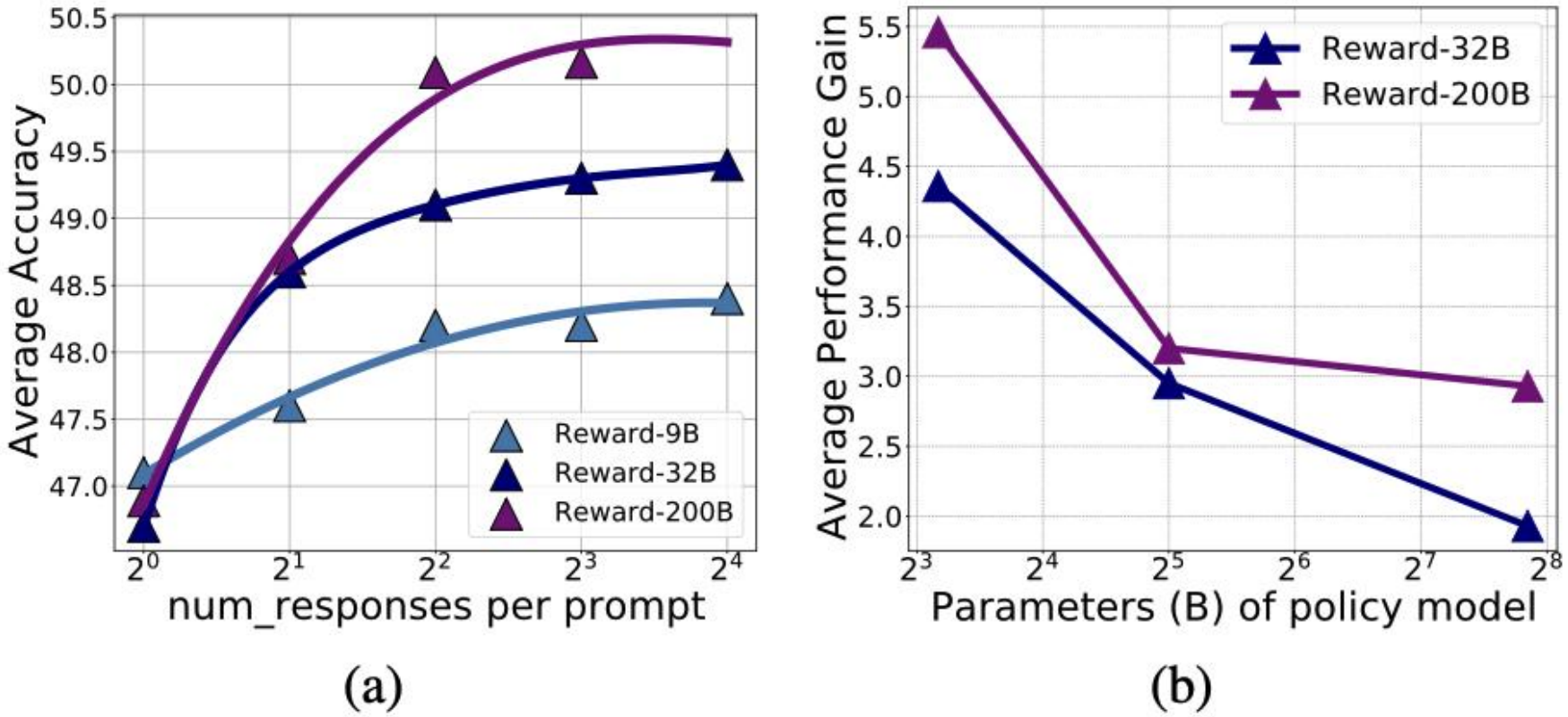
- **Scaling**: more compute leads to significantly more intelligent models
- And has been viewed as key to the success of the current LLM Pretraining





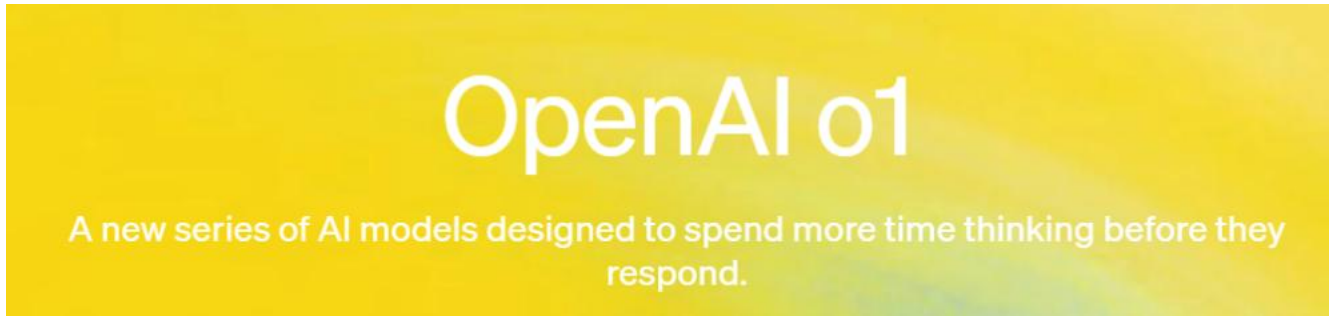
Does RLHF scale ?

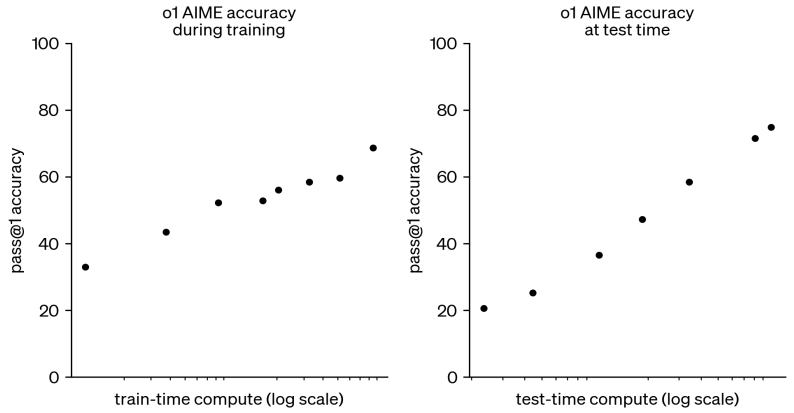
- Previous answer: **No**
- Deserve further exploration



LLMs with RL

- OpenAI's large-scale reinforcement learning algorithm teaches the model how to think productively using its chain of thought in a highly data-efficient training process.
- They had found that the performance of o1 consistently improves with more reinforcement learning (train-time compute) and with more time spent thinking (test-time compute).





- **What more can we do for LLMs with RL?** Next session：Self learning

Thank you!