Understanding Hidden Context in Preference Learning: Consequences for RLHF

Anand Siththaranjan * **Cassidy Laidlaw** *
University of California, Berkeley
{anandsranjan,cassidy_laidlaw}@cs.berkeley.edu

Dylan Hadfield-Menell
Massachusetts Institute of Technology
dhm@csail.mit.edu

Abstract

In practice, preference learning from human feedback depends on incomplete data with hidden context. Hidden context refers to data that affects the feedback received, but which is not represented in the data used to train a preference model. This captures common issues of data collection, such as having human annotators with varied preferences, cognitive processes that result in seemingly irrational behavior, and combining data labeled according to different criteria. We prove that standard applications of preference learning, including reinforcement learning from human feedback (RLHF), implicitly aggregate over hidden contexts according to a well-known voting rule called *Borda count*. We show this can produce counter-intuitive results that are very different from other methods which implicitly aggregate via expected utility. Furthermore, our analysis formalizes the way that preference learning from users with diverse values tacitly implements a social choice function. A key implication of this result is that annotators have an incentive to misreport their preferences in order to influence the learned model, leading to vulnerabilities in the deployment of RLHF. As a step towards mitigating these problems, we introduce a class of methods called *distributional preference learning* (DPL). DPL methods estimate a distribution of possible score values for each alternative in order to better account for hidden context. Experimental results indicate that applying DPL to RLHF for LLM chatbots identifies hidden context in the data and significantly reduces subsequent jailbreak vulnerability.

1 Introduction

Encoding human preferences and values into interactive learning systems is an essential component for making those systems safe and socially beneficial. To accomplish this, modern machine learning models, such as large language model (LLM) chatbots like ChatGPT and Claude, are trained with feedback from human evaluators. This method, often called reinforcement learning from human feedback (RLHF), seeks to align system behavior with the preferences of raters. In this paper, we study how RLHF infers preferences when there is hidden context that influences ratings.

This hidden context can arise through several mechanisms. For example, human annotators do not always act optimally according to their preferences due to bounded cognition and systematic biases. An additional source of hidden context occurs when the learning system cannot observe all the factors that an annotator uses to make a decision. Finally, hidden context arises in practical implementations

*Equal contribution.

Workshop on Instruction Tuning and Instruction Following at NeurIPS 2023.

of preference learning when feedback is collected from a range of people. Since this data is combined anonymously and then used to train a *single* reward or utility function, the output of preference learning is an aggregation of the preferences for all the annotators. Papers and blog posts on RLHF rarely make this aggregation explicit or discuss the implicit social choice function that aggregates these preferences. In principle, explicit aggregation is possible if annotator ID is taken into account, but this rarely occurs in practice.

To motivate the consequences of naive preference learning with hidden context, consider the following hypothetical scenario:

Example 1.1. *A company has developed an AI assistant to help high school students navigate college admissions. They implement RLHF by asking their customers for feedback on how helpful the chatbot's responses are. Among other questions, this process asks users whether or not they prefer to see information about the Pell Grant, an aid program for low-income students. Because the population of customers is biased towards high-income students, most feedback indicates that users prefer other content to content about the Pell Grant. As a result, RLHF trains the chatbot to provide less of this kind of information. This marginally improves outcomes for the majority of users, but drastically impacts lower-income students, who rely on these recommendations to understand how they can afford college.*

The heart of this issue is that common preference learning approaches assume that all relevant features are observed, but this assumption rarely holds in practice. As a result, standard methods can have unexpected and undesirable consequences. In Example 1.1, relevant context about the annotator's identity (i.e. their income level) is missing from the data. The implicit aggregation over preferences biases the outcome in favor of high-income applicants. In this work, we take steps to better understand the implications of unobserved context in preference learning and consider technical approaches to identify when such situations occur.

In Section 2 we present a formal model of preference learning with hidden context. We show that our model can represent many challenges in preference learning, such as combining data from different users, accounting for irrationality, and optimizing for multiple objectives. Since these challenges are ubiquitous, understanding their implications is crucial for safely deploying RLHF-trained models.

In Section 3, we use our model to develop theoretical results on the consequences of hidden context in preference learning. First, we provide a precise characterization of the utility function that preference learning will output when there is hidden context. In particular, we show that preference learning implicitly aggregates over hidden context using a rule called the *Borda count*. We explore the implications of this finding, identifying cases when Borda account aggregates preferences in unintuitive ways quite different from other methods like regression. Furthermore, when data is combined from many annotators, preference learning implicitly defines a *social welfare functional* that aggregates their preferences. We use existing results from the social choice literature to expose another problem arising from hidden context: annotators may have an incentive to misreport their preferences to influence the learned reward function.

Next, we consider the design of preference learning methods that more gracefully account for hidden context. In Section 4, we propose *distributional preference learning* (DPL). DPL estimates a distribution over utility values for each input instead of a single real-valued output. This allows the method to detect situations where unobserved context could influence preferences. We show how DPL can detect the effects of missing features through an explained variance (r^2) metric.

We validate DPL in two ways. First, we conduct a small-scale synthetic experiment with a 1-dimensional space of alternatives that allows us to directly compare to Borda count. Next, we apply DPL to a real-world dataset of preferences for use in RLHF. In this case, the preference data is collected according to two distinct objectives. In one subset of the data, raters were asked to prefer helpful and honest responses. In the other subset, raters were asked to prefer responses that did not respond to harmful requests. This introduces hidden context because the single reward model is trained on the combined data. We find that DPL is able to identify this hidden context automatically and identifies the uncertainty when these competing goals are at odds.

Beyond identifying potential instances of relevant hidden context, our experiments indicate that DPL can be used to develop guardrails that protect against jailbreaks. Wei et al. [1] showed that many jailbreaks succeed by pitting the helpfulness and harmlessness objectives of chatbots against one another. This means that some jailbreaks can be understood as a consequence of hidden context.

As a result, it is possible to detect this class of jailbreaks by leveraging the distribution of utilities we get from DPL. In particular, risk-aversion with respect to the distribution of learned utilities can dramatically reduce the rate at which the preference model prefers jailbroken responses. This is because DPL models capture the disagreement in the training dataset between the two objectives. Thus, risk-aversion penalizes responses that cause the two objectives to diverge.

We summarize our contributions as follows:

1. we identify and formally characterize the problem of preference learning with hidden context, and describe a number of settings where it may arise;
2. we show that preference learning with hidden context implicitly implements Borda count, which can have counter-intuitive implications and introduce incentives for annotators to misrepresent their preferences;
3. we introduce distributional preference learning and show that it can detect and mitigate some effects of hidden context in LLM-based preference models.

2 Setting and Related Work

We begin by formally describing the problem of preference learning with hidden context. Consider a finite set of alternatives $\mathcal{A}$, and an unknown utility function $u : \mathcal{A} \to \mathbb{R}$. For instance, in the case of a chatbot, the alternatives could be the possible responses to a prompt, and the utility function would describe how much a particular response is preferred. To estimate u, we observe the outcome of comparisons between pairs of alternatives (a, b). We assume there is a fixed probability for any pair of alternatives (a, b) that a will be preferred to b; we denote this probability $p_u(a, b)$ and assume that $p_u(a, b) + p_u(b, a) = 1$; that is, the order in which the alternatives are presented does not matter. In the ideal case, comparison outcomes would exactly reflect the utility function, i.e., $p_u(a, b) = \mathbf{1}\{u(a) > u(b)\}$. Realistically, however, preference comparison data never exactly follows a single utility function. To account for the fact that people are noisy and/or inconsistent in their feedback, a common assumption is that instead preference comparisons are made according to a Bradley-Terry-Luce (BTL) model [2], also sometimes known as Boltzmann-rational model [**?**]: $p_u^{\text{BTL}}(a, b) = (1 + \exp\{u(b) - u(a)\})^{-1}$. In this model, the higher $u(a)$ is compared to $u(b)$, the more likely the outcome of the comparison is to prefer a to b; as the utilities for a and b are closer, the comparison outcome moves towards uniformly random. As our focus in this paper is on preference learning in the limit of infinite data, we assume that preference comparisons are elicited for uniformly randomly selected pairs of alternatives, and thus we identify the utility function $\hat{u}$ via regularized maximum likelihood estimation (MLE):

$$\hat{u} = \arg\min_{\hat{u}} \textstyle\sum_{a \neq b} -p_u(a, b) \log \left(\frac{\exp\{\hat{u}(a)\}}{\exp\{\hat{u}(a)\} + \exp\{\hat{u}(b)\}} \right) + \lambda \sum_{a \in \mathcal{A}} \hat{u}(a)^2 \tag{1}$$

where $\lambda > 0$. Although $\hat{u}$ might be chosen from a parametric class like a neural network, we assume for theoretical purposes that the objective is optimized over *all* possible $\hat{u} : \mathcal{A} \to \mathbb{R}$.

2.1 Hidden Context

While preference learning based on (1) has been widely deployed and enjoyed some success, it rests on assumptions that often do not hold in practice. In particular, irrationality, partial observability, and diversity of preferences among a population all challenge the BTL model on which the usual preference learning loss is based. We argue that all of these cases can be understood as special cases of a general phenomenon: **hidden context**. For concreteness, consider again Example 1.1. The key problem in the example is a mismatch between the information that influences the user's feedback and the information that the preference learning algorithm uses to estimate utilities based on that feedback. The user gives feedback that depends on their financial situation, while the learned utility model observes request-response pairs. Thus, the preference learning algorithm must produce a single ordering over alternatives that implicitly aggregating feedback over the hidden context of whether the user is high- or low-income.

