Why DPO is a Misspecified Estimator and How to Fix it

Aditya Gopalan (IISc) (aditya@iisc.ac.in)

Sayak Ray Chowdhury (IIT Kanpur)

Debangshu Banerjee (HP AI Research)

ICLR 2026

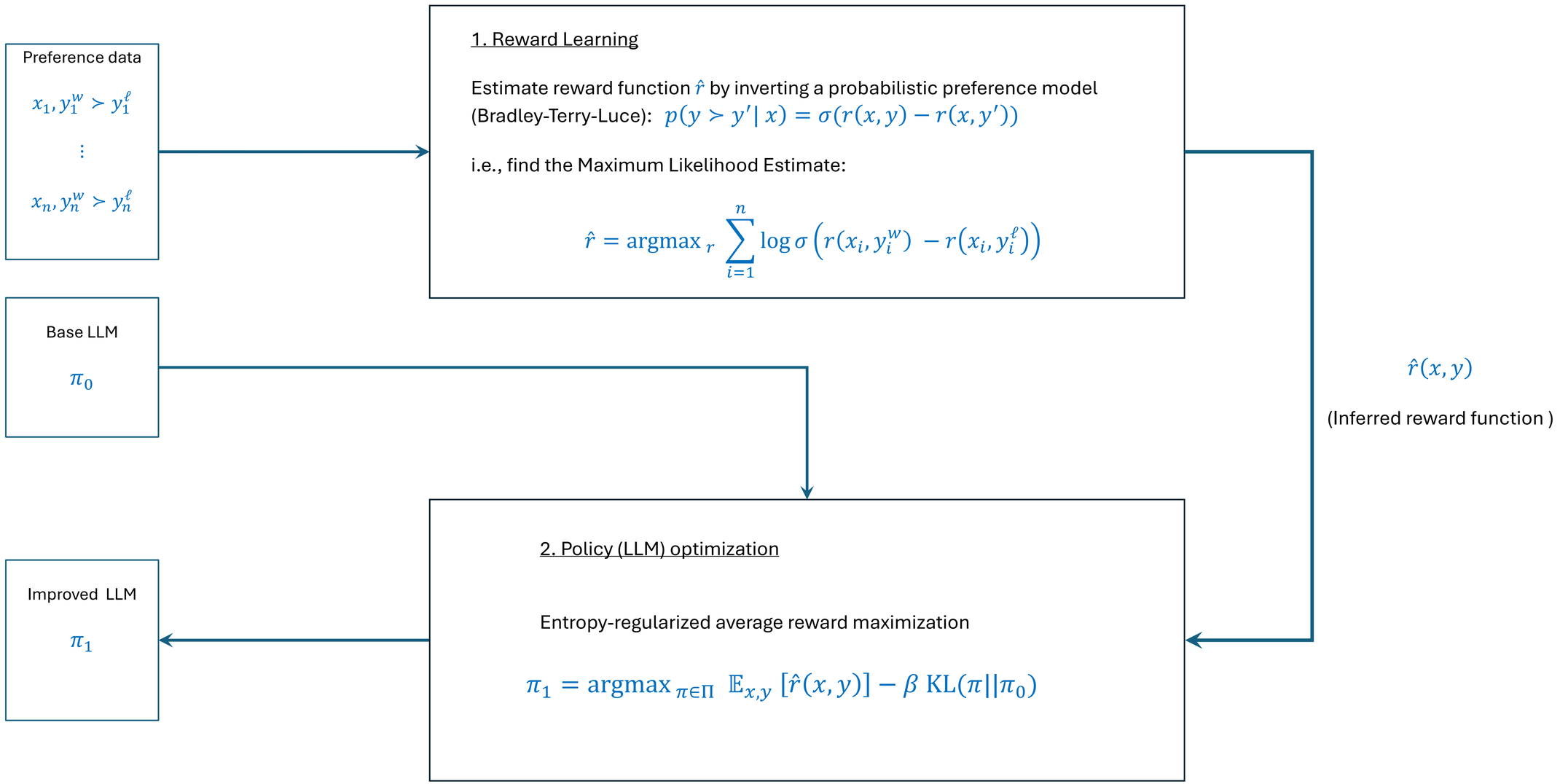
Preference-based Alignment Problem

- LLM π: Collection of conditional probabilities $\pi(y|x)$, where x is any prompt and y is any response

