BA-DPO: BIAS-ADJUSTED DIRECT PREFERENCE OPTIMIZATION FOR LANGUAGE MODEL ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Preference-based alignment methods such as Direct Preference Optimization (DPO) use pairwise preferences labeled by human annotators to fine-tune language models. However, annotators carry systematic biases toward some attributes: a name that signals a gender or an ethnicity, a persona, a language variety, a formatting convention, or length. If not properly addressed, these systematic biases can be absorbed and amplified during alignment. Existing methods address length bias or annotator disagreement, but fail to eliminate biases toward arbitrary attributes. To address this limitation, we propose *Bias-Adjusted DPO* (BA-DPO), a generalization of DPO that adds one bias parameter per annotator toward responses carrying a declared attribute. We prove that the objective is convex in the bias parameters and that the votes identify each annotator's bias up to a shared constant. The remaining constant is what fixes the aligned model's attribute rate: by default the rate of the reference model, or a target rate, which we use to bring a biased policy to statistical parity. On a corpus with planted biases, DPO drives the attribute from a balanced start to probability 0.96 and BA-DPO removes 81 to 95% of that shift; on MultiPref with real annotators it removes about half of DPO's lengthening. Both hold at 0.5B with full fine-tuning and at 8B with LoRA, at no higher KL than DPO and no loss in judged quality.

1 INTRODUCTION

Preference-based alignment has become the dominant paradigm for shaping language model behavior from pairwise preference labels produced by human annotators. Standard techniques, whether they rely on reinforcement learning with a reward model (Christiano et al., 2017; Ouyang et al., 2022; Bai et al., 2022) or optimize a cross-entropy loss over preference pairs directly, as in Direct Preference Optimization (DPO) (Rafailov et al., 2023), inherit the foundational assumption of the *Bradley-Terry model* (BT) (Bradley & Terry, 1952): that every human-annotated pairwise comparison is a noisy but unbiased measurement of the quality gap between two responses. This assumption rarely holds. Human evaluators carry systematic preferences for attributes unrelated to quality, such as a name that signals a gender or an ethnicity (Bertrand & Mullainathan, 2004), a persona, a language variety, a formatting convention, or response length (Park et al., 2024; Lu et al., 2024; Liu et al., 2024). Because standard preference optimization treats every label as evidence of quality, systematic biases shared across annotators do not average out; instead, they are absorbed and amplified by the aligned policy as a preference for the attribute itself.

In practice, preference optimization cannot discern why a label was assigned. For instance, if annotators consistently favor answers signed with names signaling a specific demographic group, the objective simply treats the signature as a sign of response quality. Under standard DPO the effect is large: a policy that initially signs across demographics with equal probability shifts to placing probability 0.96 on the favored marker. The same happens with real annotators who favor longer answers or Markdown styling (Section 5).

In this paper, we address the following alignment problem: *given preference judgments from annotators and a declared attribute of the responses, how can we train a policy that does not inherit the annotators' shared bias toward that attribute?*

We introduce **Bias-Adjusted DPO (BA-DPO)**, which generalizes DPO to possibly biased annotators, where each annotator carries a parameter for how much they favour the declared attribute. The

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.

parameter is added to the reward and does not enter the partition function, so the reparameterization that lets DPO train without a reward model goes through unchanged (Proposition 2).
We then characterise what the labels determine. The votes fix every annotator's bias relative to the others. What they leave open is one direction: raising all the biases together while lowering how much the policy itself favours the attribute changes no judgment (Theorem 3, Corollary 4). That direction is what sets the attribute rate of the trained policy. Training from the reference leaves the rate the reference had, and stating a target rate moves it instead.

BA-DPO fundamentally differs from existing alignment corrections in its underlying formulation. Single-attribute heuristics like R-DPO (Park et al., 2024) and SamPO (Lu et al., 2024) modify or regularize length distributions; the attribute is fixed in the objective, and as our experiments show, correcting it leaves another one in place or substitutes one spurious shortcut for another (Lamparth et al., 2026). Heterogeneity techniques like Group-DRO (Sagawa et al., 2020) and EM-DPO (Chidambaram et al., 2026) represent or reweight evaluator disagreement, which addresses how annotators differ rather than what they hold in common. Methods like ADPO (Wang, 2025) remove unindexed response-level offsets, while BiasDPO (Allam, 2024) addresses the converse problem of cleansing biased model outputs using clean preference data. By contrast, BA-DPO models label bias directly, as additive per-annotator scalars on declared attributes, which separates the bias the annotators share from the quality signal and removes it rather than accommodating it. A more detailed discussion of prior related literature is provided in Appendix A.

The primary contributions of this paper are summarized as follows:

1. **Problem formulation and method:** We formalize the challenge of removing shared, attribute-directed annotator bias from preference optimization and propose BA-DPO, an efficient objective requiring one parameter per attribute when annotators are pooled and one per annotator and attribute otherwise, with no reward model and no dedicated network (Section 3).
2. **Theoretical characterization:** We prove that the objective is convex in the bias parameters and that the votes fix every annotator's bias relative to the others, leaving one direction that sets the trained policy's attribute rate (Section 4).
3. **Empirical validation:** On a corpus with planted biases and on MultiPref (Miranda et al., 2025) with real annotators, at 0.5B with full fine-tuning and at 8B with LoRA, we show that BA-DPO removes 81 to 95% of DPO's attribute shift on the planted corpus and about half of its lengthening on MultiPref, at no higher KL than DPO and no loss in judged quality (Section 5).

2 Background

Given a prompt x and two responses y_1, y_2, the Bradley-Terry (BT) model expresses the probability that y_1 is preferred over y_2 (denoted $y_1 \succ y_2$) in terms of the difference in intrinsic quality between y_1 and y_2:

$$p(y_1 \succ y_2 \mid x) = \sigma\left(r^\star(x, y_1) - r^\star(x, y_2)\right) \tag{1}$$

where $r^\star$ is the latent reward function and σ is the logistic sigmoid.

Reinforcement Learning with Human Feedback (RLHF) fits a reward model to pairwise comparisons and subsequently maximizes expected reward under a KL-divergence penalty of strength $\beta > 0$ relative to a reference policy π_{ref}:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(\cdot|x)}\left[r(x, y)\right] - \beta\mathbb{D}_{\text{KL}}\left[\pi_\theta(y \mid x) \,\|\, \pi_{\text{ref}}(y \mid x)\right]. \tag{2}$$

Rafailov et al. (2023) observe that this constrained optimization problem has an exact closed-form solution

$$\pi_\theta(y \mid x) = \frac{1}{Z(x)}\pi_{\text{ref}}(y \mid x)\exp\left(\frac{1}{\beta}r(x, y)\right) \tag{3}$$

where $Z(x) = \sum_y \pi_{\text{ref}}(y \mid x)\exp\left(\frac{1}{\beta}r(x, y)\right)$ is the partition function. Rearranging terms yields the implicit reward:

$$r(x, y) = \beta\log\frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta\log Z(x). \tag{4}$$

When substituted back into the BT preference model, the partition function $Z(x)$ cancels out in the reward difference. Maximum likelihood estimation over preference pairs thus simplifies to a direct

loss over the policy parameters, bypassing explicit reward modeling and reinforcement learning:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta\log\frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right]. \tag{5}$$

The BARP model (Ferrara et al., 2024) extends Bradley-Terry to account for evaluator bias in pairwise ranking. Item i has a true quality score s_i and a group attribute indicator $\delta_{g_i} \in \{0, 1\}$. Evaluator k carries a scalar bias θ_k and perceives the score as $s_i + \theta_k\delta_{g_i}$, so that:

$$p_k(i \succ j) = \sigma\left(s_i + \theta_k\delta_{g_i} - s_j - \theta_k\delta_{g_j}\right). \tag{6}$$

Scores and bias parameters are estimated jointly via maximum likelihood. The bias terms vanish for same-group comparisons ($\delta_{g_i} = \delta_{g_j}$), which isolate true item quality, whereas cross-group comparisons identify evaluator bias. We adopt this formulation as our foundation for developing BA-DPO.

