

Detecting Prefix Bias in LLM-based Reward Models

Ashwin Kumar^{*†}
Washington University in St Louis
St Louis, USA
ashwinkumar@wustl.edu

Yuzi He^{*}
Meta Platforms, Inc.
Menlo Park, USA
yuzihe12@gmail.com

Aram H. Markosyan
Meta Platforms, Inc.
Menlo Park, USA
aram.math@gmail.com

Bobbie Chern
Meta Platforms, Inc.
Sunnyvale, USA
bgchern@meta.com

Imanol Arrieta-Ibarra[‡]
Independent
San Mateo, USA
imanol.arrieta.ibarra@gmail.com

Abstract

Reinforcement Learning with Human Feedback (RLHF) has emerged as a key paradigm for task-specific fine-tuning of language models using human preference data. While numerous publicly available preference datasets provide pairwise comparisons of responses, the potential for biases in the resulting reward models remains underexplored. In this work, we introduce novel methods to detect and evaluate prefix bias—a systematic shift in model preferences triggered by minor variations in query prefixes—in LLM-based reward models trained on such datasets. We leverage these metrics to reveal significant biases in preference models across racial and gender dimensions. Our comprehensive evaluation spans diverse open-source preference datasets and reward model architectures, demonstrating susceptibility to this kind of bias regardless of the underlying model architecture. Furthermore, we propose a data augmentation strategy to mitigate these biases, showing its effectiveness in reducing the impact of prefix bias. Our findings highlight the critical need for bias-aware dataset design and evaluation in developing fair and reliable reward models, contributing to the broader discourse on fairness in AI.

CCS Concepts

• **Computing methodologies** → *Natural language processing*; Learning from demonstrations.

Keywords

Bias, Reinforcement Learning from Human Feedback, LLM Fine-tuning, Reward Models

ACM Reference Format:

Ashwin Kumar, Yuzi He, Aram H. Markosyan, Bobbie Chern, and Imanol Arrieta-Ibarra. 2025. Detecting Prefix Bias in LLM-based Reward Models. In

^{*}Corresponding Author

[†]Work done as an intern at Meta Platforms, Inc.

[‡]Work done while at Meta Platforms, Inc.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

FACCT '25, Athens, Greece

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-XXXX-X/2018/06

<https://doi.org/10.1145/3715275.3732204>

Proceedings of ACM Conference on Fairness, Accountability and Transparency (FAccT '25). ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3715275.3732204>

1 Introduction

Large Language Models (LLMs) have seen a large growth in research and usage [1, 26]. Their ability to understand context and generate and analyze text has proven their utility in applications from code completion [20] to medical diagnosis [23]. This is thanks to the large amounts of pre-training that models undergo, which deployers can leverage for different use cases. However, to efficiently use LLMs for different tasks, we need to fine-tune them on task-specific information [2, 18, 24].

This finetuning can be done in multiple ways, the simplest being Supervised Fine Tuning (SFT), where domain data is used to train the model with next-token prediction. Reinforcement Learning with Human Feedback (RLHF) is a paradigm popularized by the Instruct-GPT paper [18], and further used by Anthropic [2], Llama [25, 26] and other popular language models. When using RLHF, a preference model or Reward Model (RM) is trained to discriminate between pairwise responses to an input prompt to give higher scores for responses that are *preferred* by human reviewers [18]. These reward models are used for reinforcement learning to guide fine-tuning, using algorithms like proximal policy optimization (PPO) [21].

There have been studies [3, 10, 22] looking at bias in language models. However, to the best of our knowledge, no one has studied bias in reward models in the context of RLHF fine-tuning. Further, there has been limited study of *prefix bias*, where changing the prefix to a prompt can change the model's behavior. RMs are typically used to evaluate prompt-response pairs and assign a value to them, which then trains the finetuned model to prefer the response that receives a higher reward. If the reward model's preferences are biased, it may poison the downstream model and cause undesirable learned behavior. Thus, it is important to create ways to identify and mitigate bias in the reward model stage.

In this paper, we study bias in reward models by explicitly conditioning candidate responses on different demographic identifiers using short textual prefixes. Our contributions are as follows:

- (1) We design **auto-influence** and **cross-influence**, two methods for evaluating the prefix-bias in a learned reward model. We also present **winrate deviation** and **accuracy deviation**, metrics for quantifying auto-influence and cross-influence respectively.

- (2) We train and evaluate reward models on a variety of open datasets and show the presence of racial and gender bias using these metrics. We also use different model architectures and show that this bias is present for all of them, suggesting that the bias is learned from the datasets.
- (3) We conduct experiments to identify the source of the bias and uncover patterns based on the training dataset used, and present a solution using data augmentation to fix this issue.

2 Related Work

Bias in LLMs: The study of bias in LLMs has become a pressing and active field of research [10, 12, 14, 17], driven by their widespread use in sensitive domains. Early work focused on how training data can encode systemic biases [3, 7], while more recent research has highlighted specific downstream harms, such as racial disparities in clinical advice [17] and gendered patterns in name-based completions [12, 14]. A recent survey [10] provides a broad overview of these challenges. Prefix bias has also been studied in recent work; for instance, Chaudhary et al. [5] use prefix perturbations to measure LLM robustness in a certification setting, but they do not consider the reward modeling stage.

Several recent studies have taken an identity-grounded approach to evaluating bias in LLMs and reward models. Kantharuban et al. [13] find that chatbots often reflect racial stereotypes based on inferred user identity, blurring the line between personalization and harm. Eloundou et al. [8] introduce the concept of *first-person fairness*, showing how name-based probes can surface demographic biases in chatbot behavior. These works underscore the importance of evaluating LLM behavior through demographic context. However, in this paper, we focus on developing metrics to evaluate and quantify prefix bias in Reward Models. We explore the biases that may exist and appear in the Reward Model stage, which can introduce biases to the fine-tuned model due to an inaccurate learned reward function. As such, this paper is orthogonal to the rich field of analyzing bias in pre-trained and deployed LLMs.

Language Model Fine-tuning: Trained on vast corpora, language models also possess vast information. However, to constrain their outputs to suit a particular use case, and to ensure their outputs are aligned with our expectations, LLMs need to be fine-tuned on curated data [2, 18, 24]. Reinforcement Learning from Human Feedback (RLHF) [2, 18] is a popular technique for achieving LLM alignment. RLHF aims to collect human preferences in terms of pairwise rankings, which are then used to learn a preference model or Reward Model (RM). The RM is then used to fine-tune the LLM using Proximal Policy Optimization (PPO) [21], which is a popular reinforcement learning algorithm. Reward models may also be used for best-of-n (BoN) sampling of the model outputs. While there exist other finetuning methods like supervised finetuning (SFT) and direct preference optimization (DPO) [19], we focus on methods that rely on reward models, because of their widespread use.