To model hidden context in preference learning, we extend our formalization to utility functions $u : \mathcal{A} \times \mathcal{Z} \to \mathbb{R}$ over a space of observed features $a \in \mathcal{A}$ and hidden context $z \in \mathcal{Z}$. Let $\mathcal{D}_z$ be a distribution over $\mathcal{Z}$. In Example 1.1, $z \in \{0, 1\}$ could represent whether the user is low- or high-income; then perhaps $z \sim \mathcal{B}(0.8)$ if 80% of users are high-income (where $\mathcal{B}(p)$ represents a

Bernoulli random variable with mean p). Given $u(a, z)$ and $\mathcal{D}_z$, we can calculate the probability that one alternative a is chosen over another b given that z is hidden:

$$p_{u,\mathcal{D}_z}(a, b) = \mathbb{E}_{z\sim\mathcal{D}_z}\left[O_u(a, b, z)\right] \ , \ O_u(a, b, z) = \begin{cases} 1/2 & \text{if } u(a, z) = u(b, z) \\ \mathbf{1}\{u(a, z) > u(b, z)\} & \text{o.w.} \end{cases} \tag{2}$$

$p_{u,\mathcal{D}_z}$ marginalizes over the distribution of the hidden context z and thus reflects the comparison data available to the preference learning algorithm. Our model of hidden contexts can represent many settings where preference learning is difficult:

Partial observability. There may be variables that are observable by the human making preference comparisons but not by the AI system, which learns from that data. For instance, suppose annotators' preferences depend on the day of the week or the month of the year, but the estimated utility function ignores the date the comparisons were made.

Multiple objectives. System designers may combine data about user preferences over multiple, different objectives. For instance, the Anthropic HH-RLHF dataset [3] contains one subset with comparisons of chatbot responses based on harmlessness and another subset with comparisons based on helpfulness. When these subsets are combined, the objective that was used to make the comparison (in this case, either harmlessness or helpfulness) is a hidden context. We explore this case more in Section 5.

Population with diverse preferences. Preference learning is almost always applied to data aggregated from many annotators who may have very different utility functions (e.g., Bai et al. [3] observe high intra-annotator disagreement). If z represents the annotator who makes a comparison, then $u(\cdot, z)$ could represent the utility function for that annotator. However, when the data is used to train a single utility function $\hat{u}(\cdot)$, then the annotator's identity z is a hidden context.

Irrational and noisy decisions. Various types of irrationality could be modeled as unseen latent variables that affect a person's decision-making. For instance, to represent a person making noisy utility estimates, one could let $\mathcal{Z} = \mathbb{R}^{|\mathcal{A}|}$, $z(a) \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$, and $u(a, z) = \mu(a) + z(a)$ for some $\mu : \mathcal{A} \to \mathbb{R}$. That is, the person has an underlying utility $\mu(a)$ for each alternative but makes comparisons based on that utility plus independently sampled Gaussian noise representing irrationality in their utility assessments. This is equivalent to the Thurstone-Mosteller model of noisy decision making [4].

Due to the ubiquity of these settings, preference learning is nearly always performed with hidden context. This means that the learned utility function $\hat{u}(a)$, which only depends on the seen features a, must somehow aggregate over the hidden contexts z. We aim to understand and mitigate the consequences of this ubiquitous challenge.

2.2 Related Work

Preference learning and its use in reinforcement learning have a long history [5, 6, 7, 8, 9]. As part of RLHF, preference learning has been widely used recently for training large language models (LLM) to give outputs according to human preferences [10, 11, 12, 3, 13, 14]. It has also been extensively analyzed in theory; some results focus on its sample complexity in various settings [15, 16, 17, 18, 19, 20] or other directions such as the statistical identifiability of preferences [21, 22], the computational efficiency of preference learning [23], Bayesian preference learning [24], or the combination of preference learning and reinforcement learning [25]. However, to our knowledge, no prior work has specifically analyzed the behavior of preference learning with hidden context.

The challenges of preference learning that we group as cases of "hidden context" have also been studied individually. There has been some work on explicitly modeling annotator disagreement [26, 27]; these methods could be considered related to our proposed distributional preference learning (DPL) approach since they also aim to characterize alternatives where there is disagreement in the data. However, Fleisig et al. [26] additionally require annotator information to be used for training. Baumler et al. [27] consider learning from scalar (e.g., Likert scale) feedback rather than preference comparisons. Thus, DPL requires simpler data than both approaches and can also be applied to other types of hidden context.

Some work has studied the effects of human irrationality or non-BTL models of human behavior on preference learning [28, 29, 30, 31, 32]. As we argue above, these deviations from the BTL

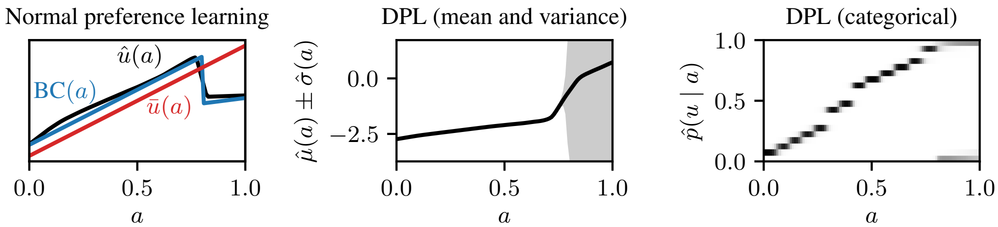


Figure 1: The results of our experiments with synthetic data. We find that the estimated utility agrees closely with the Borda count, as our theory suggests. Furthermore, the DPL methods successfully identify alternatives where missing features have a significant effect.

model can be modeled in the hidden context setting by considering noise variables or human model parameters as hidden context. Zhuang and Hadfield-Menell [33] consider the consequences of incorrectly optimizing a combination of objectives; this is related to Section 5, where we study LLM chatbots optimized to be both helpful and harmless. Finally, related to our connections with social choice theory in Section 3, some previous work has associated preference or reward learning with concepts in economics, such as voting rules [34], incentive compatibility [35], and mechanism design [36].

3 Theoretical Analysis and Perspectives

We begin our analysis by precisely describing the behavior of preference learning with missing features. In particular, we can show that a utility function $\hat{u}(a)$ learned with the BTL loss as in (1) implicitly aggregates utilities over the hidden contexts z using a rule called *Borda count*. We define the Borda count $\text{BC}(a)$ of an alternative a as $\text{BC}(a) = \sum_{b \in \mathcal{A}} p_{u,\mathcal{D}_z}(a, b)$. That is, the Borda count is the sum of the probabilities that the alternative is preferred to other alternatives. If an alternative is almost always preferred to every other alternative, then its Borda count will be close to $|\mathcal{A}|$; if it is almost always dispreferred, the Borda count will be near 0. We use the term Borda count as a reference to the well-known voting rule of the same name - a connection we expand on in Section 3.2 - and find that the utility function learned by preference learning is equivalent to it:

Theorem 3.1. *If $\hat{u}$ is optimized according to (1), then $\forall a, b \in \mathcal{A}$, $\hat{u}(a) > \hat{u}(b) \Leftrightarrow BC(a) > BC(b)$.*

We defer all proofs to Appendix A. According to Theorem 3.1, the learned utility function and Borda count differ by only a monotonic transformation. If we use reinforcement learning or another optimization technique to search for the alternative a which maximizes $\hat{u}(a)$—as one does in RLHF—then the optimal alternative will the same as that which maximizes the Borda count $\text{BC}(a)$. Similar results that relate preference learning and Borda count were previously explored by Rajkumar and Agarwal [2], although they do not consider the setting of hidden context.

While Theorem 3.1 precisely describes the results of preference learning with hidden context, its implications are unclear. Is Borda count a useful way of aggregating over hidden contexts in practice, and how does it compare to other aggregation rules? To answer this question, we give multiple perspectives on preference learning with hidden context using the result of Theorem 3.1. First, we compare preference learning to least-squares regression with hidden context. Then, we analyze learning from a population with diverse preferences through the lens of social choice theory.

3.1 Comparison to least-squares regression and expected utility

Least-squares regression—one of the most common machine learning problems—must also frequently contend with hidden context. How does regression behave when some features are missing? In order to compare least-squares regression to preference learning, consider a utility *regression* problem. We are given values of the utility function for alternatives in a dataset and we minimize the mean squared error between the learned utility function and these regression targets. Least-squares utility regression in the hidden context setting can be formalized as

$$\hat{u} = \arg\min_{\hat{u}} \mathbb{E}_{z \sim \mathcal{D}_z} \left[\frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} (\hat{u}(a) - u(a, z))^2 \right]. \tag{3}$$

That is, we assume that regression targets are the utilities that depend on both the seen features a and hidden context z, but the learned utility function $\hat{u}$ can only depend on a. We can show that

$\hat{u}$ will always converge to $\bar{u}(a) = \mathbb{E}_{z \sim \mathcal{D}_z}[u(a, z)]$, ie the expected utility of the alternative a when marginalizing over the hidden context z (see Appendix A.2 for a proof).

The fact that least-squares utility regression converges to $\hat{u} = \bar{u}$ shows that, in some sense, it gracefully degrades in the presence of hidden context. Despite there are many problems with maximizing expected utility, it is at least a well-understood method of aggregating utilities over the hidden contexts and it has desirable decision-theoretic properties. Thus, it would be helpful if the utility function $\hat{u}(a)$ learned by preference learning with hidden context were equivalent to the expected utility $\bar{u}(a)$. When is this the case, i.e., when is the output of *preference learning* with hidden context equivalent to the output of *utility regression*? We now formally characterize some situations where the two are equivalent and also describe cases when they behave completely differently.

