MITIGATING COGNITIVE BIAS IN RLHF BY ALTERING RATIONALITY

Anonymous authors
Paper under double-blind review

ABSTRACT

How can we make models robust to even imperfect human feedback? In reinforcement learning from human feedback (RLHF), human preferences over model outputs are used to train a reward model that assigns scalar values to responses. Because these rewards are inferred from pairwise comparisons, this learning depends on an assumed relationship between latent reward differences and observed preferences, typically modeled using a Boltzmann formulation in which a rationality parameter β informs how consistently preferences reflect these reward differences. In practice, β is typically treated as a fixed constant that assumes uniform annotator reliability. However, human feedback is not actually this simplistic: real human judgments are shaped by cognitive biases, leading to systematic deviations from reward-consistent behavior that arise in context. To address this, we treat rationality as context- and annotation-dependent. We design an approach to dynamically adjust the rationality parameter β during reward learning by effectively downweighting comparisons that likely reflect biased or unreliable judgments. Empirically, we show this approach learns a useful and more rational downstream model, even when finetuning on datasets with strongly biased preferences.

1 INTRODUCTION

RLHF has emerged as a prominent technique for aligning LLMs with human preferences. This is often framed as progress toward "value alignment," the challenge of ensuring that learned objectives faithfully reflect human intent (Russell, 2019). However, RLHF adopts the premise that expressed human preferences reflect human intentions. But, humans are fallible: we regularly make mistakes and are subject to cognitive biases, so our expressed preferences may not capture our intentions. Standard RLHF pipelines do not differentiate between more or less reliable forms or instances of feedback.

A famous example of cognitive bias is the Linda problem (Tversky & Kahneman, 1983), a conjunction fallacy. People are told, "Linda is 31 years old, single, outspoken, and very bright. She majored in philosophy. As a student, she was deeply concerned with issues of discrimination and social justice, and also participated in anti-nuclear demonstrations" They must then assess whether A: "Linda is a bank teller" or B: "Linda is a bank teller and is active in the feminist movement" is more probable. Most people pick B despite the impossibility of any subset of bank tellers being greater than the entire set of bank tellers.

The Linda problem may seem trivial, but cognitive biases arise often in high-stakes domains such as medicine. For example, physicians are known to exhibit anchoring bias. This can lead a physician to potentially overweight an initial diagnosis (e.g., asthma) and discount subsequent evidence that may point to a more serious condition, like heart failure. These are not hypothetical concerns: empirical studies document systematic cognitive biases in clinical decision-making with serious consequences for patient outcomes (Ke et al., 2024). When cognitive biases are present in the judgments used as training signals, as in RLHF, these systematic deviations become embedded in the learned reward model, leading downstream models to reproduce or even amplify these biases in their outputs.

To address the risk of cognitive biases in preference data, we intervene directly on the RLHF objective. In the standard formulation, human preferences are modeled using a Boltzmann-rational model with a rationality parameter β, which is typically fixed across annotators and across comparisons.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

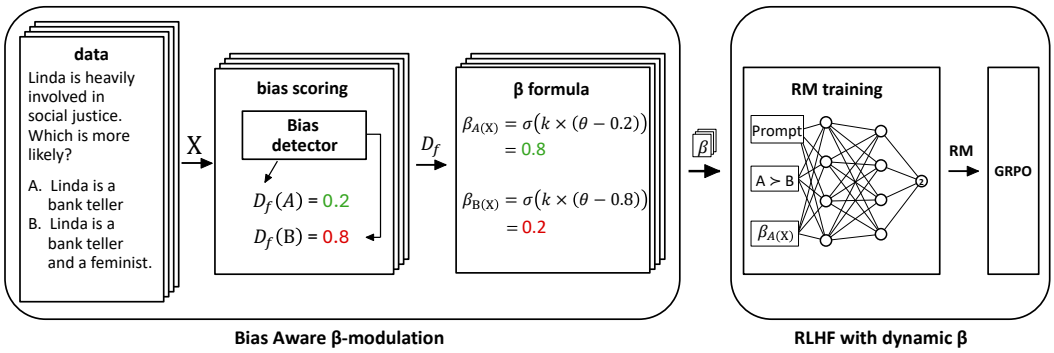

Figure 1: Overview of the intervention pipeline to reduce cognitive bias in the finetuned LLM. Humans provide preferences over responses to a prompt, like the "Linda is a bank teller" example. A bias detector then judges whether each paired prompt and response is likely to be subject to a cognitive bias (D_f). From this, we compute a dynamic value of β. These prompts, responses, and β values are used to learn a reward model and to finetune an LLM. Higher bias = lower value of β.

This implicitly assumes uniform reliability in human judgments. We relax this assumption by treating β as *feedback-dependent*, as some responses are more likely to be unreliable than others. To do so, we estimate the likelihood that a given comparison is affected by cognitive bias, and we use this estimate to dynamically adjust β during reward learning to downweight feedback that is likely to be biased; see Figure 1. This preserves informative signals from human preferences while reducing the influence of systematic cognitive biases, enabling the learned reward model and downstream policy to better reflect underlying preferences rather than observed, potentially biased judgments.

We evaluate our approach on two datasets known to elicit cognitive bias. We find that dynamically adjusting β by assessing response bias propensity yields the following contributions:

1. **Reduction in bias propagation.** Relative to baseline LLMs without this intervention, our method produces models that are significantly less likely to reproduce cognitive biases.
2. **Robustness to biased feedback.** Even when the preference dataset contains many cognitively biased responses, our approach still constructs a useful reward model by effectively downweighting the bias-prone annotations in the preference data.
3. **Preservation of general performance.** Despite operating on heavily biased training data, the intervention does not degrade performance on unrelated tasks, indicating that the method selectively reduces bias without sacrificing overall model capabilities.

2 Background

2.1 Cognitive bias

RLHF can inherit systematic irrationality from human judgment (Casper et al., 2023). One contributing factor is that human annotators rely on simplified, often myopic reasoning when expressing preferences. Annotators may neglect longer-term consequences or cumulative effects and instead rely on heuristics. For example, prospect theory shows that humans are more sensitive to relative losses than to gains of the same magnitude, leading to asymmetric outcome evaluations (Kahneman & Tversky, 2013), which may manifest in preferences. Likewise, temporal preferences are often inconsistent with exponential discounting; people exhibit hyperbolic discounting and disproportionately favor immediate rewards over delayed ones (Moore et al., 2025). Additional biases such as framing effects and scope insensitivity can further shape expressed preferences in ways that are not aligned with stable or reflective intentions (D'Alonzo et al., 2026; Hatgis-Kessell et al., 2025).

For humans, cognitive biases serve a purpose: they enable rapid decision-making through heuristics, trading off accuracy for speed. There is an established literature that shows humans are more likely to experience cognitive biases under certain conditions and therefore act in a less rational way (Macmillan-Scott & Musolesi, 2023; Kahneman, 2013–2011), e.g., under time pressure. Because LLMs are trained with human preferences, LLMs tend to adopt our cognitive biases during pretraining (Malberg et al., 2024) or in finetuning (Cheung et al., 2025; Itzhak et al., 2024), whether through

instruction tuning or RLHF. Unlike humans, however, LLMs are not subject to the same time and resource constraints that encourage the use of such heuristics. This creates an opportunity: rather than inheriting human biases as an unavoidable feature of decision-making, we can design learning procedures that identify and mitigate them, enabling better deliberative reasoning with LLMs.