Reward Models: It is known that reward models can have generalization and misspecification issues [4]. Recent work on reward model overoptimization [11] has shown that larger reward models exhibit lower susceptibility to reward hacking, and increasing training data reduces overoptimization. This study considered reward

models with up to 3 Billion parameters. A follow-up paper showed that using ensembling in reward models helps to reduce overoptimization [6]. However, these papers do not consider any bias resulting from overoptimization. In a more related vein, Mire et al. [16] demonstrated that reward models penalize African American Language (AAL), leading to representational disparities.

To the best of our knowledge, bias in LLM-based reward models using prefix-based attacks has not been studied, and our paper is the first to show its existence when training popular open-weight language models [15, 25–28] on popular RLHF datasets [2, 9].

3 Learning Reward Models

Reward models are an integral part of reinforcement learning systems. They endow the models with the ability to reason about their actions and provide an optimization target. A reward model takes as input a combination of states and actions and tells the RL agent the associated reward. Since the objective of RL is typically to maximize the total or expected reward, agents learn to take actions with the best-expected returns given the current state. If the reward is misspecified, the policy may not reflect the intended behavior. In the context of LLM fine-tuning, the RM is intended to provide feedback about the LLM’s generated output, and RLHF aims to use the RM to make the model learn to produce output that has a higher reward.

The reward model is initialized using a pre-trained LLM with an added value head. The RM then takes a sequence as input and predicts a scalar reward for it. To learn a reward model, the technique proposed by InstructGPT [18] is to get human reviewers to rank a set of responses to an input based on the criteria of choice (e.g. helpfulness, harmlessness, instruction following). Then, using this ranking, we can learn a reward model by minimizing the loss in Equation 1, where σ is the sigmoid function.

$$\mathcal{L} = -\log(\sigma(S(c) - S(r))) \quad (1)$$

Here, $S(t)$ is the reward model’s score for text t , and c and r are the chosen (preferred) response text and rejected (less preferred) response text respectively, based on the human reviewer’s ranking. We describe how c and r are constructed in Section 3.1.

3.1 Using Reward Models for Evaluation

In this section, we describe our preprocessing steps and evaluation methodology. Reward models are used to compare different responses to the same query. Given a dataset D , we define an input data point as the tuple $D_i = \langle q, a_1, a_2 \rangle$ containing a query q and two responses a_1 and a_2 . Without loss of generality, we use the convention that a_1 is the preferred/better response. Many RLHF datasets [2, 9] datasets are available in this format. For training and evaluation using such datasets, we preprocess the dataset using the following template:

$$T(q, a_1, a_2) = \text{“Prompt:”} + q + \text{“Response1:”} + a_1 + \quad (2)$$

$$\text{“Response2:”} + a_2 + \quad (3)$$

$$\text{“Is response 1 better than response 2? A:”} \quad (4)$$

This template, formatted as a natural language question, leverages the reward model’s linguistic capabilities. However, the

model’s output ($S(t)$) remains a scalar value. Sequences c (chosen) and r (rejected) are constructed for each data point by flipping the order of responses to mitigate input-order bias:

$$c = T(q, a_1, a_2) \quad (5)$$

$$r = T(q, a_2, a_1) \quad (6)$$

Then, $S(c) - S(r)$ is the reward model’s preference for a_1 versus a_2 . We then compute the scores c and r and plug them into the loss (Eq.1) to train the reward model. This technique of explicitly asking the model to compare both responses results in a better agreement rate compared to directly scoring each response and can be adapted during PPO training by making a_2 be the empty string to get independent scores [9]¹.

To measure how well a reward model performs, we use the **agreement rate**, which encodes how often the reward model’s preferences align with the human preferences. We concisely represent the above process for a trained reward model M as $M(q, a_1, a_2)$, a shorthand for preprocessing the sample, computing rewards for each input and comparing them (Eq.7). This also serves as the accuracy function.

$$M(q, a_1, a_2) = \mathbb{I}[S(c) > S(r)] \quad (7)$$

4 Evaluating Prefix Bias in Trained Reward Models

Reward models are intended to assign higher scores to responses that better align with human preferences, ideally focusing on the semantic quality of those responses. In this work, we investigate whether reward models’ preferences can systematically shift when the same response is framed with different demographic indicators—such as race or gender—even when the underlying content remains unchanged.

To test this, we introduce a controlled intervention: we prepend short identity-related prefixes (e.g., “I am a woman.” or “I am a man.”) to responses before passing them to the reward model. These prefixes serve as stand-ins for demographic context. They do not reflect how users typically phrase inputs or how RLHF data is collected, but allow us to isolate whether the reward model’s scoring function changes based solely on these surface-level cues.

While this setup uses explicit markers, similar forms of demographic information—whether mentioned earlier in a conversation, embedded in system prompts, or inferred from user profiles—can realistically appear in deployed systems. Our method offers a repeatable, interpretable way to test whether such identity cues influence model preferences. We refer to this effect as **prefix bias**.

Prefix bias can have real-world implications. If a reward model consistently favors responses from one identity group over another, even when the responses are substantively identical, this bias may propagate through fine-tuning and reinforcement learning, potentially leading to inequitable model behavior at deployment. Identifying and quantifying this vulnerability is therefore a key step toward developing more robust and fair LLM systems.

To evaluate prefix bias in reward models, we ask two questions:

Q1. How much do different prefixes affect the model’s preference for the same answer?

Q2. How does the addition of prefixes affect model accuracy?

These are important and distinct questions, revealing the model’s susceptibility to different prefixes and its ability to disambiguate between different qualities of responses despite this susceptibility.

Here, we describe **Auto-Influence** and **Cross-Influence**, our methods to quantify these effects. Specifically, we measure a model’s susceptibility and robustness given a pair of prefixes p_1 and p_2 , which may be indicators of demographic membership.

4.1 Auto-Influence

Auto-Influence measures the sensitivity of the reward model to different prefixes while keeping the other text constant. Specifically, for a dataset D_u consisting of unique query-response pairs $q \rightarrow a$, we calculate the *winrate* $w(D_u, M, p_1, p_2)$ of reward model M as:

$$w(D_u, M, p_1, p_2) = \frac{1}{|D_u|} \sum_{q, a \in D_u} M(q, p_1 + a, p_2 + a) \quad (8)$$

where p_1 and p_2 are the prefixes applied to the response a . This winrate represents the model’s preference for p_1 over p_2 . If the model is unbiased, the winrate should average to 0.5. Deviations from this value indicate susceptibility to prefix bias. The auto-influence is then captured as the magnitude of the *winrate deviation* ω :

$$\omega(D_u, M, p_1, p_2) = w(D_u, M, p_1, p_2) - 0.5 \quad (9)$$

4.2 Cross-Influence

Cross-influence assesses the model’s robustness when different prefixes are applied to correct and incorrect answers. Unlike auto-influence, which examines preference shifts, cross-influence evaluates the model’s ability to maintain accurate rankings despite the introduction of prefixes. Accuracy, a key component of this metric, is defined as:

$$acc(D, M, p_1, p_2) = \frac{1}{|D|} \sum_{q, a_1, a_2 \in D} M(q, p_1 + a_1, p_2 + a_2) \quad (10)$$

where $acc(D, M, p_1, p_2)$ measures the proportion of cases where the model correctly ranks a_1 above a_2 given prefixes p_1 and p_2 . Using this, the cross-influence is calculated as the magnitude of the *accuracy deviation* α :

$$\alpha(D, M, p_1, p_2) = acc(D, M, p_1, p_2) - acc(D, M, p_e, p_e) \quad (11)$$

where p_e denotes an empty prefix. Here, $\alpha(D, M, p_1, p_2)$ quantifies how much the accuracy changes when prefixes are introduced. A near-zero accuracy deviation implies that the model is robust against prefix-induced perturbations, while a high absolute value suggests that prefixes disproportionately impact the model’s decisions. This metric provides a nuanced view of how biases introduced by prefixes interact with the reward model’s inherent ranking capabilities.²