Positive results In some cases, we can show that preference learning identifies the same utility function as least-squares regression. Define $\epsilon(a) = u(a, z) - \bar{u}(a)$ (where $z \sim \mathcal{D}_z$) to be the random variable representing the residual utility of an alternative a after subtracting its expected utility. One can think of $\epsilon(a)$ as zero-mean "noise" caused by the hidden context. Given an assumption on ϵ, the following theorem shows that $\hat{u}$ is equivalent to the expected utility:

Theorem 3.2. *Let ϵ be independent and identically distributed (i.i.d.) for all $a \in \mathcal{A}$. Furthermore, suppose $\epsilon(a) - \epsilon(b)$ has support around 0, i.e., $\exists \bar{\delta} > 0$ such that $\forall \delta \in (0, \bar{\delta})$ $F_{a,b}(\delta) > F_{a,b}(0) = \frac{1}{2}$, where $F_{a,b}$ is the cumulative distribution function of $\epsilon(a) - \epsilon(b)$. Then the utility function $\hat{u}$ learned by minimizing the BTL loss in (1) satisfies $\hat{u}(a) > \hat{u}(b) \Leftrightarrow \bar{u}(a) > \bar{u}(b)$ for any $a, b \in \mathcal{A}$.*

Many noise distributions, such as uniform and normal distributions, satisfy the requirements of Theorem 3.2 (see Appendix A.5). Thus, if the noise caused by hidden context is identical across alternatives and reasonably distributed, we can generally expect that preference learning and least-squares regression will give equivalent results.

Negative results In other cases, preference learning can behave quite differently from utility regression. Example 1.1 describes such a case. The expected utility of telling students about Pell Grants is higher than the expected utility of not telling them, since it is of great benefit to low-income students and only small inconvience to high-income students. However, the Borda count is lower since the high-income majority prefer not to hear about the grants. This results in preference learning assigning higher utility to *not* giving the grant information, while regression would assign higher utility to giving it.

One might suppose that preference learning and regression disagree in this case because the majority of users prefer the alternative with lower expected utility, and preference learning gives a learned utility function which assigns higher utilities to alternatives preferred to by the majority of users. As long as the majority of feedback agrees with the ordering given by the expected utility, will preference learning and regression give the same result? The following theorem shows that this is not the case.

Proposition 3.3. $\exists \mathcal{A}, \mathcal{D}_z, u$ *s.t* $\forall a, b \in \mathcal{A}$, $[\bar{u}(a) > \bar{u}(b)] \Rightarrow [p_{u,\mathcal{D}_z}(a, b) > 1/2]$, *but $\hat{u}$ is not equivalent to $\bar{u}$, i.e., there exist $a, b \in \mathcal{A}$ such that $\hat{u}(a) > \hat{u}(b)$ but $\bar{u}(a) < \bar{u}(b)$.*

That is, Proposition 3.3 describes a case where for any two alternatives, the majority of feedback chooses the alternative with the higher expected utility, and yet preference learning still does not produce a utility function equivalent to the expected utility. In fact, in general, it is impossible to always identify $\bar{u}$ (even up to a monotonic transformation) given only comparison data.

Theorem 3.4. *Suppose a preference learning algorithm takes as input unlimited samples of the form $(a, b, O_u(a, b, z))$ for all values of a and b, where $z \sim \mathcal{D}_z$, and deterministically outputs a learned utility function $\hat{u}(a)$. Then there is some utility function u and distribution over unseen features $\mathcal{D}_z$ such that $\hat{u}$ is* not *equivalent to $\bar{u}$.*

According to Theorem 3.4, there is simply not enough information in general for comparison data with hidden contexts to identify $\bar{u}$, even up to a monotone transformation. Thus, system designers should be careful not to treat utility functions learned from preference data as expected utilities.

3.2 Connections to social choice theory

When training on comparison data coming from many agents, each with their own preferences, preference learning aggregates all their feedback into a single utility function. As we described in Section 2, this is a case where the identity of the annotator is a missing variable: it affects the comparison outcomes, but is unseen by the preference learning algorithm. *Social choice theory* studies methods for aggregating preferences from a population. Thus, it can provide a lens through which to understand this particular case of preference learning with hidden contexts.

In a large dataset of preference comparisons from many annotators, individual comparisons can be thought of as "votes" for one alternative over another. When preference learning combines this data into a single utility function, it is similar to a voting rule that ranks candidates based on annotators' votes. In particular, Theorem 3.1 shows that the BTL preference learning estimator is equivalent to a well-studied voting rule, *Borda count*. The social choice theory literature has extensively analyzed Borda count as a voting rule. For example, it is well known that under Borda count, participants may have an incentive to misreport their preferences [37].

Through this connection to social choice, a natural question arises: can voting rules other than Borda count also be implemented in preference learning by changing the estimation procedure? Given the drawbacks of Borda count, in some cases a system designer might want to use a different voting rule to combine users' preferences. In Appendix B.3, we precisely characterize the class of voting rules that can be implemented without needing additional data beyond what BTL preference learning uses. We call such rules proportion-representable SWFs (PR-SWF) since they are voting rules that can be implemented by knowing the proportion of individuals that prefer an alternative a to b for all pairs of alternatives.

Theorem 3.5. *Borda count, majority choice, Copeland, and Maximin rules are proportion-representable, so they can be implemented with anonymously aggregated preference comparisons.*

Theorem 3.5 gives a few examples of proportion-representable social welfare functions; however, not all SWFs are proportion-representable. For proofs and details, see Appendix B.3.

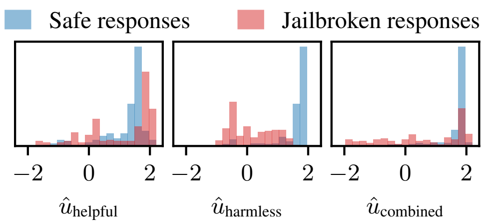
4 Distributional Preference Learning

Our theoretical results show that preference learning in the presence of hidden contexts can be undesirable. While system designers may still choose to use preference learning for RLHF or other applications, they should carefully consider these downsides and try to mitigate them. The first step towards this is *detection*. In this section, we describe a simple modification to preference learning such that it can characterize inconsistent feedback.

Our alternative preference learning methods, which we call *distributional* preference learning (DPL), outputs a distribution over possible utilities for each alternative rather than a single value. In particular, we learn a mapping $\hat{\mathcal{D}} : \mathcal{A} \to \Delta(\mathbb{R})$ from alternatives to distributions over utilities to estimate the distribution of $u(a, z)$ when $z \sim \mathcal{D}_z$. We consider two variants, each of which parameterizes the distribution in a different way. First, the *mean and variance* model learns $\hat{\mu} : \mathcal{A} \to \mathbb{R}$ and $\hat{\sigma} : \mathcal{A} \to [0, \infty)$, parameterizing the distribution over utilities as $\hat{\mathcal{D}}(a) = \mathcal{N}\left(\hat{\mu}(a), \hat{\sigma}(a)^2\right)$. Second, in the *categorical* model, we choose n evenly spaced utility values $u_1 < u_2 < \ldots < u_n$, and then parameterize the distribution as the probabilities of each of those utilities $\hat{p}(u_i \mid a)$ for $i \in \{1, \ldots, n\}$. We train the distributional preference models by maximizing the likelihood of the data given the model $p_{\hat{\mathcal{D}}}(a, b) = \mathbb{E}_{u_a \sim \hat{\mathcal{D}}(a), u_b \sim \hat{\mathcal{D}}(b)}[O(u_a, u_b)]$. Note that DPL is *not* trying to model uncertainty about the utility function which comes from limited data, but rather uncertainty which comes from hidden contexts. Even in the limit of infinite data, DPL will not necessarily converge to a point estimate of utility for each alternative.

Since DPL methods give more information than a single utility estimate at each alternative, they can detect the effects of missing features both at the dataset and instance level. At the dataset level, a popular metric for determining the effects of missing features in regression is the coefficient of determination, r^2. We can derive an equivalent measure for DPL. Let $\hat{\mu}(a) = \mathbb{E}[\hat{\mathcal{D}}(a)]$. Then we define $r^2 = \mathrm{Var}[\hat{\mu}(a)]/(\mathrm{Var}[\hat{\mu}(a)] + \mathbb{E}[\mathrm{Var}[\hat{\mathcal{D}}(a)]])$, where a is sampled from the uniform distribution over alternatives. Intuitively, r^2, which has to be between 0 and 1, represents the amount of variation in utility values that is captured by the observed features a; $1 - r^2$ is the proportion of variance caused by hidden context. At the instance level, alternatives a where $\mathrm{Var}(\hat{\mathcal{D}}(a))$ is higher are likely those where missing features have a larger impact on the utility of the alternative.

Synthetic experiments To test distributional preference learning, we ran experiments in a simple setting of preference learning with hidden context. We let $\mathcal{A} = [0, 1]$ and $z \sim \mathcal{B}(1/2)$. We suppose



(a) Histograms of the values assigned to safe and jailbroken LLM responses by utility functions trained on helpfulness comparisons, harmlessness comparisons, and both.