3 BIAS-ADJUSTED DPO

Given a prompt x, an item (a response to the prompt by the language model) y has a latent reward $r^\star(x, y)$ and a group indicator $\delta_{g(x,y)} \in \{0, 1\}$ recording whether it exhibits the attribute whose influence we wish to remove. Annotator $k \in \{1, \ldots, m\}$ carries a bias parameter $\theta_k \in \mathbb{R}$. Under the BARP model (Ferrara et al., 2024), annotator k prefers y_1 to y_2 with probability

$$p_k(y_1 \succ y_2 \mid x) = \sigma\big(r^\star(x, y_1) + \theta_k\,\delta_{g(x,y_1)} - r^\star(x, y_2) - \theta_k\,\delta_{g(x,y_2)}\big), \tag{7}$$

so that the annotator acts on a perceived reward $\tilde{r}_k(x, y) = r^\star(x, y) + \theta_k\,\delta_{g(x,y)}$ rather than on $r^\star$ itself. Substituting the DPO reparameterization (4) into (7), the two copies of $\beta\log Z(x)$ cancel exactly as they do in standard DPO, because the bias terms are additive to the reward and independent of the partition function. Given a dataset $\mathcal{D} = \{(k^{(i)}, x^{(i)}, y_w^{(i)}, y_l^{(i)})\}_{i=1}^N$ in which each comparison is attributed to its annotator; by applying maximum likelihood over $\mathcal{D}$ we obtain the BA-DPO loss:

$$\mathcal{L}_{\text{BA-DPO}}(\pi_\theta, \{\theta_k\}; \pi_{\text{ref}}) = -\mathbb{E}_{(k,x,y_w,y_l)\sim\mathcal{D}}\Big[\log\sigma\big(u + b_k\big)\Big], \tag{8}$$

where we write

$$u \coloneqq \beta\log\frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta\log\frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}, \qquad b_k \coloneqq \theta_k\big(\delta_{g(x,y_w)} - \delta_{g(x,y_l)}\big). \tag{9}$$

We write $\Delta\delta \coloneqq \delta_{g(x,y_w)} - \delta_{g(x,y_l)} \in \{-1, 0, 1\}$ for the group difference of a comparison, so that $b_k = \theta_k\,\Delta\delta$. The loss is optimised jointly over the policy parameters and the m scalar bias parameters.

Remark 1 (BA-DPO reduces to DPO for unbiased annotators)**.** *Setting $\theta_k = 0$ for every annotator in* (8) *recovers the DPO loss* (5) *exactly. DPO is therefore the special case of BA-DPO in which the annotators are assumed unbiased, which is the assumption of the Bradley-Terry model.*

Gradient analysis. The gradient of (8) with respect to the policy parameters is the DPO gradient with the importance weight $\sigma(-u)$ replaced by $\sigma(-u - b_k)$,

$$\nabla_\phi\mathcal{L}_{\text{BA-DPO}} = -\beta\,\mathbb{E}_{(k,x,y_w,y_l)\sim\mathcal{D}}\Big[\sigma(-u - b_k)\cdot\big(\nabla_\phi\log\pi(y_w \mid x) - \nabla_\phi\log\pi(y_l \mid x)\big)\Big]. \tag{10}$$

When the annotator's bias favours the winner ($b_k > 0$) the weight shrinks and the comparison teaches less, since part of its outcome is explained by bias; when the winner was preferred against the bias ($b_k < 0$) the weight grows; when the two responses share a group, $b_k = 0$ and the update is DPO's. The gradient with respect to a bias parameter is the scalar

$$\frac{\partial\mathcal{L}_{\text{BA-DPO}}}{\partial\theta_k} = -\frac{n_k}{N}\,\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}_k}\Big[\sigma(-u - b_k)\cdot\big(\delta_{g(x,y_w)} - \delta_{g(x,y_l)}\big)\Big], \tag{11}$$

where $\mathcal{D}_k$ are the comparisons provided by annotator k and n_k/N their share of the data. The bias channel adds m scalar parameters, a negligible cost next to the policy; the factor n_k/N is decisive for the optimisation dynamics (see Proposition 6 in Appendix B.5).

Per-annotator, pooled, and shared-mean. The model above requires annotator identities. When they are unavailable the bias channel collapses to a single population-level scalar $\theta \in \mathbb{R}$ shared by every comparison (the *pooled* variant). When they are available there are two ways to use them: the *naive per-annotator* variant with free parameters $\{\theta_k\}$, and the *shared-mean* variant $\theta_k = \bar{\theta} + \varepsilon_k$ in which the mean is a single parameter updated on every cross-group comparison and the deviations ε_k carry the individual structure. When the annotators belong to known classes, a class offset can be inserted between the two, $\theta_k = \bar{\theta} + \delta_c + \varepsilon_k$. All variants span the same model class where they overlap; they differ in optimisation dynamics, and Section 4 and Appendix F show that this difference, not the identity information itself, decides how much bias is removed.

Multiple attributes. For q binary or continuous attributes with group vectors $\vec{g}_y \in \mathbb{R}^q$, the bias term generalises to $b_k = \vec{\theta}_k^\top(\vec{g}_{y_w} - \vec{g}_{y_l})$ with $\vec{\theta}_k \in \mathbb{R}^q$ per annotator. The Signed-UltraFeedback corpus of Section 5 uses $q = 2$ binary attributes, and Appendix F.1 shows that declaring only one of the two leaves the other in the policy and removes only part of the declared one.

4 Theoretical Characterization

This section presents the theoretical characterization of BA-DPO (proofs in Appendix B).

The first result states precisely the substitution already presented in Section 3.

Proposition 2 (Compatibility with the DPO reparameterization)**.** *Let $r(x, y)$ be any reward function for which $Z(x) = \sum_y \pi_{\text{ref}}(y \mid x) \exp(r(x, y)/\beta)$ is finite for every prompt, and let $\{\theta_k\}_{k=1}^m$ be arbitrary evaluator bias parameters. Then the bias-aware preference model* (7) *can be written equivalently as*

$$p_k(y_1 \succ y_2 \mid x) = \sigma\left(\beta \log \frac{\pi(y_1 \mid x)}{\pi_{\text{ref}}(y_1 \mid x)} - \beta \log \frac{\pi(y_2 \mid x)}{\pi_{\text{ref}}(y_2 \mid x)} + \theta_k\big(\delta_{g(x,y_1)} - \delta_{g(x,y_2)}\big)\right)$$

with $\pi(y \mid x) = \pi_{\text{ref}}(y \mid x) \exp(r(x, y)/\beta)/Z(x)$.

Proposition 2 is what keeps the construction at one policy: the bias term is added to the reward and does not involve the partition function $Z(x)$, so it passes through the reparameterization unchanged and stays outside the policy. Methods for annotator disagreement differ in how much freedom they give each annotator's reward. DPO gives none: one reward is fit for everyone, so a bias the annotators share is represented nowhere except in the policy. Mixture methods give each of K annotator types its own reward, and since each reward has its own optimal policy, they train K policies (Chidambaram et al., 2026). BA-DPO lets annotators differ only in the coefficient θ_k on a declared direction δ_g, a fixed function of (x, y) with no parameters of its own. A single policy then carries the reward the annotators share, the annotator-specific part enters the logit as the scalar θ_k, and modelling annotator bias costs m scalars, not K policies.

We now characterise what the observed votes determine about the bias parameters. Part of what they leave undetermined is inherited from DPO, which cannot see a reward shift that is constant within a compared pair, because rewards enter the likelihood only through the difference between the two responses of a pair. The rest is what the bias channel adds, and the theorem holds for an otherwise arbitrary reward function.

