What Does Scalar RLHF Optimize?

Aaron Roth

University of Pennsylvania

Fall 2026

Overview

- Reward modeling forces pairwise human comparisons to be compressed into a scalar reward.

Overview

- Reward modeling forces pairwise human comparisons to be compressed into a scalar reward.
- Uniform logistic reward fitting implements the population Borda ordering.

Overview

- Reward modeling forces pairwise human comparisons to be compressed into a scalar reward.
- Uniform logistic reward fitting implements the population Borda ordering.
- KL-regularized policy optimization exponentially tilts a reference policy toward the fitted rewards.

Overview

- Reward modeling forces pairwise human comparisons to be compressed into a scalar reward.
- Uniform logistic reward fitting implements the population Borda ordering.
- KL-regularized policy optimization exponentially tilts a reference policy toward the fitted rewards.
- Changing how comparison labels are processed can change the social-choice rule.

RLHF Does Not Observe Harsanyi's Cardinal Utilities

The previous lecture began with preferences over lotteries satisfying the VNM axioms and derived cardinal preferences represented on different interpersonal scales:

$$u_1, \ldots, u_n.$$

Harsanyi then characterized weak-Pareto aggregates as nonnegative weighted sums of these utilities.

RLHF Does Not Observe Harsanyi's Cardinal Utilities

The previous lecture began with preferences over lotteries satisfying the VNM axioms and derived cardinal preferences represented on different interpersonal scales:

$$u_1, \ldots, u_n.$$

Harsanyi then characterized weak-Pareto aggregates as nonnegative weighted sums of these utilities.
Ordinary RLHF data look different. A label says only that one completion was preferred to another:

$$i \succ j.$$

RLHF Does Not Observe Harsanyi's Cardinal Utilities

The previous lecture began with preferences over lotteries satisfying the VNM axioms and derived cardinal preferences represented on different interpersonal scales:

$$u_1, \ldots, u_n.$$

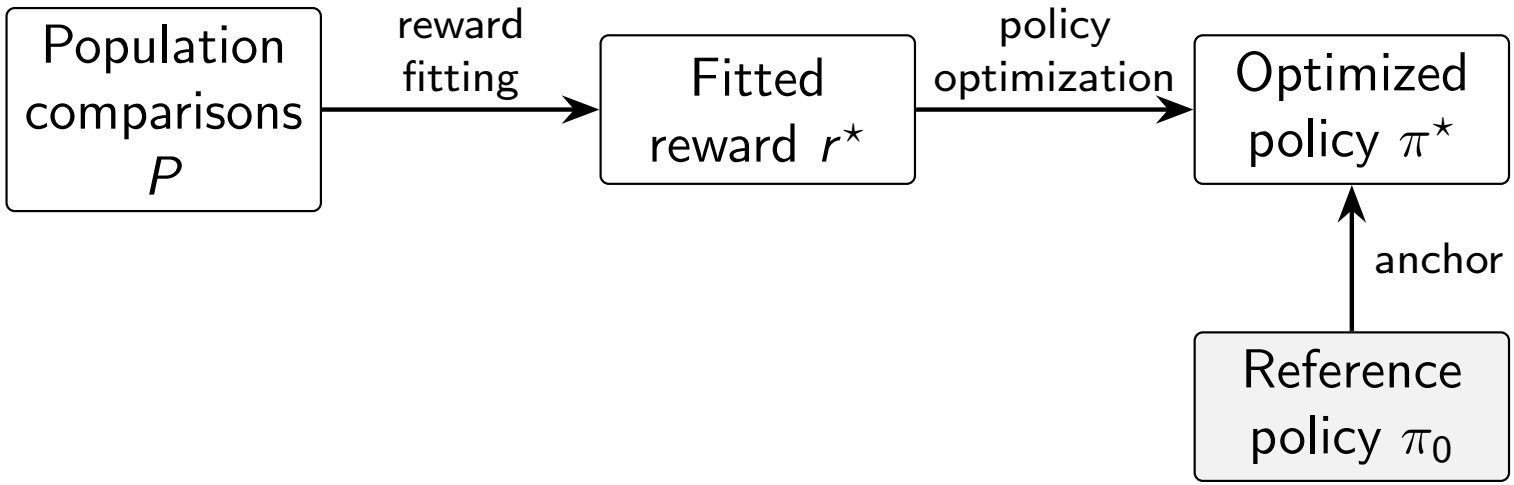
Harsanyi then characterized weak-Pareto aggregates as nonnegative weighted sums of these utilities.
Ordinary RLHF data look different. A label says only that one completion was preferred to another:

$$i \succ j.$$

RLHF nevertheless constructs and optimizes a scalar reward.
Which aggregate objective has it constructed?

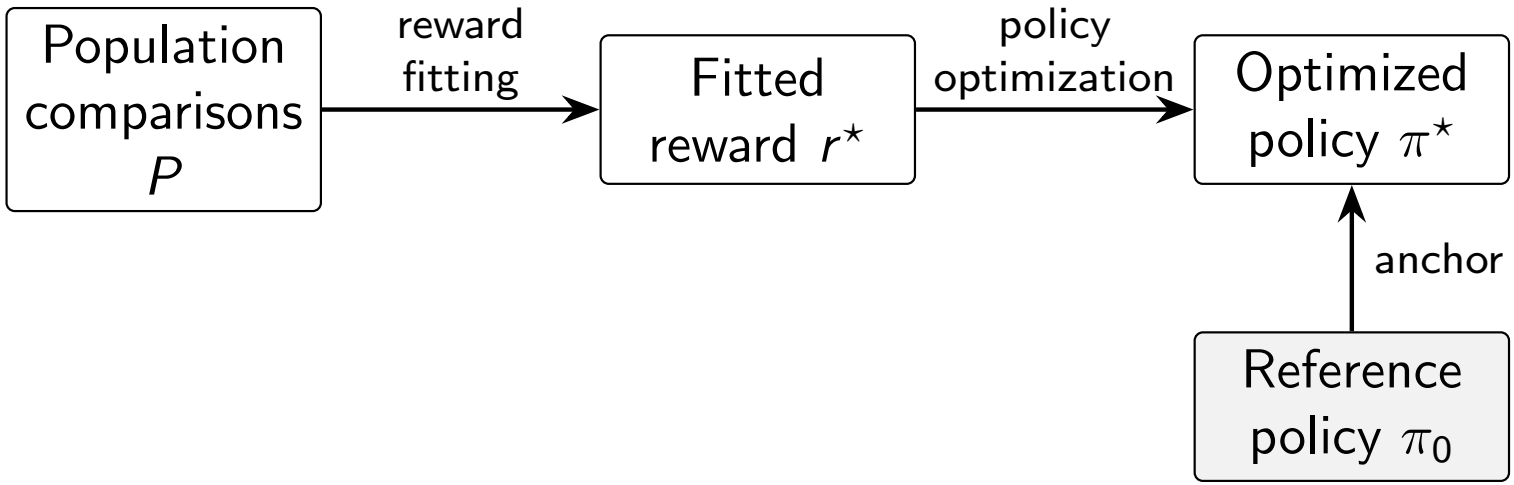
The Pipeline has two parts

Fix one prompt and a finite set C of candidate completions.