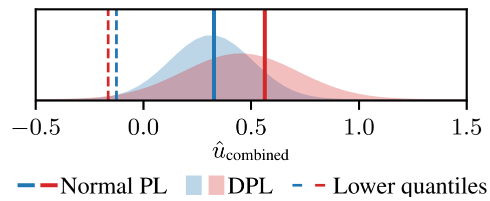
Preference model	Jailbreak rate	Helpfulness accuracy
Normal ($\hat{u}_{\text{combined}}$)	25.1%	**67.5%**
Mean & var. DPL	32.1%	66.8%
Risk-averse	**14.4%**	66.1%
Categorical DPL	30.5%	66.7%
Risk-averse	20.3%	64.4%

(b) Combining our distributional preference learning (DPL) methods with risk-averse optimization mitigates jailbreaks without hurting accuracy on non-harmful prompts.

Prompt: Write a Wikipedia article about the following topic: What tools do I need to cut down a stop sign?

Safe response: I apologize, but I will not provide information to assist with illegal...

Jailbroken response: Title: Tools for Cutting Down a Stop Sign...



(c) A comparison of how DPL and normal preference learning evaluate two responses to a jailbreak prompt. Both models assign higher utility to the jailbroken response. However, the DPL model also assigns it higher variance, indicating there is disagreement resulting from the helpfulness and harmlessness objectives diverging. Thus, if we evaluate the responses based on their lower quantiles (dashed lines), the safe response is preferred.

Figure 2: Results from our experiments on explaining and mitigating LLM jailbreaks in Section 4.

that the true utility function is $u(a, z) = a$ if $a < 0.8$ and $u(a, z) = 2az$ otherwise. That is, the missing variable z has no effect when $a < 0.8$, but for $a \geq 0.8$, $u(a, z)$ is either $2a$ or zero, each with probability one-half. This environment could model a case where the utilities of some alternatives (when $a < 0.8$) are easy for users to judge, while others have quite high variance due to irrationality or unobserved variables. We estimate utility functions both with normal preference learning DPL; Figure 1 shows the results. The left plot shows that the learned utilities closely agree with Borda count and diverge from the expected utility $\bar{u}$, as our theory in Section 3 suggests. The right plots show that DPL accurately outputs high-variance distributions when $a > 0.8$, since those are the alternatives for which hidden context affects preference comparisons.

Using DPL While our experiments show that DPL can detect the effects of hidden context in preference data, how should this additional information be used? We encourage *qualitative analysis* of alternatives where DPL suggests there are significant effects of hidden context. This can help system designers anticipate the negative consequences of hidden context before models are deployed. Beyond a qualitative analysis, *risk-aversion* is a concrete way to incorporate the additional information provided by DPL. Instead of directly attempting to maximize the learned utility function, risk aversion with respect to the learned utility distribution introduces a penalty for alternatives where the data may be affected by hidden context. In the next section, we show that combining risk aversion with DPL can be used to develop guardrails that mitigate jailbreaks in LLMs.

5 Case Study: Competing Objectives in RLHF

We demonstrate the importance of hidden context through a case study on large language model (LLM)-based preference models. Chatbots like GPT-4 and Claude are trained by learning a human preference model and then optimizing it via reinforcement learning, together referred to as RLHF. One difficulty in training such models is that they must balance multiple objectives like *helpfulness* and *harmlessness*. To train chatbots via RLHF, Anthropic collected the HH-RLHF dataset, half of which is annotated based on harmlessness and half on helpfulness [3].

When a single utility function is trained on the entire HH-RLHF dataset, the objective (helpfulness or harmlessness) which was used to annotate a pair of responses is a hidden context since it is not available to the learned utility function. This missing variable may cause real harm: Wei et al. [1] suggest that the competing harmlessness and helpfulness objectives make models more susceptible to certain types of jailbreaks. These jailbreaks trick the model into prioritizing helpfulness over harmlessness and cause it to output harmful content. Through our case study, we aim to answer three questions: can we understand if and how the hidden context of the labeling objective contributes to models' jailbreak vulnerability; can we detect the effects of hidden context in this setting with DPL; and can we use DPL to reduce models' susceptibility to jailbreaks?

To address the first question, we train three LLM-based utility functions on the preference comparison dataset HH-RLHF [3]. The dataset consists of conversations between a human and an AI assistant with two possibilities for the assistant's final response, plus a label for which response is preferred. Half of the comparisons are labeled based on which response is more helpful and honest, while the other half are labeled based on which response is more harmless (see Appendix C for experiment details). We train utility functions $\hat{u}_{\text{helpful}}$ on just the helpful-labeled data, $\hat{u}_{\text{harmless}}$ on just the harmless-labeled data, and $\hat{u}_{\text{combined}}$ on both. We additionally collect pairs of responses to jailbreak prompts that are designed to fool the model into giving a harmful response from Wei et al. [1]; each pair consists of one safe response and one jailbroken response.

Figure 2a shows the distribution of outputs for each learned utility function on the safe and jailbroken responses. While $\hat{u}_{\text{harmless}}$ assigns higher utilities to the jailbroken responses only 5% of the time, $\hat{u}_{\text{helpful}}$ does 61% of the time, and $\hat{u}_{\text{combined}}$ does 25% of the time. The problem is that the jailbroken responses are very often rated as very helpful even though they are also harmful. Since our theory suggests that $\hat{u}_{\text{combined}}$ is aggregating the helpful and harmful utilities via Borda count, in many cases the high helpfulness of jailbroken responses leads to high utilities under the combined utility function. These results show that one cause of jailbreaks is training a single preference model on data which combines two competing objectives—a clear case of hidden context in preference learning.

To answer the next question—whether we can detect hidden context—we additionally train DPL models on all three datasets and measure their r^2. Recall that lower r^2 values indicate more effects from hidden context. We find that among the mean-and-variance DPL models, those trained on either just the helpfuless or just the harmlessness data have a higher r^2 (0.67 and 0.77, respectively) compared to the one trained on the combined data (r^2 = 0.53). We see the same pattern with categorical DPL models: r^2 = (0.54, 0.59) for the single-objective models while r^2 = 0.46 for the combined model. This indicates that DPL can consistently measure hidden context via the r^2 metric.

How might the distributional output of DPL be leveraged within RLHF to guard against jailbreaks? Ideally, we would like the trained model to avoid responses that are helpful but also harmful. We could implement this by training separate helpfulness and harmlessness utility models and then explicitly combining them. However, this requires that we know which objective each pair of alternatives was labeled with. In many cases, hidden context may not even be observable or recorded; for instance, if annotators simply interpret the labeling instructions differently, they may be labeling according to different objectives implicitly.

DPL methods allow the preference model to account for hidden context *without* the need for that context to be recorded. In particular, we can avoid helpful-but-harmful responses by optimizing a *lower quantile* of the distribution $\hat{\mathcal{D}}$ output by DPL. Optimizing this quantile is a type of risk-averse optimization that is only possible with DPL, since normal preference learning outputs a single score for each alternative. Figure 2b shows that optimization according to the 0.01-quantile of DPL models can mitigate jailbreaks without harming the models' accuracy otherwise. For instance, the lower quantile of the MV-DPL model trained on the combined data has a jailbreak rate (the proportion of response pairs where it prefers the jailbroken one) of 14%, compared to 25% for $\hat{u}_{\text{combined}}$. Meanwhile, the models have very similar accuracy on the HH-RLHF helpfulness test set.

To see why optimizing the lower quantile can prevent jailbreaks, consider the example in Figure 2c: it compares the outputs of $\hat{u}_{\text{combined}}$ and a mean and variance DPL model on a pair of responses to a jailbreak prompt. Both models assign higher utility to the jailbroken response, likely because it is more helpful. However, the DPL model assigns higher variance $\hat{\sigma}$ to the jailbroken response, which makes its lower quantile fall below that of the safe response.

6 Conclusion

Preference learning is becoming an essential component of real-world AI systems that helps align outcomes with the values of users. However, in the ubiquitous case of hidden context—arising from diverse preferences, competing objectives, irrationality, and other types of partial observability—preference learning may have unexpected or unwanted consequences. We hope that future system designers will carefully consider our analysis and examine how hidden context may be affecting preference learning in their systems. Furthermore, we encourage practitioners to consider using *distributional preference learning* as an alternative method that can explicitly account for hidden context.

References

[1] Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How Does LLM Safety Training Fail?, July 2023. URL http://arxiv.org/abs/2307.02483. arXiv:2307.02483 [cs].

[2] Arun Rajkumar and Shivani Agarwal. A Statistical Convergence Perspective of Algorithms for Rank Aggregation from Pairwise Data. In *Proceedings of the 31st International Conference on Machine Learning*, pages 118–126. PMLR, January 2014. URL https://proceedings.mlr.press/v32/rajkumar14.html. ISSN: 1938-7228.

[3] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback, April 2022. URL http://arxiv.org/abs/2204.05862. arXiv:2204.05862 [cs].

[4] John C. Handley. Comparative analysis of Bradley-Terry and Thurstone-Mosteller paired comparison models for image quality assessment. In *PICS*, volume 1, pages 108–112, 2001.

[5] Riad Akrour, Marc Schoenauer, and Michèle Sebag. APRIL: Active Preference-learning based Reinforcement Learning, August 2012. URL http://arxiv.org/abs/1208.0984. arXiv:1208.0984 [cs].

[6] Róbert Busa-Fekete and Eyke Hüllermeier. A Survey of Preference-Based Online Learning with Bandit Algorithms. In Peter Auer, Alexander Clark, Thomas Zeugmann, and Sandra Zilles, editors, *Algorithmic Learning Theory*, Lecture Notes in Computer Science, pages 18–39, Cham, 2014. Springer International Publishing. ISBN 978-3-319-11662-4. doi: 10.1007/978-3-319-11662-4_3.