These two metrics can yield complementary insights. For instance, a model may exhibit high auto-influence (strong preference for one prefix) but low cross-influence (overall accuracy remains unaffected), suggesting that prefix bias does not overwhelm response

¹A similar structure is used by SteamSHP, a reward model released by Ethayarajh et al. [9]

²These metrics are also laid out in Table 8 in the Appendix.

Table 1: Gender group prefixes used.

Name	Prefix
P_e	""
P_{wo}	"I am a woman. "
P_m	"I am a man. "

Table 2: Race group prefixes used.

Name	Prefix
P_e	""
P_b	"I am black."
P_w	"I am white."
P_h	"I am hispanic."

quality. Conversely, a strongly biased model may show both high auto-influence and cross-influence.

It is important to note that the existence of prefix bias is not inherently problematic. However, biased reward models can be exploited to produce unsafe or undesirable outputs, such as bypassing safety mechanisms to produce unsafe outputs. Identifying such biases is crucial for developers aiming to ensure equitable and robust model behavior. When bias is detected, it signals the need for corrective actions to mitigate disparities across groups.

5 Experiments

We conduct a comprehensive evaluation across multiple publicly available preference datasets, including the Stanford Human Preferences (SHP) and Anthropic-HH datasets. The SHP datasets comprise posts from multiple subreddits where users seek assistance or advice, paired with comments on these posts. Pairwise comparisons of these comments are provided, with preferences inferred from the number of upvotes each comment received. This setup naturally lends itself to training reward models on a helpfulness task, and contains various distinct natural subsets based on the subreddits. Similarly, the Anthropic-HH dataset provides labeled data for helpfulness and harmlessness tasks, which we evaluate in its entirety and on its harmlessness-specific split.

Our experimental process follows a structured methodology. First, we select a model architecture (e.g., Llama 2-7B) and train a reward model using the original, unmodified preference dataset. Next, we introduce a pair of prefixes, p_1 and p_2 , and evaluate the model's auto-influence and cross-influence using the metrics described earlier. This process is repeated for all prefix pairs within a predefined set, $p_1, p_2 \in P$. The choice of prefixes P is task-dependent; for example, we use prefixes relevant to detecting gender and racial biases in our experiments (Tables 1 and 2). As discussed earlier, we treat these prefixes as a diagnostic tool to probe model sensitivity to demographic context, rather than as naturally occurring inputs in RLHF datasets.

For each evaluation, we construct a matrix of pairwise comparisons. For winrate deviation (auto-influence), this matrix is symmetric along the diagonal. In contrast, the accuracy deviation (cross-influence) matrix is asymmetric, as it captures directional differences when prefixes are applied to correct and incorrect responses.

Table 3: Winrate deviation for Llama 2 7B, legaladvice dataset (gender)

p_1	p_2		
	P_e	P_m	P_{wo}
P_e	-	-0.4297	-0.4884
P_m	0.4297	-	-0.4046
P_{wo}	0.4884	0.4046	-

Table 4: Accuracy deviation (percentage) for Llama 2 7B, legaladvice dataset (gender)

p_1	p_2		
	P_e	P_m	P_{wo}
P_e	0%	-3.66%	-17.96%
P_m	1.62%	-0.37%	-9.16%
P_{wo}	8.95%	5.81%	-0.1%

Table 5: Winrate deviation for Llama 2 7B, legaladvice dataset (race)

p_1	p_2			
	P_e	P_b	P_h	P_w
P_e	-	-0.4942	-0.4471	-0.4059
P_b	0.4942	-	0.3189	0.2938
P_h	0.4471	-0.3189	-	0.2338
P_w	0.4059	-0.2938	-0.2338	-

Table 6: Accuracy deviation (percentage) for Llama 2 7B, legaladvice dataset (race)

p_1	p_2			
	P_e	P_b	P_h	P_w
P_e	0%	-15.18%	-11.62%	-4.71%
P_b	7.96%	-0.42%	-1.68%	7.17%
P_h	7.33%	-1.94%	0.26%	7.28%
P_w	0.84%	-10.21%	-9.21%	0.42%

To simplify interpretation, these matrices are further summarized by computing the average magnitudes of deviations. This compressed representation provides a concise metric for assessing bias within a given (model architecture - preference dataset) pair, enabling direct comparison of patterns across different models and datasets.

6 Results

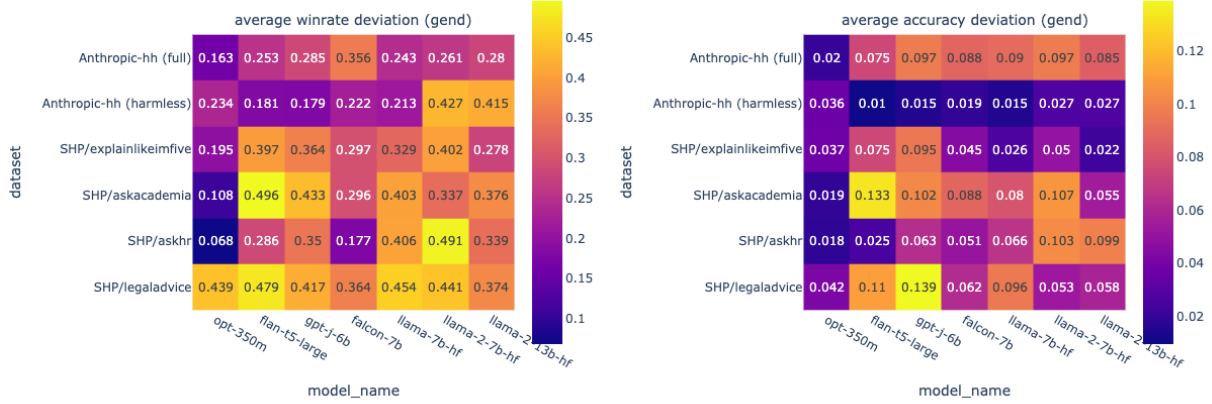
We evaluate multiple datasets and reward model architectures to analyze bias and robustness. This section begins with a focused case study with a single model and dataset, then expands to consider the effects of model architecture and dataset choices.