Consider the graph whose vertices are the annotators, with an edge between two annotators whenever they judged at least one cross-group comparison in common. We say that the annotator pool is *connected* if this graph is connected.

Theorem 3 (Identifiability of the bias parameters)**.** *The loss $\mathcal{L}$ of* (8)*, viewed as a function of the reward values $r(x, y)$ and of θ, is jointly convex in (r, θ), and all its minimisers assign the same probability to every judgment. If the annotator pool is connected and $(\hat{r}, \hat{\theta})$ is a minimiser, the minimisers are exactly the pairs*

$$\big(\hat{r} + c\,\delta_g + h,\ \hat{\theta} - c\mathbf{1}\big), \qquad c \in \mathbb{R},$$

with h any function constant on the two responses of every compared pair. Every difference $\theta_k - \theta_{k'}$ therefore takes the same value at every minimiser, while their mean $\bar{\theta} := \frac{1}{m}\sum_k \theta_k$ is not determined by the judgments.

Connectedness requires that every annotator judged at least one cross-group comparison, since the votes of an annotator who never saw the attribute on both sides say nothing about their θ_k. It holds whenever annotators are assigned to comparisons at random, as in every corpus here; if it fails, the theorem applies within each connected group of annotators, each with its own offset. For a fixed policy, a minimiser in θ exists unless some annotator chose the attribute side in every one of their cross-group comparisons, in which case their θ_k diverges.

Theorem 3 is stated in terms of the individual θ_k; it can be restated by decomposing the annotators' biases into a shared mean and deviations.

Corollary 4 (Reward-bias degeneracy)**.** *Write $\theta_k = \bar{\theta} + \varepsilon_k$ with $\sum_k \varepsilon_k = 0$. Shifting the mean by $-c$ and adding $c\,\delta_g$ to the reward leaves every judgment probability unchanged, so $\bar{\theta}$ is not determined by the votes. If the annotator pool is connected, the deviations ε_k take the same value at every minimiser.*

The degeneracy is intrinsic rather than an artefact of the parameterisation: the likelihood cannot distinguish between the attribute genuinely carrying reward and every annotator being uniformly biased toward it. The free constant c is therefore a choice, and what it decides is how much the trained policy favours the attribute: by Corollary 4, moving c from $\bar{\theta}$ into the reward multiplies the policy's odds for a response over its counterfactual without the attribute by $e^{c/\beta}$ and changes no judgment probability. Two anchors are natural. Anchoring to the reference takes c such that the policy gives a response and its counterfactual the same odds as the reference does, so the shared level of the labels sits in $\bar{\theta}$ and the policy keeps the reference's attribute rate; every experiment in the paper starts from π_{ref} with $\theta = 0$ and ends there (Appendix C.3). Anchoring to a target takes c such that the outputs meet a stated criterion, for instance statistical parity, an attribute rate of one half: the required c is found by a one-dimensional search on the rate and applied as a fixed offset (Appendix C.4; Appendix C.5 uses it to bring a biased DPO policy to parity). Which anchor is right is a normative decision, like the attribute partition itself; the labels do not make it.

Proposition 5 (Representability of the bias components)**.** *Decompose $\theta_k = \bar{\theta} + \varepsilon_k$ with $\sum_k \varepsilon_k = 0$. The shared component $\bar{\theta}\,\delta_{g(x,y)}$ is expressible as a function of (x, y) and can therefore be represented inside the policy's implicit reward, whereas the deviation term $\varepsilon_k\,\delta_{g(x,y)}$ depends on k, which the policy never observes, and is representable by no policy $\pi(y \mid x)$ unless every $\varepsilon_k = 0$, that is, unless all annotators have the same bias.*

Proposition 5 says what the policy could express, not what training makes it express. Had the policy taken up the shared component, the bias parameters would be left with the deviations alone, and the naive per-annotator and shared-mean variants would then behave alike, since they write the deviations the same way. In our experiments the shared component goes to the bias parameter instead: a mean started at zero and a mean started at the offline estimate converge to the same interior value (Appendix F.2). The two variants therefore differ only in how they write the same m parameters, as m free θ_k or as a mean plus deviations, and that decides how fast the mean moves. The shared parameter is updated by every cross-group judgment, while each free θ_k is updated only by the judgments of its own annotator. Under gradient descent the naive mean is therefore exactly m times slower (Proposition 6).

Together, these results explain the pattern of the experiments. The deviations are the only component the votes determine on their own (Theorem 3) and the only one the policy cannot express (Proposition 5). Debiasing the policy, however, requires the mean, which is the component the naive per-annotator variant learns most slowly and the shared-mean variant learns first.

5 Experiments

In this section, we test whether DPO absorbs a bias shared by the annotators and whether BA-DPO removes it, on two corpora: Signed-UltraFeedback (built from UltraFeedback (Cui et al., 2024)), where the annotators' biases are planted and the amount to remove is therefore known, and MultiPref (Miranda et al., 2025), with real annotators, under length and under formatting. DPO takes up the bias in every setting and model tested, and BA-DPO removes most of it without moving further from the reference than DPO does.

5.1 EXPERIMENTAL SETTINGS

Datasets. The method needs an attribute that can be read from each response and, for the per-annotator variants, the identity of the annotator behind each judgment. We evaluate four attributes on two corpora (Appendix D has the full details).

Length and formatting can be read from any preference corpus. We use MultiPref (Miranda et al., 2025), one of the few public corpora with annotator identities: after removing ties, $30{,}847$ disaggregated judgments on $10{,}461$ comparisons, each judged by four of 227 annotators. A response is *long* when it has at least 1.5 times the words of its partner and *formatted* when it contains Markdown.

We know of no public preference corpus with identified annotators in which the responses carry a gender- or race-coded marker. We therefore introduce *Signed-UltraFeedback*, built from the prompts and response pairs of UltraFeedback (Cui et al., 2024) by ending each response with a signature whose first name comes from published audit lists (Bertrand & Mullainathan, 2004; Caliskan et al., 2017) and is both gender-coded (woman or man) and race-coded (white or black), so that every response carries two binary attributes at once. Sixty annotators in three classes of 20 vote on each pair, four per pair, through the biased Bradley-Terry likelihood (7), each with a bias vector $\theta_k \sim \mathcal{N}(\mu_c, 0.8^2 I)$. The class means are planted so that both attributes have a population mean near 1.0 in log-odds, the classes agree on the gender-coded attribute, and on the race-coded one a class of 20 leans the other way. The θ_k are used only to generate the labels and to score the outcome, never by a training method. The reference policy signs with each name group equally often, so any bias in a trained policy comes from the preference stage. We say "gender-coded" and "race-coded" because the corpus tracks how a text marker propagates through alignment, not human prejudice.

Models and methods. We run every experiment on two models, `Qwen2.5-0.5B-Instruct` (Yang et al., 2024) and `Llama-3.1-8B-Instruct` (Grattafiori et al., 2024), which differ in scale and in family. The 0.5B model is fully fine-tuned; the 8B model is trained with LoRA (Hu et al., 2022) on all linear layers. Each model and corpus has its own SFT reference, fine-tuned on the majority-chosen response of each comparison and shared by every method and seed, so that implicit rewards are comparable (Appendix E). A third model, `Mistral-7B-Instruct-v0.3` (Jiang et al., 2023), of a different family and with a different tokenizer, repeats the comparison between DPO and BA-DPO as a robustness check (Appendix F.5).