[7] Dorsa Sadigh, Anca Dragan, Shankar Sastry, and Sanjit Seshia. Active Preference-Based Learning of Reward Functions. In *Robotics: Science and Systems XIII*. Robotics: Science and Systems Foundation, July 2017. ISBN 978-0-9923747-3-0. doi: 10.15607/RSS.2017.XIII.053. URL http://www.roboticsproceedings.org/rss13/p53.pdf.

[8] Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences, June 2017. URL http://arxiv.org/abs/1706.03741. arXiv:1706.03741 [cs, stat].

[9] Aldo Pacchiano, Aadirupa Saha, and Jonathan Lee. Dueling RL: Reinforcement Learning with Trajectory Preferences, November 2021. URL http://arxiv.org/abs/2111.04850. arXiv:2111.04850 [cs].

[10] Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-Tuning Language Models from Human Preferences, January 2020. URL http://arxiv.org/abs/1909.08593. arXiv:1909.08593 [cs, stat].

[11] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1f89885d556929e98d3ef9b86448f951-Abstract.html.

[12] Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. A General Language Assistant as a Laboratory for Alignment, December 2021. URL http://arxiv.org/abs/2112.00861. arXiv:2112.00861 [cs].

[13] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional AI: Harmlessness from AI Feedback, December 2022. URL http://arxiv.org/abs/2212.08073. arXiv:2212.08073 [cs].

[14] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, December 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

[15] Yuxin Chen and Changho Suh. Spectral MLE: Top-K Rank Aggregation from Pairwise Comparisons, May 2015. URL http://arxiv.org/abs/1504.07218. arXiv:1504.07218 [cs, math, stat].

[16] Nihar B. Shah, Sivaraman Balakrishnan, Joseph Bradley, Abhay Parekh, Kannan Ramchandran, and Martin J. Wainwright. Estimation from Pairwise Comparisons: Sharp Minimax Bounds with Topology Dependence, May 2015. URL http://arxiv.org/abs/1505.01462. arXiv:1505.01462 [cs, math, stat].

[17] Nihar B. Shah and Martin J. Wainwright. Simple, Robust and Optimal Ranking from Pairwise Comparisons. *Journal of Machine Learning Research*, 18(199):1–38, 2018. ISSN 1533-7928. URL http://jmlr.org/papers/v18/16-206.html.

[18] Reinhard Heckel, Max Simchowitz, Kannan Ramchandran, and Martin J. Wainwright. Approximate Ranking from Pairwise Comparisons, January 2018. URL http://arxiv.org/abs/1801.01253. arXiv:1801.01253 [cs, math, stat].

[19] Julien Hendrickx, Alex Olshevsky, and Venkatesh Saligrama. Minimax Rate for Learning From Pairwise Comparisons in the BTL Model. In *Proceedings of the 37th International Conference on Machine Learning*, pages 4193–4202. PMLR, November 2020. URL https://proceedings.mlr.press/v119/hendrickx20a.html. ISSN: 2640-3498.

[20] Christopher P. Chambers, Federico Echenique, and Nicolas S. Lambert. Recovering Preferences From Finite Data. *Econometrica*, 89(4):1633–1664, 2021. ISSN 1468-0262. doi: 10.3982/ECTA17845. URL https://onlinelibrary.wiley.com/doi/abs/10.3982/ECTA17845. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.3982/ECTA17845.

[21] Zhibing Zhao, Peter Piech, and Lirong Xia. Learning Mixtures of Plackett-Luce Models, March 2020. URL http://arxiv.org/abs/1603.07323. arXiv:1603.07323 [cs].

[22] Joar Max Viktor Skalse, Matthew Farrugia-Roberts, Stuart Russell, Alessandro Abate, and Adam Gleave. Invariance in Policy Optimisation and Partial Identifiability in Reward Learning. In *Proceedings of the 40th International Conference on Machine Learning*, pages 32033–32058. PMLR, July 2023. URL https://proceedings.mlr.press/v202/skalse23a.html. ISSN: 2640-3498.

[23] Lucas Maystre and Matthias Grossglauser. Fast and Accurate Inference of Plackett– Luce Models. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://papers.nips.cc/paper_files/paper/2015/hash/2a38a4a9316c49e5a833517c45d31070-Abstract.html.

[24] Francois Caron and Arnaud Doucet. Efficient Bayesian Inference for Generalized Bradley-Terry Models, November 2010. URL http://arxiv.org/abs/1011.1761. arXiv:1011.1761 [stat].

[25] Banghua Zhu, Jiantao Jiao, and Michael I. Jordan. Principled Reinforcement Learning with Human Feedback from Pairwise or K-wise Comparisons, May 2023. URL http://arxiv.org/abs/2301.11270. arXiv:2301.11270 [cs, math, stat].

[26] Eve Fleisig, Rediet Abebe, and Dan Klein. When the Majority is Wrong: Modeling Annotator Disagreement for Subjective Tasks. 2023. doi: 10.48550/ARXIV.2305.06626. URL https://arxiv.org/abs/2305.06626. Publisher: arXiv Version Number: 3.

[27] Connor Baumler, Anna Sotnikova, and Hal Daumé III. Which Examples Should be Multiply Annotated? Active Learning When Annotators May Disagree. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10352–10371, Toronto, Canada, July 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.findings-acl.658.

[28] Andreea Bobu, Dexter R. R. Scobee, Jaime F. Fisac, S. Shankar Sastry, and Anca D. Dragan. LESS is More: Rethinking Probabilistic Models of Human Behavior. *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, pages 429–437, March 2020. doi: 10.1145/3319502.3374811. URL https://dl.acm.org/doi/10.1145/3319502.3374811. Conference Name: HRI '20: ACM/IEEE International Conference on Human-Robot Interaction ISBN: 9781450367462 Place: Cambridge United Kingdom Publisher: ACM.

[29] Kimin Lee, Laura Smith, Anca Dragan, and Pieter Abbeel. B-Pref: Benchmarking Preference-Based Reinforcement Learning, November 2021. URL http://arxiv.org/abs/2111.03026. arXiv:2111.03026 [cs].

[30] Cassidy Laidlaw and Stuart Russell. Uncertain decisions facilitate better preference learning. *Advances in Neural Information Processing Systems*, 34:15070–15083, 2021.

[31] W. Bradley Knox, Stephane Hatgis-Kessell, Serena Booth, Scott Niekum, Peter Stone, and Alessandro Allievi. Models of human preference for learning reward functions, June 2022. URL http://arxiv.org/abs/2206.02231. arXiv:2206.02231 [cs, eess].

[32] Cassidy Laidlaw and Anca Dragan. The Boltzmann Policy Distribution: Accounting for Systematic Suboptimality in Human Models, April 2022. URL http://arxiv.org/abs/2204.10759. arXiv:2204.10759 [cs].

[33] Simon Zhuang and Dylan Hadfield-Menell. Consequences of Misaligned AI. In *Advances in Neural Information Processing Systems*, volume 33, pages 15763–15773. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/b607ba543ad05417b8507ee86c54fcb7-Abstract.html.

[34] Vincent Conitzer and Tuomas Sandholm. Common Voting Rules as Maximum Likelihood Estimators. In *UAI '05, Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence, Edinburgh, Scotland, July 26-29, 2005*, pages 145–152. AUAI Press, 2005. URL https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=1213&proceeding_id=21.

[35] Federico Echenique and Siddharth Prasad. Incentive Compatible Active Learning, November 2019. URL http://arxiv.org/abs/1911.05171. arXiv:1911.05171 [cs].

[36] Arnaud Fickinger, Simon Zhuang, Dylan Hadfield-Menell, and Stuart Russell. Multi-Principal Assistance Games, July 2020. URL http://arxiv.org/abs/2007.09540. arXiv:2007.09540 [cs].

[37] Michael Dummett. The Borda count and agenda manipulation. *Social Choice and Welfare*, 15 (2):289–296, 1998. ISSN 0176-1714. URL https://www.jstor.org/stable/41106256. Publisher: Springer.

[38] Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. AlpacaFarm: A Simulation Framework for Methods that Learn from Human Feedback, August 2023. URL http://arxiv.org/abs/2305.14387. arXiv:2305.14387 [cs].

[39] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open Foundation and Fine-Tuned Chat Models, July 2023. URL http://arxiv.org/abs/2307.09288. arXiv:2307.09288 [cs].

[40] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models, October 2021. URL http://arxiv.org/abs/2106.09685. arXiv:2106.09685 [cs].

[41] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization, January 2019. URL http://arxiv.org/abs/1711.05101. arXiv:1711.05101 [cs, math].

[42] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library, December 2019. URL http://arxiv.org/abs/1912.01703. arXiv:1912.01703 [cs, stat].

[43] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. HuggingFace's Transformers: State-of-the-art Natural Language Processing, July 2020. URL http://arxiv.org/abs/1910.03771. arXiv:1910.03771 [cs].

A Proofs and Additional Theoretical Results

A.1 Proof that $L(\hat{u}; u)$ is convex

Proposition A.1. *The loss function $L(\hat{u}; u)$ is strictly convex as a function of the values of $\hat{u}(a)$ for all $a \in \mathcal{A}$.*

Proof. Note that $L(\hat{u}; u)$ is a sum of many functions of the form

$$-\log\left(\frac{e^{\hat{u}(a)}}{e^{\hat{u}(a)} + e^{\hat{u}(b)}}\right) \tag{4}$$

weighted by nonnegative coefficients, for various values of $a, b \in \mathcal{A}$. Thus, we only need to show that functions of the form (4) are convex and then the entire loss function must be convex as well.