The Pipeline has two parts

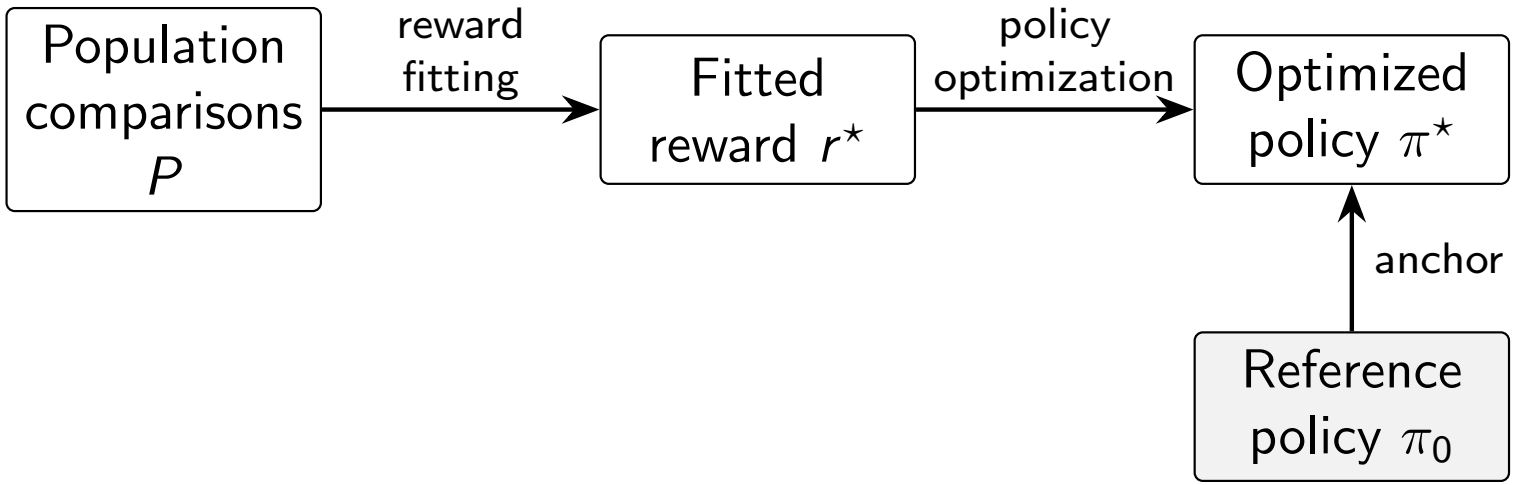
Fix one prompt and a finite set C of candidate completions.



The first map depends on the loss, pair-sampling design, label processing, and reward class.

The Pipeline has two parts

Fix one prompt and a finite set C of candidate completions.



The first map depends on the loss, pair-sampling design, label processing, and reward class.
The second map depends on the fitted reward, the reference policy π_0, and the regularization strength. In the idealized model analyzed today, π_0 enters only at this second stage: it is not part of the aggregation rule, but it shapes the resulting policy.

A Clean Population Model Isolates the Aggregation Rule

We make four simplifying assumptions:

1. There is one fixed prompt and a finite response set $C = \{1, \ldots, m\}$.

A Clean Population Model Isolates the Aggregation Rule

We make four simplifying assumptions:

1. There is one fixed prompt and a finite response set $C = \{1, \ldots, m\}$.
2. Every unordered pair is presented for labeling with equal probability.

A Clean Population Model Isolates the Aggregation Rule

We make four simplifying assumptions:

1. There is one fixed prompt and a finite response set $C = \{1, \ldots, m\}$.
2. Every unordered pair is presented for labeling with equal probability.
3. We work with the population comparison probabilities, not a finite sample.

A Clean Population Model Isolates the Aggregation Rule

We make four simplifying assumptions:

1. There is one fixed prompt and a finite response set $C = \{1, \ldots, m\}$.
2. Every unordered pair is presented for labeling with equal probability.
3. We work with the population comparison probabilities, not a finite sample.
4. The reward vector $r \in \mathbb{R}^m$ has one free score r_i for each response.

A Clean Population Model Isolates the Aggregation Rule

We make four simplifying assumptions:

1. There is one fixed prompt and a finite response set $C = \{1, \ldots, m\}$.
2. Every unordered pair is presented for labeling with equal probability.
3. We work with the population comparison probabilities, not a finite sample.
4. The reward vector $r \in \mathbb{R}^m$ has one free score r_i for each response.

These assumptions remove statistical and approximation questions so that we can see the social aggregation performed by the objective itself.

Population Comparisons Form a Probabilistic Tournament

For distinct responses $i, j \in C$, define

$$P_{ij} := \Pr(i \text{ is preferred to } j \mid \{i, j\} \text{ is presented}).$$

Because one response wins each comparison, $P_{ij} + P_{ji} = 1$.

Population Comparisons Form a Probabilistic Tournament

For distinct responses $i, j \in C$, define

$$P_{ij} := \Pr(i \text{ is preferred to } j \mid \{i, j\} \text{ is presented}).$$

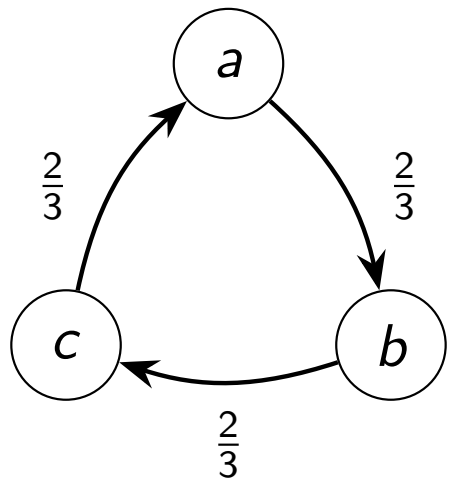
Because one response wins each comparison, $P_{ij} + P_{ji} = 1$. The randomness can come from heterogeneous labelers or from variation in repeated judgments.