We compare the reference, DPO, and the two BA-DPO variants of Section 3: *pooled*, one scalar per attribute shared by every annotator, and $\bar{\theta} + \varepsilon_k$, a shared mean per attribute plus one deviation per annotator. The remaining variants of the bias model are controls for the mechanism, reported at 0.5B in Appendix F. r3

Baselines. To the best of our knowledge, no method for preference optimisation takes the attribute as an input that the user supplies and can change between runs. We therefore compare with the two families that come closest: the corrections built for length bias, which fix that attribute in the objective, and the methods built for annotator disagreement. Each runs on the corpus of the bias it was built for and, for the length methods, also under formatting, by a rule fixed before any result was read (Appendix E.3). R-DPO (Park et al., 2024) and SamPO (Lu et al., 2024) target length bias: they are trained on MultiPref and re-scored under formatting without retraining. R-DPO uses $\alpha = 0.005$, because the published 0.02 collapses the policy on this corpus (Appendix F.2). Group-DRO (Sagawa et al., 2020) and EM-DPO with MinMax-DPO (Chidambaram et al., 2026) target annotator disagreement. Both are our own implementations (Appendix E.3). Group-DRO runs on both corpora, with the annotator classes as groups on Signed-UltraFeedback and the annotators on MultiPref; EM-DPO runs on Signed-UltraFeedback and at 0.5B only, since its four rounds over three type policies cost five times the steps of any other method.

Metrics. We evaluate each policy on 300 held-out prompts, computing five metrics from the generations and held-out judgments (formal definitions in Appendix E.4). Each configuration is trained at three seeds, and we report the mean alongside a 95% confidence interval; the reference policy is trained once and therefore has no interval.

- *Attribute rate:* How often the generations carry the declared attribute. On Signed-UltraFeedback, we compute this exactly from the policy's output probabilities rather than from sampling: the

probability mass placed on woman-coded ($p(\text{woman})$) or black-coded ($p(\text{black})$) names at the signature position. On MultiPref, this is the mean token count for length bias, and the markdown rate reweighted onto the reference's length distribution for formatting bias, so that a policy cannot lower the measured rate by producing shorter answers alone.

- *Bias removed:* The proportion of the gap in attribute rate between standard DPO and the reference policy closed by the method. This is computed per seed against the corresponding DPO seed, scaled such that 0% is DPO and 100% is the reference.
- *Held-out gap:* The difference in the policy's implicit reward accuracy between held-out judgment pairs that differ on the attribute, and pairs that do not. A biased policy predicts annotator labels better exactly when the attribute differs; a gap approaching zero, while same-group accuracy holds, shows that the bias has left the implicit reward.
- *RM and RM$^-$:* The judge's score, from `Skywork-Reward-V2-Qwen3-1.7B` (Liu et al., 2026). *RM*$^-$ is the score after uniformly stripping the attribute from all generations (that is, removing signatures in Signed-UltraFeedback or markup in formatting). No length-invariant transform exists for length bias, so there we report the raw RM score and read it with caution.
- *KL:* The per-token Kullback-Leibler divergence from the reference, estimated on the policy's own generations. This distinguishes methods that remove the bias by changing the policy from those that stay close to the reference distribution.

5.2 SIGNED-ULTRAFEEDBACK

DPO absorbs the bias on both attributes and on both models (Table 1), moving from the near-balanced reference to 0.96 to 0.99. BA-DPO removes 81 to 95% of that shift, with both variants and on both models, at no cost in judge score and at about two thirds of DPO's KL: the policy still differs from the reference, but not along the attribute.

The two BA-DPO variants give the same policy to within about 0.01 everywhere, although *pooled* has one parameter per attribute and $\bar{\theta} + \varepsilon_k$ has a mean and sixty deviations. Proposition 5 predicts this: a deviation depends on who voted, which the policy never sees, so only the shared mean can reach it. Appendix F.1 tests this prediction with the annotator identities shuffled and with the annotator classes given. The race-coded attribute is removed less than the gender-coded one at 0.5B, 81 against 89%, although both were planted at the same mean. A shared scalar is fitted to how the population votes, not to how it is biased, and here the classes disagree, so the votes partly cancel: on the pairs that differ only in the name the attribute-carrying side wins at log-odds 0.70 against a realised mean bias of 1.13, and the learned scalar settles at 0.62 to 0.68. The part of the bias the scalar does not absorb stays in the policy.

The deviations do not change the policy, but they do estimate each annotator's own bias. The learned $\theta_k = \bar{\theta} + \varepsilon_k$ correlate with the planted θ_k at $r = 0.95$ on the gender-coded attribute and 0.98 on the race-coded one, on both models. When we use them to predict each annotator's held-out votes, they are as accurate as the planted θ_k (both as 0.752 at 0.5B). The pooled scalar gives every annotator the same bias, so for the class that leans against the race-coded attribute its predictions are no better than chance (Appendix G).

Neither baseline removes the bias. Group-DRO removes 4 to 18%: it reweights the groups it is given, and a bias that every group shares is not disagreement between groups. EM-DPO does not recover the classes, one type ending up with 54 to 58 of the 60 annotators, and leaves the gender-coded attribute at DPO's level; its 61% on the race-coded one is averaging rather than removal, since part of the mixture sits on a type policy carrying the opposite bias.

5.3 MULTIPREF: LENGTH BIAS

On MultiPref the annotators are real and length has no natural target, since a longer answer is sometimes the better one. The token count therefore says how far the policy moved, not whether it stopped in the right place. We read the held-out gap for that: it compares how well the policy's implicit reward predicts these annotators' labels on held-out pairs that differ on the declared attribute with how well it predicts them on pairs that do not.

Table 2 reports the results. Under DPO the policy produces answers 96 tokens longer than the reference at 0.5B and 68 tokens longer at 8B, with held-out gaps of 0.13 and 0.09. BA-DPO removes

Table 1: Signed-UltraFeedback, two attributes and three annotator classes. Columns as defined in Section 5.1; mean and 95% interval over three seeds. RM^- is comparable within a model only.

Method	p(woman)	**Removed**	p(black)	**Removed**	**RM^-**	**KL**
Qwen2.5-0.5B-Instruct, full fine-tuning						
Reference (SFT)	0.498	—	0.517	—	-3.30	—
DPO	0.964 ± 0.005	—	0.991 ± 0.001	—	-2.63 ± 0.28	0.038 ± 0.001
BA-DPO (pooled)	0.549 ± 0.008	$89 \pm 2\%$	0.608 ± 0.017	$81 \pm 4\%$	-2.66 ± 0.29	0.026 ± 0.001
BA-DPO ($\bar{\theta} + \varepsilon_k$)	0.542 ± 0.011	$91 \pm 2\%$	0.595 ± 0.018	$83 \pm 4\%$	-2.76 ± 0.35	0.026 ± 0.001
Group-DRO DPO	0.927 ± 0.016	$8 \pm 3\%$	0.941 ± 0.018	$11 \pm 4\%$	-2.62 ± 0.08	0.035 ± 0.001
EM-DPO + MinMax-DPO	0.923 ± 0.109	$9 \pm 24\%$	0.702 ± 0.157	$61 \pm 33\%$	-2.63 ± 0.10	0.041 ± 0.007
Llama-3.1-8B-Instruct, LoRA						
Reference (SFT)	0.465	—	0.521	—	0.41	—
DPO	0.967 ± 0.012	—	0.986 ± 0.005	—	1.43 ± 0.17	0.021 ± 0.002
BA-DPO (pooled)	0.492 ± 0.002	$95 \pm <1\%$	0.555 ± 0.002	$93 \pm <1\%$	1.41 ± 0.26	0.013 ± 0.003
BA-DPO ($\bar{\theta} + \varepsilon_k$)	0.491 ± 0.002	$95 \pm <1\%$	0.552 ± 0.002	$93 \pm <1\%$	1.49 ± 0.12	0.014 ± 0.002
Group-DRO DPO	0.945 ± 0.018	$4 \pm 2\%$	0.904 ± 0.020	$18 \pm 3\%$	1.44 ± 0.25	0.020 ± 0.003

Table 2: MultiPref with length declared. Columns as in Section 5.1; mean and 95% interval over three seeds; RM is the raw judge score, which rises with answer length (Section 5.3).

