Reinforcement Learning in the Era Of LLMs

Dec. 2023

Hao Sun

University of Cambridge

hs789@cam.ac.uk

Content

- Essential Concepts in RL
- Alignment: From SFT to RLHF
- Prompt Optimization as an Inverse-Alignment

Some ‘Terms’ You May Have Heard About...

- Markov Decision Processes
- States, Observations, Transitions (Dynamics), Actions, Rewards, Discout Factors...
- Model-Based / Model Free
- Value-Based / Policy-Based / Actor-Critic
- On-Policy / Off-Policy
- Online / Offline
- Discrete Control / Continuous Control

But those are not necessary...

- ~~Markov Decision Processes~~
- ~~States, Observations, Transitions (Dynamics), Actions, Rewards, Discout Factors...~~
- ~~Model-Based / Model Free~~
- ~~Value-Based / Policy-Based / Actor-Critic~~
- ~~On-Policy / Off-Policy~~
- ~~Online / Offline~~
- ~~Discrete Control / Continuous Control~~

Essential Ideas

- What is RL?

  An agent learns from trial and error, to maximize a cumulative reward.

- agent: can be human or neural networks. It has a policy (clinical guidelines, public policies)
- trial and error: Online or Offline? Real or (data-drive) Simulator?
- cumulative reward: the *Reward Hypothesis*

Reward Can be Sparse...

- "Win the game", but how?
- Imitation Learning (IL) Can Help.
- 1. Behavior Clone

  Pros: Simple, Does not need further interactions with the environment

  Cons: Compounding error, multi-modality modeling
- 2. GAIL: Generative Adversarial Imitation Learning [Ho et al. 2016]

  Pros: Solves the above cons.

  Cons: Needs more interactions (the dynamics is accessible, but reward unknown)

Ho, Jonathan, and Stefano Ermon. "Generative adversarial imitation learning." Advances in neural information processing systems 29 (2016).

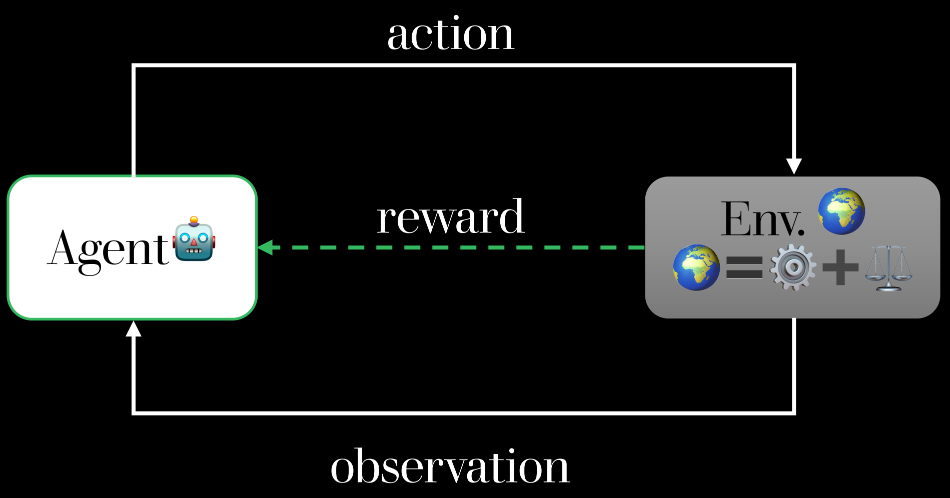
Difference between Inverse RL and IL

- Inverse-RL ≈ Imitation Learning, with an emphasis on explicit reward modeling

  Learning from logged trial and error, to find out what cumulative reward is being optimized.

- logs can be either expert decisions or non-expert decisions. Extrapolation.
- trial and error are offline data
- (the estimated) reward can be used as an evaluator of trajectories/policies