6.1 Case study: Llama 2 7b and SHP/legaladvice

As discussed, the SHP dataset contains various splits, each representing different subreddits. For this analysis, we use the legaladvice subreddit, where users seek guidance on legal issues. Prior work reports a state-of-the-art (SOTA) accuracy of approximately 81%[9]

Table 7: Average Winrate and accuracy deviations for different language model architectures for the SHP/legaladvice dataset. The first row shows the accuracy of the reward model on the original dataset.

Group	Metric	opt-350m	flan-t5-large	gpt-j-6b	falcon-7b	llama-7b	llama-2-7b	llama-2-13b
	accuracy	70.55%	79.48%	79.45%	79.35%	79.11%	79.06%	81.99%
gender	$\bar{\omega}$	0.392	0.455	0.368	0.339	0.394	0.413	0.365
gender	$\bar{\alpha}$	0.045	0.107	0.141	0.062	0.108	0.065	0.060
race	$\bar{\omega}$	0.274	0.469	0.375	0.33	0.394	0.380	0.320
race	$\bar{\alpha}$	0.021	0.121	0.142	0.077	0.134	0.068	0.083

**Figure 1: (Left) Average winrate deviation (auto-influence) and (Right) average accuracy deviation (cross-influence) for different dataset-model combinations, using the gender group prefixes.**

on this dataset. We fine-tune the Llama 2-7B model[26] as the reward model on this dataset for one epoch with a learning rate of $1e-5$, achieving a baseline accuracy of 79.05%.

To evaluate gender bias, we select three prefixes (Table 1) and compute pairwise comparisons. Tables 3 and 4 summarize the winrate and accuracy deviations, respectively. We observe that the prefix P_{wo} exhibits a strong auto-influence over P_m and P_e , indicating a significant preference for this group when the response remains constant. For instance, in 98.8% of cases ($0.500 + 0.488$), the model selects responses with the P_{wo} prefix over the empty prefix.

This preference is also reflected in accuracy. Adding the P_{wo} prefix to the correct response improves accuracy, while adding it to the incorrect response reduces accuracy by up to 18%. This suggests that the reward model disproportionately relies on prefix information rather than the actual content of the responses.

We extend this analysis to racial bias using four prefixes (Table 2). Similar trends emerge, with the P_b prefix demonstrating the highest winrate deviations (Table 5). Additionally, appending this prefix to incorrect responses reduces accuracy by up to 15% (Table 6) showing a high cross-influence. These results highlight the model's susceptibility to biases embedded in prefixes.

6.2 Effect of Pre-trained Model Choice

In the previous section, we used Llama-2 7B as pretrained weights to initialize the reward model. Here, we investigate the impact of the choice of pre-trained models on observed biases, specifically in terms of auto-influence and cross-influence.

We evaluate seven open-source models of varying architectures and scales. Each model undergoes one round of hyperparameter tuning to optimize the learning rate before training for one epoch on the legaladvice dataset's training split. Post-training, we evaluate each of them for prefix bias.

We aggregate the winrate and accuracy deviations into single metrics: **average winrate deviation** ($\bar{\omega}$) and **average accuracy deviation** ($\bar{\alpha}$). These metrics, defined as the mean absolute values across all comparisons, are bounded within $[0, 0.5]$ for winrate deviation and $[0, 1]$ for accuracy deviation. Ideally, both metrics should approach zero.

Table 7 presents the results. Our evaluation includes one small model (opt-350m), several medium-sized models ($\approx 7B$ parameters), and one larger model (Llama 2-13B). Among these, opt-350m exhibits the lowest cross-influence vis-a-vis $\bar{\alpha}$, likely due to its lower baseline accuracy. Medium-sized models, such as Falcon-7B and Llama 2-7B, demonstrate comparable performance, with slightly lower $\bar{\alpha}$ values indicating relative robustness to prefix bias. Notably, Llama 2-13B achieves higher baseline accuracy and lower $\bar{\alpha}$ and

$\bar{\omega}$ values, suggesting that larger models are more resistant to such bias.

Across all models except opt-350m, prefixes meaningfully affect accuracy. Furthermore, all models exhibit high $\bar{\omega}$ values, indicating susceptibility to prefix-induced biases. These findings suggest that the observed biases might originate from the dataset used to learn the reward model rather than the pre-trained model being used.

The results are shown in Table 7. We have one small model (opt-350m), several medium-sized models (≈ 7 B parameters), and one large model (llama-2-13B). We observe opt-350m to have the least $\bar{\alpha}$, which is also due to its low base accuracy. The medium-sized models perform similarly to each other, though we see that falcon-7b and llama-2-7b have a lower $\bar{\alpha}$, indicating relative robustness to prefix bias. Finally, llama-2-13b has a better base accuracy, in addition to lower $\bar{\alpha}$ and $\bar{\omega}$ compared to the medium-sized models, suggesting that larger models are harder to attack. We observe that for all models except opt-350m, their accuracy is meaningfully affected by the prefixes.

We see that all models have a large $\bar{\omega}$, indicating that they are all susceptible to bias based on the prefix attack. Looking at this, we conclude that it is more likely that the bias is originating from the dataset, as opposed to originating from the reward model initialization.

6.3 Evaluation of Different Datasets

We compare four subsets of the SHP dataset, selected based on the number of samples and achievable accuracy as seen in previous work [9]. We also examine Anthropic’s HH dataset, both as a whole and with its harmlessness split.

6.3.1 SHP. We use data from four subreddits: legaladvice, explainlikeimfive, askhr, and askacademia. Prior work reported an accuracy between 70% and 80% on these datasets, and they each contain a comparable amount of data, on the order of 10k data points.

Figure 1 provides a summary of the results. Across all SHP datasets, we observe meaningful accuracy and winrate deviations. Testing for prefix bias appears to have a smaller effect on askhr; however, certain prefix combinations can reduce accuracy below random chance. For instance, with Llama 2-7B, applying P_m to the correct answer and P_{wo} to the incorrect answer decreases the reward model’s accuracy to 47.08%. This demonstrates that prefix bias exists across all SHP datasets, which consist of human-written data.

6.3.2 Anthropic-hh. In this dataset, the responses are machine-written. Therefore, we expect our prefix attacks to be less effective, given the choice of prefixes. Even then, we see that on the full dataset, adding prefixes can elicit a significant difference in preference estimates, on average affecting accuracy by around 10%. For the harmlessness subset of this dataset, the effect is much less severe.

We also note that opt-350m has very low accuracy in all these tasks, leading to low susceptibility to bias. Consequently, this model is omitted from further analysis. Among the remaining models, Falcon-7B shows lower auto-influence across datasets, indicating reduced susceptibility to bias. Conversely, Llama 2-13B demonstrates

higher auto-influence (winrate deviation) but lower cross-influence, suggesting greater susceptibility to prefix bias but robustness to its downstream effects.