Method	**Tokens**	**Removed**	**Gap**	**RM**	**KL**
Qwen2.5-0.5B-Instruct, full fine-tuning					
Reference (SFT)	288.6	—	—	-1.34	—
DPO	384.5 ± 3.7	—	0.126 ± 0.022	-0.66 ± 0.16	$0.033 \pm <0.001$
BA-DPO (pooled)	341.3 ± 8.7	$45 \pm 10\%$	0.019 ± 0.045	-0.66 ± 0.20	0.033 ± 0.001
BA-DPO ($\bar{\theta} + \varepsilon_k$)	335.3 ± 7.4	$51 \pm 8\%$	0.001 ± 0.032	-0.78 ± 0.30	0.032 ± 0.001
R-DPO ($\alpha = 0.005$)	310.8 ± 10.3	$77 \pm 10\%$	-0.040 ± 0.007	-0.74 ± 0.38	0.035 ± 0.002
SamPO	420.9 ± 4.4	$-38 \pm 8\%$	0.088 ± 0.023	-0.76 ± 0.07	0.044 ± 0.002
Group-DRO DPO	389.9 ± 4.3	$-6 \pm 3\%$	0.122 ± 0.032	-0.53 ± 0.24	0.032 ± 0.002
Llama-3.1-8B-Instruct, LoRA					
Reference (SFT)	309.7	—	—	4.85	—
DPO	378.2 ± 4.7	—	0.095 ± 0.008	6.52 ± 0.11	0.018 ± 0.001
BA-DPO (pooled)	343.9 ± 2.6	$50 \pm 4\%$	0.034 ± 0.032	6.10 ± 0.02	$0.018 \pm <0.001$
BA-DPO ($\bar{\theta} + \varepsilon_k$)	346.9 ± 6.5	$46 \pm 6\%$	0.046 ± 0.013	6.10 ± 0.09	$0.018 \pm <0.001$
R-DPO ($\alpha = 0.005$)	317.4 ± 2.9	$89 \pm 4\%$	-0.038 ± 0.004	5.73 ± 0.17	0.020 ± 0.001
SamPO	382.8 ± 3.1	$-7 \pm 9\%$	0.061 ± 0.024	6.70 ± 0.54	$0.022 \pm <0.001$
Group-DRO DPO	381.4 ± 12.1	$-5 \pm 11\%$	0.097 ± 0.016	6.49 ± 0.34	0.018 ± 0.001

45 to 51% of that increase and reduces the gap to 0.02 and 0.00 at 0.5B and to 0.03 and 0.05 at 8B. Its learned scalar converges to 0.77 to 0.81 at 0.5B and to 0.69 to 0.71 at 8B, below the offline estimate of 0.99, which counts every win of the longer response as bias.

Each baseline misses the bias in a different way. R-DPO removes more tokens than BA-DPO does, 77 to 89% of DPO's increase, but its gap becomes negative, -0.04 at both scales: the policy now prefers the shorter response on the pairs that carry the attribute, which inverts the bias rather than removing it. Under SamPO the policy is longer than under DPO (-38% and -7%), and Group-DRO changes neither the length nor the gap, since a bias that every annotator shares is not disagreement between annotators. At 8B the judge ranks the methods in order of their length, from 4.9 for the reference to 6.7 for SamPO, so BA-DPO's 6.1 against DPO's 6.5 cannot be separated from the tokens removed. The held-out gap does not use the judge, which is why we report it here.

5.4 MULTIPREF: FORMATTING BIAS

The data is the same as in Section 5.3, with formatting declared in place of length: the same annotators and the same judgments. The two attributes are favoured about equally in the labels, markdown winning at log-odds 0.98 and length at 0.99, but DPO amplifies only one of them. Under length it adds 96 tokens, many times the spread across seeds, while under formatting it raises the markdown

Table 3: MultiPref with formatting declared, on the judgments and annotators of Table 2. Columns as in Table 2. No removal share is given because DPO moves the markdown rate by less than its seed interval (Section 5.4). The length baselines are their Table 2 checkpoints scored under formatting without retraining.

Method	Rate	Gap	RM$^-$	KL
Qwen2.5-0.5B-Instruct, full fine-tuning				
Reference (SFT)	0.583	—	-1.51	—
DPO	0.618 ± 0.053	0.066 ± 0.025	-0.98 ± 0.14	$0.033 \pm <0.001$
BA-DPO (pooled)	0.592 ± 0.027	0.004 ± 0.013	-0.89 ± 0.12	0.032 ± 0.002
BA-DPO ($\bar{\theta} + \varepsilon_k$)	0.581 ± 0.034	0.000 ± 0.051	-0.86 ± 0.19	0.032 ± 0.002
R-DPO ($\alpha = 0.005$, re-scored)	0.634 ± 0.003	-0.039 ± 0.039	-0.98 ± 0.34	0.035 ± 0.002
SamPO (re-scored)	0.636 ± 0.049	0.053 ± 0.024	-1.16 ± 0.15	0.044 ± 0.002
Llama-3.1-8B-Instruct, LoRA				
Reference (SFT)	0.560	—	4.67	—
DPO	0.567 ± 0.034	0.067 ± 0.016	6.18 ± 0.12	0.018 ± 0.001
BA-DPO (pooled)	0.536 ± 0.018	0.033 ± 0.033	6.05 ± 0.03	0.018 ± 0.001
BA-DPO ($\bar{\theta} + \varepsilon_k$)	0.539 ± 0.022	0.033 ± 0.046	6.08 ± 0.09	$0.018 \pm <0.001$
R-DPO ($\alpha = 0.005$, re-scored)	0.568 ± 0.041	-0.031 ± 0.029	5.50 ± 0.15	0.020 ± 0.001
SamPO (re-scored)	0.569 ± 0.013	0.040 ± 0.007	6.34 ± 0.54	$0.022 \pm <0.001$

rate by 0.035, less than the ± 0.053 interval on that rate. The rate is therefore a weak readout here, since DPO barely moves it, and we read the held-out gap instead.

DPO opens a gap of 0.066 (Table 3). BA-DPO closes it to 0.004 and 0.000 while leaving the markdown rate where the reference had it. At 8B DPO does not raise the rate beyond seed noise (0.567 ± 0.034 against 0.560), and still opens a gap of 0.067, which both BA-DPO variants halve.

Neither length method can be given another attribute, and correcting length does not correct formatting: R-DPO turns the gap negative again (-0.039) and SamPO leaves it near DPO's level (0.053). At its published strength, $\alpha = 0.02$, R-DPO raises the markdown rate to 0.70, above DPO, in answers collapsed to 138 tokens (Appendix F.2): the pressure DPO had put on length has moved to markdown. On two attributes of one corpus this is the substitution pattern of Lamparth et al. (2026), and it is what declaring the attribute avoids. Appendix F.1 shows the same pattern inside Signed-UltraFeedback when only one of its two biased attributes is declared.

6 Conclusions

We addressed the vulnerability of preference optimization to a bias that annotators share by introducing Bias-Adjusted DPO (BA-DPO), a generalization of DPO that adds one bias parameter per annotator for each declared attribute. We proved that the objective is convex in these parameters and that the votes fix every annotator's bias relative to the others. How much of the shared bias counts as quality is left to one free choice, which sets the aligned policy's attribute rate. On three models, BA-DPO removed 81 to 95% of the shift DPO absorbs on planted biases and about half of the length DPO adds on MultiPref, with no higher KL to the reference than DPO and no loss in judged quality wherever the judge can be read independently of the attribute.

Limitations and future work. BA-DPO removes only the bias it is told about: the attribute must be declared and computable from the response, and an attribute that is not declared stays in the policy (Appendix F.1). Because the labels do not fix the shared level of the bias, the default anchor keeps the reference's attribute rate, including any bias the reference already has; a target rate removes it, but the target must be chosen. That level could instead be estimated from a small set of judgments by annotators known to be unbiased. Our evidence comes from one reward-model judge, one corpus with real annotators whose two attributes are both surface features, and LoRA at 7B and 8B; a corpus with annotator identities and a social attribute, such as a persona or a language variety, is the natural next test. Since the bias term is additive to the reward, it also applies to reward-model training and to other pairwise objectives such as IPO (Azar et al., 2024), and the per-annotator estimates could flag biased annotators while a pool is still being labelled.