In decision-support settings, there is a fundamental tension between matching human-provided feedback and avoiding the systematic biases that such feedback may contain. In this work, we assume that people hold true preferences that may be obscured in their expressed preferences when cognitive bias is likely present (Hosking et al., 2024). We seek to learn the underlying preferences that a person would express were it not for the confounding factors of the bias. Prior research has shown that under some circumstances, people prefer machines not act in accordance with the literal preference given but rather with the intent behind the wording (Horter et al., 2026). This can in part address two known open problems with RLHF (Casper et al., 2023), namely (A) "Humans make simple mistakes due to limited time, attention, or care" and (B) "Humans can be misled, so their evaluations can be gamed."

2.2 Boltzmann Rationality

The premise of this work (and of RLHF more broadly) is that people have a latent reward function r^* that induces a distribution over their preferences. The objective of RLHF is to approximate this reward function from preferences. The standard model of human decision-making uses a Boltzmann-rational assumption (Jeon et al., 2020). Under this model, humans are more likely to select higher-reward choices, with the strength of this tendency increasing as reward differences grow. In this model, the rationality parameter β controls how inconsistently higher-reward responses are selected—i.e., it admits noise.

In particular, we condition on a candidate reward function r and model the likelihood of observed comparisons between outputs. Given a comparison $\sigma_1 \succ \sigma_2$, this yields the standard logistic form:

$$\mathbb{P}(\sigma_1 \succ \sigma_2 \mid r) = \texttt{logistic}\left(\beta\left(r(\sigma_1) - r(\sigma_2)\right)\right)$$

In standard approaches of approximating a reward function, whether from preferences or other modes of human feedback (Jeon et al., 2020), β is treated as a fixed parameter capturing the overall noisiness of human feedback. This interpretation assumes that deviations from reward-consistent behavior arise from uniform stochastic noise, rather than systemic bias of some form.

We instead treat the mapping from latent preferences to expressed preferences as context-dependent. Rather than assuming a fixed β, we allow β to vary across instances based on the expected reliability of the feedback. Intuitively, β should be lower in settings where cognitive biases are likely to distort judgment, and higher when responses are more likely to align with latent preferences. This can be viewed as reweighting observations by their expected fidelity, with the goal of recovering a reward model that better approximates the true r^*, rather than fitting the reward model to artifacts of bias.

2.3 Prior Work on Rationality Modeling

Accurately modeling the human's true decision process, even under cognitive bias, is critical for effective reward learning (Hong et al., 2023; Knox et al., 2023; Chan et al., 2021). In fact, explicitly modeling structured human irrationality can improve reward inference beyond assuming perfectly rational behavior, even with an oracle (Chan et al., 2021).

Prior methods have explored how rationality varies with feedback modality (e.g., comparisons, demonstrations, corrections) (Ghosal et al., 2022), as well as with annotator expertise (Daniels-Koch & Freedman, 2022). Other work addresses heterogeneity in annotator rationality by computationally varying the β parameter (Yamagata et al., 2024; Barnett et al., 2023), though these approaches differ from our approach as they do not model bias as context-dependent or as a property of the response. When considering annotators as well as the context, these methods could combine with our method for increased performance. Some approaches instead fit a global β to account for the level of systematic bias in human responses across feedback types (Ghosal et al., 2022), whereas we focus on the information held in the response rather than its format. Others attempt to learn the algorithms people use to make decisions directly, often finding that incorporating known heuristic biases improves learning performance (Shah et al., 2019).

More recent work adjusts β based on estimated difficulty of the annotation setting, such as interaction signals (e.g., clicks, time spent) or model-predicted difficulty (Singhal et al.). However, these approaches treat rationality as a function of the overall scenario rather than the specific feedback provided over responses, and do not evaluate whether such adjustments mitigate cognitive bias, focusing instead on upstream reward model behavior (e.g., sensitivity to factuality versus length). The issue with accounting for only the scenario rather than the response is that it removes or downweights *all* responses to biased scenarios, and ignores responses that avoid the pitfall of biases. We compare to this as a baseline and show improvement; see Appendix B.

3 METHOD

We build on the RLHF formulation introduced above, where the rationality parameter β governs how strongly reward differences influence observed preferences. Rather than treating β as fixed, we model this parameter as instance-dependent, reflecting the reliability of each piece of feedback.

Given an observed comparison, we estimate whether the expressed preference is likely to be influenced by cognitive bias. For this method, we could use any method that gives a measure of bias: we could use an LLM as an evaluator, a traditional classifier, or human annotations.

Using this bias estimate, we adjust β. When feedback is likely to be affected by bias, we reduce β to downweight its influence; when feedback is likely to reflect the annotator's latent preference, we increase β to place greater weight on the observation. In this way, β acts as a context-dependent measure of how faithfully expressed preferences reflect latent preferences.

We as humans don't update as strongly on information deemed likely to not be trustworthy or be biased, so we propose to define our dynamic, instance-dependent rationality parameter as

$$\beta_{new} = \texttt{logistic}(k \cdot (\theta - D_f)), \tag{1}$$

where $D_f \in [0, 1]$ denotes the estimated probability that the observed feedback is influenced by cognitive bias, θ is a threshold parameter, and k controls the steepness of the transition. This parameterization maps bias likelihoods to a continuous measure of feedback reliability. When D_f is low (i.e., the feedback is unlikely to be biased), β_{new} approaches 1, placing greater weight on the observed preference. As D_f increases, β_{new} decreases smoothly toward 0, reducing the influence of potentially biased feedback. The use of a logistic transform ensures a bounded mapping, while allowing for a tunable transition around the threshold θ.

The hyperparameters k and θ control the sensitivity of this mapping. The threshold θ determines the bias likelihood at which feedback begins to be substantially downweighted, while k governs how sharply this transition occurs. In practice, we select these parameters via a 2-D traversal of the validation set, selecting values that maximize the model's ability to distinguish between ground-truth-consistent and cognitively biased preferences. These hyperparameters may shift across datasets or annotator populations and reflect how susceptible to bias the annotator population is likely to be.

Importantly, D_f is defined with respect to the ***human's chosen response*** rather than the scenario as a whole. This distinction allows the model to differentiate between contexts that are conducive to bias where annotators fall into those biases and instances in which annotators nevertheless provide reliable feedback. When a scenario is bias-inducing but the annotator prefers an unbiased response, the corresponding D_f remains low, and the feedback is assigned a higher β_{new}. Such a scenario might occur in someone with statistical training who encounters the Linda example, who would be primed to recognize that having a larger subset than the original set is impossible. In the reverse case, we have situations like the patient with heart failure who presents to the hospital with chest pain. In this example, the context is not primed to trigger cognitive bias, but the doctor's response only mentions having seen asthma on the chart, indicating a likely anchoring bias. Considering the bias of the chosen response avoids discarding informative examples from difficult or ambiguous settings, and instead treats such cases as high-value signals for learning under challenging conditions.

3.1 DATA

To validate this intervention, we used two datasets designed to induce biased responses. The first is the BRU Dataset (Zhong et al., 2025) which was developed with a psychologist and a medical data

expert to ensure validity. This small dataset served as an exploratory small-scale case study, and consists of 205 multiple choice questions with a ground truth answer, covering 8 cognitive biases including anchoring bias and conjunction fallacy. The second dataset is the "Comprehensive Evaluation of Cognitive Biases in LLMs: Dataset" (CogBias dataset (Malberg et al., 2025)) that includes 30,000 choice selection test cases and 30 biases (Malberg et al., 2025). Stereotyping bias examples were excluded as they reflect social bias rather than a cognitive bias. Across both datasets, to be consistent with RLHF framing, we convert each example from the initial multiple choice options into a binary preference pair consisting of a ground-truth response and a bias-consistent response. The former aligns with the unbiased or intended evaluation of the task; this is the preference a human might arrive at if they employed deliberative reasoning instead of heuristic-driven reasoning (Guthrie et al., 2007; Kahneman, 2013–2011). The latter reflects the direction of error induced by a cognitive bias: for example, in the Linda problem, selecting an example that exhibits the conjunction fallacy (that it is more likely she is a bank teller and active in the feminist movement). Since neither dataset contained train/val/test splits, we split the data into train, val and test.