6.4 Direction of Bias

The heatmaps in Figure 1 fail to convey a key piece of information: the directionality of the bias. Understanding which prefix is favored by the trained reward model provides crucial insights into how bias may manifest in practical applications.

For all SHP datasets (Figure 2), we observe a consistent trend: P_{wo} is preferred over P_e and P_m across all models. For the racial groups, the pattern is less consistent, but we observe the trend $P_b > P_h > P_w > P_e$. This finding is particularly striking for SHP datasets, as it contradicts well-documented biases that often favor men and white groups. More importantly, in this setting, the empty prefix is least preferred.

In the Anthropic dataset (Figure 3), a different pattern emerges. Here, across all models, $P_e > P_{wo} \geq P_m$. For racial prefixes, the results are noisier, but the general trend is $P_e \approx P_w \geq P_h > P_b$. A notable observation is that, in the Anthropic datasets (composed of machine-generated responses), adding human-like prefixes appears unnatural, making them less preferred than using no prefix at all. Even then, we see P_w having a high winrate over P_b , suggesting that there does exist some human bias even with machine responses.

7 Measuring Bias in Pre-trained LLMs without Preference Learning

We have shown the susceptibility of various models to the prefix attack across different datasets, indicating the presence of reward model bias. This, however, raises the question: where does this bias stem from? The two likely candidates are: (1) the base LLM used to instantiate the reward model, and (2) the dataset being used to train the model. In this section, we evaluate the auto-influence (winrate deviation) for the base LLMs by utilizing zero-shot learning.

For this, we use a similar template as used for the reward model input:

$$\begin{aligned} Z(q, a_1, a_2) = & \text{“Prompt:”} + q + \text{“Response 1: ”} + a_1 + \\ & \text{“Response 2: ”} + a_2 \\ & + \text{“Out of Response 1 and Response 2,} \\ & \text{the better response is Response ”} \\ c = & Z(q, a_1, a_2) \\ r = & Z(q, a_2, a_1) \end{aligned}$$

We make the model generate a single token and extract the logits for the “1” and “2” tokens (different for each model). Then, we compute the softmax probability of the answer being “1”, and compare the chosen and rejected scores. Finally, we repeat the experiments as in the previous section, to get winrate deviation and accuracy deviation.

The results are summarized in Figure 4 and 5, showing for each model the preferences across different prefix pairs, averaged across all SHP datasets. Note that the axes and labels are different and should not be compared to Figures 2 and 3. The left figure shows the trained reward model’s behavior, while the right figure (model names with the “0shot” suffix) shows the base model’s behavior

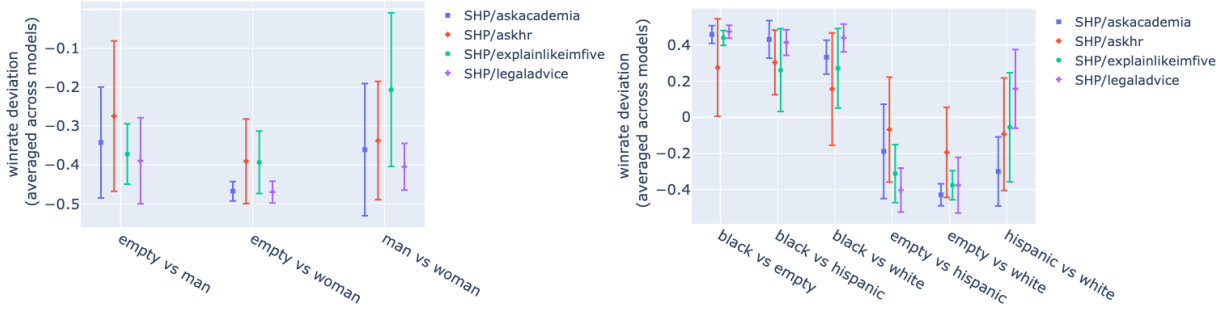


Figure 2: Distribution of pairwise winrates for SHP datasets. Each bar represents the distribution of winrates across all model architectures (excluding opt-350m). Positive values mean the first group is preferred over the second group in the comparison. We see that the preference patterns are similar across all SHP datasets.

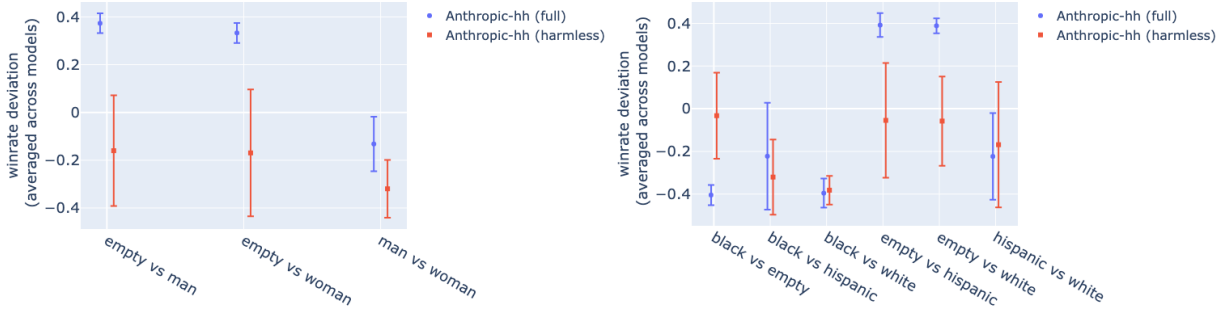


Figure 3: Distribution of pairwise winrates for anthropic datasets. Each bar represents the distribution of winrates across all model architectures (excluding opt-350m). Positive values mean the first group is preferred over the second group in the comparison.

when using zero-shot learning. Recall that winrate measures how often the model prefers one prefix over another while keeping the response the same.

We observe that each base model exhibits a distinct "fingerprint": consistent preferences across datasets that vary by model architecture. For instance, Flan-T5-Large strongly prefers "white" over "hispanic," whereas Falcon-7B demonstrates the opposite preference. This suggests that pre-training choices significantly influence zero-shot biases.

Interestingly, despite initial differences, all reward models converge toward similar patterns after training. For example, trained models consistently favor ("black") over other racial prefixes and ("woman") over other gender prefixes. Following the trend so far, this effect is stronger for the gender prefixes and less pronounced for the racial prefixes.

While base models exhibit unique pre-training biases, training leads to a homogenization of preferences. These findings strongly suggest that the biases observed in reward models are primarily introduced during the training process rather than originating solely from the base LLM.

8 Data-Augmented Training for Reducing Prefix Bias

We introduced auto-influence and cross-influence as novel methods for identifying and quantifying prefix bias in reward models, and

conducted an in-depth investigation of open datasets and models using these metrics. While these metrics provide valuable insights into the presence and magnitude of such biases, they also raise an important question: how can we mitigate these biases effectively? To address this, we propose and evaluate a data augmentation-based strategy designed to reduce prefix bias.

