CS639 Deep Learning for NLP

Learning from Human Feedback II

Aligning LLMs with Human Intent and Values

Junjie Hu

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

https://junjiehu.github.io/cs639-spring26/

Outline

- Introduction to reinforcement learning with human feedback (RLHF)
- Policy Gradient Fundamentals: **REINFORCE**
- Policy Optimization: **PPO** (RLHF Standard)
- Model-Free Preference Optimization: **DPO**
- Algorithm Comparison and Future Directions

Proximal Policy Optimization (PPO)

PPO uses Actor-Critic (AC)

— A more stable variant of Policy Gradient

- In a classic **actor-critic** objective, the classic policy gradient theorem gives:

$$\nabla_\theta J(\theta) = \mathbb{E}_t \left[\nabla_\theta \log \pi_\theta(a_t \mid s_t) A_t \right]$$

- To perform gradient ascent, we maximize the surrogate loss:

$$J(\theta) = \mathbb{E}_t \left[\log \pi_\theta(a_t \mid s_t) A_t \right]$$

- where the **advantage** uses $A_t = G_t - V_\phi(s_t)$.
- A **critic (value function)** estimates the value function $V_\phi(s_t)$ that quantifies the expected reward (scalar) for a given state.
- A **actor (policy)** estimates the probability of an action given a state $\pi_\theta(a_t \mid s_t)$

PPO

- **Motivation**: REINFORCE is high-variance and unstable. Even with a baseline (like in Actor-Critic), when we take a large gradient step, the new policy can deviate too far from the old policy, causing performance forgetting.
- So, we want a **trust region**: “Don’t move the policy too far in one update.” This leads to a thread of methods called **Trust Region Policy Optimization (TRPO)**.
- **PPO** is a simpler, practical version of TRPO that **constrains updates to avoid drastic policy changes**.

Trust Region Idea

- TRPO also uses advantages in an actor-critic style.
- But to enforce stability, TRPO introduces
    1. A objective that measures the performance difference between new and old policies
    2. A constraint that limits the KL divergence between new and old polices

In RLHF for LLMs, the **new policy** is the **RL policy**, the **old policy** is the **SFT pre-trained** policy.

$$\max_{\theta} \quad \mathbb{E}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\text{sft}}(a_t \mid s_t)} A_t\right] \qquad \text{s.t.} \quad \mathbb{E}_t\left[D_{\text{KL}}(\pi_{\text{sft}} \,||\, \pi_\theta)\right] \le \delta$$

Challenge: But solving this constrained optimization exactly is complicated.

PPO Simplification

- PPO simplifies TRPO by replacing the hard KL constraint with a clipped surrogate objective.
- Define the probability ration:

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

- Then the PPO clipped objective is:

$$J^{\text{clip}}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta)A_t,\ \text{clip}(r_t(\theta),1-\epsilon,1+\epsilon)A_t\right)\right].$$

Intuition for Clipping

- If $r_t(\theta)$ stays close to 1 (small change in policy), the update is normal.
- If $r_t(\theta)$ tries to move too far away (e.g., $r_t(\theta) > 1 + \epsilon$ or $r_t(\theta) < 1 - \epsilon$), the term is clipped — limiting the objective to prevent destructive updates.

$$J^{\text{clip}}(\theta) = \mathbb{E}_t\left[J_t^{\text{clip}}(\theta)\right]$$

$$= \mathbb{E}_t\left[\min\left(r_t(\theta)A_t,\ \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t\right)\right].$$

The min operator ensures that the policy does not move further in a direction that *reduces* the objective beyond the clipped region.

This acts like a **soft trust region**.