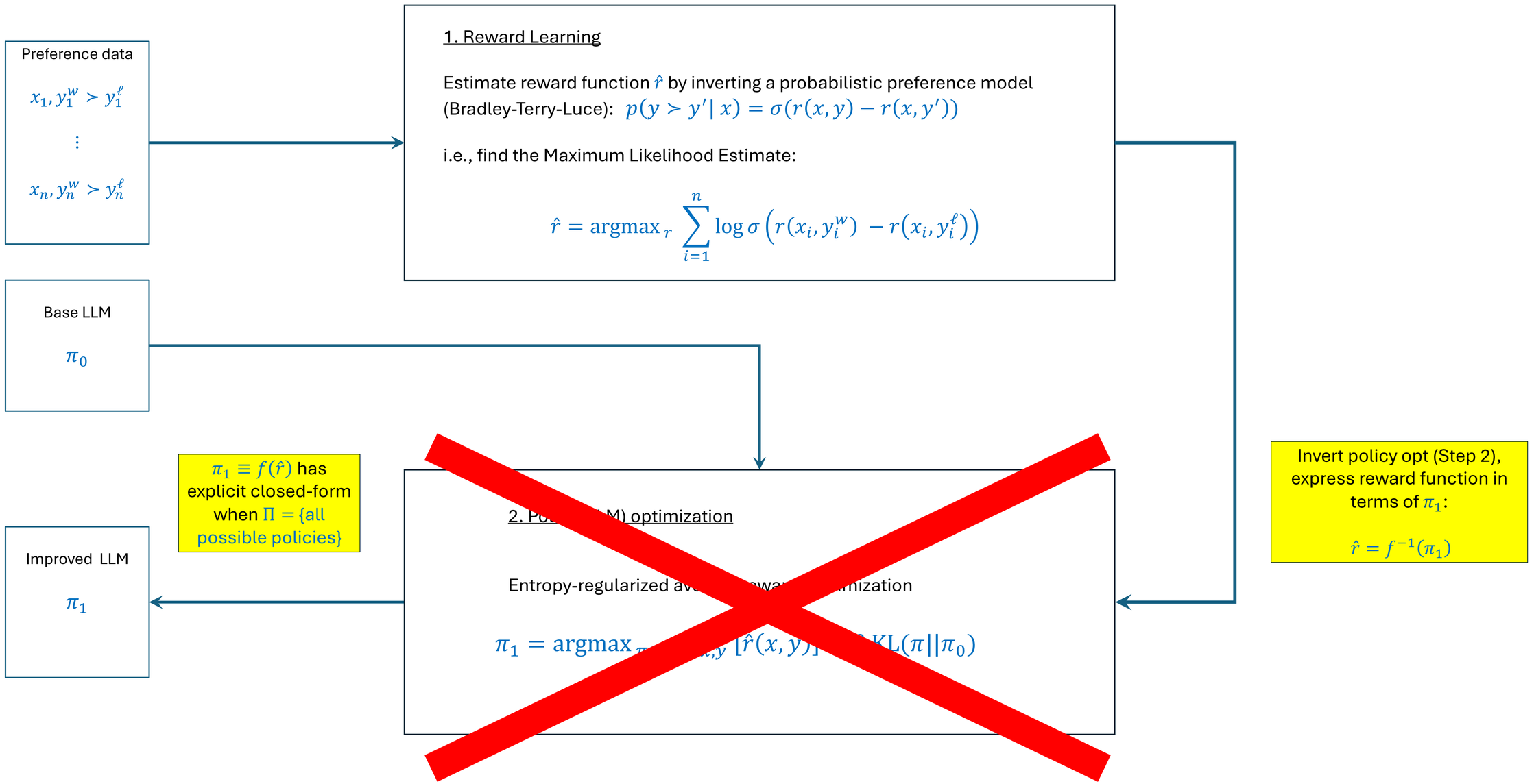
- Given preference data $\left(x_i, y_i^w \succ y_i^\ell\right)_{i=1}^n$ and a base LLM π_0, find new LLM params π_1 that are "more aligned" with the expressed preferences