Recall the Cyclic Majority Example

Three equally common preference types rank three responses as

$$a \succ b \succ c, \qquad b \succ c \succ a, \qquad c \succ a \succ b,$$

so that $P_{ab} = P_{bc} = P_{ca} = \frac{2}{3}$:

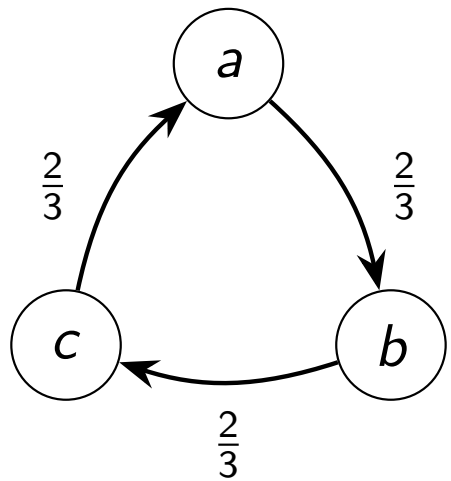


Recall the Cyclic Majority Example

Three equally common preference types rank three responses as

$$a \succ b \succ c, \qquad b \succ c \succ a, \qquad c \succ a \succ b,$$

so that $P_{ab} = P_{bc} = P_{ca} = \frac{2}{3}$:



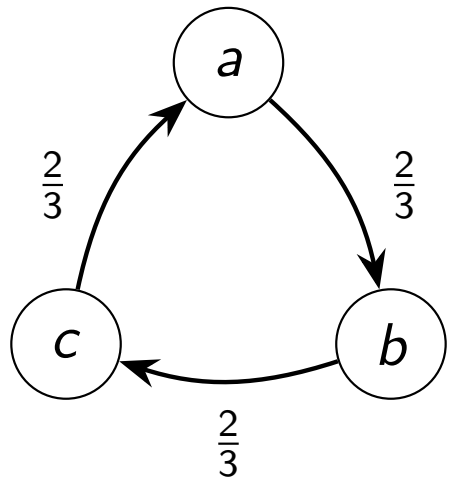
Thus a defeats b, b defeats c, and c defeats a by population majority.

Recall the Cyclic Majority Example

Three equally common preference types rank three responses as

$$a \succ b \succ c, \qquad b \succ c \succ a, \qquad c \succ a \succ b,$$

so that $P_{ab} = P_{bc} = P_{ca} = \frac{2}{3}$:



Thus a defeats b, b defeats c, and c defeats a by population majority.

No single scalar ordering can reproduce all three comparisons.

A Scalar Reward Predicts a Transitive Tournament

Let

$$\sigma(z) := \frac{1}{1 + e^{-z}}$$

denote the logistic (sigmoid) function. A Bradley–Terry reward vector $r \in \mathbb{R}^m$ predicts

$$Q^r_{ij} := \sigma(r_i - r_j).$$

A Scalar Reward Predicts a Transitive Tournament

Let

$$\sigma(z) := \frac{1}{1 + e^{-z}}$$

denote the logistic (sigmoid) function. A Bradley–Terry reward vector $r \in \mathbb{R}^m$ predicts

$$Q_{ij}^r := \sigma(r_i - r_j).$$

Because σ is strictly increasing,

$$Q_{ij}^r > \frac{1}{2} \quad \Longleftrightarrow \quad r_i > r_j.$$

A Scalar Reward Predicts a Transitive Tournament

Let

$$\sigma(z) := \frac{1}{1 + e^{-z}}$$

denote the logistic (sigmoid) function. A Bradley–Terry reward vector $r \in \mathbb{R}^m$ predicts

$$Q^r_{ij} := \sigma(r_i - r_j).$$

Because σ is strictly increasing,

$$Q^r_{ij} > \frac{1}{2} \quad \Longleftrightarrow \quad r_i > r_j.$$

The population matrix P can cycle, but the fitted matrix Q^r must inherit one transitive ordering from r.

The fitting loss decides how the cycle is compressed.

A Comparison Label Has a Bernoulli Likelihood

Fix two responses i and j. Let $Y = 1$ when i is preferred and $Y = 0$ when j is preferred, and write

$$p := \Pr(Y = 1), \qquad z := r_i - r_j.$$

Thus $Y \sim \text{Bernoulli}(p)$ in the population.

A Comparison Label Has a Bernoulli Likelihood

Fix two responses i and j. Let $Y = 1$ when i is preferred and $Y = 0$ when j is preferred, and write

$$p := \Pr(Y = 1), \qquad z := r_i - r_j.$$

Thus $Y \sim \text{Bernoulli}(p)$ in the population.
The Bradley–Terry model assigns

$$\Pr_z(Y = 1) = \sigma(z), \qquad \Pr_z(Y = 0) = \sigma(-z).$$

Hence the likelihood of an observed label $y \in \{0, 1\}$ is

$$\text{Lik}(z; y) := \Pr_z(Y = y) = \sigma(z)^y \sigma(-z)^{1-y}.$$

Expected Negative Log-Likelihood Is Logistic Loss

Taking logs gives

$$\log \mathrm{Lik}(z; y) = y \log \sigma(z) + (1 - y) \log \sigma(-z).$$

Expected Negative Log-Likelihood Is Logistic Loss

Taking logs gives

$$\log \mathrm{Lik}(z; y) = y \log \sigma(z) + (1 - y) \log \sigma(-z).$$

Taking the negative expectation under the population label $Y \sim \mathrm{Bernoulli}(p)$ gives

$$\begin{aligned} \ell(p, z) &:= -\mathbb{E}[\log \mathrm{Lik}(z; Y)] \\ &= -p \log \sigma(z) - (1 - p) \log \sigma(-z) \\ &= p \log(1 + e^{-z}) + (1 - p) \log(1 + e^{z}). \end{aligned}$$