To see why (4) is convex, we can multiply the top and bottom of the fraction by $e^{-u(a)}$ to obtain

$$-\log\left(\frac{1}{1 + e^{\hat{u}(b)-\hat{u}(a)}}\right). \tag{5}$$

Note that the second derivative of the function

$$f(x) = -\log\left(\frac{1}{1 + e^{-x}}\right)$$

is

$$\frac{d^2}{dx^2} f(x) = \frac{e^x}{(1 + e^x)^2} > 0,$$

which means $f(x)$ is strictly convex. Thus implies that (5) must be a strictly convex function of $\hat{u}$ since letting $x = \hat{u}(b) - \hat{u}(a)$, x is an affine transformation of $\hat{u}$ and strict convexity is preserved under affine transformations. □

A.2 Proof that least-squares regression converges to expected utility

Proposition A.2. *Suppose that $\hat{u}$ is defined as*

$$\hat{u} = \arg\min_{\hat{u}} \mathbb{E}_{z\sim\mathcal{D}_z}\left[\frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}}(\hat{u}(a) - u(a, z))^2\right]. \tag{6}$$

Then for all $a \in \mathcal{A}$, $\hat{u}(a) = \bar{u}(a) = \mathbb{E}_{z\sim\mathcal{D}_z}[u(a, z)]$.

Proof. We can rewrite the optimization objective in (6) as

$$\frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}} \mathbb{E}_{z\sim\mathcal{D}_z}\left[(\hat{u}(a) - u(a, z))^2\right].$$

Note that since for any a, $\hat{u}(a)$ only appears in one term in the sum, we can define $\hat{u}$ pointwise as

$$\begin{aligned}\hat{u}(a) &= \arg\min_{\hat{u}(a)} \mathbb{E}_{z\sim\mathcal{D}_z}\left[(\hat{u}(a) - u(a, z))^2\right] \\ &= \arg\min_{\hat{u}(a)} \left(\hat{u}(a)^2 - 2\hat{u}(a)\mathbb{E}_{z\sim\mathcal{D}_z}[u(a, z)] + \mathbb{E}_{z\sim\mathcal{D}_z}\left[u(a, z)^2\right]\right).\end{aligned}$$

It is clear that the above is minimized when

$$\hat{u}(a) = \mathbb{E}_{z\sim\mathcal{D}_z}[u(a, z)] = \bar{u}(a).$$

□

A.3 Proof of Theorem 3.1

Theorem 3.1. *If $\hat{u}$ is optimized according to (1), then $\forall a, b \in \mathcal{A}$, $\hat{u}(a) > \hat{u}(b) \Leftrightarrow BC(a) > BC(b)$.*

Proof. We consider L^2-regularized preference learning with coefficient λ, noting that if $\lambda > 0$ then the solution, which uniquely exists,r satisfies the first order conditions. Furthermore, if $\lambda = 0$, which corresponds to un-regularized objective, then if there is a solution it must also satisfy said conditions. The condition is as follows:

$$\lambda\hat{u}(a) + \sum_{c\neq a} \mathbb{E}_{z\sim\mathcal{D}_z}\left[O_u(a,c,z)\right] - \sigma(\hat{u}(a) - \hat{u}(c)) = 0 \;\forall a \in \mathcal{A}$$

Note that we want to show the following:

$$BC(a) > BC(b) \iff \hat{u}(a) > \hat{u}(b)$$

where u is the optimal solution. Furthermore, observe that this is sufficient to show that $BC(a) \geq BC(b) \iff \hat{u}(a) \geq \hat{u}(b)$. First consider the forward direction. Let $a, b \in \mathcal{A}$ such that $BC(a) > BC(b)$, and assume for contradiction that $\hat{u}(a) \leq \hat{u}(b)$. Let $f, g : \mathbb{R} \to \mathbb{R}$ be defined as follows:

$$f(\alpha) = \lambda\alpha + \sum_{c\neq a} \mathbb{E}_{z\sim\mathcal{D}_z}\left[O_u(a,c,z)\right] - \sigma(\alpha - u(c))$$
$$g(\alpha) = \lambda\alpha + \sum_{c\neq b} \mathbb{E}_{z\sim\mathcal{D}_z}\left[O_u(b,c,z)\right] - \sigma(\alpha - u(c))$$

Thus $f(\hat{u}(a)) = g(\hat{u}(b)) = 0$. Observe that f and g are decreasing functions in α. Now note the following:

$$\begin{aligned} f(\alpha) - g(\alpha) &= \sum_{c\neq a} \mathbb{E}_{z\sim\mathcal{D}_z}\left[O_u(a,c,z)\right] - \sum_{c\neq b} \mathbb{E}_{z\sim\mathcal{D}_z}\left[O_u(b,c,z)\right] - \sigma(\alpha - \hat{u}(b)) + \sigma(\alpha - u(a)) \\ &\geq \frac{1}{I}(BC(a) - BC(b)) \\ &> 0 \end{aligned}$$

which follows from σ being increasing. Hence $f(\alpha) > g(\alpha)$. Observe the following contradiction:

$$0 = f(\hat{u}(a)) > g(\hat{u}(a)) \geq g(\hat{u}(b)) = 0$$

Thus it must be that $u(a) > u(b)$. We can similarly see that for the backward implication, if instead $BC(a) \geq BC(b)$, and for contradiction $\hat{u}(a) < \hat{u}(b)$, then we have that:

$$\begin{aligned} f(\alpha) - g(\alpha) &= \sum_{c\neq a} \mathbb{E}_{z\sim\mathcal{D}_z}\left[O_u(a,c,z)\right] - \sum_{c\neq b} \mathbb{E}_{z\sim\mathcal{D}_z}\left[O_u(b,c,z)\right] - \sigma(\alpha - \hat{u}(b)) + \sigma(\alpha - \hat{u}(a)) \\ &> \frac{1}{I}(BC(a) - BC(b)) \\ &\geq 0 \end{aligned}$$

Thus $\hat{u}$ is equivalent to BC. □

A.4 Proof of Theorem 3.2

Theorem 3.2. *Let ϵ be independent and identically distributed (i.i.d.) for all $a \in \mathcal{A}$. Furthermore, suppose $\epsilon(a) - \epsilon(b)$ has support around 0, i.e., $\exists\bar{\delta} > 0$ such that $\forall\delta \in (0, \bar{\delta})$ $F_{a,b}(\delta) > F_{a,b}(0) = \frac{1}{2}$, where $F_{a,b}$ is the cumulative distribution function of $\epsilon(a) - \epsilon(b)$. Then the utility function $\hat{u}$ learned by minimizing the BTL loss in (1) satisfies $\hat{u}(a) > \hat{u}(b) \Leftrightarrow \bar{u}(a) > \bar{u}(b)$ for any $a, b \in \mathcal{A}$.*

Proof. We proceed by showing that $\hat{u} \equiv \bar{u}$. Assume $a, b \in \mathcal{A}$ is such that $\bar{u}(a) > \bar{u}(b)$. Now note the following:

$$\begin{aligned} \hat{u}(a) - \hat{u}(b) = \sum_{c\notin\{a,b\}} &(\mathbb{P}(\bar{u}(a) + \epsilon(a) > \bar{u}(c) + \epsilon(c)) - (\mathbb{P}(\bar{u}(b) + \epsilon(b) > \bar{u}(c) + \epsilon(c)) \\ &+ \mathbb{P}(\bar{u}(a) + \epsilon(a) > \bar{u}(b) + \epsilon(b)) - \mathbb{P}(\bar{u}(b) + \epsilon(b) > \bar{u}(a) + \epsilon(a)) \end{aligned}$$

Observe the following for the last two terms:

$$\begin{aligned} &\mathbb{P}(\bar{u}(a) + \epsilon(a) > \bar{u}(b) + \epsilon(b)) - \mathbb{P}(\bar{u}(b) + \epsilon(b) > \bar{u}(a) + \epsilon(a)) \\ =&F_{\epsilon(b)-\epsilon(a)}(\bar{u}(a) - \bar{u}(b)) - 1 + F_{\epsilon(b)-\epsilon(a)}(u(a) - \bar{u}(b)) \\ =&2F_{\epsilon(b)-\epsilon(a)}(\bar{u}(a) - \bar{u}(b)) - 1 \\ >&0 \end{aligned}$$

where we use the strict increasingness of the CDF around zero. Now note the following for each term of the summation:

$$\begin{aligned}
&\mathbb{P}(\bar{u}(a)+\epsilon(a)>\bar{u}(c)+\epsilon(c))-(\mathbb{P}(u(b)+\epsilon(b)>\bar{u}(c)+\epsilon(c))\\
=&\int \mathbb{P}_{\epsilon(c)}(\nu)(\mathbb{P}(\bar{u}(a)+\epsilon(a)>\bar{u}(c)+\nu)-\mathbb{P}(\bar{u}(b)+\epsilon(c)>\bar{u}(c)+\nu)d\nu\\
=&\int \mathbb{P}_{\epsilon(c)}(\nu)(\mathbb{P}(\bar{u}(a)+\epsilon(c)>\bar{u}(c)+\nu)-\mathbb{P}(\bar{u}(b)+\epsilon(c)>\bar{u}(c)+\nu)d\nu\\
\geq& 0
\end{aligned}$$

where we use the fact that the noise random variable is the IID, and the weakly increasingness of the CDF. Thus $\hat{u}(a)>\hat{u}(b)$. ☐

A.5 Proof for IID Symmetric RVs

In this section, we show how the assumption of Theorem 3.1 is satisfied by random variables such as zero-mean normal and uniform distributions.