- Broadly, the LLM π_1 should have higher (resp. lower) probability of outputting response y_i^w (resp. response y_i^ℓ) than the original LLM π_0 for prompt x_i

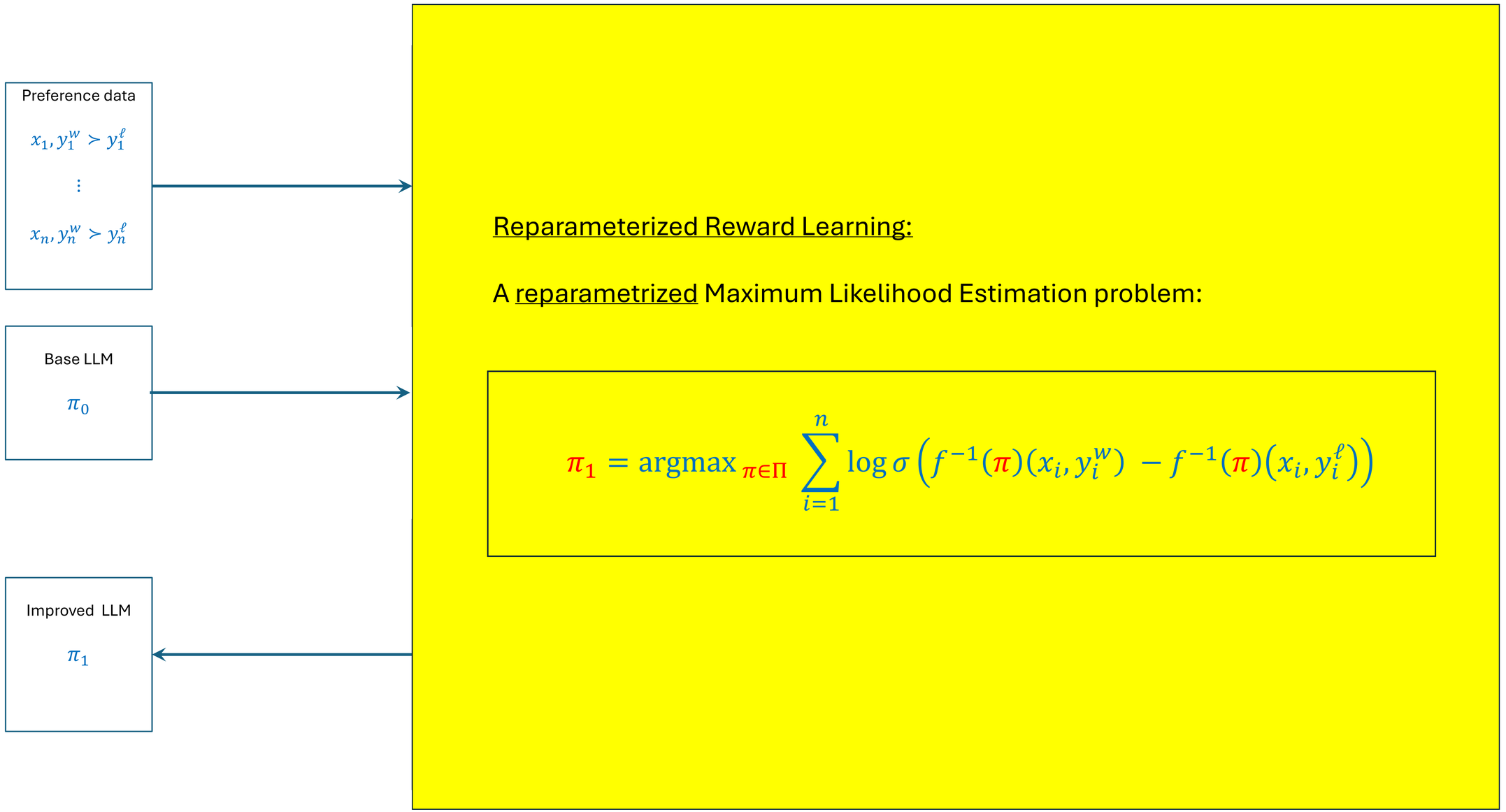
Reinforcement Learning with Human Feedback (RLHF)