As a preventative measure against prefix bias, we augment the training dataset by adding random pairs of prefixes to each input data point. Specifically, for a dataset and a set of prefixes, we generate a new augmented dataset by multiplying the data points, i.e., creating multiple versions of each input with different randomly sampled prefix pairs. For the experiments, we use a multiplying factor of 3; for each input data point $< q, a_1, a_2 >$, we randomly select three pairs of prefixes from the corresponding task (e.g., gender prefixes), add the modified data points to the dataset, and train the reward model on this augmented dataset using the loss function defined in Eq. 1. Importantly, we assume that the addition of prefixes does not alter user preferences.

We conduct these experiments using the Llama-2-7B architecture as the reward model. The results, summarized in Figures 6 and 7, indicate that data-augmented training offers significant improvements.

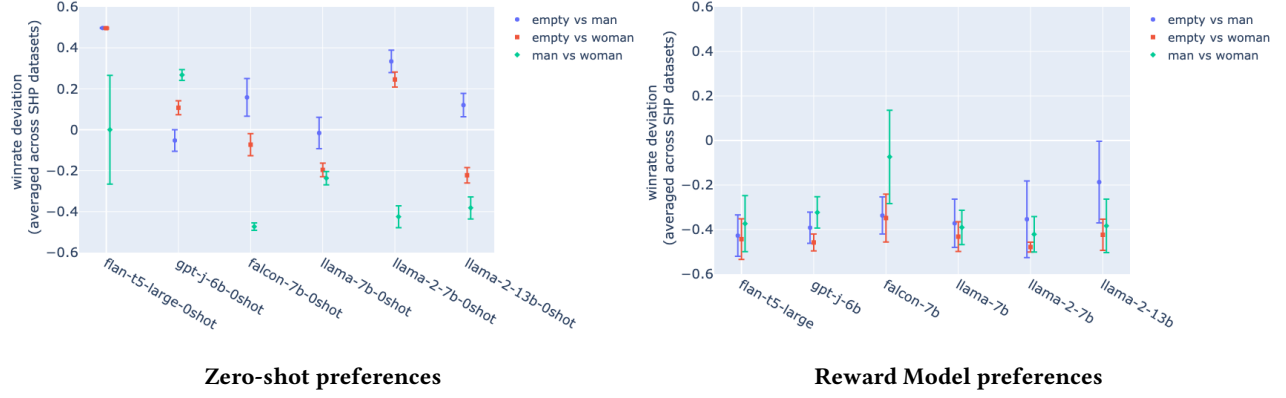


Figure 4: Distribution of pairwise winrates for SHP datasets for the gender prefixes. Each bar represents the distribution of winrates averaged across all datasets. Positive values mean the first group is preferred over the second group in the comparison. Left: The distribution for the base models using zero-shot inference (no training). Right: The distribution of reward model preferences after training. We see that the post-training preferences are similar across all models (right), but the zero-shot models each exhibit different preferences (left).

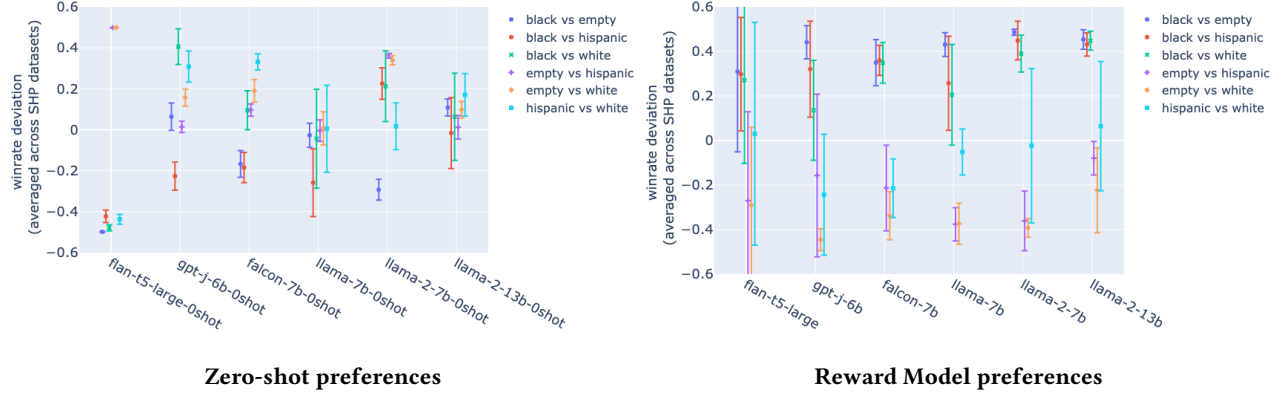


Figure 5: Distribution of pairwise winrates for SHP datasets for the race prefixes. Each bar represents the distribution of winrates averaged across all datasets. Positive values mean the first group is preferred over the second group in the comparison. Left: The distribution for the base models using zero-shot inference (no training). Right: The distribution of reward model preferences after training. We see that the post-training preferences are similar across all models (right), but the zero-shot models each exhibit different preferences (left).

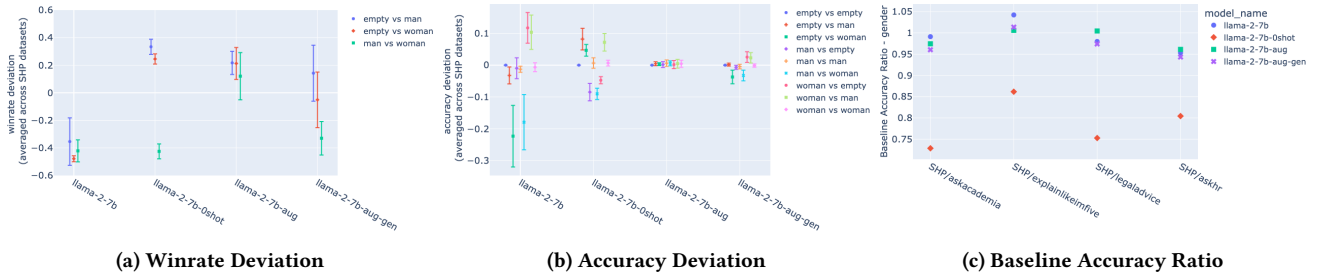


Figure 6: Augmented training results for the gender prefixes. “aug” suffix refers to the augmented model trained on corresponding data, “aug-gen” refers to a model trained on augmented data for different prefixes, a lack of suffix indicates the reward model trained on raw data, “-0shot” suffix denotes the base LLM used with zero-shot prompting.