AI USE STATEMENT

In this work, we used generative AI tools (Claude Code with Fable 5.1 and Opus 5) as a coding assistant and to polish the writing of the manuscript. We have reviewed all AI-assisted code and text, and we take full responsibility for the final content of this work.

ETHICS STATEMENT

This work aims to reduce the influence of systematic annotator bias on aligned language models. Two cautions apply. First, the partition of response attributes into bias to be removed and quality to be preserved is a declared input, since preference data determine the biases only up to a common offset; the choice is normative and should involve the stakeholders affected. Second, per-annotator modelling requires annotator identifiers and produces per-person bias estimates; we recommend treating both as personal data, and we use only datasets whose identifiers were pseudonymised by their publishers.

REPRODUCIBILITY STATEMENT

All models (`Qwen2.5-0.5B-Instruct`, `Llama-3.1-8B-Instruct` and `Mistral-7B-Instruct-v0.3`, with `Skywork-Reward-V2-Qwen3-1.7B` as the judge) and datasets (`ultrafeedback_binarized` and `allenai/multipref`) are public. Every reported number derives from a logged run keyed by (method, parameters, seed); every cell of the main tables is a mean over seeds 42, 123 and 456 with a 95% interval, and single-seed appendix rows are labelled as controls. Complete data-generation procedures and hyperparameters are in Appendices D.1 to E. All runs used a single NVIDIA A100 80GB GPU; the peak memory and duration of each kind of run are given in Appendix E. The source code is in the supplementary material, with the configuration of every experiment and the commands that reproduce each table; it will be made public with the final version.

REFERENCES

Ahmed Allam. BiasDPO: Mitigating bias in language models through direct preference optimization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pp. 42–50, 2024.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 238 of *Proceedings of Machine Learning Research*, pp. 4447–4455. PMLR, 2024.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Marianne Bertrand and Sendhil Mullainathan. Are Emily and Greg more employable than Lakisha and Jamal? a field experiment on labor market discrimination. *American Economic Review*, 94 (4):991–1013, 2004.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 2017.

Haoxian Chen, Hanyang Zhao, Henry Lam, David Yao, and Wenpin Tang. MallowsPO: Fine-tune your LLM with preference dispersions. In *International Conference on Learning Representations (ICLR)*, 2025.

Lichang Chen, Chen Zhu, Jiuhai Chen, Davit Soselia, Tianyi Zhou, Tom Goldstein, Heng Huang, Mohammad Shoeybi, and Bryan Catanzaro. ODIN: Disentangled reward mitigates hacking in RLHF. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 7935–7952, 2024.

David Chhan, Ellen Novoseller, and Vernon J. Lawhern. Crowd-PrefRL: Preference-based reward learning from crowds. *arXiv preprint arXiv:2401.10941*, 2024.

Keertana Chidambaram, Karthik Vinay Seetharaman, and Vasilis Syrgkanis. Direct preference optimization with unobserved preference heterogeneity: The necessity of ternary preferences. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 300 of *Proceedings of Machine Learning Research*, pp. 4555–4563. PMLR, 2026.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *NeurIPS*, 2017.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4069–4082. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1418.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. UltraFeedback: Boosting language models with scaled AI feedback. In *International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 9722–9744. PMLR, 2024.

Antonio Ferrara, Francesco Bonchi, Francesco Fabbri, Fariba Karimi, and Claudia Wagner. Bias-aware ranking from pairwise comparisons. *Data Mining and Knowledge Discovery*, 38:2062–2086, 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

He He, Sheng Zha, and Haohan Wang. Unlearn dataset bias in natural language inference by fitting the residual. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo)*, pp. 132–142. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-6115.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

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. *arXiv preprint arXiv:2310.06825*, 2023.

Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. End-to-end bias mitigation by modelling biases in corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8706–8716. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.769.

Max Lamparth, Daniel Fein, Andreas Haupt, Marcel Hussing, and Mykel J. Kochenderfer. Reward bias substitution: Single-axis bias mitigations redirect optimization pressure. *arXiv preprint arXiv:2605.27996*, 2026.

Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, Yang Liu, and Yahui Zhou. Skywork-Reward-V2: Scaling preference data curation via human-AI synergy. In *International Conference on Learning Representations (ICLR)*, 2026.

Shunyu Liu, Wenkai Fang, Zetian Hu, Junjie Zhang, Yang Zhou, Kongcheng Zhang, Rongcheng Tu, Ting-En Lin, Fei Huang, Mingli Song, Yongbin Li, and Dacheng Tao. A survey of direct preference optimization. *arXiv preprint arXiv:2503.11701*, 2025.

Wei Liu, Yang Bai, Chengcheng Han, Rongxiang Weng, Jun Xu, Xuezhi Cao, Jingang Wang, and Xunliang Cai. Length desensitization in direct preference optimization. *arXiv preprint arXiv:2409.06411*, 2024.

Junru Lu, Jiazheng Li, Siyu An, Meng Zhao, Yulan He, Di Yin, and Xing Sun. Eliminating biased length reliance of direct preference optimization via down-sampled KL divergence. In *EMNLP*, 2024.

Lester James V. Miranda, Yizhong Wang, Yanai Elazar, Sachin Kumar, Valentina Pyatkin, Faeze Brahman, Noah A. Smith, Hannaneh Hajishirzi, and Pradeep Dasigi. Hybrid preferences: Learning to route instances for human vs. AI feedback. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 7162–7200, 2025.

Ignavier Ng, Patrick Blöbaum, Siddharth Bhandari, Kun Zhang, and Shiva Kasiviswanathan. Debiasing reward models by representation learning with guarantees. *arXiv preprint arXiv:2510.23751*, 2025.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. Disentangling length from quality in direct preference optimization. In *Findings of ACL*, 2024.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, 2023.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations (ICLR)*, 2020.

Wei Shen, Rui Zheng, Wenyu Zhan, Jun Zhao, Shihan Dou, Tao Gui, Qi Zhang, and Xuanjing Huang. Loose lips sink ships: Mitigating length bias in reinforcement learning from human feedback. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 2859–2873. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.188.

Chaojie Wang, Haonan Shi, Long Tian, Bo An, and Shuicheng Yan. Removing prompt-template bias in reinforcement learning from human feedback. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 24110–24122. Association for Computational Linguistics, 2025a. doi: 10.18653/v1/2025.findings-acl.1237.

Chaoqi Wang, Zhuokai Zhao, Yibo Jiang, Zhaorun Chen, Chen Zhu, Yuxin Chen, Jiayi Liu, Lizhu Zhang, Xiangjun Fan, Hao Ma, and Sinong Wang. Beyond reward hacking: Causal rewards for large language model alignment. *arXiv preprint arXiv:2501.09620*, 2025b.

Zixian Wang. ADPO: Anchored direct preference optimization. *arXiv preprint arXiv:2510.18913*, 2025.

Jiancong Xiao, Ziniu Li, Xingyu Xie, Emily Getzen, Cong Fang, Qi Long, and Weijie J. Su. On the algorithmic bias of aligning large language models with RLHF: Preference collapse and matching regularization. *Journal of the American Statistical Association*, 120(552):2154–2164, 2025. doi: 10.1080/01621459.2025.2555067.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Xuanchang Zhang, Wei Xiong, Lichang Chen, Tianyi Zhou, Heng Huang, and Tong Zhang. From lists to emojis: How format bias affects model alignment. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 26940–26961. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1308.

Kangwen Zhao, Jianfeng Cai, Jinhua Zhu, Ruopei Sun, Dongyun Xue, Wengang Zhou, Li Li, and Houqiang Li. Bias fitting to mitigate length bias of reward model in RLHF. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 2912–2927. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.133.