Direct Policy Optimization (DPO) (Rafailov et al ‘23)



Direct Policy Optimization (DPO) (Rafailov et al ‘23)



Direct Policy Optimization (DPO) (Rafailov et al ‘23)

$$\underset{\substack{\theta\text{: LLM params}}}{\arg\max} \sum_i \log \sigma \left(\beta \log \frac{\pi_\theta(y_i^{\mathrm{w}} \mid x_i)}{\pi_{\theta_0}(y_i^{\mathrm{w}} \mid x_i)} - \beta \log \frac{\pi_\theta(y_i^{\ell} \mid x_i)}{\pi_{\theta_0}(y_i^{\ell} \mid x_i)} \right)$$

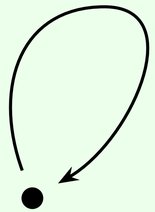
$r_\theta(x_i, y_i^{\mathrm{w}})$ $r_\theta(x_i, y_i^{\ell})$

Implicit reward function parameterized by θ (LLM)

DPO in the ideal case (tabular LLMs)

All reparameterized reward functions are denoted by green points.
When $\Pi = \{all\ possible\ policies\}$, all possible reward functions are covered by reparameterization

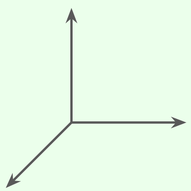
DPO projects r^* to closest reparameterized reward function $\Rightarrow$ to itself