Full PPO Objective

(with Value function + Entropy)

- In practice, PPO uses a combined objective that includes:
  1. **Policy loss** (clipped surrogate)
  2. **Value function loss** (squared error)
  3. **Entropy bonus** (for exploration)

$$J^{\text{ppo}}(\theta) = \mathbb{E}_t\Big[J_t^{\text{clip}}(\theta) - c_1(V_\theta(s_t) - V_t^*)^2 + c_2 S[\pi_\theta](s_t)\Big],$$

1.Policy loss **2. Value function loss** **3. Entropy bonus**

where c_1, c_2 are multi-objective coefficients, $S[\pi_\theta](s_t)$ is the entropy of the policy at state s_t.
In most implementation, the value function V_θ and policy model π_θ may share some parameters θ. For example, take the embedding from the last-hidden layer of a LLM policy π_θ, and add a linear layer to estimate the state-value $V_\theta(s_t)$

Summary: PPO Training

for $k = 1,\ldots, K$:

Sample trajectories using current policy π_θ

Compute advantages $A_t = G_t - V_\theta(s_t)$

Update θ by maximizing $J_{\text{ppo}}(\theta)$

Why PPO Works

- The clipping ensures **stability** — avoids overly large steps.
- It's **simple** (no second-order approximations or constrained optimization).
- It retains **sample efficiency** of on-policy methods.
- The **advantage baseline** keeps the gradient unbiased, while reducing variance.

Direct Preference Optimization (DPO)

Motivation and Intuition

- DPO removes the need to train a separate reward model and then run RL. Instead, it directly learns from human preference comparisons (pairs of chosen vs rejected) in a supervised-style optimization.
- The idea: if human preference data says for prompt x, response y_w is preferred over y_l, then adjust the policy so that $\pi(y_w|x)$ is favored over $\pi(y_l|x)$ in a way consistent with a hidden "reward" interpretation.

DPO Loss

- For pairwise preference data y_w (winner, loser), DPO defines its objective:

$$\max_{\phi} \mathbb{E}_{(x,y_w,y_l)} \left[\log \sigma \left(\beta \log \frac{\pi_\phi(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\phi(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right) \right]$$

- where β is a temperature (alignment strength) and π_{ref} is a reference policy (SFT model).
- Intuitively, it encourages the policy to increase the log odds of preferred vs non-preferred response, relative to the reference.
- The reference model ensures that the update is modest and regularized.
- One can derive that the optimal policy in DPO corresponds to:

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

- where $r^*(x, y)$ is the (implicit) reward function.
- **Key takeaway**: no explicit reward model, no RL loop, no value critic — it's just a pairwise preference-based update.

Theoretical Connection & Limitations

- Advantages:
  - Offline RL algorithms on collected pairwise preferences
  - Simpler implementation (no reward model training, no RL iterations)
  - More stable (fewer moving parts)
  - Empirically competitive with PPO-based RLHF in many tasks
- Limitations
  - Assumes pairwise preference data (cannot trivially incorporate more general feedback)
  - Might be less flexible in incorporating more complex reward structures