Graph: RL and Offline-RL

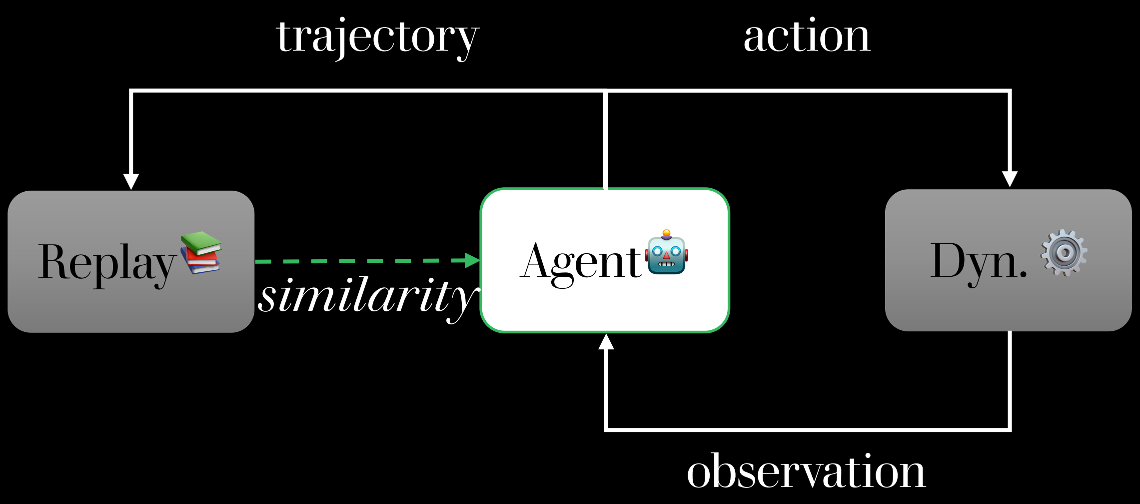


RL

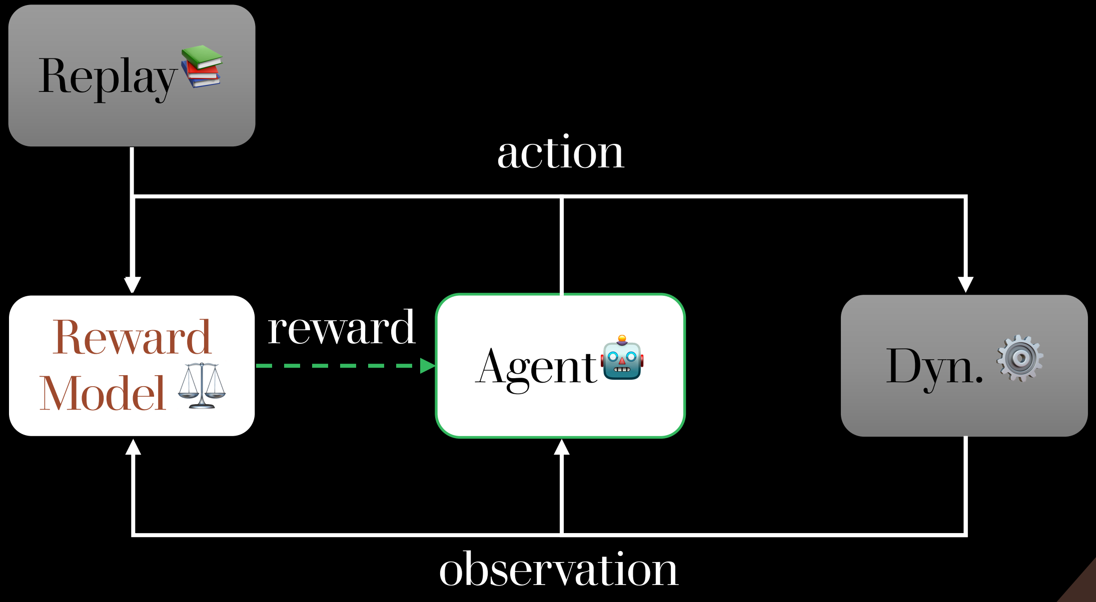
Offline-RL

Env. 🌍 = Dyn.⚙️ + Rew.⚖️

Graph: IL and IRL

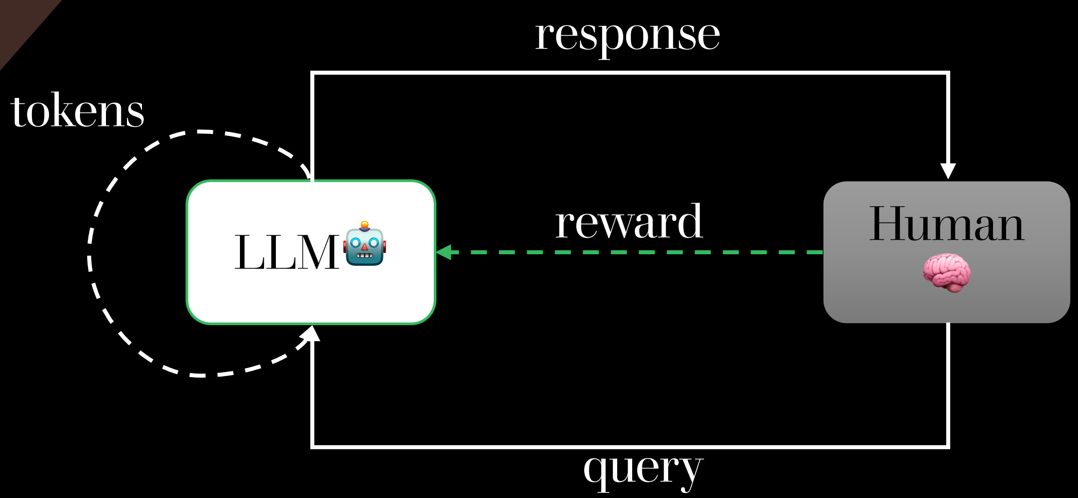


IL



IRL: Model Reward Explicitly

LLM Alignment with Human Feedback



Alignment as RL

- Why RL?

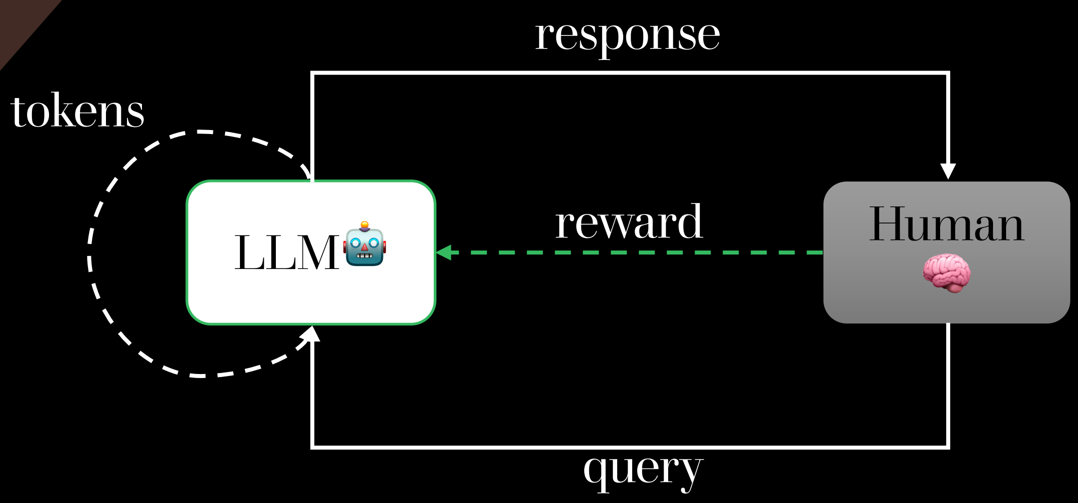
  We can not define the desired objective as a metric function.

  We can not do back-prop through a black-box 🧠

- Why not?

  Too expensive.

LLM Alignment with Human Feedback



Alignment as RL

- Why RL?

  We can not define the desired objective as a metric function.

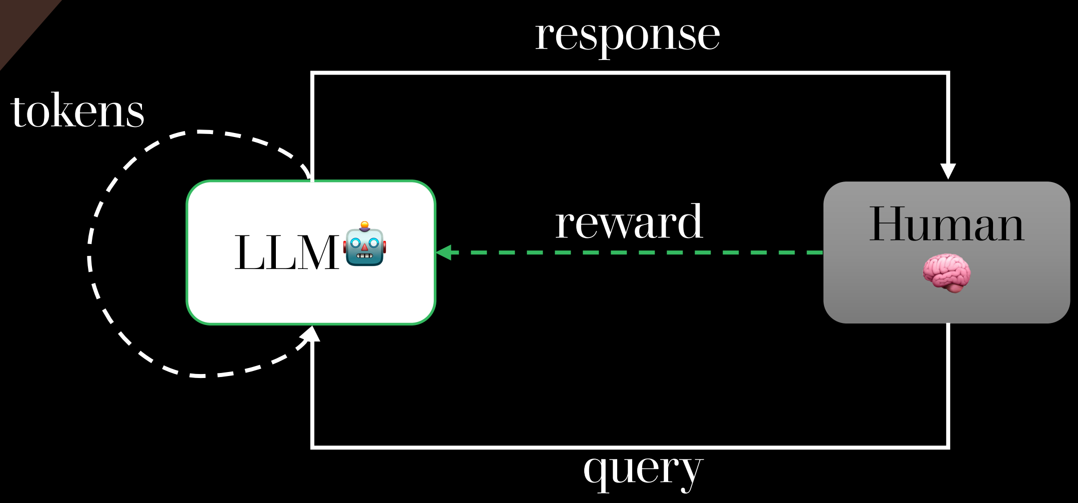
  We can not do back-prop through a black-box 🧠

- Why not?