$r^* = r(\pi_1^*)$

$r(\pi_1^a) \equiv f^{-1}(\pi_1^a)$

$r(\pi_1^b) \equiv f^{-1}(\pi_1^b)$

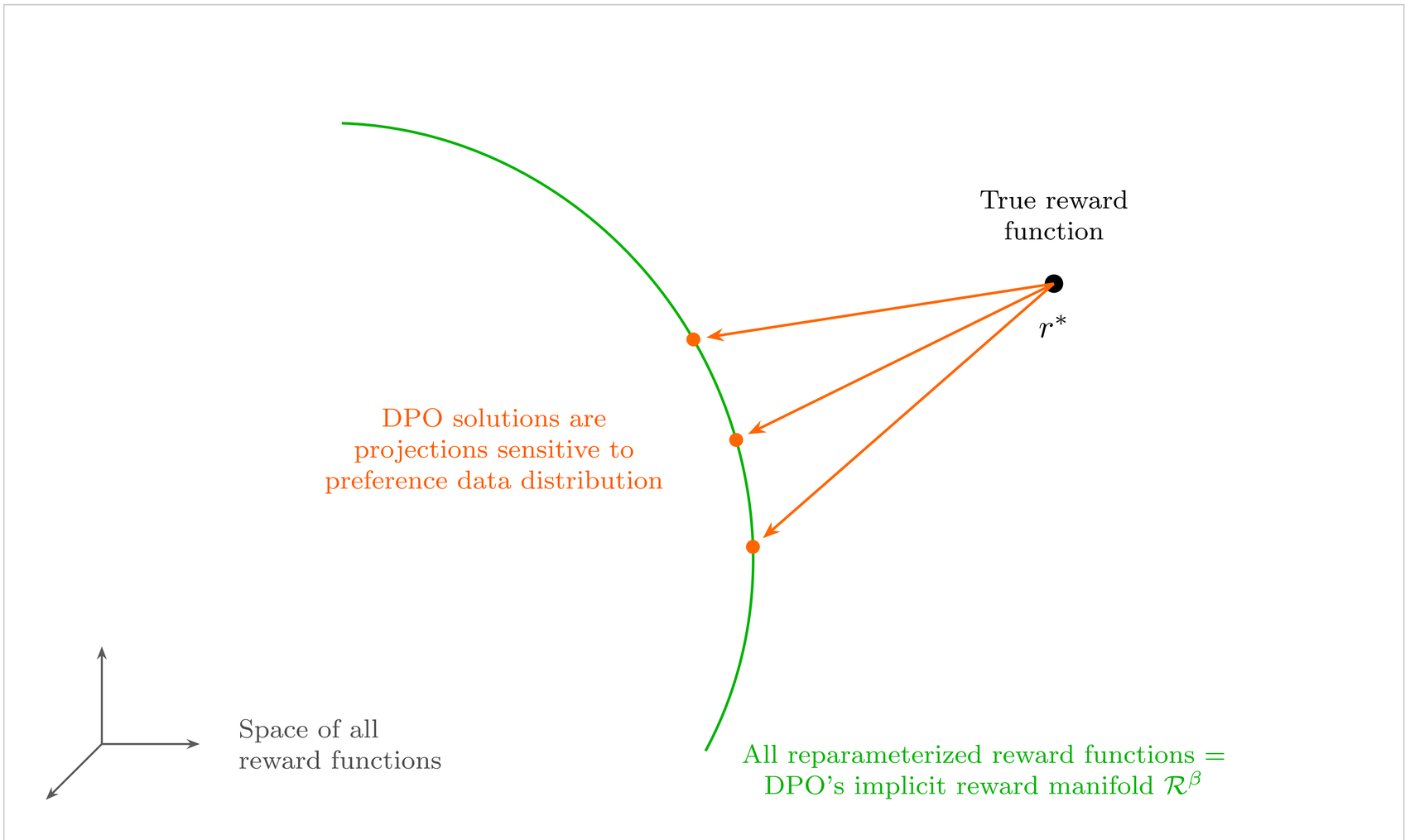


Space of all reward functions

DPO is a data-weighted reverse-KL projection of true reward function (r^*) to nearest reparameterized reward function (itself)