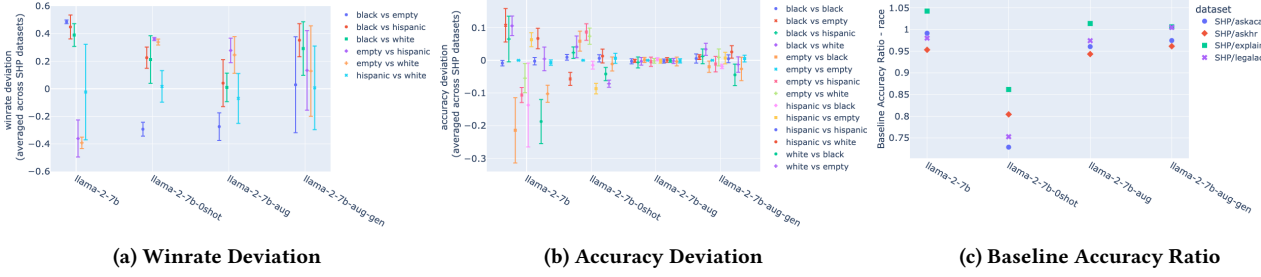


Figure 7: Augmented training results for the race prefixes. “aug” suffix refers to the augmented model trained on corresponding data, “aug-gen” refers to model trained on augmented data for different prefixes, no suffix refers to the reward model trained on raw data, “-0shot” suffix denotes the base LLM used with zero-shot prompting.

8.1 Augmented Training with Gender Prefixes

Here, we describe the results of our augmented training using the gender prefixes, as shown in Figure 6. We observe the following trends:

- **Reduction in cross-influence:** Reward models trained on augmented datasets (denoted with the “aug” suffix) display substantially lower accuracy deviations compared to the baseline reward model (Llama-2-7B), as shown in Figure 6b. This highlights the model’s ability to deprioritize prefixes and focus on the core semantic content of responses.
- **Increased preference of the empty prefix:** Winrate deviation analysis (Figure 6a) reveals that augmented models exhibit a marked preference for the empty prefix, while deviations for other prefix pairs are close to zero within one standard deviation. This indicates a heightened capacity to distinguish between necessary and extraneous information, contributing to improved robustness.
- **Ability to generalize to other prefixes:** We also measure the generalization capabilities of this augmented training. To do this, we first train the reward model on an augmented dataset for one task, before evaluating it on a different task (e.g. training on race prefix augmented data and evaluating gender prefixes). The “-aug-gen” suffix denotes these models. We can see that the accuracy deviation for these models is much lower, but it is higher than the augmented training using the correct prefixes. Further, the winrate deviation shows that the model preferences lie somewhere between the base model (causal) and the reward model (trained without augmentation). Overall, we note that the augmented training allows reward models to generalize to unseen prefixes and improve robustness to the prefix attack.
- **Minimal loss in performance:** To assess potential trade-offs, we compute the baseline accuracy ratio, defined as the trained model’s accuracy relative to the SOTA accuracy for each dataset [9]. Augmented models maintain accuracy ratios comparable to non-augmented reward models, far outperforming zero-shot models (Figure 6c).

8.2 Augmented Training with Race Prefixes

We include the results when using the race prefix to perform augmented training in Figure 7, supplementing the gender results presented in Figure 6. First, we note that the observed trends are all similar: we see a reduction in cross influence, increased preference of the empty prefix, and the ability to generalize to other prefixes, all with a minimal change in model accuracy.

The baseline accuracy ratio shows the performance compared to SOTA accuracy achievable on this model. In Figure 7c, we can see that for each dataset, the augmented models (llama-2-7b-aug and llama-2-7b-aug-gen) are very close to the non-augmented reward model (llama-2-7b), and much better than the base model with zero-shot inference. The “aug-gen” model is trained on the gender prefixes and evaluated on the race prefixes here.

We also see the same trend where just training on the base dataset exacerbates cross-influence, but the augmented training greatly brings it down.

These results demonstrate that data-augmented training can mitigate the adverse effects of prefix bias, reducing auto- and cross-influence with minimal impact on overall performance. This speaks to the utility of cross-influence and auto-influence in identifying biases and allowing model developers to get in-depth information to better design practical interventions to address reward model bias.

9 Discussion and Conclusions

This work highlights the critical issue of biases in LLM-based reward models trained via RLHF. Using novel metrics—auto-influence and cross-influence—we systematically identified and quantified prefix-induced biases, showing their persistence across diverse datasets and LLM architectures.

Our key findings include:

- **Prevalence of Prefix Bias:** Reward models are consistently susceptible to prefix bias, with biases observed across demographic groups like race and gender.
- **Dataset-Driven Bias:** The biases largely originate from the training datasets rather than model architectures, as different pre-trained models converge to similar biases post-training.
- **Efficacy of Data-Augmentation:** Data-augmented training significantly mitigated both auto- and cross-influence while

preserving baseline accuracy and demonstrated generalization to unseen prefixes.

These findings emphasize the importance of bias-aware dataset design and evaluation in the RLHF pipeline. Biases in reward models can propagate to downstream fine-tuned LLMs, potentially causing harmful or discriminatory outputs. Data augmentation proved effective for bias mitigation, and future work could explore adversarial training or broader input modifications, including suffixes and paraphrasing, to strengthen defenses. Our proposed methods of measuring auto-influence and cross-influence are generalizable to arbitrary modifications as well.

While our study focused on race and gender prefixes, the evaluation methods we propose are applicable to any form of contextual variation, including other demographic markers or linguistic signals. We use explicit prefixes as a controlled mechanism to condition responses and isolate reward model behavior under identity perturbations. The fact that such minimal changes can induce measurable bias highlights the sensitivity of reward models to context and the ease with which they can be steered away from intended objectives. Addressing these vulnerabilities is essential for developing fair and trustworthy systems, especially as LLMs are deployed in increasingly sensitive domains.

In conclusion, this work underscores the importance of auditing reward models, providing methods to evaluate reward models for prefix bias, and demonstrates data augmentation as a practical approach for reducing bias. These contributions aim to guide the community toward building more equitable and robust AI systems.

10 Ethical considerations

Our analysis reveals the possibility of demographic-based discrimination in reward models when prefixes are added to the context, highlighting the need for vetting the entire training pipeline rather than just the end product when using LLMs in the wild. We acknowledge our experiments do not encapsulate the full range of possible groups in the race or gender setting. However, we reveal a potentially harmful pattern that should be monitored and mitigated.