Comparison

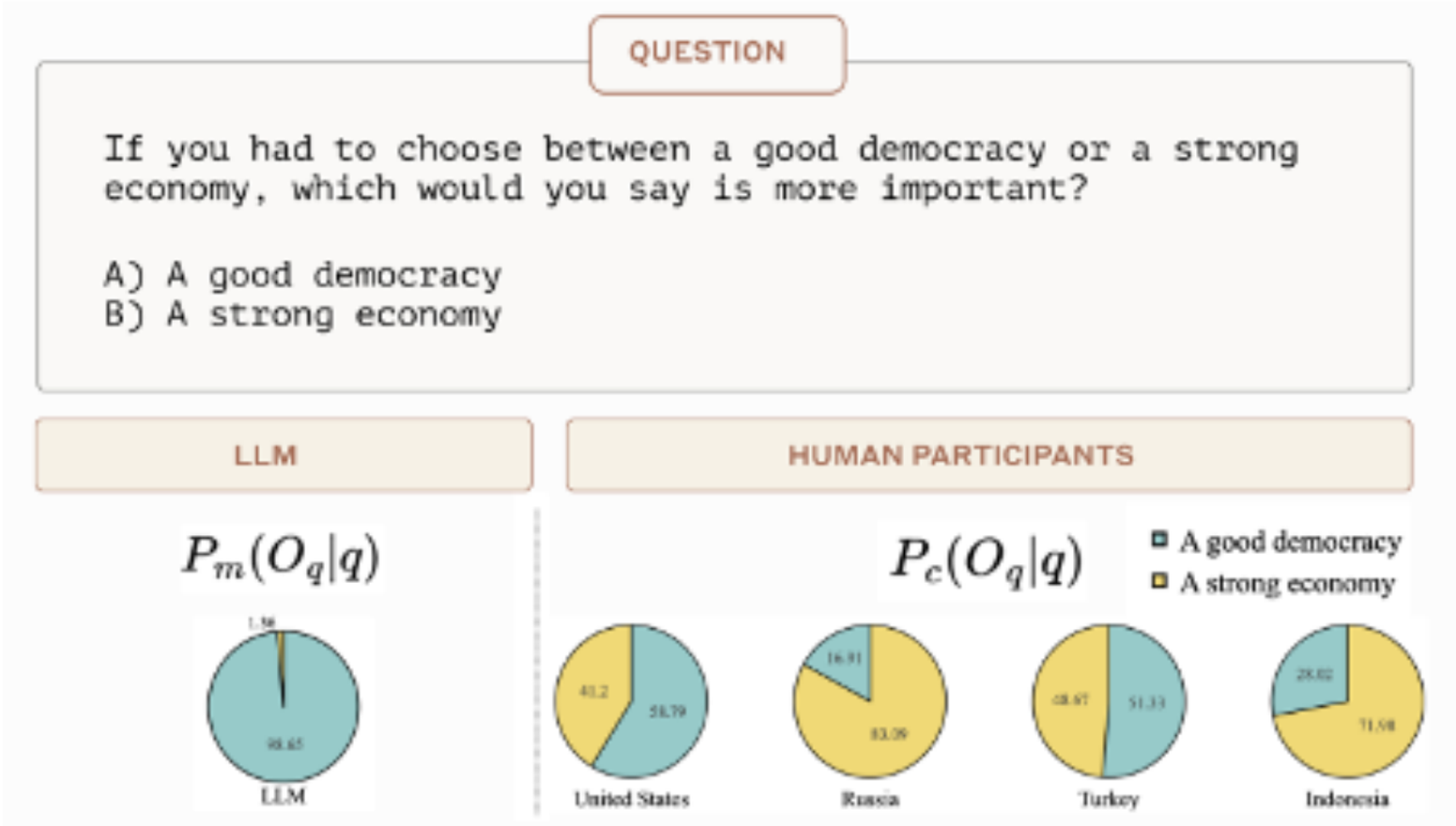
Method	Requires Reward Model?	Critic / Value Network?	Stability & Variance	Simplicity / Implementation	Use of preference pairs vs full reward	Strong points	Weak points
REINFORCE (vanilla)	Usually yes (score responses)	No	High variance, unstable	Simple	Uses sampled reward	Conceptually simple	Hard to scale, unstable
PPO-based RLHF	Yes	Yes	More stable, robust	Moderate complexity	Uses reward model + RL loop	Widely used, good empirical stability	More complexity, hyperparameter tuning
GRPO	No explicit reward model (direct preference)	Usually none	Robust across groups	More complex weighting logic	Preference-labeled data with group weights	Fairness, robustness across groups	Requires group structure, weight tuning
DPO	No	No	More stable (less moving parts)	Simple, supervised-style	Pairwise preferences	Simplicity, speed	Less flexibility, may not capture richer reward signals

Future Directions

- Better reward modeling / generalization
- Combining direct preference methods with RL-based ones
- Scalability and efficiency
- Fairness, robustness, group-aware alignment
- Multi-objective alignment & safety constraints
- Open benchmarks & evaluation
- Theory & convergence analysis