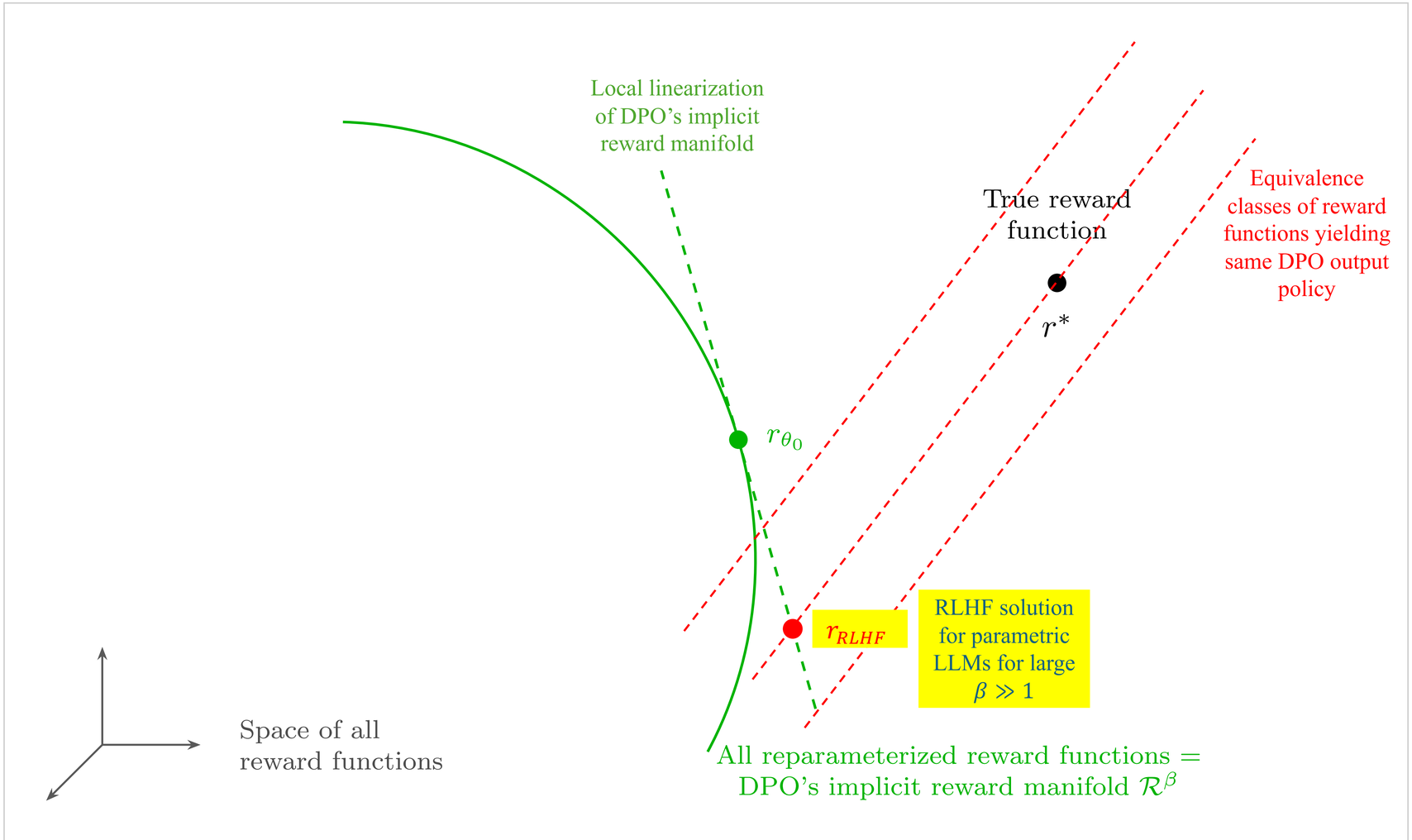
DPO in the practical case ≡ Misspecified MLE

- Practical LLMs are parametric ⇒ reparameterizable reward functions live on a manifold
- DPO projects r^* to manifold depending on preference data input
- The associated policy π_1 need not enjoy any guarantees
  - Worse policy than at initialization
  - Preference order reversal
  - Sensitivity to preference data distribution