Expected Negative Log-Likelihood Is Logistic Loss

Taking logs gives

$$\log \mathrm{Lik}(z; y) = y \log \sigma(z) + (1 - y) \log \sigma(-z).$$

Taking the negative expectation under the population label $Y \sim \mathrm{Bernoulli}(p)$ gives

$$\begin{aligned} \ell(p, z) &:= -\mathbb{E}[\log \mathrm{Lik}(z; Y)] \\ &= -p \log \sigma(z) - (1 - p) \log \sigma(-z) \\ &= p \log(1 + e^{-z}) + (1 - p) \log(1 + e^{z}). \end{aligned}$$

Since

$$\log(1 + e^{-z}) = \log(1 + e^{z}) - z,$$

we obtain the useful form

$$\ell(p, z) = \log(1 + e^{z}) - pz.$$

Uniform Pair Sampling Gives One Population Objective

When every unordered pair receives equal weight, reward fitting minimizes

$$L(r) := \sum_{1 \le i < j \le m} \ell(P_{ij}, r_i - r_j).$$

Uniform Pair Sampling Gives One Population Objective

When every unordered pair receives equal weight, reward fitting minimizes

$$L(r) := \sum_{1 \le i < j \le m} \ell(P_{ij}, r_i - r_j).$$

Only reward differences matter:

$$L(r + c\mathbf{1}) = L(r).$$

Uniform Pair Sampling Gives One Population Objective

When every unordered pair receives equal weight, reward fitting minimizes

$$L(r) := \sum_{1 \le i < j \le m} \ell(P_{ij}, r_i - r_j).$$

Only reward differences matter:

$$L(r + c\mathbf{1}) = L(r).$$

Thus the fitted scores are identified only up to an additive constant. This does not affect their ordering or any predicted comparison probability.

Borda Measures Average Pairwise Success

Definition

The *population Borda total* of response i is

$$S_i := \sum_{j \neq i} P_{ij}.$$

Borda Measures Average Pairwise Success

Definition

The *population Borda total* of response i is

$$S_i := \sum_{j \neq i} P_{ij}.$$

Its normalized Borda score is

$$B_i := \frac{S_i}{m-1}.$$

Borda Measures Average Pairwise Success

Definition

The *population Borda total* of response i is

$$S_i := \sum_{j \neq i} P_{ij}.$$

Its normalized Borda score is

$$B_i := \frac{S_i}{m-1}.$$

Thus B_i is the probability that i defeats an opponent drawn uniformly from the other responses. The totals S_i and scores B_i induce the same ordering.

Uniform Logistic Fitting Implements Borda

Theorem

Let $r^\star$ be any minimizer of the uniformly weighted logistic loss L. Then, for every i, k,

$$r_i^\star > r_k^\star \quad \Longleftrightarrow \quad S_i > S_k.$$

The same equivalence holds with $>$ replaced by $=$.

Proof Roadmap: Optimality Becomes Moment Matching

1. Differentiate the logistic loss: $\partial_z \ell(p, z) = \sigma(z) - p$, the model win probability minus the population win probability.

Proof Roadmap: Optimality Becomes Moment Matching

1. Differentiate the logistic loss: $\partial_z \ell(p, z) = \sigma(z) - p$, the model win probability minus the population win probability.
2. Use the first-order conditions to match each response's population and predicted average win rate.

Proof Roadmap: Optimality Becomes Moment Matching

1. Differentiate the logistic loss: $\partial_z \ell(p, z) = \sigma(z) - p$, the model win probability minus the population win probability.
2. Use the first-order conditions to match each response's population and predicted average win rate.
3. Use monotonicity of σ: a higher fitted score gives a higher predicted win probability against every fixed comparison score, and hence a larger average win rate.

The Derivative Measures the Probability Gap

Recall that $\ell(p, z) = \log(1 + e^z) - pz$. Treating p as fixed,

$$\begin{aligned}
\frac{\partial}{\partial z}\ell(p, z) &= \frac{\partial}{\partial z}\left[\log(1 + e^z) - pz\right] \\
&= \frac{e^z}{1 + e^z} - p \\
&= \frac{1}{1 + e^{-z}} - p \\
&= \sigma(z) - p.
\end{aligned}$$

The Derivative Measures the Probability Gap

Recall that $\ell(p, z) = \log(1 + e^z) - pz$. Treating p as fixed,

$$\begin{aligned}\frac{\partial}{\partial z}\ell(p, z) &= \frac{\partial}{\partial z}\left[\log(1 + e^z) - pz\right] \\ &= \frac{e^z}{1 + e^z} - p \\ &= \frac{1}{1 + e^{-z}} - p \\ &= \sigma(z) - p.\end{aligned}$$

Here $\sigma(z)$ is the model probability of $Y = 1$, while p is its population probability.

Each Pair Contributes a Probability Gap

For a pair written as $i < j$, its loss is $\ell(P_{ij}, r_i - r_j)$. The chain rule first separates the derivative into two factors:

$$\frac{\partial}{\partial r_i}\ell(P_{ij}, r_i - r_j) = \left.\frac{\partial \ell(p, z)}{\partial z}\right|_{\substack{p=P_{ij} \\ z=r_i-r_j}} \frac{\partial(r_i - r_j)}{\partial r_i}.$$

Each Pair Contributes a Probability Gap

For a pair written as $i < j$, its loss is $\ell(P_{ij}, r_i - r_j)$. The chain rule first separates the derivative into two factors:

$$\frac{\partial}{\partial r_i}\ell(P_{ij}, r_i - r_j) = \left.\frac{\partial \ell(p, z)}{\partial z}\right|_{\substack{p=P_{ij} \\ z=r_i-r_j}} \frac{\partial(r_i - r_j)}{\partial r_i}.$$

The preceding calculation evaluates the first factor; the second is immediate:

$$\left.\frac{\partial \ell(p, z)}{\partial z}\right|_{\substack{p=P_{ij} \\ z=r_i-r_j}} = \sigma(r_i - r_j) - P_{ij}, \qquad \frac{\partial(r_i - r_j)}{\partial r_i} = 1.$$

Each Pair Contributes a Probability Gap

For a pair written as $i < j$, its loss is $\ell(P_{ij}, r_i - r_j)$. The chain rule first separates the derivative into two factors:

$$\frac{\partial}{\partial r_i}\ell(P_{ij}, r_i - r_j) = \left.\frac{\partial \ell(p, z)}{\partial z}\right|_{\substack{p=P_{ij} \\ z=r_i-r_j}} \frac{\partial(r_i - r_j)}{\partial r_i}.$$