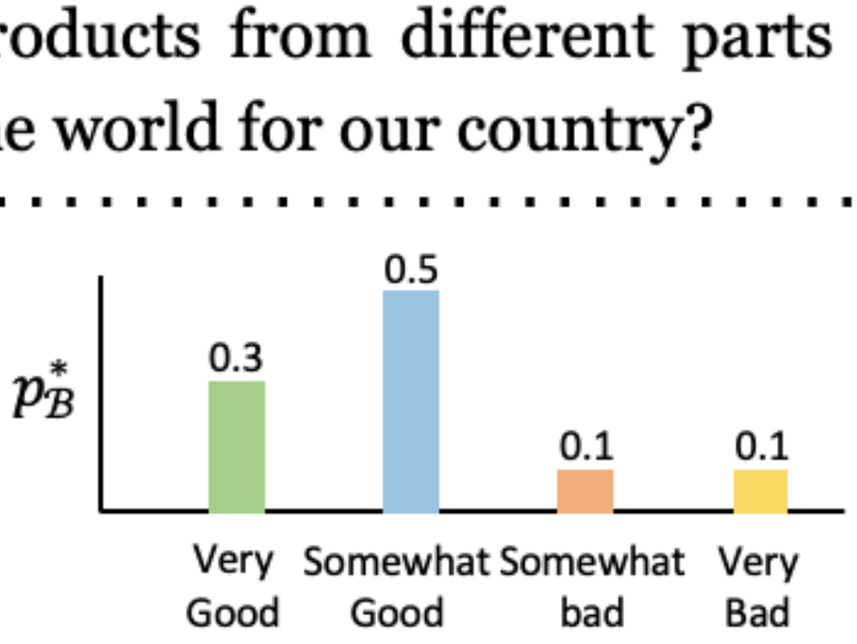
Pluralistic Preference Learning

- Human preferences are diverse and distributional
- But LLM Skews Towards Majority Opinions (Durmus, , et al., 2024)



Instance-level Preference Learning

x: In your opinion, what is the significance of the availability of products from different parts of the world for our country?



Distribution of Opinions

$b_c \sim p^*_{\mathcal{B}}$ | $b_r \sim \mathcal{B} \setminus \{b_c\}$

Somewhat Good | *Somewhat Bad*

y_c > y_r

I believe that, in many circumstances, the access to a wide variety of products from different parts of the world is a sign of growth and evolution.

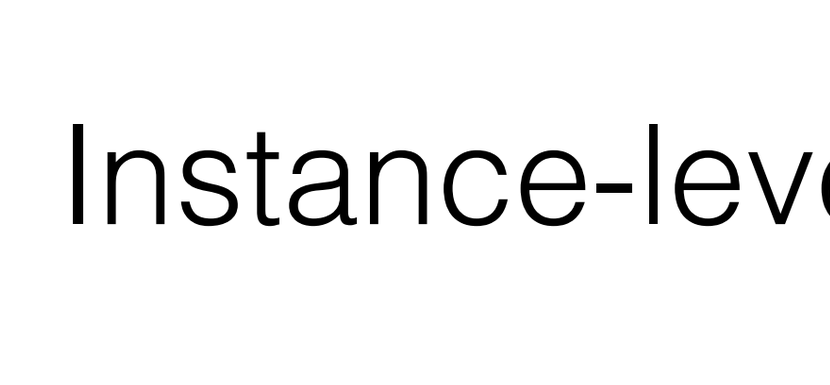
It definitely has its pros and cons. It's nice to have access to all these products, but we need to make sure we're supporting our local economy as well.

People's preference, shaped by their values, also follows a **distribution**.

Yao, Cai, Chuang, Yang, Jiang, Yang, Hu. GDPO. ICLR 2025.

Instance-level Preference Learning

x: In your opinion, what is the significance of the availability of products from different parts of the world for our country?



Distribution of Opinions

$b_c \sim p^*_B$

Somewhat Good

y_c

I believe that, in many circumstances, the access to a wide variety of products from different parts of the world is a sign of growth and evolution.

>

$b_r \sim \mathcal{B} \backslash \{b_c\}$

Somewhat Bad

y_r

It definitely has its pros and cons. It's nice to have access to all these products, but we need to make sure we're supporting our local economy as well.