  ~~Too expensive.~~

  OpenAI: unless you have enough volunteers (users)...

LLM Alignment with GPT-4 Feedback?



Alignment as RL

- Why RL?

  We can not define the desired objective as a metric function.

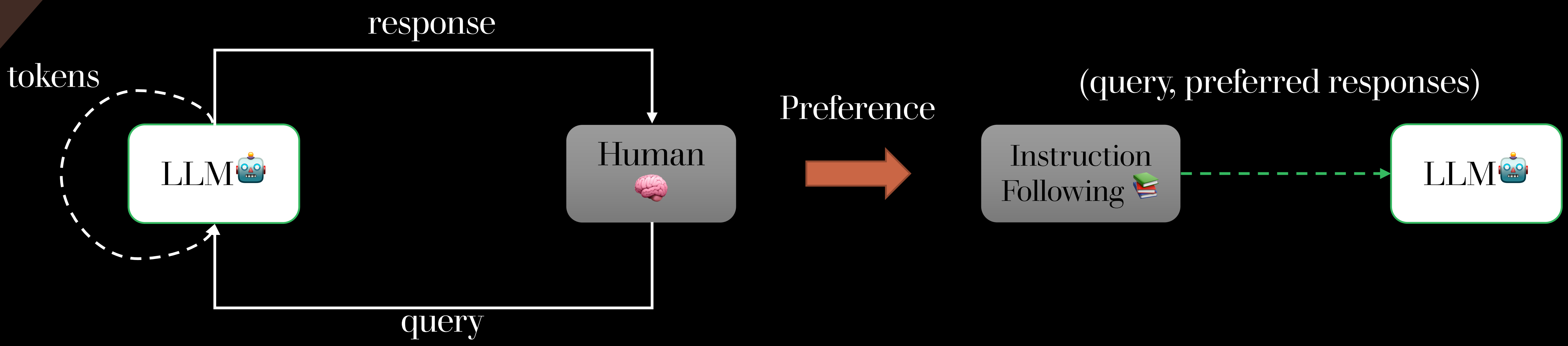
  We can not do back-prop through a black-box 🧠

- Why not?

  ~~*Too expensive.*~~

  OpenAI: unless you have enough volunteers (users)...

LLM Alignment with Human Feedback Logs



Preference Data Generation

Alignment as Offline-RL

Alignment as Offline-RL: How to Learn?

Alignment as Offline-RL

- We can always use behavior clone.

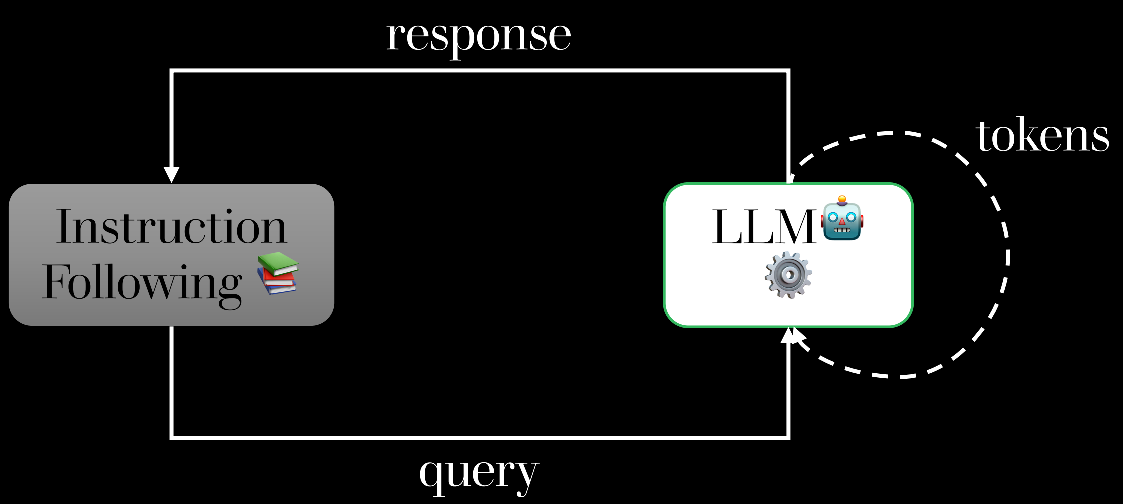
  BC = SFT, supervised-fine-tuning

- It is simple, stable, efficient.
- But language modeling is not 1-step decision.

  Compounding errors

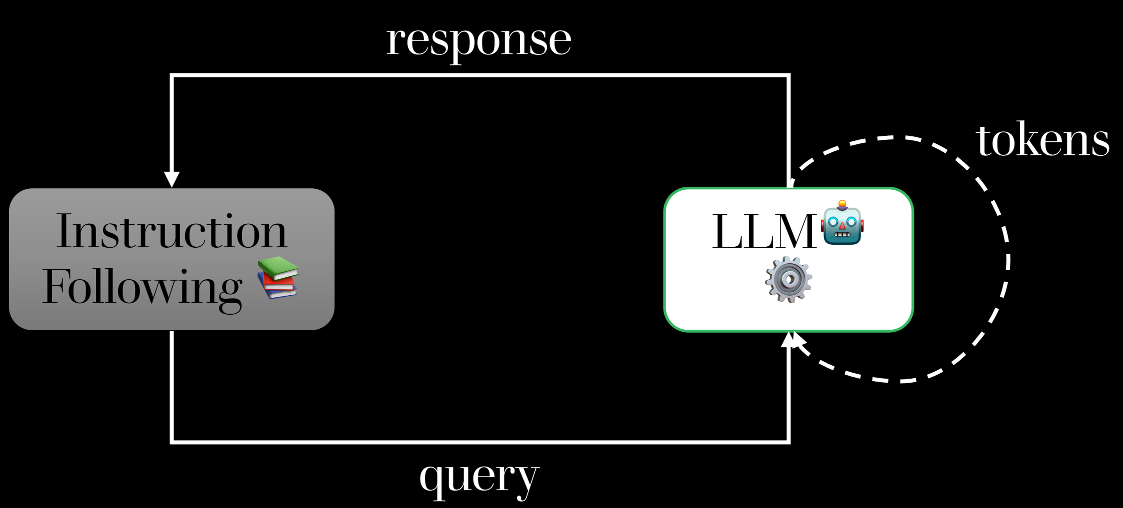
What Makes LLM Alignment Special?