The preceding calculation evaluates the first factor; the second is immediate:

$$\left.\frac{\partial \ell(p, z)}{\partial z}\right|_{\substack{p=P_{ij} \\ z=r_i-r_j}} = \sigma(r_i - r_j) - P_{ij}, \qquad \frac{\partial(r_i - r_j)}{\partial r_i} = 1.$$

Multiplying the two factors therefore gives

$$\frac{\partial}{\partial r_i}\ell(P_{ij}, r_i - r_j) = \sigma(r_i - r_j) - P_{ij}.$$

The Pair Formula Does Not Depend on Its Written Order

If $j < i$, the pair appears as $\ell(P_{ji}, r_j - r_i)$. Its contribution is

$$\frac{\partial}{\partial r_i}\ell(P_{ji}, r_j - r_i) = \big(\sigma(r_j - r_i) - P_{ji}\big)(-1).$$

The Pair Formula Does Not Depend on Its Written Order

If $j < i$, the pair appears as $\ell(P_{ji}, r_j - r_i)$. Its contribution is

$$\frac{\partial}{\partial r_i}\ell(P_{ji}, r_j - r_i) = \big(\sigma(r_j - r_i) - P_{ji}\big)(-1).$$

Now use the two symmetry identities

$$\sigma(r_j - r_i) = 1 - \sigma(r_i - r_j), \qquad P_{ji} = 1 - P_{ij}.$$

The Pair Formula Does Not Depend on Its Written Order

If $j < i$, the pair appears as $\ell(P_{ji}, r_j - r_i)$. Its contribution is

$$\frac{\partial}{\partial r_i}\ell(P_{ji}, r_j - r_i) = \big(\sigma(r_j - r_i) - P_{ji}\big)(-1).$$

Now use the two symmetry identities

$$\sigma(r_j - r_i) = 1 - \sigma(r_i - r_j), \qquad P_{ji} = 1 - P_{ij}.$$

Substituting and simplifying gives

$$\frac{\partial}{\partial r_i}\ell(P_{ji}, r_j - r_i) = -\big((1 - \sigma(r_i - r_j)) - (1 - P_{ij})\big) = \sigma(r_i - r_j) - P_{ij}.$$

The Pair Formula Does Not Depend on Its Written Order

If $j < i$, the pair appears as $\ell(P_{ji}, r_j - r_i)$. Its contribution is

$$\frac{\partial}{\partial r_i}\ell(P_{ji}, r_j - r_i) = \big(\sigma(r_j - r_i) - P_{ji}\big)(-1).$$

Now use the two symmetry identities

$$\sigma(r_j - r_i) = 1 - \sigma(r_i - r_j), \qquad P_{ji} = 1 - P_{ij}.$$

Substituting and simplifying gives

$$\frac{\partial}{\partial r_i}\ell(P_{ji}, r_j - r_i) = -\big((1 - \sigma(r_i - r_j)) - (1 - P_{ij})\big) = \sigma(r_i - r_j) - P_{ij}.$$

Thus, for a fixed coordinate r_i, every opponent contributes the same formula regardless of how the unordered pair was written:

$$\frac{\partial L}{\partial r_i}(r) = \sum_{j \neq i}\big(\sigma(r_i - r_j) - P_{ij}\big).$$

Optimal Fitting Matches Total Wins

Because $r^\star$ minimizes the differentiable function L, its gradient vanishes. The ith first-order condition is therefore

$$\begin{aligned} 0 &= \frac{\partial L}{\partial r_i}(r^\star) \\ &= \sum_{j \neq i} \left(Q_{ij}^{r^\star} - P_{ij} \right). \end{aligned}$$

Rearranging gives

$$\sum_{j \neq i} Q_{ij}^{r^\star} = \sum_{j \neq i} P_{ij} = S_i.$$

Optimal Fitting Matches Total Wins

Because $r^\star$ minimizes the differentiable function L, its gradient vanishes. The ith first-order condition is therefore

$$\begin{aligned} 0 &= \frac{\partial L}{\partial r_i}(r^\star) \\ &= \sum_{j \neq i} \left(Q_{ij}^{r^\star} - P_{ij} \right). \end{aligned}$$

Rearranging gives

$$\sum_{j \neq i} Q_{ij}^{r^\star} = \sum_{j \neq i} P_{ij} = S_i.$$

To compare two rows termwise, give the diagonal entries their natural values $P_{ii} = Q_{ii}^{r} = \frac{1}{2}$ and add the same term to both sides:

$$\sum_{j \in C} Q_{ij}^{r^\star} = \sum_{j \in C} P_{ij} = S_i + \frac{1}{2}.$$

Bradley–Terry Row Totals Reveal the Scores

Suppose $r_i > r_k$. For every fixed opponent j,

$$Q^r_{ij} = \sigma(r_i - r_j) > \sigma(r_k - r_j) = Q^r_{kj}.$$

Bradley–Terry Row Totals Reveal the Scores

Suppose $r_i > r_k$. For every fixed opponent j,

$$Q^r_{ij} = \sigma(r_i - r_j) > \sigma(r_k - r_j) = Q^r_{kj}.$$

Summing over j gives

$$\sum_j Q^r_{ij} > \sum_j Q^r_{kj}.$$

Bradley–Terry Row Totals Reveal the Scores

Suppose $r_i > r_k$. For every fixed opponent j,

$$Q^r_{ij} = \sigma(r_i - r_j) > \sigma(r_k - r_j) = Q^r_{kj}.$$

Summing over j gives

$$\sum_j Q^r_{ij} > \sum_j Q^r_{kj}.$$

Equal scores give equal row totals, while reversing the score inequality reverses every termwise comparison. Therefore

$$r_i > r_k \quad \Longleftrightarrow \quad \sum_j Q^r_{ij} > \sum_j Q^r_{kj}.$$

Moment Matching Proves the Borda Ordering

Combine the preceding two conclusions:

$$
\begin{aligned}
r_i^\star > r_k^\star \quad &\Longleftrightarrow \quad \sum_j Q_{ij}^{r^\star} > \sum_j Q_{kj}^{r^\star} \\
&\Longleftrightarrow \quad S_i + \frac{1}{2} > S_k + \frac{1}{2} \\
&\Longleftrightarrow \quad S_i > S_k .
\end{aligned}
$$

Moment Matching Proves the Borda Ordering

Combine the preceding two conclusions:

$$\begin{aligned} r_i^\star > r_k^\star \quad &\Longleftrightarrow \quad \sum_j Q_{ij}^{r^\star} > \sum_j Q_{kj}^{r^\star} \\ &\Longleftrightarrow \quad S_i + \frac{1}{2} > S_k + \frac{1}{2} \\ &\Longleftrightarrow \quad S_i > S_k. \end{aligned}$$

The same chain with equality shows that $r_i^\star = r_k^\star$ exactly when $S_i = S_k$.