APPENDIX CONTENTS

A Related Work **14**

B Proofs **14**

B.1 Proof of Proposition 2 15

B.2 Proof of Theorem 3 15

B.3 Proof of Corollary 4 17

B.4 Proof of Proposition 5 17

B.5 The speed of the mean under the two parameterisations 18

C The common offset and the choice of anchor **19**

C.1 The policy's tilt toward the attribute 19

C.2 What the labels determine 19

C.3 Reference anchoring 20

C.4 Target anchoring 20

C.5 Experiment: a fixed offset cures the DPO policy 21

D Datasets **22**

D.1 The MultiPref corpus and the length attribute 23

D.2 The NAMES corpus (Signed-UltraFeedback) 23

E Experimental details **25**

E.1 Models and training 25

E.2 Arms 25

E.3 Baselines 26

E.4 Readouts 26

F Ablations and controls **27**

F.1 Variants of the bias model on Signed-UltraFeedback 27

F.2 MultiPref: variants, warm starts and the remaining baselines 28

F.3 Reading the signature from probabilities, not samples 29

F.4 LoRA calibration at 0.5B 29

F.5 Robustness to the model family: Mistral-7B 30

G Vote prediction with the learned bias parameters **30**

A Related Work

Length bias in RLHF and DPO. A substantial line of work identifies the tendency of preference-trained models to exploit response length as a proxy for quality. Park et al. (2024) study length exploitation in DPO specifically and propose a regularisation strategy, Lu et al. (2024) eliminate length reliance through down-sampled KL divergence, and Liu et al. (2024) desensitise DPO to length by decoupling explicit length preference from implicit quality signals. Wang et al. (2025a) show that length is not the only such bias: reward models also favour the response format that the prompt template invites, and correcting length alone leaves that preference in place. On the reward-model side, Zhao et al. (2026) learn and correct length-bias patterns before preference optimisation, while Shen et al. (2023) and Chen et al. (2024) give the reward model a separate head for length and discard it when the policy is trained. All of them build length into the objective; BA-DPO treats length as one instance of a per-annotator additive bias on an attribute the user declares, and our experiments show that correcting length leaves a second attribute in place.

Format and other spurious features. Beyond length, Zhang et al. (2025) show that format features such as lists, markdown and emojis bias alignment in the same way, and Wang et al. (2025b) propose causal reward models addressing several bias types jointly. Ng et al. (2025) learn a representation that separates the factors the reward should depend on from the spurious ones, with identifiability guarantees. These operate at the reward-model level, whereas BA-DPO corrects at the preference-likelihood level and so remains compatible with the reward-model-free setting DPO defines. Allam (2024) address a different problem under a similar name: they use DPO to make a model produce less biased text, with a curated dataset in which the less biased of two completions is always the preferred one. There the bias is in the model's outputs and the labels are the correction; here the bias is in the annotators' labels and the outputs are what it corrupts. Their preferred completions carry the quality signal to be learned, whereas our declared attribute carries the bias to be removed.

Bias-only models. Training a separate term on a declared biased feature alongside the main model and dropping it at test time is the standard way to keep a classifier from relying on a known dataset bias (Clark et al., 2019; He et al., 2019; Karimi Mahabadi et al., 2020). BA-DPO is that construction inside preference optimisation: the bias term is additive in the reward, so it survives the DPO reparameterisation and needs no second network, and its coefficient is per annotator, which those methods have no analogue for because their corpora carry no annotator identity.

Annotator heterogeneity. Standard RLHF treats all annotators as sampling from one latent reward. Xiao et al. (2025) analyse the algorithmic bias this induces and show that RLHF can exhibit preference collapse under annotator disagreement. In the taxonomy of Liu et al. (2025), BA-DPO belongs to the data-quality heterogeneity branch, alongside methods that model disagreement as a mixture over latent preference types (Chidambaram et al., 2026) or as preference dispersion (Chen et al., 2025); Chhan et al. (2024) use annotator identifiers to infer per-user reliability in preference-based reward learning, though not within DPO. ADPO (Wang, 2025) anchors the policy's logits to the reference so that a response's prior popularity is separated from its quality; the offset it removes is common to every response and carries no annotator identity, so it does not address a preference for an attribute that only some responses carry. Those methods represent or balance disagreement, whereas BA-DPO models it as an additive bias on a declared attribute, which is what makes removal rather than accommodation possible, and separates what its parameters do: the mean removes the bias and the deviations describe the annotators.

B Proofs

Throughout, judgments are indexed by $j = 1, \dots, N$; judgment j was made by annotator $k(j)$ on prompt x_j, with winner $y_{w,j}$ and loser $y_{l,j}$, and $\Delta\delta_j := \delta_{g(x_j, y_{w,j})} - \delta_{g(x_j, y_{l,j})} \in \{-1, 0, 1\}$. We write

$$\phi(t) := -\log \sigma(t) = \log\bigl(1 + e^{-t}\bigr), \qquad \phi'(t) = -\sigma(-t), \qquad \phi''(t) = \sigma(t)\,\sigma(-t) > 0,$$

so that the negative log-likelihood of any model whose logit on judgment j is L_j equals $\sum_j \phi(L_j)$.

B.1 Proof of Proposition 2

Proposition 2 (Compatibility with the DPO reparameterization)**.** *Let $r(x, y)$ be any reward function for which $Z(x) = \sum_y \pi_{\text{ref}}(y \mid x) \exp(r(x, y)/\beta)$ is finite for every prompt, and let $\{\theta_k\}_{k=1}^m$ be arbitrary evaluator bias parameters. Then the bias-aware preference model* (7) *can be written equivalently as*

$$p_k(y_1 \succ y_2 \mid x) = \sigma\Bigg(\beta \log \frac{\pi(y_1 \mid x)}{\pi_{\text{ref}}(y_1 \mid x)} - \beta \log \frac{\pi(y_2 \mid x)}{\pi_{\text{ref}}(y_2 \mid x)} + \theta_k\big(\delta_{g(x,y_1)} - \delta_{g(x,y_2)}\big)\Bigg)$$

with $\pi(y \mid x) = \pi_{\text{ref}}(y \mid x) \exp(r(x, y)/\beta)/Z(x)$.

Since $Z(x)$ is finite and positive, $\pi(y \mid x) = \pi_{\text{ref}}(y \mid x) \exp(r(x, y)/\beta)/Z(x)$ is a probability distribution over responses for every prompt. Taking logarithms and solving for the reward,

$$r(x, y) = \beta \log \frac{\pi(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x),$$

which is the reparameterisation of Equation 5 of Rafailov et al. (2023).

Now substitute this expression for r into the logit of the biased Bradley-Terry model (7):

$$\begin{aligned}
&r(x, y_1) + \theta_k\, \delta_{g(x,y_1)} - r(x, y_2) - \theta_k\, \delta_{g(x,y_2)} \\
&\quad = \beta \log \frac{\pi(y_1 \mid x)}{\pi_{\text{ref}}(y_1 \mid x)} + \beta \log Z(x) - \beta \log \frac{\pi(y_2 \mid x)}{\pi_{\text{ref}}(y_2 \mid x)} - \beta \log Z(x) + \theta_k\big(\delta_{g(x,y_1)} - \delta_{g(x,y_2)}\big) \\
&\quad = \beta \log \frac{\pi(y_1 \mid x)}{\pi_{\text{ref}}(y_1 \mid x)} - \beta \log \frac{\pi(y_2 \mid x)}{\pi_{\text{ref}}(y_2 \mid x)} + \theta_k\big(\delta_{g(x,y_1)} - \delta_{g(x,y_2)}\big).
\end{aligned}$$

The two copies of $\beta \log Z(x)$ cancel because both responses share the prompt, and the bias terms pass through untouched because they are added to the reward and do not involve $Z(x)$. Applying σ to the last line gives the stated form. A bias that entered multiplicatively, or through the partition function, would not cancel in this way. □

B.2 Proof of Theorem 3

Theorem 3 (Identifiability of the bias parameters)**.** *The loss $\mathcal{L}$ of* (8)*, viewed as a function of the reward values $r(x, y)$ and of θ, is jointly convex in (r, θ), and all its minimisers assign the same probability to every judgment. If the annotator pool is connected and $(\hat{r}, \hat{\theta})$ is a minimiser, the minimisers are exactly the pairs*

$$\big(\hat{r} + c\,\delta_g + h,\ \hat{\theta} - c\mathbf{1}\big), \qquad c \in \mathbb{R},$$

with h any function constant on the two responses of every compared pair. Every difference $\theta_k - \theta_{k'}$ therefore takes the same value at every minimiser, while their mean $\bar{\theta} \coloneqq \frac{1}{m}\sum_k \theta_k$ is not determined by the judgments.