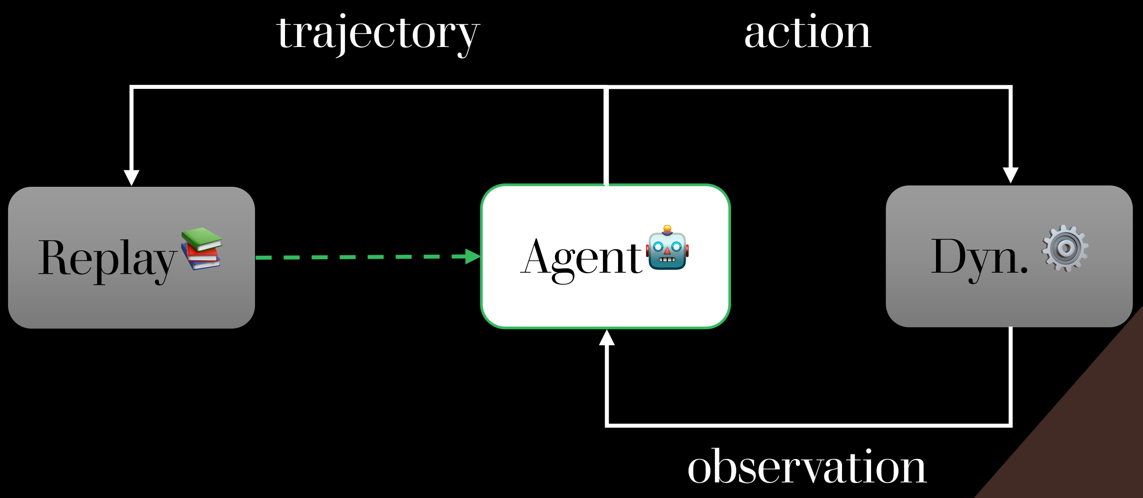
- The transition dynamics is deterministic and known!



What Makes LLM Alignment Special?

- The transition dynamics is deterministic and known!
- Recall the framework of Imitation Learning.

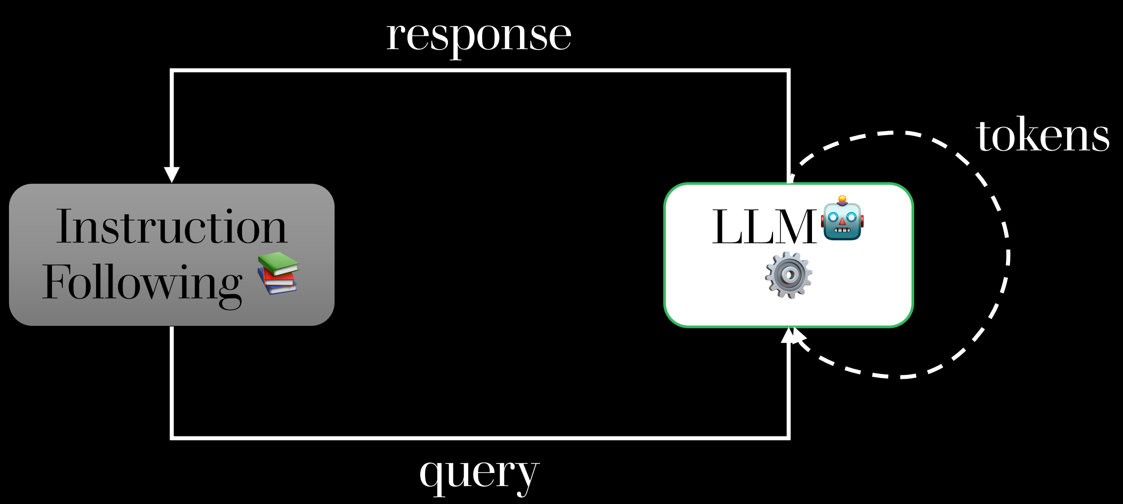


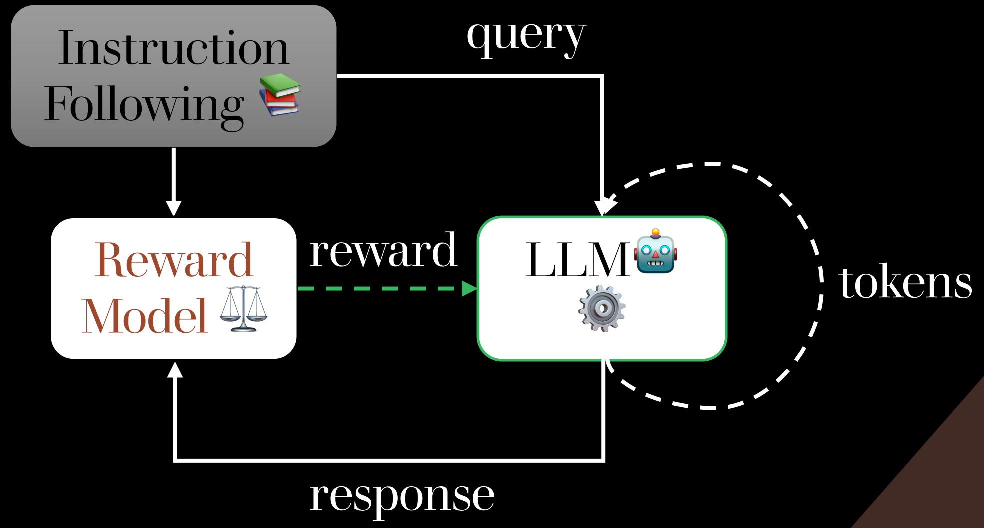


IL

RLHF: Solving Offline-RL via Online Inverse RL

- Inverse RL: learn the reward model, then optimize the policy.





SFT vs RLHF*: from the RL Perspective

- LLM alignment with logged human feedback (preference) can be interpreted as
  1. *Offline-RL --- solve it with behavior clone --- SFT*
  2. *Imitation Learning --- solve it with IRL --- RLHF*

- We can always do both:

  RLHF uses SFT as a warm-start *(OpenAI Alignment Paper)*

- Potential Alternatives? Someone would try GAIL...

---

* OpenAI's SFT is based on a separated high-quality response written by human.