Moment Matching Proves the Borda Ordering

Combine the preceding two conclusions:

$$\begin{aligned} r_i^\star > r_k^\star \quad &\Longleftrightarrow \quad \sum_j Q_{ij}^{r^\star} > \sum_j Q_{kj}^{r^\star} \\ &\Longleftrightarrow \quad S_i + \frac{1}{2} > S_k + \frac{1}{2} \\ &\Longleftrightarrow \quad S_i > S_k. \end{aligned}$$

The same chain with equality shows that $r_i^\star = r_k^\star$ exactly when $S_i = S_k$.

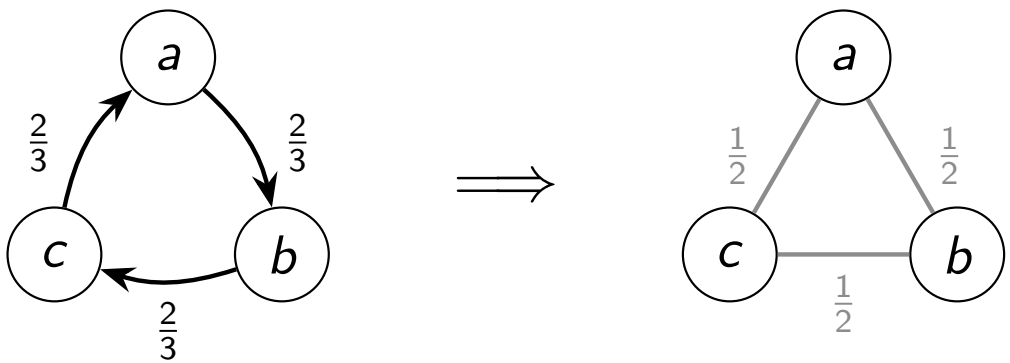
Uniform logistic reward fitting is the Borda rule in disguise.

The Running Cycle Is Compressed to a Tie

For the equal mixture of the three cyclic rankings,

$$S_a = P_{ab} + P_{ac} = \frac{2}{3} + \frac{1}{3} = 1,$$

and cyclically $S_a = S_b = S_c = 1$, so $r_a^\star = r_b^\star = r_c^\star$ and the fitted model predicts $Q_{ij}^{r^\star} = \frac{1}{2}$ on every pair.



The Theorem Gives an Ordering, Not More

The Borda theorem identifies its ordering, but generally

$$r_i^\star - r_k^\star \neq S_i - S_k.$$

The numerical gaps are determined jointly by all pairwise comparisons.

The Theorem Gives an Ordering, Not More

The Borda theorem identifies its ordering, but generally

$$r_i^\star - r_k^\star \neq S_i - S_k.$$

The numerical gaps are determined jointly by all pairwise comparisons.
Nor has the fit recovered Harsanyi welfare or a representative person's utility. Pairwise comparison data did not contain interpersonal utility scales in the first place.

The Second Stage Balances Reward and Policy Change

Reward fitting produces a score vector r. The next RLHF stage chooses a policy $\pi \in \Delta(C)$, meaning a distribution over responses.

The Second Stage Balances Reward and Policy Change

Reward fitting produces a score vector r. The next RLHF stage chooses a policy $\pi \in \Delta(C)$, meaning a distribution over responses. If this stage only maximized expected fitted reward, it would solve

$$\max_{\pi \in \Delta(C)} \sum_i \pi(i) r_i.$$

Every optimizer would put all its probability on responses with maximal fitted reward.

The Second Stage Balances Reward and Policy Change

Reward fitting produces a score vector r. The next RLHF stage chooses a policy $\pi \in \Delta(C)$, meaning a distribution over responses. If this stage only maximized expected fitted reward, it would solve

$$\max_{\pi \in \Delta(C)} \sum_i \pi(i) r_i.$$

Every optimizer would put all its probability on responses with maximal fitted reward.
RLHF instead begins from a reference policy π_0, representing the model before reward optimization, and trades reward against changing that policy. We therefore need a numerical measure of departure from π_0.

KL Penalizes Departure from the Reference Policy

For a policy $\pi \in \Delta(C)$ and a full-support reference policy $\pi_0 \in \Delta(C)$, define

$$D_{\mathrm{KL}}(\pi \| \pi_0) := \sum_i \pi(i) \log \frac{\pi(i)}{\pi_0(i)}.$$

Terms with $\pi(i) = 0$ are defined to be zero. The full-support assumption on π_0 keeps every denominator positive.

KL Penalizes Departure from the Reference Policy

For a policy $\pi \in \Delta(C)$ and a full-support reference policy $\pi_0 \in \Delta(C)$, define

$$D_{\mathrm{KL}}(\pi \| \pi_0) := \sum_i \pi(i) \log \frac{\pi(i)}{\pi_0(i)}.$$

Terms with $\pi(i) = 0$ are defined to be zero. The full-support assumption on π_0 keeps every denominator positive.
KL divergence is nonnegative and equals zero only when $\pi = \pi_0$.

Policy Optimization Trades Reward Against Policy Change

Let $\pi_0 \in \Delta(C)$ be a full-support reference policy, and let $\tau > 0$. The policy-optimization stage solves

$$\max_{\pi \in \Delta(C)} \left\{ \sum_i \pi(i) r_i - \tau D_{\mathrm{KL}}(\pi \| \pi_0) \right\}.$$

Policy Optimization Trades Reward Against Policy Change

Let $\pi_0 \in \Delta(C)$ be a full-support reference policy, and let $\tau > 0$. The policy-optimization stage solves

$$\max_{\pi \in \Delta(C)} \left\{ \sum_i \pi(i) r_i - \tau D_{\mathrm{KL}}(\pi \| \pi_0) \right\}.$$

The first term rewards high-scoring responses.