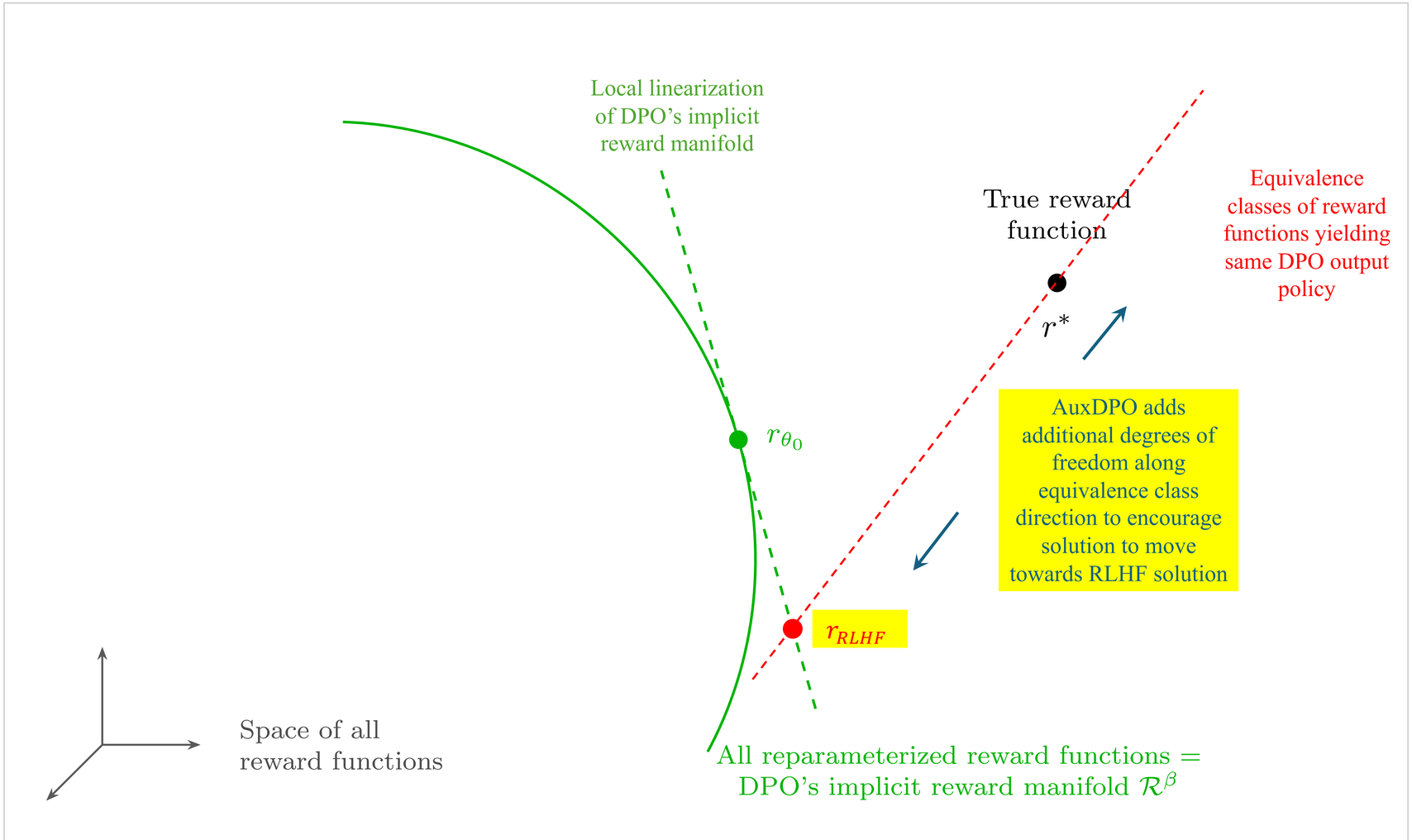
Local analysis of RLHF for practical (parametric) LLMs

- DPO with ‘small changes’ ($\beta \to \infty$) ‘stretches’ and linearizes the implicit reward manifold near r_{θ_0}
- RLHF $\equiv$ Natural policy gradient step on the tangent space
- Equivalence classes of reward functions w.r.t. DPO’s output



Algorithm AuxDPO

- Exploits equivalence class structure of RLHF solution to move DPO towards it
- Introduces auxiliary optimization variables constrained along the equivalence class direction



Numerics: Qwen-0.6B on RewardBench-v2

Percentage accuracy improvement over base policy

DPO	AuxDPO	IPO	DPOP
<u>55.10</u>	**65.31**	53.06	51.02

In-domain (In-distribution)

DPO	AuxDPO	IPO	DPOP
−8.16	**18.36**	−8.23	<u>−6.25</u>

Cross-domain transfer (Out-of-distribution)

Summary

1. DPO (Rafailov et al ‘23) is a revolutionary alignment algorithm paving the way for direct preference-based fine-tuning

    - Ideally equivalent to 2-stage RLHF (Reward Learning + Policy Opt.)
    - But equivalence breaks with (parametric) LLMs in general
    - Strange outcomes:
        - Worse policy than at initialization
        - Preference order reversal
        - Sensitivity to preference data distribution

2. A remedy via information geometry: AuxDPO algorithm

Thank you

Gopalan, Chowdhury, Banerjee, ICLR ‘26 (oral)