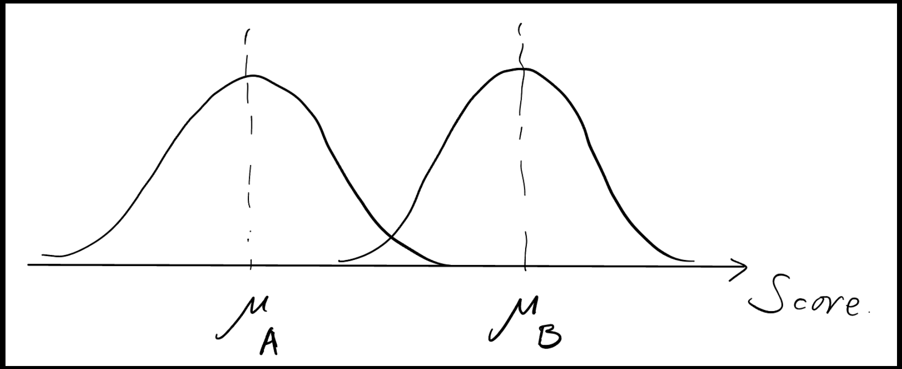
RLHF

- Underlying assumptions:
  1. *Learning a reward model is statistically easier than directly learning aligned LLMs.*
  2. *There are some higher-level metrics that can not be captured by token-level distances.*

- Two steps
  1. *Reward Learning (Response Evaluation)*
  2. *LLM Optimization (Response Optimization)*

Step 1: RM --- The Bradley-Terry Model

- Ranking is better than scoring, because the latter is noisier.
- Bradley-Terry Model is used to turn preferences into scores.



$$P(x_A > x_B | x_A \sim N(\mu_A, \sigma_A^2), x_B \sim N(\mu_B, \sigma_B^2))$$

$$P(x_A > x_B) = \frac{1}{2} + \frac{1}{2}\mathrm{erf}\left(\frac{\mu_A - \mu_B}{\sqrt{2(\sigma_A^2 + \sigma_B^2)}}\right)$$

The Bradley-Terry Model

- In RLHF (and also many MOBA games), people use a slightly different function form

$$S_i = \exp[\beta_i]$$

$$P(i \succ j) = \frac{1}{1+S_j/S_i} = \frac{1}{1+e^{-(\beta_i-\beta_j)}} = \sigma(\beta_i - \beta_j) = \frac{1}{2} + \frac{1}{2}tanh(\frac{\beta_i-\beta_j}{2})$$

- Equivalently: [DPO paper, Equation (1)]

$$p^*(y_1 \succ y_2 \mid x) = \frac{\exp\left(r^*(x, y_1)\right)}{\exp\left(r^*(x, y_1)\right) + \exp\left(r^*(x, y_2)\right)}.$$

- Practical Optimization: Binary Classification

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

Is B-T Model a Good Choice?

- In paired games, BT-model is used to attribute scores to different players

  evaluating player ability

  fairness of game (trade off between waiting time)

  number of players << number of games

  applied in an online manner --- error correction!

  reward is comparable: the same game

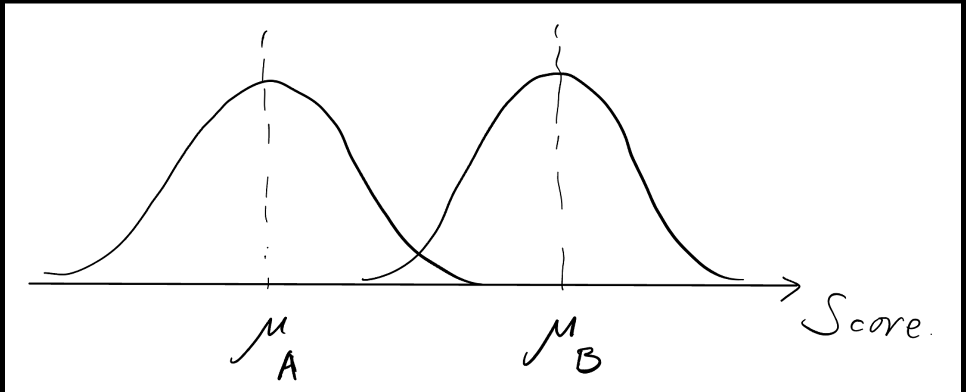
- In RLHF, labor annotation is noisy, and can be biased

  every labor + query- (paired) response = a game

  number of players (query) ~ number of games

  offline manner --- no error correction

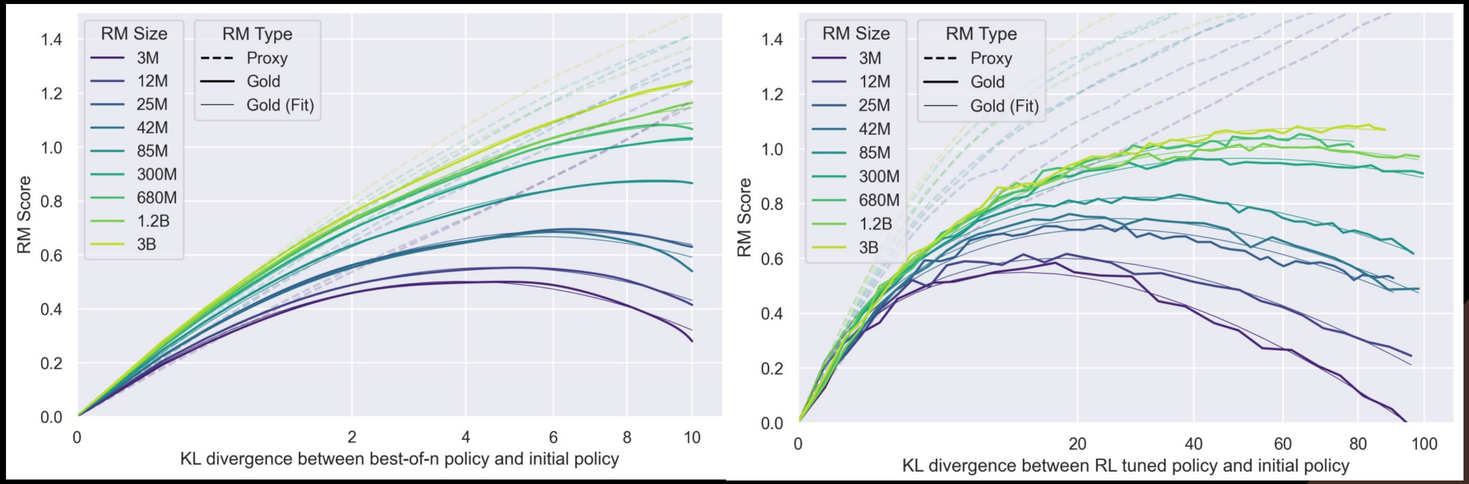
  is the reward value really comparable? E.g., Some queries can be toxic.