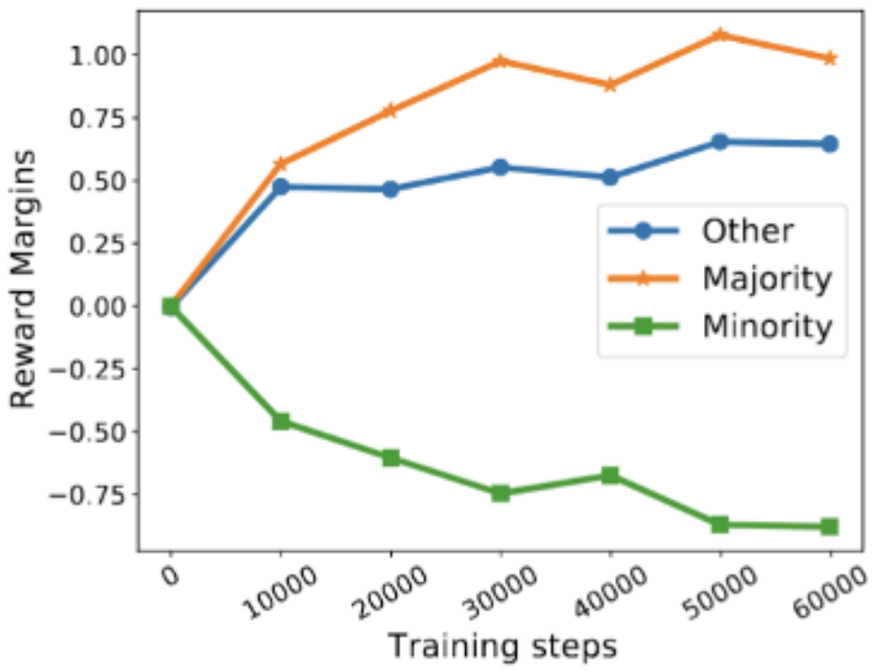
What if $y_r > y_c$ co-exists in the data?

Yao, Cai, Chuang, Yang, Jiang, Yang, Hu. GDPO. ICLR 2025.

Why not DPO?

Reward Margins = Chosen Rewards - Reject Rewards

DPO Skews Towards Majority Preferences !!!



(a) GPT-2 Large

(b) Pythia-2.8B

No Preference Left Behind:

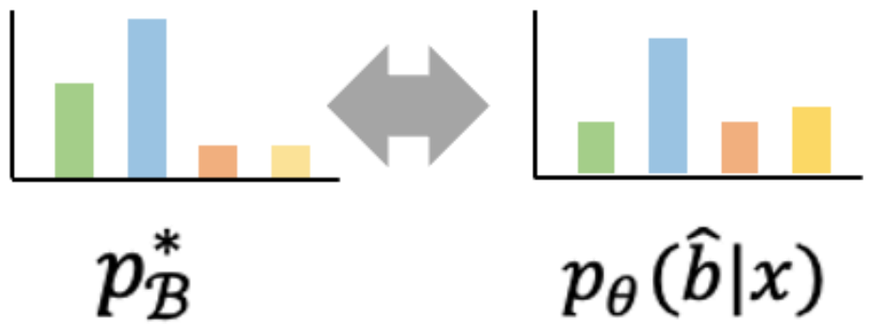
Group Distributional Preference Optimization

- from Instance-level to Distributional-level Alignment

Group Distributional Preference Optimization (GDPO)

$$\ell_{\text{GDPO}} = \ell_{\text{cal.}}(p_\theta(\hat{b}|x), p_B^*) + \ell_{\text{pref.}}(y_c > y_r, b_c, x)$$

Belief Calibration



Belief-Conditioned Preference Alignment

$$p_\theta(y_i|x, b_i) > p_\theta(y_j|x, b_i)$$

Belief is the degree to which individuals agree with a particular stance.
p_B^* the label distribution of beliefs, which are statistics from the dataset.

Yao, Cai, Chuang, Yang, Jiang, Yang, Hu. GDPO. ICLR 2025.

Questions?