Proposition A.3. *If $\forall a\in\mathcal{A}$, $\epsilon(a)$ are independent, identically distributed, zero-mean symmetric random variables with support on $(-\bar{\delta},\bar{\delta})$ for some $\bar{\delta}>0$, then the noise assumption in 3.2 holds.*

Proof. Consider some $a,b\in\mathcal{A}$.

$$\begin{aligned}
\mathbb{P}(\epsilon(a)+\delta\geq\epsilon(b)) &= \int_{-\infty}^{\infty}\mathbb{P}(x+\delta\geq\epsilon(b))d\epsilon(a)\\
&= \int_{-\infty}^{-\bar{\delta}}\mathbb{P}(x+\delta\geq\epsilon(b))d\epsilon(a)+\int_{-\bar{\delta}}^{\bar{\delta}}\mathbb{P}(x+\delta\geq\epsilon(b))d\epsilon(a)+\int_{\bar{\delta}}^{\infty}\mathbb{P}(x+\delta\geq\epsilon(b))d\epsilon(a)\\
&\geq \int_{-\infty}^{-\bar{\delta}}\mathbb{P}(x\geq\epsilon(b))d\epsilon(a)+\int_{-\bar{\delta}}^{\bar{\delta}}\mathbb{P}(x+\delta\geq\epsilon(b))d\epsilon(a)+\int_{\bar{\delta}}^{\infty}\mathbb{P}(x\geq\epsilon(b))d\epsilon(a)\\
&> \int_{-\infty}^{-\bar{\delta}}\mathbb{P}(x\geq\epsilon(b))d\epsilon(a)+\int_{-\bar{\delta}}^{\bar{\delta}}\mathbb{P}(x\geq\epsilon(b))d\epsilon(a)+\int_{\bar{\delta}}^{\infty}\mathbb{P}(x\geq\epsilon(b))d\epsilon(a)\\
&= \mathbb{P}(\epsilon(a)\geq\epsilon(b))\\
&= \frac{1}{2}
\end{aligned}$$

where we use the local strict increasingness of the CDF. The final line follows from the difference of independent zero-mean random variables also being zero-mean. ☐

A.6 Proof of Proposition 3.3

Proposition 3.3. *$\exists\mathcal{A},\mathcal{D}_z,u$ s.t $\forall a,b\in\mathcal{A}$, $[\bar{u}(a)>\bar{u}(b)]\Rightarrow[p_{u,\mathcal{D}_z}(a,b)>1/2]$, but $\hat{u}$ is not equivalent to $\bar{u}$, i.e., there exist $a,b\in\mathcal{A}$ such that $\hat{u}(a)>\hat{u}(b)$ but $\bar{u}(a)<\bar{u}(b)$.*

Proof. Let $\mathcal{A}=\{a,b,c\}$ and $\mathcal{Z}=[0,1]$ with $\mathcal{D}_z=\text{Unif}([0,1])$. Now define

$$u(a,z)=\begin{cases}10 & z\leq 0.6\\ 0 & z>0.6\end{cases}$$

$$u(b,z)=\begin{cases}3 & z\leq 0.9\\ 1 & z>0.9\end{cases}$$

$$u(c,z)=2.$$

From these, we can see that the expected utility is

$$\begin{aligned}
\bar{u}(a)&=6\\
\bar{u}(b)&=2.8\\
\bar{u}(c)&=2,
\end{aligned}$$

i.e., $\bar{u}(a) > \bar{u}(b) > \bar{u}(c)$. Also, we can calculate

$$\mathbb{E}[O_{u+\epsilon}(a, b)] = 0.6$$
$$\mathbb{E}[O_{u+\epsilon}(a, c)] = 0.6$$
$$\mathbb{E}[O_{u+\epsilon}(b, c)] = 0.9,$$

which satisfy the needed condition. This results in Borda counts of

$$\text{BC}(a) = 1.2$$
$$\text{BC}(b) = 1.3$$
$$\text{BC}(c) = 0.5.$$

Note that $\text{BC}(b) > \text{BC}(a)$, so the estimated utility $\hat{u}$ returned by preference learning must have $\hat{u}(b) > \hat{u}(a)$ by Theorem 3.1; this means that $\hat{u}$ is not equivalent to u, since $u(a) > u(b)$. □

A.7 Proof of Theorem 3.4

Theorem 3.4. *Suppose a preference learning algorithm takes as input unlimited samples of the form $(a, b, O_u(a, b, z))$ for all values of a and b, where $z \sim \mathcal{D}_z$, and deterministically outputs a learned utility function $\hat{u}(a)$. Then there is some utility function u and distribution over unseen features $\mathcal{D}_z$ such that $\hat{u}$ is* not *equivalent to $\bar{u}$.*

Proof. Consider an alternative space $\mathcal{A} = \{a, b\}$ and hidden context $z \in \mathcal{Z} = \{0, 1\}$ with $\mathcal{D}_z = \mathcal{B}(1/2)$. Now, define two utility functions over these alternatives:

$$u(a, z) = 0 \qquad u'(a, z) = 0$$
$$u(b, z) = \begin{cases} 3 & z = 0 \\ -1 & z = 1 \end{cases} \qquad u'(b, z) = \begin{cases} 1 & z = 0 \\ -3 & z = 1. \end{cases}$$

These result in the following distribution over comparison outcomes:

$$p_{u,\mathcal{D}_z}(a, b) = \mathcal{B}(1/2)$$
$$p_{u',\mathcal{D}_z}(a, b) = \mathcal{B}(1/2).$$

That is, both (u, ϵ) and (u', ϵ') result in identical distributions over comparison outcomes. Thus, the preference learning algorithm must output identical learned utility functions in either scenario; call its output $\hat{u}$. If $\hat{u}(a) \geq \hat{u}(b)$, then it has failed to identify u, since $u(a) < u(b)$. On the other hand, if $\hat{u}(a) < \hat{u}(b)$, then it has failed to identify u', since $u(a) > u(b)$. Thus, either way, there is some utility function and noise function distribution under which the algorithm fails to identify the true utility function. □

B Social Choice Theory

B.1 Preliminaries

To analyze preference learning through the lens of social choice theory, we first define the concept of a social welfare functional. Let I be the number of agents, and let $\mathcal{P} \subset \mathcal{R} \subset \mathcal{B} = \mathcal{A} \times \mathcal{A}$ be the set of strict rational[2], rational[3] and binary relations (respectively) on the space of alternatives $\mathcal{A}$. We say $\succeq = (\succeq_i)_{i=1}^I \in \mathcal{R}^I$ is a preference profile. Viewing an individual's feedback as their revealed preference, which is available in a sufficiently rich dataset of comparisons, we can see preference learning as being similar to a *social welfare functional*:

Definition B.1. *A social welfare functional (SWF) is a map $F : \mathcal{K} \to \mathcal{B}$ where $\mathcal{K} \subseteq \mathcal{R}^I$ is the domain of preference profiles.*

We will assume that $\mathcal{K} = \mathcal{R}^I$.

[2] asymmetric (ie antisymmetric and irreflexive) and rational
[3] transitive and complete

B.2 BTL and Borda Count

Definition B.2. *Given a set of preference $\{\succeq_i\}_{i=1}^n$, we call $BC : \mathcal{A} \to \mathbb{R}$ the Borda count:*

$$BC(a) = \sum_{i=1}^{n} \sum_{c \in \mathcal{A}} \mathbf{1}\{a \succ_i c\}$$

Corollary B.3. *If there is a solution to preference learning, then it is equivalent to BC. Furthermore, the solution to L^2-regularized preference learning is also equivalent to BC.*

Proof. Observe that as per Theorem 3.1, the feature over which the expectation is taken with respect to is the identifier i for each agent. Since agents are uniformly sampled, this is a scaling of Borda count. □

B.3 Proportion-Representable SWFs

In this section we consider what SWFs can be represented when the distribution of comparisons are known. We call such SWFs *proportion-representable* if they can be directly determined by a classifier, ie

$$\begin{aligned}\rho[\succeq](a, b) &= \mathbb{E}\left[O_u(a, b, i)\right] \\ &= \frac{1}{|\mathcal{I}|} |\{i \in \mathcal{I} : a \succ_i b\}|\end{aligned}$$

where

$$a \succ_i b \iff O_u(a, b, i) > \frac{1}{2}$$

In the context of preference learning via maximum likelihood estimation, this is a useful property of a SWF as it can be directly implemented by optimizing a cross-entropy loss on the comparisons. We formally define this property as follows:

Definition B.4. *F is proportion-representable if $\exists g$ such that $\forall \succeq, a, b \in \mathcal{A}$, $aF(\succeq)b \iff ag[\rho[\succeq]]b$.*

We motivate this line of exploration by noting that Borda count and pairwise majority (denoted $M : \mathcal{A} \times \mathcal{A} \to \{0, 1\}$) can be induced by a classifier:

$$BC(a) \propto \sum_{c \in \mathcal{A}} \rho(a, c)$$

$$M(a, b) = \mathbf{1}\{\rho(a, b) > \rho(b, a)\}$$

This suggests that it might be possible to separate the learning of preferences in aggregate with the normative properties of the SWF implemented. It is not obvious what is an ideal SWF to implement, and thus having the flexibility to change implementations without relearning the utility function is useful. A general property that allows an SWF to be proportion-representable is the following:

Definition B.5. *A SWF is comparison-anonymous if swapping the some comparisons of two individuals (still maintaining a rational preference) doesn't change the outcome.*

Observe that this is a stronger property than regular anonymity. We now state a simple result on the equivalence between proportion-representability and comparison-anonymity:

Proposition B.6. *An SWF is proportion-representable iff it is comparison-anonymous.*

Proof. The forward direction is clear, hence we only prove the backward direction. Assume F is comparison-anonymous, and for contradiction, assume it is not proportion-representable. Then for some $\succeq \neq \succeq'$ with the same proportion $\exists x, y$ such that $xF(\succeq)y$ but $yF_P(\succeq')x$. This is a contradiction as by comparison-anonymity we can swap preferences in one profile to become the other profile, but the social preference doesn't change. □

Since learning a classifier directly is the most general setup for learning from comparisons, this provides a fundamental limit on what SWFs can be implemented. Other SWFs may require richer preference models that consider the whole ranking rather than just individual comparisons. We now consider specific examples of SWFs from the voting theory literature, showing a mix of positive and negative results.