In the BRU dataset, the unbiased choice ground-truth labels are provided. We constructed bias-consistent responses by prompting a language model to select the response most aligned with a specified bias type among distractor answers given. An ablation showing roughly equivalent performance whether given the chosen responses from the language model or ones provided by a human annotator is provided in App. A. In the CogBias dataset, we use the provided bias metric to select the least biased option as the ground-truth response and the most biased option as the bias-consistent response.

To evaluate robustness to biased annotations, we simulate systematic bias in human feedback by constructing a dataset that consists of preference pairs where the biased response is more often (but not universally) preferred over the ground truth answer. Specifically, we construct this dataset by marking the bias-consistent response as preferred over the ground-truth-consistent response a tunable fraction (> 0.5) of the time. This models a common failure mode in RLHF, where annotators systematically favor more salient or cognitively appealing options rather than the correct ones. By default, we use a 3:1 ratio of bias-consistent to ground-truth-consistent labels in our experimentation, reflecting settings in which biased responses are more likely but not universal. We additionally vary this ratio from 1:1 to 5:1 in Section 4.3 to study sensitivity to the level of bias.

3.2 BIAS DETECTOR

The first step of our method is to estimate, for each training example, the likelihood that the observed feedback is influenced by cognitive bias. This estimate, denoted D_f, is used to modulate the contribution of each example during reward learning. This approach uses an LLM-based evaluator (though alternate methods could be used) to estimate the likelihood that a given preference comparison reflects cognitive bias. Here, we frame bias detection as a local judgment over a single comparison reflecting its contextual basis rather than a consistent aggregation across many examples. As we show empirically, the method remains effective even when the bias detector is imperfect, indicating that even coarse estimates of bias likelihood are sufficient to guide the intervention. Beyond the LLM-as-judge, much good prior work has been done in the field of bias detection (even without LLMs such as classifier (Jin et al., 2022) based models). It would be possible to replace the LLM-as-judge with any of these methods without fundamentally changing the takeaways of this paper. Though the LLM may inherit many of the same cognitive biases as humans when producing text, it also may have quite a few training examples in its corpus that specifically label cognitive biases and show examples of it. This makes the problem of detection easier than the problem of avoiding generating biased text as human feedback on identifying the bias will be much more likely to be unaffected.

To demonstrate that our approach is compatible with different bias detection strategies, we instantiate D_f using two LLM-based evaluators to score the likelihood that a given preference reflects the influence of cognitive bias. For the BRU dataset, we use ChatGPT 5.2 Thinking (OpenAI, 2025) to score the likelihood that a given preference reflects the influence of cognitive bias. For the CogBias dataset, owing to the size of the dataset, we used a local model as the evaluator, Mistral-7B-Instruct (Jiang et al., 2023). We construct D_f using a pairwise LLM-as-judge procedure. Given two candidate responses, the model is prompted to identify which response is more biased. To mitigate positional bias, we evaluate each pair in both orderings and aggregate the results. We then compute a probability by applying a softmax over the final-token logits of the two choices. Because bias de-

tection performance varies across bias types (i.e., a model may be able to reliably detect anchoring bias but not conjunction fallacies), we apply a lightweight calibration step. Specifically, we estimate a per-bias-type transformation using a minimal annotated held-out validation set (20 examples per bias in the CogBias dataset) to correct systematic directionality errors via sign inversion. Importantly, the LLM bias detector is not trained or finetuned with ground-truth correctness labels; the outputs are simply calibrated by this inversion. This design reflects a realistic setting in which large-scale bias annotations are unavailable, while still enabling effective estimation of D_f from a small amount of labeled data. These two instantiations provide evidence that the approach is not tightly coupled to a particular bias detection mechanism. Despite differences in model scale and prompting strategy, both yield effective estimates of D_f, suggesting the method is robust to variation in how likelihood is operationalized.

3.3 Reward Model & GRPO Finetuning

Given the training set consists of a mix of groundtruth-consistent and bias-consistent preference pairs that simulate cognitively-biased annotation, we train a reward model with a loss function that treats each pair's weight as conditional on how biased the chosen response appears to be. Concretely, we replace the global Bradley–Terry rationality constant β with a per-datapoint weight β_i that decays smoothly when the chosen response's bias score D_f is high.

The reward model is initialized from Mistral-7B-v0.1 (Jiang et al., 2023). The θ and k parameters are selected via a two dimensional sweep to maximize accuracy over the validation set. The final k, θ values per dataset, as well as reward model training hyperparameters are in Appendix D and E. The models were then finetuned with Group Relative Policy Optimization (GRPO) (Shao et al., 2024), where rewards were determined by a frozen reward model. All hyperparameters were held identical across runs for the baselines and debiased models. Our standard baseline for the finetuning was to use $\beta = 1.0$ (Barnett et al., 2023; Ghosal et al., 2022). This simulates annotators being equally rational in all scenarios. We also tested a variety of fixed β values (0.1, 0.5, 0.9) and a random β to confirm that there was not a hidden fixed β value that was best or that per-example modulation was the key to the improvement mechanism.

Testing was conducted on a held out test split from each dataset. We generated n samples per prompt in the test set to reduce dependence on a random draw. These answers were then compared to the ground truth answers and those that matched the ground truth were marked as correct. Our main metric to assess the performance improvements of dynamic β is the accuracy (the direct ground-truth rate) since it is RM independent. A significance level of $\alpha = 0.05$ was used globally.

4 Evaluation on the BRU & CogBias Datasets

We evaluate the performance of this method on two datasets. One is a small scale study on the BRU dataset (205 questions total, 21 in the test split) to validate the promise of this method. The other is the CogBias dataset (29,000 test cases). Across these experiments, we found support for the hypothesis that dynamic β selection can reduce an LLM's replication of cognitive biases.

4.1 Reduction in bias propagation & preservation of general performance

To investigate whether the intervention could reduce bias propagation, we compared the performance of the debiased model with the base model. Despite exposure to a substantial number of biased samples, the debiased model still outperforms the base model (which was never exposed to the biased data) on the test set of both the CogBias and BRU dataset. On the BRU dataset, the debiased model improves by 8% over the base vanilla (untrained) LLM despite having been finetuned on $3:1$ cognitively biased data, though it doesn't reach $\alpha < 0.05$ with $n = 21$ prompts. The CogBias results show significant improvement of 25.2% (95% CI $[+23.3, +26.9]$, $d = 0.51$). This lends support to our claim of a reduction in bias propagation through this dynamic β intervention. See figure 2.

Further, this increase in performance on cognitive bias tasks does not come at the cost of general performance. To test this, we compared the vanilla LLM and the debiased model (trained with the CogBias dataset and the dynamic β intervention) on two baselines: the TruthfulQA dataset (Lin et al., 2022), and the MMLU benchmark (Hendrycks et al., 2021). On the TruthfulQA dataset, it did not significantly degrade performance (non-significant change of -0.5% with 95% CI $[-0.011, 0.000]$). For MMLU, it had an essentially undetectable performance drop of -0.63% with 95% CI $[-0.0111, -0.0016]$.