Reward Model Overoptimization

- Size of RM?

- LM with different sizes are used:
  OpenAI: 6B RM for 175B LM
  DeepMind: 75B RM for 75B LM
  Larger Model is Better

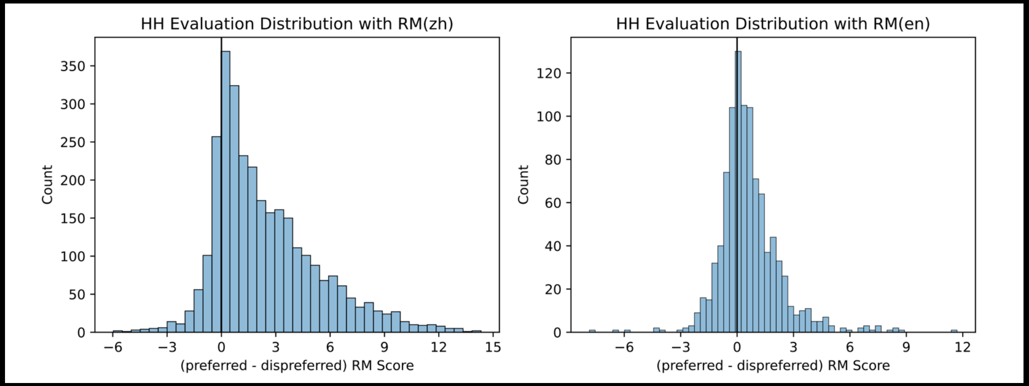


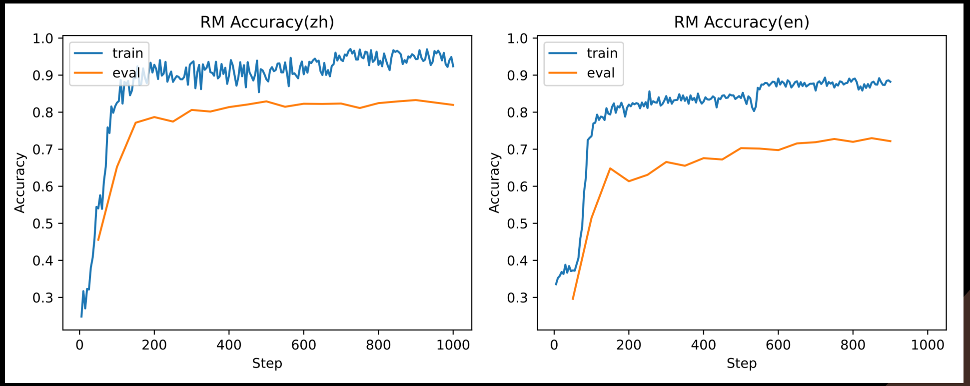
- Core Idea: The RM should be able to **understand** responses.

[Figs: Scaling Laws in Reward Model Overoptimization]

Some Evidence: RM Quality & Data Quality

- Model: LLaMA-7B / OpenChineseLLaMA
- Dataset (en): HH-RLHF 118k helpful + 42k harmless as train, 7.5k as val., 1k as test.
- Dataset (zh): annotated 31k helpful + 8k harmless, 30k train, 6k val., 3k as test.
- Results:





[Zheng, Rui, et al. "Secrets of rlhf in large language models part i: Ppo." *arXiv preprint arXiv:2307.04964* (2023).]

Step 2. Learning with RM

- RL Algorithms
  - *PPO: ([Secrets of RLHF in Large Language Models](#))*
  - *Best-of-N: Empirically better than PPO (but is not parameterized)*

  Challenges:
  - *multiple LLMs required. e.g., reference model, actor, critic, reward model.*
  - *not stable, hard to train, sensitive to hyper-params & seeds.*
  - *episodic return is a very sparse reward. (trajectory-level return)*
  - *no dynamic programming structure*
- Evolution Strategies can be a scalable alternative [Salimans et al. 2017]
  - *RAFT*
  - *RRHF*

  Sample a batch, and select the best using RM

Salimans, Tim, et al. "Evolution strategies as a scalable alternative to reinforcement learning." arXiv preprint arXiv:1703.03864 (2017).

Challenges of RLHF

- RM:
  - *Reward Overoptimization --- use larger model (other efficient choices?)*
  - *Noisy and Offline dataset --- use clean dataset (how to clean-up existing ones?)*
  - *The Bradley-Terry Model is Mostly a Wrong Assumption.*

- Policy Learning:
  - *Sparse reward --- dense reward ?*
  - *The credit assignment is not exploited (TD-nature is unclear)*
  - *Conservative update: do not need to change the LLM too much (e.g., BoN KL-div)*

DPO: Implicit Bradley-Terry Model

- Soft-Q-Learning: Not using arg-max, but soft-max. (exponential sum over q-values)
- Motivation: better exploration (max-ent RL)

$$\pi^*_{\text{MaxEnt}} = \arg\max_\pi \sum_t \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \rho_\pi} \left[r(\mathbf{s}_t, \mathbf{a}_t) + \alpha \mathcal{H}(\pi(\cdot | \mathbf{s}_t)) \right],$$

$$\eta(\pi) = \mathbb{E}_{a \sim \pi, r}[r] - \tau D_{\text{KL}}[\pi \parallel \overline{\pi}]$$

- A similar objective in RLHF (Eqn.3 in DPO):

RL Fine-Tuning Phase: During the RL phase, we use the learned reward function to provide feedback to the language model. In particular, we formulate the following optimization problem

$$\max_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(y|x)} \left[r_\phi(x, y) \right] - \beta \mathbb{D}_{\text{KL}} \left[\pi_\theta(y \mid x) \mid\mid \pi_{\text{ref}}(y \mid x) \right] \tag{3}$$

Haarnoja, Tuomas, et al. "Reinforcement learning with deep energy-based policies." *International conference on machine learning*. PMLR, 2017.

Schulman, John, Xi Chen, and Pieter Abbeel. "Equivalence between policy gradients and soft q-learning." *arXiv preprint arXiv:1704.06440* (2017).