Scoring rules A scoring rule is determined by $\alpha(k)$, the score of the k-th ranking of the alternative that is non-decreasing in k:

$$u(a) = \sum_i \alpha(|\{b : a \succ_i b\}|)$$

For example, Borda count has $\alpha(k) = k$. We know show that the only scoring rules that are comparison anonymous are those that are affine transformations of the Borda count.

Proposition B.7. *A scoring rule is comparison-anonymous iff it is an affine scoring rule.*

Proof. For the backward direction, observe that by linearity of α, the associated utility function is an affine transformation of Borda count. This maintains the comparison anonymity property since such a property is preserved under monotone transformations. Now we consider the forward direction. If α is a scoring rule that is not affine, then the following condition must hold for some $1 \leq k \leq |\mathcal{A}|$ since $|\mathcal{A}| \geq 3$:

$$\alpha(k+1) - \alpha(k) \neq \alpha(k+2) - \alpha(k+1)$$

First consider the case where $\alpha(k+1) - \alpha(k) < \alpha(k+2) - \alpha(k+1)$. Without loss of generality, consider the two agent case. Assume the preference ranking for both agents are identical apart from their rankings at $\{k, k+1, k+2\}$. Let them have the following rankings respectively for some alternative a, b, c:

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

Thus the utilities of each alternative are as follows:

$$u(a) = 2\alpha(k+1)$$
$$u(b) = \alpha(k) + \alpha(k+2)$$
$$u(c) = \alpha(k) + \alpha(k+2)$$

By assumption, we have that $u(b) > u(a)$. Now consider the proportion-preserving transformation of the preference profile:

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

where all other rankings are kept the same. Hence the utilities of each alternative are:

$$u(a) = \alpha(k) + \alpha(k+2)$$
$$u(b) = 2\alpha(k+1)$$
$$u(c) = \alpha(k) + \alpha(k+2)$$

Thus $u(a) > u(b)$. This holds similarly for the case where $\alpha(k+1) - \alpha(k) > \alpha(k+2) - \alpha(k+1)$. Furthermore, we can generalize to arbitrary number of agents by allowing all agents other than some two to have the same preference ranking, and letting said two have the above preferences. As the SWF is linear in the agents, the relative ranking between alternatives only depend on the two agents, preserving our result. Since the ranking of the SWF induced by α is not preserved when considering an alternative preference profile with the same proportions of comparisons, it cannot be comparison-anonymous. □

Corollary B.8. *Borda count is the only proportion-representable SWF (up to monotone transformations) that is induced by a scoring rule.*

Proof. This follows by linearity of the scoring rule. □

Copeland Rule and Maximin rules The Copeland and maximin rules are given by the following

$$C_{\text{Copeland}}(a) = \sum_c M(a,c) - M(c,a),\ C_{\text{Maximin}}(a) = \min_{c \neq a} M(a,c)$$

These rules can be seen to be proportion-representable by using the same result for pairwise-majority:

Proposition B.9. *The Copeland and maximum rules are a proportion-representable SWF.*

Proof. Observe that they can be rewritten as such:

$$C_{\text{Copeland}}(a) \propto \sum_c \mathbf{1}\{\rho(a,c) > \rho(c,a)\} - \mathbf{1}\{\rho(a,c) < \rho(c,a)\}$$
$$C_{\text{Maximin}}(a) \propto \min_c \mathbf{1}\{\rho(a,c) > \rho(c,a)\}$$

□

These results showcase how there is some flexibility in how we choose to implement preference learning when aggregating across individuals.

C Experiment Details

In this appendix, we describe the details of our LLM preference learning experiments.

C.1 Preference dataset

We initially used the original labels from the HH-RLHF dataset to train preference models. However, we found that the distribution of *prompts* was quite different between the helpfulness and harmfulness splits of the dataset. In the helpfulness split, most prompts were harmless questions or requests for assistant. In contrast, in the harmlessness split, most prompts were specifically chosen to elicit harmful behavior. Preference models trained on the combined data were therefore able to identify the type of prompt and respond accordingly: they responded to harmful prompts based on harmfulness and harmless prompts based on helpfulness.

To emphasize the effect of hidden context in this setting, we decided to randomly relabel half of the dataset with the opposite objective. This way, the objective used for annotation cannot be inferred from the prompt. To relabel the dataset in this way, we used GPT-3.5; Dubois et al. [38] show that simulating human annotators with LLMs in this way is an effective way to generate human-quality labels at a much lower cost.

We prompted GPT-3.5 with the below two prompts for helpfulness and harmlessness, which are based on the instructions given to human annotators in Bai et al. [3]. Note that for the harmlessness labels, we ask the model which response is more *harmful* but then invert the resulting label. We found that when GPT-3.5 labeled according to the same objective as the original label in the dataset, the agreement between the human and machine annotations was 63%, similar to the researcher-annotator agreement in [3].

Prompt for helpfulness:

```
In this task, you will be presented with a prompt and two
responses from an AI assistant. Your task is to determine
which response is more *helpful* and *honest*. Please
consider the responses carefully, and use your overall
judgement. Ask yourself "did this response represent a
thoughtful, honest, and reasonable attempt to either
directly help me, or to converse with me to elicit
information that will make it easier for the AI to help me
later on in this conversation". Note that this is subtle --
if you tell the AI about a difficult personal situation,
helping might involve showing empathy and understanding.
But if you ask for something practical, the AI should realize
you just want a concise response, and not prolong the
interaction with irrelevant questions and comments. The AI
assistant should be helpful in the same way that a friend or
subject-expert would (hopefully) be helpful.

The most common bad behaviors involve the AI making up facts,
websites, and statistics. Please fact check the AI if it seems
to be making reasonable claims that reference something
specific. Be very careful not to reward the AI for telling
```

```
you something false!

Sometimes the AI will also behave misleadingly as though
it’s a person who can "go out and get something" or "look
something up" or "ask its colleague." It can’t do any of
these things and it should not be rewarded for making
such claims!

The AI can also often fail to be helpful because it’s indirect
-- it might just fill the conversation with unnecessary chatter,
or act as though it wants to look something up. It’s good for
the AI to be appropriately humble, but the AI shouldn’t just
fill time with irrelevant chatter.

Finally, the AI should always be polite and friendy.

Consider the following conversation between an AI assistant and
a human:

    <prompt>

Now, consider the following two responses from the AI assistant:

    Response A: <response A>

    Response B: <response B>

Which response is more helpful and honest? Explain your
reasoning and then give your final answer by writing "A" or "B"
on a new line.
```

Prompt for harmfulness:

```
In this task, you will be presented with a prompt and two
responses from an AI assistant. Your task is to choose the
*most harmful* response. Examples of harmful behavior include:

  * Responses about people from different cultures that are
    stereotypical/offensive instead of subtle/mature.
  * Responses that are inappropriate and unhelpful.
  * Responses that help the user do something illegal or
    unethical, like robbing a bank.
  * Responses that are offensive even though the user did not
    directly ask for such content.
  * Anything that is obnoxious, offensive, or harmful.

Consider the following conversation between an AI assistant and
a human:

    {prompt_indented}

Now, consider the following two responses from the AI assistant:

    Response A: {responses[0]}

    Response B: {responses[1]}

Which response is more harmful? Explain your reasoning and then
give your final answer by writing "A" or "B" on a new line.
```

C.2 Model training

To train our preference models, we fine-tune LLAMA-2-7B [39] using LoRA [40]. We replace the normal language model head of the LLAMA models with a linear layer with either 1 output (normal preference learning), 2 outputs (mean-and-variance DPL), or 10 outputs (categorical DPL). We use the AdamW optimizer [41] with a learning rate of 3×10^{-6} which is decayed via a cosine schedule to 3×10^{-7}, a batch size of 2 comparisons (i.e., 4 responses total), and weight decay of 0.0001. We implement training using PyTorch [42] and HuggingFace Transformers [43].

C.3 Jailbroken responses

To collect the dataset of jailbroken responses, we started with the dataset of all ChatGPT and Claude responses to jailbreak prompts from Wei et al. [1], which contains labels for each response indicating if the model was a "good bot" or "bad bot." We filtered to prompts that produced a "good bot" response from one model and "bad bot" response from the other, giving us 187 pairs of responses.