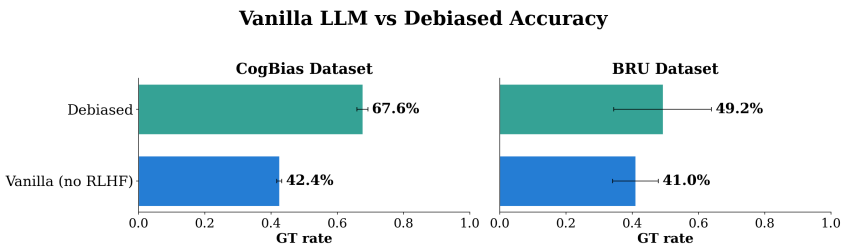

Figure 2: For CogBias (left), the debiased model's accuracy of 67.6% correctly choosing ground truth is a significant improvement over the vanilla un-finetuned LLM with ground truth rate 42.4%.

4.2 Robustness to Biased Feedback

The performance on both datasets shows that adding a model of cognitive bias to the reward model tuning makes it robust to even high levels of human error due to cognitive bias; see Figure 3. When finetuned on a dataset where each question had three biased and one unbiased responses to model the distribution of cognitive bias, we found that our debiased method (67.6% accurate for CogBias, 49.2% for BRU) far outperformed the baselines (Cohen's d ¿= 1.48 for CogBias) that trained on the same data but with fixed values for $\beta \in \{0.1, 0.5, 0.9, 1.0\}$ or where β was assigned a random value for each question. All baselines produce GT rates roughly indistinguishable from the collapsed baseline ($\approx 9 - 10\%$ for BRU dataset, $\approx 1.7\%$ for CogBias dataset). The baselines' degradation in performance is expected, as each has been exposed to imperfect data and has learned to predict that imperfect data. Only our dynamic β improves robustness with biased human data, with bootstrap $p < 0.01$ for all baselines in both datasets (for BRU fixed-β: one-sided). This is promising, as the system learned to avoid biases without structured heuristics or privileged information about the bias.

4.3 Robustness to Different Levels of Biased Feedback

To represent the issue of humans perhaps having different frequencies of bias (which as a baseline can be $\approx 80\%$ in humans for problems like the Linda problem (Tversky & Kahneman, 1983)), we showed that the method is robust at four different ratios of bias as shown in Table 1. To study this, we finetuned with the baseline $\beta = 1.0$ and dynamic β on training data with different ratios of preferring biased or unbiased responses. We trained both on fixed ratios (1:1, 3:1, 5:1) and on one dataset where the number of biased pairs was randomly selected for each question. In all cases, despite finetuning on preference pairs that did not match people's unbiased preferences, the debiased model is still more accurate in selecting more rational responses than the un-finetuned model.

4.4 Robustness to Model Architectures

We also tested to assess whether this intervention applied across model architectures. We tested with Mistral-7b-0.1 (Jiang et al., 2023), Qwen3-8B-Base (Team, 2025) and Microsoft phi-4 (Abdin et al., 2024). Due to compute limitations, Qwen3-8B-Base and phi-4 were quantized to use a 4-bit policy

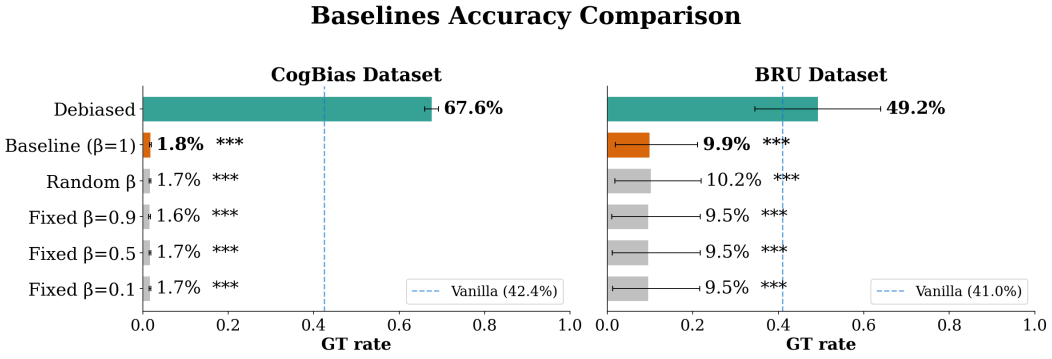

Figure 3: Comparison of CogBias (left) and BRU (right) datasets accuracy compared across different baselines ($\beta \in \{1, 0.9, 0.5, \texttt{random}\}$) and our dynamic β method (Debiased, top row). Our method shows a strong improvement in accuracy compared to the baselines. With either fixed or random choices of β, these models were highly susceptible to cognitive biases.

Table 1: CogBias: Comparing different ratios of cognitively biased preferences to ground-truth preferences by ground-truth answer rate and effect size. All values use n=10 samples per prompt over 2,900 test prompts. Intervention shows strong improvement across different levels of cognitive bias. BL GT: baseline ground truth. DB GT: debiased ground truth.

Ratio	BL GT%	DB GT%	GT diff	Cohen's d	p-val
1:1	47.9%	82.7%	+34.8%	0.81	<0.01
3:1	1.8%	67.6%	+65.8%	1.48	<0.01
5:1	1.8%	53.5%	+51.7%	1.08	<0.01
random:3	49.3%	83.7%	+34.4%	0.64	<0.01

quantization for GRPO (and reward training for phi-4); hyperparameters in Appendix F. Across all models, the debiased method showed a statistically significant improvement (Cohen's d, Table 2).

4.5 Comparison to data filtering

We also compare the impact of varying β with a more standard data filtering approach of dropping low-quality (i.e. highly biased) data. When preferences labeled as more biased than the threshold used for our method (θ) were dropped from the reward model training with the CogBias dataset, the resulting model achieved 52.4% accuracy rate on selecting ground truth answers in the CogBias test set. While this outperformed the vanilla model (42.4%), it was still lower than the performance of our debiased model which achieved 67.6% accuracy. We hypothesize this is because our method allows some signal to come through from even biased pairs rather than discarding it completely.

4.6 Robustness to noisy judging

To confirm the bias detection worked on the BRU dataset, we tested the pairwise accuracy ($D_f(cb) > D_f(gt)$) of the ChatGPT 5.2-Thinking model; this was 98.0% accurate. All 8 types of bias in this dataset scored above 93% accuracy in detection and therefore did not require calibration. On the CogBias dataset, at training time, the per-pair β values were able to cleanly separate ground truth pairs from cognitive biased ones. On the held-out validation set, the calibrated out-of-the-box Mistral-7b-0.1-Instruction-Tuned (Jiang et al., 2023) judge achieves 82.2% pairwise accuracy (Cohen's $d = 0.696$) across bias types, with $24/29$ types significantly above chance (Wilcoxon $p < 0.001$).

Given the intervention relies on bias scores from a potentially noisy judge, it is important that the method can tolerate noise when the judge is wrong. To probe this, we did an algebraic test to perturb the bias scores by adding zero-mean Gaussian noise in logit space to bias scores and recalculate β on the $n = 2900$ CogBias validation pairs with the $3:1$ ratio of cb to gt. When the judge is correct ($D_f(cb) > D_f(gt)$), $\beta_{cb} < \beta_{gt}$ correctly downweights the cb-chosen pairs; when the judge is incorrect, it flips so $\beta_{cb} > \beta_{gt}$ incorrectly downweights the ground-truth-chosen pairs. We summarize this trade-off as the benefit/damage ratio: avg($\beta_{gt} - \beta_{cb}$) on judge-correct pairs divided by $avg(\beta_{cb} - \beta_{gt})$ on judge-flipped pairs. A ratio greater than 1.0 indicates that the β-modulation is helping more than it is harming. Across the judge accuracy (induced by the noise addition) from 83% (no noise) down to 57% (near chance), the benefit/damage ratio stays > 1 ($1.84 \rightarrow 1.12$); we conclude the intervention is therefore relatively insensitive to adjudication noise.