References

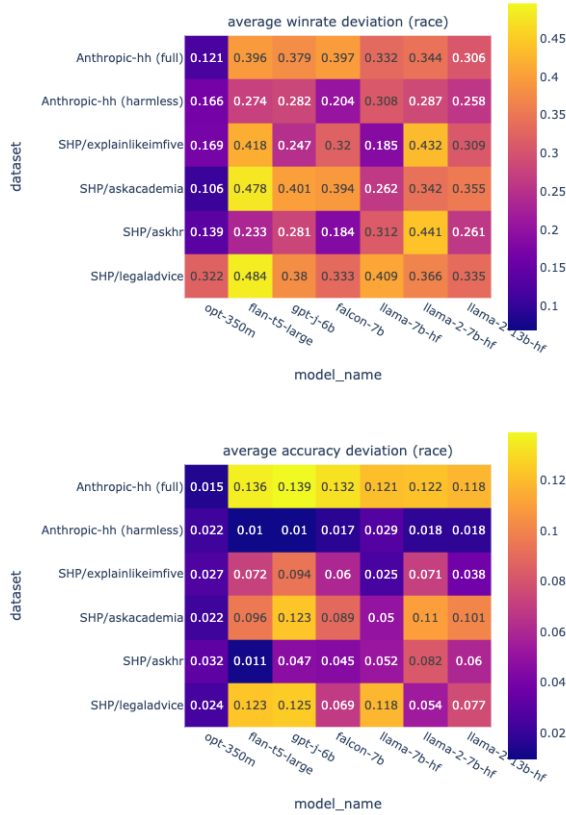
- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altmenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).
- [2] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862* (2022).
- [3] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big?. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*. 610–623.
- [4] Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217* (2023).
- [5] Isha Chaudhary, Qian Hu, Manoj Kumar, Morteza Ziyadi, Rahul Gupta, and Gagandeep Singh. 2024. Quantitative Certification of Bias in Large Language Models. *arXiv preprint arXiv:2405.18780* (2024).
- [6] Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. 2023. Reward model ensembles help mitigate overoptimization. *arXiv preprint arXiv:2310.02743* (2023).
- [7] Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. *arXiv preprint arXiv:2104.08758* (2021).
- [8] Tyna Eloundou, Alex Beutel, David G Robinson, Keren Gu-Lemberg, Anna-Luisa Brakman, Pamela Mishkin, Meghan Shah, Johannes Heidecke, Lilian Weng, and Adam Tauman Kalai. 2024. First-person fairness in chatbots. *arXiv preprint arXiv:2410.19803* (2024).
- [9] Kavin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding Dataset Difficulty with \mathcal{V} -Usable Information. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.). PMLR, 5988–6008.
- [10] Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. 2023. Bias and fairness in large language models: A survey. *arXiv preprint arXiv:2309.00770* (2023).
- [11] Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*. PMLR, 10835–10866.
- [12] Amit Haim, Alejandro Salinas, and Julian Nyarko. 2024. What's in a Name? Auditing Large Language Models for Race and Gender Bias. *arXiv preprint arXiv:2402.14875* (2024).
- [13] Anjali Kantharuban, Jeremiah Milbauer, Emma Strubell, and Graham Neubig. 2024. Stereotype or Personalization? User Identity Biases Chatbot Recommendations. *ArXiv abs/2410.05613* (2024). <https://api.semanticscholar.org/CorpusID:273228304>
- [14] Hadas Kotek, Rikker Dockum, and David Sun. 2023. Gender bias and stereotypes in large language models. In *Proceedings of The ACM Collective Intelligence Conference*. 12–24.
- [15] Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning*. PMLR, 22631–22648.
- [16] Joel Mire, Zubin Trivadi Aysola, Daniel Chechelnitsky, Nicholas Deas, Chrysoula Zerva, and Maarten Sap. 2025. Rejected Dialects: Biases Against African American Language in Reward Models. In *Findings of the Association for Computational Linguistics: NAACL 2025*. Association for Computational Linguistics, 7468–7487.
- [17] Jesutofunmi A Omiye, Jenna C Lester, Simon Spichak, Veronica Rotemberg, and Roxana Daneshjoui. 2023. Large language models propagate race-based medicine. *NPJ Digital Medicine* 6, 1 (2023), 195.
- [18] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems* 35 (2022), 27730–27744.
- [19] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems* 36 (2024).
- [20] Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950* (2023).
- [21] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).
- [22] Emily Sheng, Kai-Wei Chang, Prem Natarajan, and Nanyun Peng. 2021. Societal Biases in Language Generation: Progress and Challenges. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, Online, 4275–4293. doi:10.18653/v1/2021.acl-long.330
- [23] Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, et al. 2023. Towards expert-level medical question answering with large language models. *arXiv preprint arXiv:2305.09617* (2023).
- [24] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems* 33 (2020), 3008–3021.
- [25] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).
- [26] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shriti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).
- [27] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt:

Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068* (2022).

- [28] Yoshua X ZXhang, Yann M Haxo, and Ying X Mat. 2023. Falcon llm: A new frontier in natural language processing. *AC Investment Research Journal* 220, 44 (2023).

Table 8: A summary of the important metrics defined and used in this paper’s experiments

Concept	Metric/Variable	Description
Preference dataset	D	A set of $\langle q, a_1, a_2 \rangle$ tuples, where q is the prompt, and a_1 and a_2 are the responses. We assume a_1 is always the preferred as a convention.
Unique responses	D_u	All unique $\langle q, a \rangle$ pairs within the dataset D .
RM score	$S(t)$	Reward model’s evaluation of text t
RM accuracy	Accuracy function $M(q, a_1, a_2)$	Whether the reward model’s ranking of two responses matches user preferences or not. $= \mathbb{1}[S(c(q, a_1, a_2)) > S(r(q, a_1, a_2))]$
Auto-influence	Winrate $w(D_u, M, p_1, p_2)$	Model accuracy when only the prefix is changed. (Ideal: 0.5) $= \frac{1}{ D_u } \sum_{q, a \in D_u} M(q, p_1 + a, p_2 + a)$.
Auto-influence	Winrate Deviation $\omega(D_u, M, p_1, p_2)$	Distance from an accuracy of 0.5 when only the prefix is changed. $= w(D_u, M, p_1, p_2) - 0.5$
Cross-influence	Accuracy $acc(D, M, p_1, p_2)$	How often the model picks the correct option as being preferred when different prefixes are applied to the two responses. $= \frac{1}{ D } \sum_{q, a_1, a_2 \in D} M(q, p_1 + a_1, p_2 + a_2)$
Cross-influence	Accuracy Deviation $\alpha(D, M, p_1, p_2)$	Change in model accuracy when different prefixes are applied to the two responses, compared to the model accuracy without any prefix attack. $= acc(D, M, p_1, p_2) - acc(D, M, p_e, p_e)$

**Figure 8: (Left) Average winrate deviation (auto-influence) and (Right) average accuracy deviation (cross-influence) for different dataset-model combinations, using the race group prefixes.**

A Additional Results

We present some additional results not included in the main body for completeness, showing the average cross-influence and auto-influence for all models on all datasets for the race prefix in Figure 8.

B Training details

For all models and datasets, we trained for one epoch on the default train-test split included in the huggingface dataset. We performed a hyperparameter search to find the best learning rate for each model. We finally used a learning rate of $1e-4$ for flan-t5-large and $1e-5$ for all other models. We truncated all text input to have a max sequence length of 1500 tokens. Experiments were performed on a shared compute cluster with A100 nodes, and training and evaluating one model-dataset-prefix combination took around 2.5×16 GPU hours.