DPO: Implicit Bradley-Terry Model

- The optimal policy has a closed-form (expressed as function of reference policy and reward)

$$\eta(\pi) = \mathbb{E}_{a\sim\pi,r}\left[r\right] - \tau D_{\mathrm{KL}}\left[\pi \parallel \overline{\pi}\right]$$

$$\pi^{\mathcal{B}}_{\bar{r}}(a) = \overline{\pi}(a)\exp(\bar{r}(a)/\tau) / \underbrace{\mathbb{E}_{a'\sim\overline{\pi}}\left[\exp(\bar{r}(a')/\tau)\right]}_{\text{normalizing constant}}.$$

- DPO put this result in preference-based learning, and cancel-out the normalizing constant

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

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

- “End-to-end”—direct optimization / a smart idea / overoptimization? (B-o-128>DPO)

Schulman, John, Xi Chen, and Pieter Abbeel. "Equivalence between policy gradients and soft q-learning." *arXiv preprint arXiv:1704.06440* (2017).

IPO: A Generalized Framework

- ΨPO:

  Ψ: Non-Decreasing Function

$$\max_{\pi} \mathop{\mathbb{E}}_{\substack{x\sim\rho \\ y\sim\pi(.|x) \\ y'\sim\mu(.|x)}} \left[\Psi(p^*(y \succ y'|x))\right] - \tau D_{\mathrm{KL}}(\pi \,||\, \pi_{\mathrm{ref}}) \,.$$

- DPO & RLHF --- $\Psi(q) = \log(\frac{q}{1-q})$
- IPO --- Ψ(q) = q (Without the B-T Assumption)
- Empirical Loss:

$$\mathop{\mathbb{E}}_{(y_w,y_l)\sim D}\left[\left(h_\pi(y_w,y_l) - \frac{\tau^{-1}}{2}\right)^2\right]$$

Azar, Mohammad Gheshlaghi, et al. "A general theoretical paradigm to understand learning from human preferences." *arXiv preprint arXiv:2310.12036* (2023).

Nash Learning from Human Feedback

- Directly Learn a Preference Model.

$$\mathcal{P}(\pi \succ \pi') \stackrel{\text{def}}{=} \mathbb{E}_{x\sim\rho}\mathbb{E}_{y\sim\pi(\cdot|x),y'\sim\pi'(\cdot|x)}\left[\mathcal{P}(y \succ y'|x)\right].$$

- Policy Learning as a Game.

$$\pi^* \stackrel{\text{def}}{=} \arg\max_{\pi}\min_{\pi'}\mathcal{P}(\pi \succ \pi').$$

Munos, Rémi, et al. "Nash Learning from Human Feedback." *arXiv preprint arXiv:2312.00886* (2023).

Instruction Following by Prompting

- Prompt engineering is an effective approach in eliciting the abilities of LLMs
- In-context Learning/Fine-tuning

  Few-Shot Prompting + In-Context Learning

  Zero-Shot Prompting
- Examples:

  CoT: let's think step by step...

  OPRO: *take a deep breath ...*
- How to design? Previous approaches: learning from trial and error.

Prompt-OIRL

- Offline Prompt Evaluation and Optimization with IRL [Sun et al. 2023]

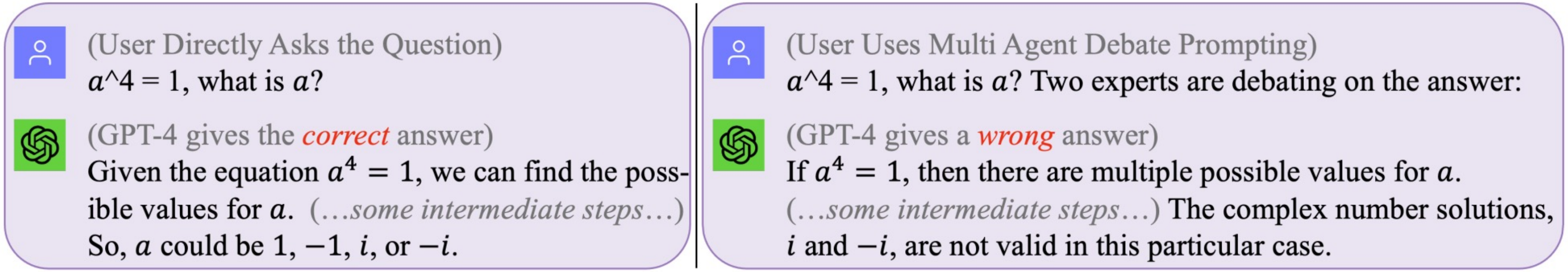


Figure 1: *A motivating example.*(Left, Right) No prompt is perfect that works for all queries. The optimal prompt is query-dependent. Yet the seeking of such prompts is hindered by the **Challenges 1-2** we identified. Our method optimizes prompt during inference on a **query-dependent** level effectively and cost-efficiently.

Sun, Hao et al. "Offline Prompt Evaluation and Optimization with Inverse Reinforcement Learning." arXiv preprint arXiv:2309.06553 (2023).

Prompting Using RL?

- RL is Expensive.

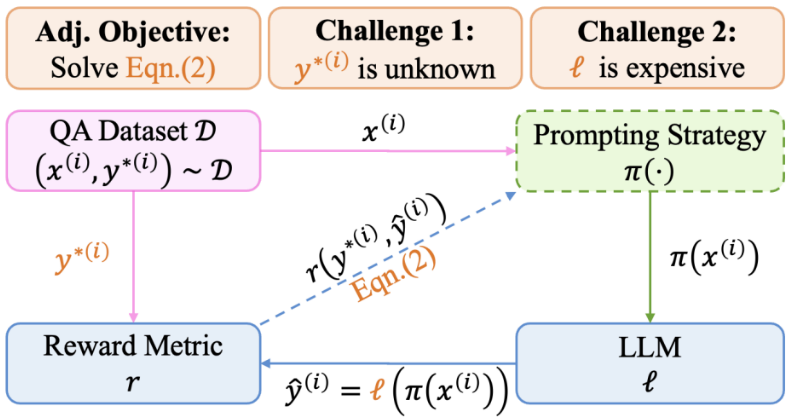


Figure 2: *The Adjusted Objective and Challenges in prompt optimization.* We use blue to denote fixed functions, pink for datasets, and green for functions to be optimized. Solid lines show the flow of outputs, and dashed lines denote the learning process.

How about using existing expert demonstrations?