4.7 Transferring learned knowledge of cognitive biases to other datasets

Does the model learn a transferrable representation of the cognitive biases? We tested the debiased model trained on the BRU dataset on the subset of test prompts from the CogBias dataset that shared

Table 2: Cross-architecture robustness on CogBias dataset. Each row is a base model finetuned with the beta modulation pipeline; OTB is the un-tuned base. All $n = 2900$ prompts $\times$ 10 samples. CIs from 10,000-iteration paired bootstrap; p-values from one-sided Wilcoxon signed-rank test ($H_1: \Delta > 0$); Cohen's d on per-prompt paired differences.

Base model	OTB GT	Debiased GT	Δ (pp)	95% CI	Wilcoxon p	Cohen's d
Mistral-7B-v0.1	42.4%	67.6%	+25.2	[+23.3, +26.9]	<0.001	+0.509
Qwen3-8B-Base	45.8%	60.0%	+14.2	[+12.5, +15.9]	<0.001	+0.31
phi-4	21.6%	51.5%	+29.9	[+28.5, +31.7]	<0.001	+0.61

a common cognitive bias: anchoring bias (100 samples, randomly assigned equal numbers of A/B labels). We find that the BRU-trained debiased model is successfully able to transfer that knowledge of anchoring bias onto the new dataset, having a 44.3% GT accuracy improving over the vanilla LLM's 37.7% GT accuracy (p=0.001, paired Wilcoxon). Given the small scale of the BRU dataset, it is surprising that it was able to transfer structural knowledge. This suggests that this method has promise for being used more broadly on datasets drawn from a different distribution.

5 Limitations

One limitation of this method is its dependence on the LLM-as-judge's ability to detect bias. Because detector accuracy varied across bias types, we examined its association with downstream performance after finetuning. On CogBias, all 18 bias types with judge accuracy above 87% showed improvements exceeding 55 percentage points. For the four bias types with judge accuracy below 50%, three showed improvements and one remained unchanged, suggesting that judge accuracy alone does not fully explain downstream outcomes. Across bias types, judge accuracy was positively associated with downstream improvement (Spearman $\rho = 0.84$, $p < 0.01$; Fig. 4), consistent with detector quality constraining the method's effectiveness. Whether the benefits observed with less accurate judges extend to other settings remains an open question. Although we did not observe performance regressions in these experiments, inaccurate bias estimates could suppress useful feedback or amplify biased judgments. Establishing when such errors undermine downstream learning is necessary to assess the method's reliability beyond the settings studied here.

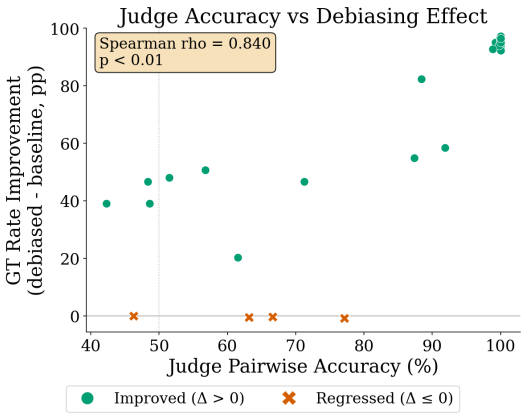

Figure 4: Does more accurate judging (i.e., D_f) result in more effective debiasing? Judge accuracy compared to ground truth rate improvement (debiased - baseline). All low-performing (~ 0 improvement) have judge accuracy under 80%.

6 Conclusion

In this work, we revisit an old assumption in preference-based reward learning: that deviations from reward-consistent behavior can be captured by a fixed, global rationality parameter. Although mathematically convenient, this assumption overlooks the systematic biases and contextual effects that shape expressed preferences. The same concern extends to other signals of human supervision, including demonstrations, binary feedback, and corrections, which can also be interpreted through a reward-rational model of human decision-making (Jeon et al., 2020). A noise parameter allows these models to accommodate human errors without accounting for why they occur. Yet those errors often follow patterns that can inform how feedback should be interpreted.

We use estimates of cognitive bias to vary β across comparisons, and find that this change improves downstream performance on cognitive bias tasks, even when bias detection is imperfect. In our experiments, adjusting β also outperformed the tested filtering approach, suggesting that comparisons flagged as biased can still contribute useful information. Together, these findings show that knowledge about the reliability of feedback can improve learning without requiring either the removal of all suspect comparisons or a complete model of human decision-making.

Varying β addresses how strongly a comparison informs reward learning, but does not explicitly model the reasoning that produced it. A natural extension is to learn β as a function of the annotator, task, and comparison. Models of particular cognitive biases could go further by predicting which alternatives people systematically favor and why. Interventions on the dataset, such as resampling or asking annotators to reconsider their judgments, offer additional ways to address these errors. We invite future research examining when treating feedback as less reliable is sufficient, and when recovering intended objectives requires a more explicit account of how judgment is distorted.

AI USE STATEMENT

In this work, we used generative AI tools for selecting responses from a dataset, providing feedback on research methodology or experiments, and implementing methods (writing code for experiments). We have not used generative AI tools for proposing or refining hypotheses, and the rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for writing code, including code to create visualizations, and surface relevant literature, which we then went on to carefully assess before citing in this work. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

There is, of course, always a moral dilemma when we work on the question of how can we reinterpret humans' expressed preferences: what does it mean to say that we are able to recover the true or accurate preference when that preference is not the one we express or choose? We run the risk of paternalism when we allow a model to determine truth despite explicit feedback. However, there is a key distinction here, as we are only applying this reinterpretation in situations where long-running psychological experiments have shown that humans make less rational choices. The biases and heuristics that may be helpful for humans to short circuit computations should not be unthinkingly passed on to a model that people will then use and be themselves impacted by cognitive bias (i.e., confirmation bias) to accept its prescriptions and responses with little questioning.

Less abstractly, the "Comprehensive Evaluation of Cognitive Biases in LLMs: Dataset" (CogBias dataset (Malberg et al., 2025)) that includes 30,000 choice selection test cases and 30 biases was designed only for testing models for bias (Malberg et al., 2025); we received special permission from the authors to finetune with this dataset to proactively reduce replicated cognitive bias in models.

REPRODUCIBILITY STATEMENT

On publication, we will release the code we used to train the models. Hyperparameters used are provided in the appendix.

REFERENCES

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. Phi-4 technical report, 2024. URL https://arxiv.org/abs/2412.08905.

Peter Barnett, Rachel Freedman, Justin Svegliato, and Stuart Russell. Active reward learning from multiple teachers, 2023.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, J'er'emy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro J Freire, Tony Wang, Samuel Marks, Charbel-Raphaël Ségerie, Micah Carroll, Andi Peng, Phillip J. K. Christoffersen, Mehul Damani, Stewart Slocum, Usman Anwar, Anand Siththaranjan, Max Nadeau, Eric J. Michaud, Jacob Pfau, Dmitrii Krasheninnikov, Xin Chen, Lauro Langosco di Langosco, Peter Hase, Erdem Biyik, Anca D. Dragan, David Krueger, Dorsa Sadigh, and Dylan Hadfield-Menell. Open problems and fundamental limitations of reinforcement learning from human feedback. *ArXiv*, abs/2307.15217, 2023.

Lawrence Chan, Andrew Critch, and Anca Dragan. Human irrationality: both bad and good for reward inference, 2021. URL https://arxiv.org/abs/2111.06956.

Vanessa Cheung, Maximilian Maier, and Falk Lieder. Large language models show amplified cognitive biases in moral decision-making. *Proceedings of the National Academy of Sciences*, 122 (25):e2412015122, 2025. doi: 10.1073/pnas.2412015122.

Samantha D'Alonzo, Frauke Kreuter, and Serena Booth. Helpful, harmless, honest? rlhf as survey design and content moderation. In *Proceedings of the 2026 ACM Conference on Fairness, Accountability, and Transparency (FAccT '26)*. ACM, 2026. ISBN 979-8-4007-2596-8/2026/06. doi: 10.1145/3805689.3806444.