Policy Optimization Trades Reward Against Policy Change

Let $\pi_0 \in \Delta(C)$ be a full-support reference policy, and let $\tau > 0$. The policy-optimization stage solves

$$\max_{\pi \in \Delta(C)} \left\{ \sum_i \pi(i) r_i - \tau D_{\mathrm{KL}}(\pi \| \pi_0) \right\}.$$

The first term rewards high-scoring responses.
The second penalizes movement away from π_0. The temperature τ controls the strength of this penalty.

The Optimized Policy Is an Exponential Tilt

Theorem

The unique optimizer of the KL-regularized policy objective is

$$\pi_r(i) = \frac{\pi_0(i) e^{r_i/\tau}}{\sum_j \pi_0(j) e^{r_j/\tau}}.$$

The Optimized Policy Is an Exponential Tilt

Theorem

The unique optimizer of the KL-regularized policy objective is

$$\pi_r(i) = \frac{\pi_0(i) e^{r_i/\tau}}{\sum_j \pi_0(j) e^{r_j/\tau}}.$$

We first establish the formula, then use it to connect the learned social order to the optimized policy's behavior.

Proof: Rewrite the Candidate's Log Ratio

Define

$$Z_r := \sum_j \pi_0(j) e^{r_j/\tau}, \qquad \pi_r(i) := \frac{\pi_0(i) e^{r_i/\tau}}{Z_r}.$$

Then

$$\log \frac{\pi_r(i)}{\pi_0(i)} = \frac{r_i}{\tau} - \log Z_r.$$

Proof: KL Nonnegativity Gives Optimality

For each response i, inserting $\pi_0(i)$ gives $\pi(i)/\pi_r(i) = [\pi(i)/\pi_0(i)]/[\pi_r(i)/\pi_0(i)]$. The denominator ratio is divided, so logarithms subtract. Therefore

$$\begin{aligned} D_{\mathrm{KL}}(\pi\|\pi_r) &= \sum_i \pi(i) \log \frac{\pi(i)}{\pi_r(i)} \\ &= \sum_i \pi(i) \log \frac{\pi(i)}{\pi_0(i)} - \sum_i \pi(i) \log \frac{\pi_r(i)}{\pi_0(i)} \\ &= D_{\mathrm{KL}}(\pi\|\pi_0) - \frac{1}{\tau} \sum_i \pi(i) r_i + \log Z_r. \end{aligned}$$

Proof: KL Nonnegativity Gives Optimality

For each response i, inserting $\pi_0(i)$ gives $\pi(i)/\pi_r(i) = [\pi(i)/\pi_0(i)]/[\pi_r(i)/\pi_0(i)]$. The denominator ratio is divided, so logarithms subtract. Therefore

$$\begin{aligned} D_{\mathrm{KL}}(\pi\|\pi_r) &= \sum_i \pi(i) \log \frac{\pi(i)}{\pi_r(i)} \\ &= \sum_i \pi(i) \log \frac{\pi(i)}{\pi_0(i)} - \sum_i \pi(i) \log \frac{\pi_r(i)}{\pi_0(i)} \\ &= D_{\mathrm{KL}}(\pi\|\pi_0) - \frac{1}{\tau} \sum_i \pi(i) r_i + \log Z_r. \end{aligned}$$

Multiplying by $-\tau$ and rearranging yields

$$\sum_i \pi(i) r_i - \tau D_{\mathrm{KL}}(\pi\|\pi_0) = \tau \log Z_r - \tau D_{\mathrm{KL}}(\pi\|\pi_r).$$

The right side is uniquely maximized at $\pi = \pi_r$ because KL divergence is nonnegative, with equality only when its arguments agree.

The Theorem Connects the Social Order to Behavior

Taking ratios in the exponential-tilt formula gives

$$\log \frac{\pi_r(i)}{\pi_r(j)} = \log \frac{\pi_0(i)}{\pi_0(j)} + \frac{r_i - r_j}{\tau}.$$

The Theorem Connects the Social Order to Behavior

Taking ratios in the exponential-tilt formula gives

$$\log \frac{\pi_r(i)}{\pi_r(j)} = \log \frac{\pi_0(i)}{\pi_0(j)} + \frac{r_i - r_j}{\tau}.$$

- At positive temperature, fitted reward gaps update the reference policy's log odds. Borda determines the ordering, but not these gaps.

The Theorem Connects the Social Order to Behavior

Taking ratios in the exponential-tilt formula gives

$$\log \frac{\pi_r(i)}{\pi_r(j)} = \log \frac{\pi_0(i)}{\pi_0(j)} + \frac{r_i - r_j}{\tau}.$$

- At positive temperature, fitted reward gaps update the reference policy's log odds. Borda determines the ordering, but not these gaps.
- As $\tau \downarrow 0$, the policy concentrates on the reward maximizers. If several responses tie for the maximum, the limit is π_0 renormalized to that set.

The Theorem Connects the Social Order to Behavior

Taking ratios in the exponential-tilt formula gives

$$\log \frac{\pi_r(i)}{\pi_r(j)} = \log \frac{\pi_0(i)}{\pi_0(j)} + \frac{r_i - r_j}{\tau}.$$

- At positive temperature, fitted reward gaps update the reference policy's log odds. Borda determines the ordering, but not these gaps.
- As $\tau \downarrow 0$, the policy concentrates on the reward maximizers. If several responses tie for the maximum, the limit is π_0 renormalized to that set.
- Thus an example with a unique low-welfare Borda winner becomes an example for aggressively optimized scalar RLHF. We will quantify this distortion later.

Majority Preprocessing Turns Borda into Copeland

Copeland RLHF replaces each population probability by its majority outcome:

$$H_{ij} := \begin{cases} 1, & P_{ij} > 1/2, \\ 1/2, & P_{ij} = 1/2, \\ 0, & P_{ij} < 1/2. \end{cases} \qquad \mathrm{Cop}(i) := \sum_{j \neq i} H_{ij}.$$

Majority Preprocessing Turns Borda into Copeland

Copeland RLHF replaces each population probability by its majority outcome:

$$H_{ij} := \begin{cases} 1, & P_{ij} > 1/2, \\ 1/2, & P_{ij} = 1/2, \\ 0, & P_{ij} < 1/2. \end{cases} \qquad \mathrm{Cop}(i) := \sum_{j \neq i} H_{ij}.$$

Our Borda calculation ranks responses by the row sums of the target matrix. Replacing P by H therefore replaces Borda totals by Copeland scores.