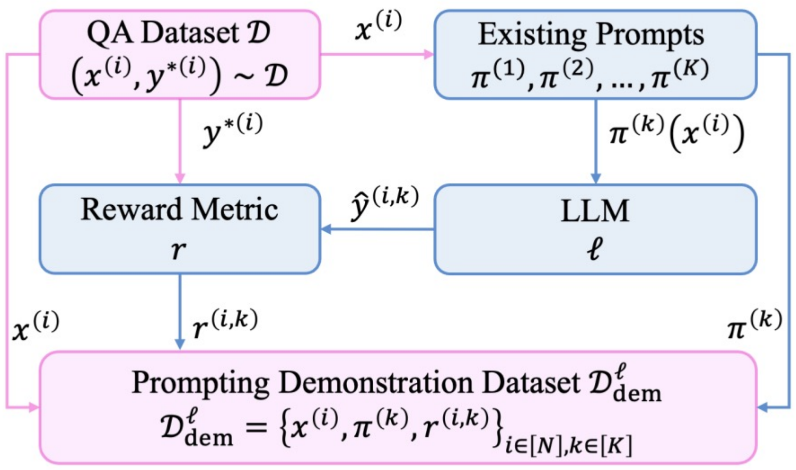


Figure 3: *The offline demonstration dataset is generated as a by-product of evaluating existing (query-agnostic) prompts.*

Sun, Hao et al. "Offline Prompt Evaluation and Optimization with Inverse Reinforcement Learning." arXiv preprint arXiv:2309.06553 (2023).

Offline Inverse RL --- RLAIF

- Reward model estimate the *preference* of LLMs

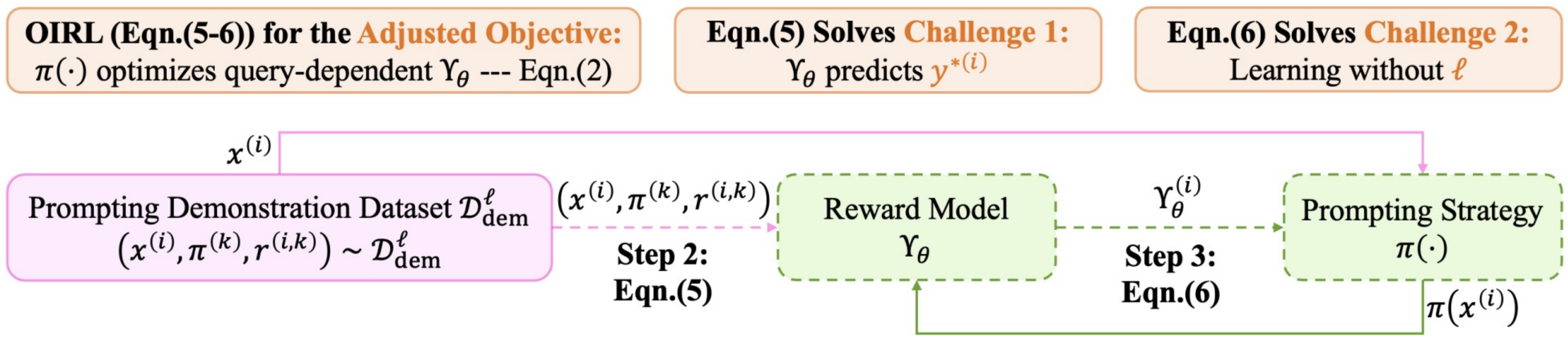


Figure 4: *Prompt-OIRL addresses the specified Objective and challenges*. It first learns a proxy reward model from the offline demonstration dataset we created in the last section. Such a learned reward model can be applied in inference time to evaluate prompts in a query-dependent manner without access to the language model, hence optimizing prompt w.r.t. such a proxy reward model solves all issues identified.

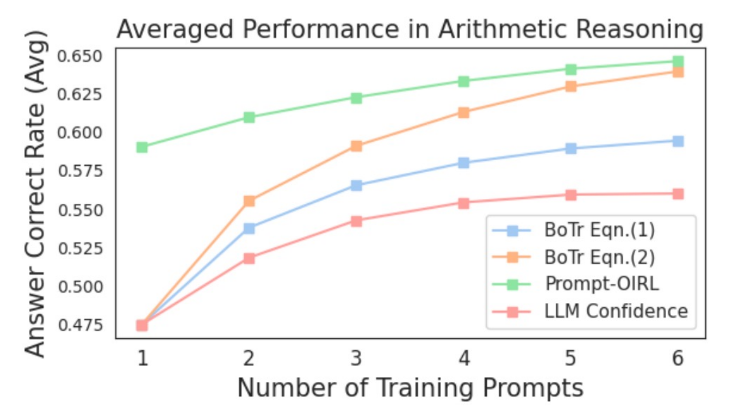
Sun, Hao et al. "Offline Prompt Evaluation and Optimization with Inverse Reinforcement Learning." arXiv preprint arXiv:2309.06553 (2023).

Results

- Experiments on Arithmetic Reasoning Datasets (GSM8K, SVAMP, MAWPS)
- TakeAways:
  1. *Prompt-OIRL further improve the ability of LLMs in inference.*
  2. *It is extremely cheap to train and deploy Prompt-OIRL.*

Table 2: prompts used in offline training dataset collection.

No.	Effective Prompts Discovered by Experts and Algorithms	Explanation
1	"The answer is:"	direct prompting
2	"Let's think step by step:"	zero-shot CoT (Kojima et al., 2022)
3	"Let's work this out in a step by step way to be sure we have the right answer:"	APE discovered (Zhou et al., 2022b)
4	"First, decompose the question into several sub-questions that need to be solved, and then solve each question step by step:"	Least-to-most (Zhou et al., 2022a)
5	"Imagine three different experts are answering this question. All experts will write down 1 step of their thinking, and then share it with the group. Then all experts will go on to the next step, etc. If any expert realizes they're wrong at any point then they leave."	Tree-of-thought (Hulbert, 2023)
6	"3 experts are discussing the question, trying to solve it step by step, and make sure the result is correct:"	multi-agent debate (Liang et al., 2023)



Summary / hs789@cam.ac.uk

- RL is learning from trial and errors to maximize a cumulative reward.
- Define reward function can be easy, but exploration of RL is hard.
- With expert demonstrations, IL can improve learning efficiency.
- Behavior Clone is the simplest IL, but it suffers from compounding errors.
- IRL first learns a RM, and then use the learned RM to optimize policy.
- SFT is behavior clone, RLHF is online IRL.
- Assumptions under Reward Model Learning --- The Bradley-Terry Model
- Overoptimization in Reward Model Learning
- Given an RM, there are multiple approaches to optimize LLMs to align with human.
- DPO, and Soft-Q-Learning, IPO, Nash-LHF
- Prompt-OIRL is able to effectivly and efficiently perform offline promt evaluation and optimization.