Oliver Daniels-Koch and Rachel Freedman. The expertise problem: Learning from specialized feedback. In *NeurIPS ML Safety Workshop*, 2022.

Gaurav R. Ghosal, Matthew Zurek, Daniel S. Brown, and Anca D. Dragan. The effect of modeling human rationality level on learning rewards from multiple feedback types. In *AAAI Conference on Artificial Intelligence*, 2022.

Chris Guthrie, Jeffrey J Rachlinski, and Andrew J Wistrich. Blinking on the bench: How judges decide cases. *Cornell L. Rev.*, 93:1, 2007.

Stephane Hatgis-Kessell, W Bradley Knox, Serena Booth, Scott Niekum, and Peter Stone. Influencing humans to conform to preference models for RLHF. *arXiv preprint arXiv:2501.06416*, 2025.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Joey Hong, Kush Bhatia, and Anca Dragan. On the sensitivity of reward inference to misspecified human models, 2023. URL https://arxiv.org/abs/2212.04717.

Tiffany Horter, Andrew Markham, Niki Trigoni, and Serena Booth. Should robots comply with our instructions or intentions? In *Proceedings of the 21st ACM/IEEE International Conference on Human-Robot Interaction*, HRI '26, pp. 246–254, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400721281. doi: 10.1145/3757279.3785553. URL https://doi.org/10.1145/3757279.3785553.

Tom Hosking, Phil Blunsom, and Max Bartolo. Human feedback is not gold standard, 2024. URL https://arxiv.org/abs/2309.16349.

Itay Itzhak, Gabriel Stanovsky, Nir Rosenfeld, and Yonatan Belinkov. Instructed to bias: Instruction-tuned language models exhibit emergent cognitive bias, 2024.

Hong Jun Jeon, Smitha Milli, and Anca D. Dragan. Reward-rational (implicit) choice: A unifying formalism for reward learning. *ArXiv*, abs/2002.04833, 2020.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Zhijing Jin, Abhinav Lalwani, Tejas Vaidhya, Xiaoyu Shen, Yiwen Ding, Zhiheng Lyu, Mrinmaya Sachan, Rada Mihalcea, and Bernhard Schölkopf. Logical fallacy detection, 2022. URL https://arxiv.org/abs/2202.13758.

Daniel Kahneman. Thinking, fast and slow. In *Thinking, Fast and Slow*. Farrar, Straus and Giroux, New York, first paperback edition. edition, 2013–2011. ISBN 978-0-374-53355-7.

Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. In *Handbook of the fundamentals of financial decision making: Part I*, pp. 99–127. World Scientific, 2013.

Yuhe Ke, Rui Yang, Sui An Lie, Taylor Xin Yi Lim, Yilin Ning, Irene Li, Hairil Rizal Abdullah, Daniel Shu Wei Ting, and Nan Liu. Mitigating cognitive biases in clinical decision-making through multi-agent conversations using large language models: Simulation study. *J Med Internet Res*, 26:e59439, Nov 2024. ISSN 1438-8871. doi: 10.2196/59439. URL https://www.jmir.org/2024/1/e59439.

W. Bradley Knox, Stephane Hatgis-Kessell, Serena Booth, Scott Niekum, Peter Stone, and Alessandro Allievi. Models of human preference for learning reward functions, 2023.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods, 2022. URL https://arxiv.org/abs/2109.07958.

Olivia Macmillan-Scott and Mirco Musolesi. (Ir)rationality in AI: State of the art, research challenges and open questions. *Artificial Intelligence Review*, 58:352, 2023.

Simon Malberg, Roman Poletukhin, Carolin M. Schuster, and Georg Groh. A comprehensive evaluation of cognitive biases in llms, 2024.

Simon Malberg, Roman Poletukhin, Carolin Schuster, and Georg Groh Groh. A comprehensive evaluation of cognitive biases in LLMs. In Mika Hämäläinen, Emily Öhman, Yuri Bizzoni, So Miyagawa, and Khalid Alnajjar (eds.), *Proceedings of the 5th International Conference on Natural Language Processing for Digital Humanities*, pp. 578–613, Albuquerque, USA, May 2025. Association for Computational Linguistics. ISBN 979-8-89176-234-3. doi: 10.18653/v1/2025.nlp4dh-1.50. URL https://aclanthology.org/2025.nlp4dh-1.50/.

Ian M Moore, Eura Nofshin, Siddharth Swaroop, Susan Murphy, Finale Doshi-Velez, and Weiwei Pan. When and why hyperbolic discounting matters for reinforcement learning interventions. In *Reinforcement Learning Conference*, 2025.

OpenAI. Update to GPT-5 System Card: GPT-5.2. https://openai.com/index/gpt-5-system-card-update-gpt-5-2/, 2025.

Stuart J. Russell. Human compatible : AI and the problem of control. In *Human Compatible : AI and the Problem of Control*. Penguin Books, London, 2019. ISBN 978-0-241-33524-6.

Rohin Shah, Noah Gundotra, Pieter Abbeel, and Anca D. Dragan. On the feasibility of learning, rather than assuming, human biases for reward inference, 2019.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Shivam Singhal, Cassidy Laidlaw, and Anca Dragan. Scalable oversight by accounting for unreliable feedback.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Amos Tversky and Daniel Kahneman. Extensional versus intuitive reasoning: The conjunction fallacy in probability judgment. *Psychological review*, 90(4):293–315, 1983. ISSN 0033-295X.

Taku Yamagata, Tobias Oberkofler, Timo Kaufmann, Viktor Bengs, Eyke Hüllermeier, and Raul Santos-Rodriguez. Relatively rational: Learning utilities and rationalities jointly from pairwise preferences. August 2024. URL https://icml.cc/virtual/2024/events/workshop. ICML 2024 Workshop on Models of Human Feedback for AI Alignment.

H. Zhong, L. Wang, Wenting Cao, and Zeyuan Sun. Balancing rigor and utility: Mitigating cognitive biases in large language models for multiple-choice questions. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 47. Cognitive Science Society, 2025. URL https://escholarship.org/uc/item/2vr690cx.

Table 3: Comparison of performance of debiased and baseline models on GT when trained with LLM-given labels of most-cognitively biased answer compared to the human-given labels.

Labeler	Debiased GT rate	Baseline GT rate (β = 1.0)
Human-labeled	46.8%	10.2%
LLM-labeled	49.2%	9.9%

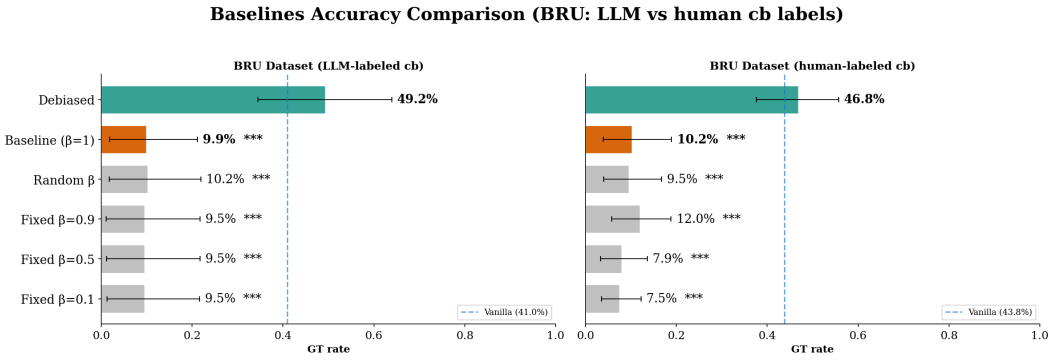

Figure 5: Comparison of BRU with LLM-labels of cognitively biased (left) and BRU with human-given labels (right) datasets accuracy compared across different baselines ($\beta \in \{1, 0.9, 0.5, \texttt{random}\}$) and our dynamic β method (Debiased, top row). Our method shows a strong improvement in accuracy compared to the baselines. With either fixed or random choices of β, these models were highly susceptible to cognitive biases.