Majority Preprocessing Turns Borda into Copeland

Copeland RLHF replaces each population probability by its majority outcome:

$$H_{ij} := \begin{cases} 1, & P_{ij} > 1/2, \\ 1/2, & P_{ij} = 1/2, \\ 0, & P_{ij} < 1/2. \end{cases} \qquad \mathrm{Cop}(i) := \sum_{j \neq i} H_{ij}.$$

Our Borda calculation ranks responses by the row sums of the target matrix. Replacing P by H therefore replaces Borda totals by Copeland scores.

Technical note. Exact zero–one targets need not have a finite optimum. Replacing H_{ij} by $\delta + (1 - 2\delta)H_{ij}$ for $0 < \delta < 1/2$ moves the targets into $(0, 1)$ without changing their Copeland ordering.

Borda and Copeland Can Select Different Winners

Borda asks which response has the greatest average pairwise support; Copeland asks which response wins the most pairwise-majority contests.

Borda and Copeland Can Select Different Winners

Borda asks which response has the greatest average pairwise support; Copeland asks which response wins the most pairwise-majority contests.

Consider three population comparison probabilities:

$$P_{ab} = 0.51, \qquad P_{ac} = 0.51, \qquad P_{bc} = 0.99.$$

Borda and Copeland Can Select Different Winners

Borda asks which response has the greatest average pairwise support; Copeland asks which response wins the most pairwise-majority contests.

Consider three population comparison probabilities:

$$P_{ab} = 0.51, \qquad P_{ac} = 0.51, \qquad P_{bc} = 0.99.$$

The Borda totals are

$$(S_a, S_b, S_c) = (1.02, 1.48, 0.50),$$

so fitting the original probabilities ranks $b \succ a \succ c$.

Borda and Copeland Can Select Different Winners

Borda asks which response has the greatest average pairwise support; Copeland asks which response wins the most pairwise-majority contests.

Consider three population comparison probabilities:

$$P_{ab} = 0.51, \qquad P_{ac} = 0.51, \qquad P_{bc} = 0.99.$$

The Borda totals are

$$(S_a, S_b, S_c) = (1.02, 1.48, 0.50),$$

so fitting the original probabilities ranks $b \succ a \succ c$.

Copeland counts majority wins: a has two, b has one, and c has none. Majority-preprocessed fitting therefore ranks

$$a \succ b \succ c.$$

Borda and Copeland Can Select Different Winners

Borda asks which response has the greatest average pairwise support; Copeland asks which response wins the most pairwise-majority contests.

Consider three population comparison probabilities:

$$P_{ab} = 0.51, \qquad P_{ac} = 0.51, \qquad P_{bc} = 0.99.$$

The Borda totals are

$$(S_a, S_b, S_c) = (1.02, 1.48, 0.50),$$

so fitting the original probabilities ranks $b \succ a \succ c$.

Copeland counts majority wins: a has two, b has one, and c has none. Majority-preprocessed fitting therefore ranks

$$a \succ b \succ c.$$

Same raw comparisons and fitting machinery; different preprocessing, different social choice rule.

Summary

- A scalar reward model is also a social aggregation rule.

Summary

- A scalar reward model is also a social aggregation rule.
- In the clean tabular population model, uniform logistic fitting produces the Borda ordering by matching total wins.

Summary

- A scalar reward model is also a social aggregation rule.
- In the clean tabular population model, uniform logistic fitting produces the Borda ordering by matching total wins.
- KL-regularized optimization converts fitted reward gaps into log-odds updates; aggressive optimization therefore selects Borda maximizers.

Summary

- A scalar reward model is also a social aggregation rule.
- In the clean tabular population model, uniform logistic fitting produces the Borda ordering by matching total wins.
- KL-regularized optimization converts fitted reward gaps into log-odds updates; aggressive optimization therefore selects Borda maximizers.
- The theorem delivers an *ordering* only. Fitted gaps encode model log-odds, not welfare, and no interpersonal utility scale has been recovered.

Summary

- A scalar reward model is also a social aggregation rule.
- In the clean tabular population model, uniform logistic fitting produces the Borda ordering by matching total wins.
- KL-regularized optimization converts fitted reward gaps into log-odds updates; aggressive optimization therefore selects Borda maximizers.
- The theorem delivers an *ordering* only. Fitted gaps encode model log-odds, not welfare, and no interpersonal utility scale has been recovered.
- Label preprocessing changes the social objective: fitting probabilities gives Borda, while fitting majority outcomes gives Copeland.

Summary

- A scalar reward model is also a social aggregation rule.
- In the clean tabular population model, uniform logistic fitting produces the Borda ordering by matching total wins.
- KL-regularized optimization converts fitted reward gaps into log-odds updates; aggressive optimization therefore selects Borda maximizers.
- The theorem delivers an *ordering* only. Fitted gaps encode model log-odds, not welfare, and no interpersonal utility scale has been recovered.
- Label preprocessing changes the social objective: fitting probabilities gives Borda, while fitting majority outcomes gives Copeland.

Next time: keep the cycle intact instead of compressing it, and ask what a randomized policy can guarantee against every head-to-head challenger.

Thanks!

See you next class!

References: RLHF and Reward Fitting

- Long Ouyang et al. "Training Language Models to Follow Instructions with Human Feedback." *NeurIPS*, 2022.
- Ralph A. Bradley and Milton E. Terry. "Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons." *Biometrika*, 1952.
- R. Duncan Luce. *Individual Choice Behavior: A Theoretical Analysis*, 1959.
- Anand Siththaranjan, Cassidy Laidlaw, and Dylan Hadfield-Menell. "Distributional Preference Learning: Understanding and Accounting for Hidden Context in RLHF." *ICLR*, 2024.
- Arun Rajkumar and Shivani Agarwal. "A Statistical Convergence Perspective of Algorithms for Rank Aggregation from Pairwise Data." *ICML*, 2014.

References: Social Choice and Distortion

- Jiancong Xiao et al. "Theoretical Tensions in RLHF: Reconciling Empirical Success with Inconsistencies in Social Choice Theory." arXiv:2506.12350, 2025.
- Daniel Halpern et al. "AI Alignment From Social Choice Perspectives." *ACM SIGecom Exchanges*, 2026.
- Paul Gölz, Nika Haghtalab, and Kunhe Yang. "Distortion of AI Alignment: Does Preference Optimization Optimize for Preferences?" arXiv:2505.23749, 2025.