A BRU WITH HUMAN LABELING

To settle the question of circularity with how we select the most cognitively-biased answer for the BRU dataset, we also constructed bias-consistent responses with a human annotator marking the response most aligned with a specified bias type. To match the scenario based format of the CogBias dataset, 15 questions were dropped from the BRU dataset that asked the person to name the psychological effect at play rather than inducing a bias, leaving 190 questions total with 19 in the test split. We excluded questions that simply asked what cognitive bias was at play, as those questions are not likely to induce cognitive bias themselves. On the same questions, the human labeled responses matched the LLM choice 69.5% of the time, far above chance (25%).

The bias detector method used for the human-labeled data was the same as for the CogBias dataset. This led to similar performance as the LLM-selected cognitively biased responses. The debiased method was 46.8% accurate for the human-labeled BRU dataset, while all baselines produced GT rates similar to the collapsed baseline ($\approx 7.5 - 12\%$).

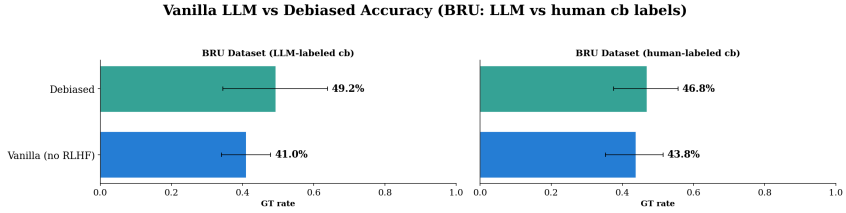

Figure 6: Comparison of the untrained vanilla LLM compared to the debiased model's accuracy for both models trained with the different selections of cognitively biased options for the BRU dataset.

B β_{new} FORMULA ABLATION

As an alternative to equation 1, we instead considered the following formula for setting β_{new}:

$$\beta_{new} = \texttt{logistic}(k * D_s * (c - D_f))$$

where D_s is the likelihood from the bias detector that the situation is one in which bias may be present, and D_f is the likelihood that the given feedback is biased. This uses a logit transform to make it smooth around the ends. However, we determined using an ablation study on the BRU dataset with the LLM bias detector (shown in Table 4) that the inclusion of the D_s factor in fact degraded performance. This is likely because the scenario is essentially repeated as context for the bias detector in the D_f factor but lacks the information about whether a biased choice was actually made or not; as discussed throughout the text, if a preference reflects a rational choice in the face of bias, that information is highly useful for learning and should not be discarded through a low β. Using the D_s factor only while holding the D_f factor constant at the pooled mean of D_f made it collapse to near the baseline, indicating the potential weakness of a β that relies only on the scenario.

Table 4: Mistake-function ablation on BRU dataset. "Full" uses both s (scenario) and r (response) bias scores; "response only" holds s=1; "scenario only" pools gt/cb response scores to their mean. Diff and p are vs. the GRPO baseline (β=1.0, GT = 9.9%); Cohen's d is paired across test prompts; p is from a 10 000-sample paired bootstrap.

Variant	GT %	CB %	NA %	Δ GT	Cohen d	p
Both	50.2	47.1	2.7	+40.3	0.99	<0.001
Response only (s=1)	54.6	43.4	2.0	+44.7	1.30	<0.001
Scenario only (pooled r)	9.4	89.4	1.1	−0.5	−0.11	0.716

C IMPACT OF DEBIASING BY BIAS TYPE

Here we present the results of the debiased model and consider the effect on each bias type. 95% CI width is reported.

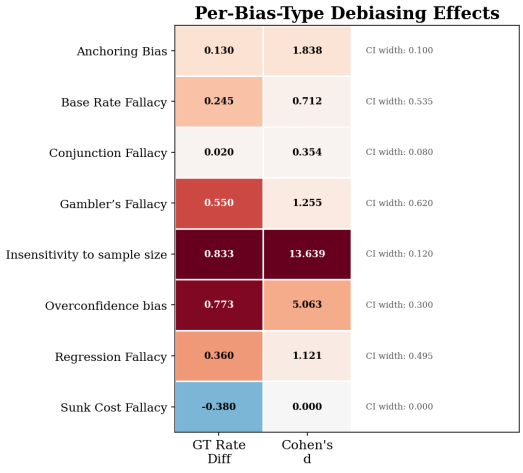

Figure 7: Heat map of debiasing effects by bias type for BRU dataset.

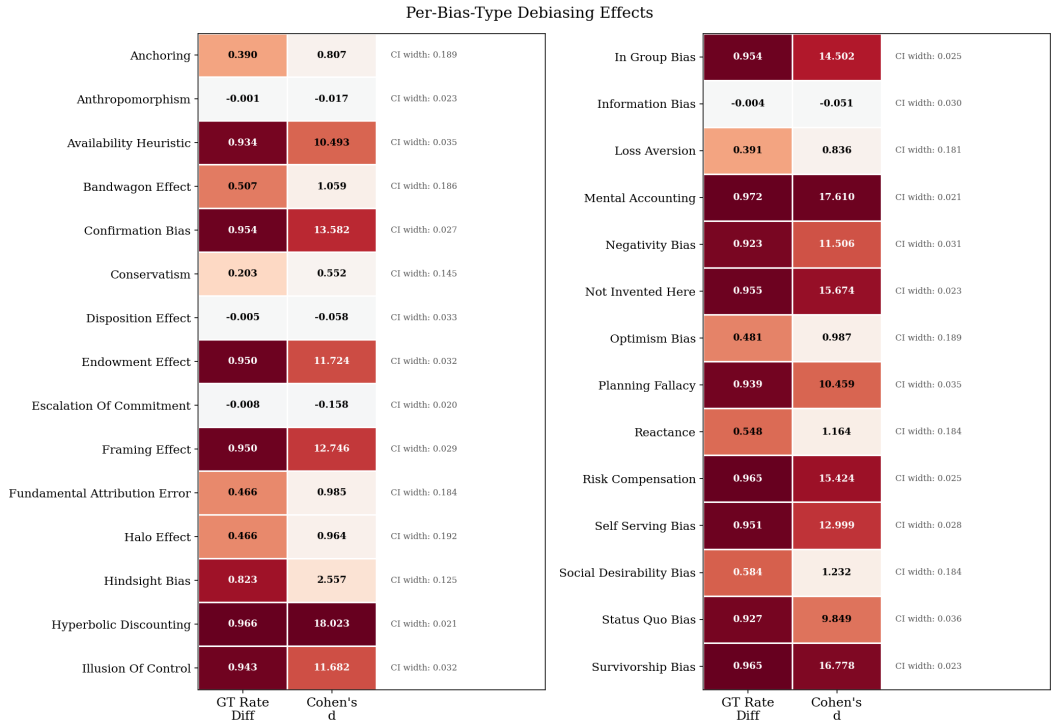

Figure 8: Heat map of debiasing effects by bias type for CogBias dataset.

D TRAINING HYPERPARAMETERS FOR COGBIAS DATASET

On publication, we will release the code we used to train the models.

All values reflect `config/judge.yaml` and the source files under `src/`.

Setting	Value
Base model	`mistralai/Mistral-7B-v0.1`
`torch_dtype`	`bfloat16`
Attention impl.	`flash_attention_2` (CUDA); SDPA fallback (MPS/CPU)
RM quant. during GRPO	4-bit (`bnb_4bit_compute_dtype=bf16`)
Policy quant. during GRPO	full bf16 (4-bit only if `model.use_4bit=true`)

Table 5: CogBias: Base-model loading and numeric precision.

Setting	Value
`r`	16
`lora_alpha`	32
`lora_dropout`	0.05
`target_modules`	`q_proj, k_proj, v_proj, o_proj`
`bias`	`none`
`task_type`	`SEQ_CLS` (RM) / `CAUSAL_LM` (policy)
`modules_to_save`	none

Table 6: CogBias: LoRA adapter configuration. Identical for the reward model and the policy except for `task_type`.

Setting	Value
Format	`dataset2` (control/treatment, 29 bias types)
Splits	`data/dataset2_{train,val,test}.csv`
`option_mode`	`ab` (binary A/B reduction)
`stratify_by`	`bias_type`
`train_sample_frac`	1.0 (no downsampling)
`reduce_test`	`true`
Shuffle / A–B alternation seed	42
Stratified-downsample seed	42 (used only when `train_sample_frac<1.0`)
Score calibration	applied at load time from `outputs/judge/score_calibration.json` if present

Table 7: CogBias: Data loading.

D.1 Distributed-training scaling

When the pipeline is launched under accelerate / torch DDP with `WORLD_SIZE>1` (e.g. `scripts/run_grpo_parallel.py`), `maybe_scale_batch_for_ddp` (`main.py:42--55`) preserves the single-GPU effective batch size by:

1. dividing `gradient_accumulation_steps` by `WORLD_SIZE` when cleanly divisible (memory-neutral); otherwise
2. dividing per-device `batch_size` by `WORLD_SIZE`.

The rule is applied to both the `training.*` (GRPO) and `reward_model.*` sections so that ablations launched on different machines remain comparable.

D.2 Evaluation

Setting	Value
Formula	$\beta = \sigma\big(\mathrm{rc} \cdot (\tau - s)\big)$, s = per-response bias score
`k` (rc in code)	30.0
`threshold` (τ)	0.40 (`auto` falls back to pooled-median)
Range	$\beta \in [0, 1]$, no clamping
Baseline-branch override	$\beta = 1.0$ for every pair
Bias-score source	per-example LLM-judge scores in CSV

Table 8: CogBias - Mistake function (dynamic β, debiased branch only).

Setting	Value
`pair_sampling`	`fixed` (default)
`gt_pairs` (correct)	1 per question
`cb_pairs` (biased)	3 per question
Effective ratio (correct:biased)	1:3
Random-mode seed (if enabled)	42 (validation: 43)
Suffix appended to prompt	`"\nChoose A or B.\nAnswer: "`

Table 9: CogBias - Pair generation per training row.

Setting	Value
Loss	$-\log \sigma\big(\beta \cdot (r_{\text{chosen}} - r_{\text{rejected}})\big)$
Epochs	3
Per-device batch size	8
`gradient_accumulation_steps`	4 (inherits `training.*`)
Effective batch (single-GPU)	32
Learning rate	1×10^{-5}
`max_length`	512
`gradient_checkpointing`	`true`
`bf16` / `fp16`	auto from `torch_dtype` (bf16=true)
`save_strategy`/`_steps`/`_total_limit`	`steps` / 50 / 2
`remove_unused_columns`	`true`
`center_rewards_coefficient`	`None` (term disabled)
Eval-set cap	first 200 validation samples
`report_to`	`tensorboard`

Table 10: CogBias: Reward-model training arguments. Both baseline ($\beta = 1$) and debiased branches share these; only the per-row β column differs.

Setting	Value
Procedure	mean RM logit over `prompt+suffix+gt` and `prompt+suffix+cb` for every train row
Batch size	8
Tokenization `max_length`	512
Cache	`<rm_path>/calibration.json` (skipped if present)
Used as	`reward = (rm_score − offset) + format_bonus`

Table 11: CogBias - Reward-model offset calibration (pre-GRPO). Zero-centering offset computed once per RM and cached; ensures the GRPO advantage term is not biased by the RM's mean logit.

Setting	Value
`num_generations` per prompt	4
`max_completion_length`	15 tokens
`temperature`	0.7
`beta` (KL coef. vs ref. policy)	0.04
`num_iterations`	1
`loss_type`	`grpo`
Learning rate	1×10^{-5}
Per-device batch size	4
`gradient_accumulation_steps`	4
Effective batch (single-GPU)	16
Epochs	2
`gradient_checkpointing`	`true`
`train_order`	`debiased_first`
Tokenizer `padding_side`	`left` (forced for generation)
Seed	42 (default arg of `_grpo_finetune`)
`save_strategy`/`_steps`/`_total_limit`	`steps` / 50 / 2
`report_to`	`tensorboard`

Table 12: CogBias - GRPO trainer configuration.

Setting	Value
Answer regex (AB mode)	`[ABab]`
Reward (parseable answer)	`(rm_logit − offset) + format_reward_bonus`
Reward (unparseable)	`−format_reward_bonus`
`format_reward_bonus`	2.0
RM scoring `max_length`	512 (truncation, dynamic padding)
Clipping / normalization	none

Table 13: CogBias - GRPO Reward function

Setting	Value (HF default; not overridden)
Optimizer	`adamw_torch`
`lr_scheduler_type`	`linear`
`warmup_ratio` / `warmup_steps`	0 / 0
`weight_decay`	0.0
`max_grad_norm`	1.0
AdamW (β_1, β_2) / ϵ	$(0.9, 0.999)$ / 1×10^{-8}

Table 14: CogBias - Trainer defaults inherited from `transformers.TrainingArguments`; not set in `config/judge.yaml`.

Setting	Value
`logging.log_every_n_steps`	10
`logging.tensorboard`	`true`
`evaluation.n_samples`	10 generations / prompt
`evaluation.max_new_tokens`	15

Table 15: CogBias - Logging and post-training evaluation

E TRAINING HYPERPARAMETERS FOR BRU DATASET

BRU dataset is run via `config/default.yaml`:

All settings not listed below match the CogBias dataset tables.

Setting	Value
Source	single CSV `data/BRUdataset3.csv`
Split policy	stratified `train_test_split` (seed 42) on `bias_type`
`train` / `val` / `test` ratio	0.8 / 0.1 / 0.1
Score calibration applied	no – `_apply_score_calibration` is dataset 2 only
Judge held-out exclusion	not applied - no calibration

Table 16: BRU dataset - Data

Setting	Value
`rationality_constant` (rc)	100.0
`threshold` (τ)	`auto` – median of all training response bias scores (label-blind)

Table 17: BRU dataset - Mistake function

Setting	Value
Epochs	4
Per-device batch size	4
`gradient_accumulation_steps`	1
Effective batch (single-GPU)	4

Table 18: BRU dataset - Reward-model training.

Setting	Value
Per-device batch size	16
`gradient_accumulation_steps`	1
Effective batch (single-GPU)	16
`num_generations` per prompt	8
`evaluation.n_samples`	50 generations / prompt

Table 19: BRU dataset - GRPO finetuneing and evaluation

F ROBUSTNESS HYPERPARAMETERS

For Qwen3-8B-Base run:

```
"model": {"use_4bit": True},
"training": {"batch_size": 1, "gradient_accumulation_steps": 16},
"grpo": {"num_generations": 8, "temperature": 1.0},
```

G COMPUTE RESOURCES

All experiments were able to be run on a maximum of 4 A10s. To reproduce an experiment where a model passes through the reward model training and then GRPO from end to end to be finetuned would take approximately 36 hours on 2 A10s. Multiply this number by the number of variants present in the experiment for an estimate of wall-clock time per experiment.