We first write the logit. For judgment j the logit of the bias-aware model is

$$L_j(r, \theta) = r(x_j, y_{w,j}) - r(x_j, y_{l,j}) + \theta_{k(j)}\, \Delta\delta_j.$$

As a function of the reward values and the bias vector together, L_j is a sum of parameters with coefficients $+1$, -1 and $\Delta\delta_j$, hence affine. For a fixed policy the reward difference is a constant u_j and $L_j = u_j + \theta_{k(j)}\Delta\delta_j$ is affine in θ alone. The negative log-likelihood is

$$\mathcal{L} = \sum_{j=1}^{N} \phi\big(L_j\big).$$

Convexity follows. ϕ is convex because $\phi'' > 0$. A convex function composed with an affine map is convex, and a sum of convex functions is convex. Hence $\mathcal{L}$ is convex in (r, θ) jointly for an arbitrary reward, and in particular convex in θ for a fixed policy.

All minimisers give the same probabilities. Let (r, θ) and (r', θ') both minimise $\mathcal{L}$, with logits L_j and L'_j. Their midpoint has logits $\frac{1}{2}(L_j + L'_j)$, because L_j is affine, and by convexity it minimises $\mathcal{L}$ as well. Since ϕ is strictly convex ($\phi'' > 0$),

$$\sum_j \phi\Big(\tfrac{L_j + L'_j}{2}\Big) \;\le\; \tfrac{1}{2}\sum_j \phi(L_j) + \tfrac{1}{2}\sum_j \phi(L'_j),$$

with equality if and only if $L_j = L'_j$ for every j. All three quantities equal the minimum of $\mathcal{L}$, so equality holds, the logits agree, and so do the probabilities $\sigma(L_j)$.

For the strictness of the dependence on a single θ_k, take the direction that moves only θ_k. Since $\partial L_j / \partial \theta_k = \Delta\delta_j$ when $k(j) = k$ and 0 otherwise, the second derivative of $\mathcal{L}$ along this direction is

$$\frac{\partial^2 \mathcal{L}}{\partial \theta_k^2} = \sum_{j:\, k(j)=k} \phi''(L_j)\, \Delta\delta_j^2.$$

Every term is non-negative, and a term is positive exactly when $\Delta\delta_j \neq 0$, that is, when judgment j is a cross-group comparison. So the second derivative is positive at every point as soon as annotator k has one cross-group judgment, which is strict convexity in θ_k.

For the characterisation of the minimisers, let (r, θ) and (r', θ') be two parameter vectors and write $dr := r' - r$ and $d\theta := \theta' - \theta$ for their difference. The probability assigned to judgment j is $\sigma(L_j)$, and σ is strictly increasing, so the two vectors assign the same probability to every judgment if and only if $L_j(r', \theta') = L_j(r, \theta)$ for every j. Because L_j is affine, this is the linear system

$$dr(x_j, y_{w,j}) - dr(x_j, y_{l,j}) + d\theta_{k(j)}\, \Delta\delta_j = 0 \qquad \text{for all } j = 1, \ldots, N. \tag{12}$$

It remains to show that the solutions of (12) are exactly the pairs $(dr, d\theta)$ with $d\theta = -c\mathbf{1}$ and $dr - c\,\delta_g$ constant on the two responses of every compared pair, for some $c \in \mathbb{R}$.

We solve the system one compared pair at a time. Fix a compared pair (y_1, y_2) on prompt x and let K be the set of annotators who judged it. If the pair is same-group, then $\Delta\delta_j = 0$ for each of these judgments and (12) reads

$$dr(x, y_1) = dr(x, y_2).$$

If the pair is cross-group, then for each annotator $k \in K$, whichever response that annotator chose, (12) reads

$$dr(x, y_1) - dr(x, y_2) = -\,d\theta_k \left(\delta_{g(x,y_1)} - \delta_{g(x,y_2)}\right). \tag{13}$$

(If the annotator chose y_2, both sides of (12) are the negatives of those in (13), so the equation is the same.) The left side of (13) does not depend on k, and the bracket on the right is ± 1, so $d\theta_k$ takes the same value for every $k \in K$: annotators who judged the same cross-group pair have the same $d\theta_k$.

Connectedness spreads this common value. Two annotators joined by an edge of the graph, that is, who judged a cross-group pair in common, share the same $d\theta$. When the pool is connected, any two annotators are joined by a chain of such edges, so the value is the same along the chain and hence for the whole pool: there is a single constant c with

$$d\theta_k = -c \quad \text{for every } k, \qquad \text{that is,} \qquad d\theta = -c\mathbf{1}.$$

Next the reward part. Substitute $d\theta_k = -c$ back into (13): every cross-group pair satisfies

$$dr(x, y_1) - dr(x, y_2) = c\left(\delta_{g(x,y_1)} - \delta_{g(x,y_2)}\right), \qquad \text{that is,} \qquad \big(dr - c\,\delta_g\big)(x, y_1) = \big(dr - c\,\delta_g\big)(x, y_2).$$

Every same-group pair satisfies the same equality, because there $dr(x, y_1) = dr(x, y_2)$ and $\delta_{g(x,y_1)} = \delta_{g(x,y_2)}$. So $dr - c\,\delta_g$ is constant on the two responses of every compared pair. This proves that every solution of (12) has the stated form.

Conversely, take any c, any $d\theta = -c\mathbf{1}$, and any dr such that $dr - c\,\delta_g$ is constant within every compared pair. For judgment j,

$$dr(x_j, y_{w,j}) - dr(x_j, y_{l,j}) = c\left(\delta_{g(x_j, y_{w,j})} - \delta_{g(x_j, y_{l,j})}\right) = c\,\Delta\delta_j = -\,d\theta_{k(j)}\,\Delta\delta_j,$$

so (12) holds, and the solutions are exactly as claimed.

It remains to put the two halves together. Any two minimisers have the same logits, so their difference solves (12), and conversely every solution leaves all logits and hence $\mathcal{L}$ unchanged. The minimisers are therefore exactly the stated family. Any two of them have $d\theta = -c\mathbf{1}$, so $d\theta_k - d\theta_{k'} = 0$: they agree on every difference $\theta_k - \theta_{k'}$, while the common level of θ shifts by c. □

B.3 Proof of Corollary 4

Corollary 4 (Reward-bias degeneracy). *Write $\theta_k = \bar{\theta} + \varepsilon_k$ with $\sum_k \varepsilon_k = 0$. Shifting the mean by $-c$ and adding $c\,\delta_g$ to the reward leaves every judgment probability unchanged, so $\bar{\theta}$ is not determined by the votes. If the annotator pool is connected, the deviations ε_k take the same value at every minimiser.*

The transformation $r \mapsto r + c\,\delta_g$, $\theta \mapsto \theta - c\mathbf{1}$ is the case of Theorem 3 in which the within-pair-constant part of the reward shift is zero, so the invariance follows from the theorem. The direct check is short. The logit for a comparison (y_w, y_l) by annotator k is

$$L = r(x, y_w) - r(x, y_l) + \theta_k\big(\delta_{g(x,y_w)} - \delta_{g(x,y_l)}\big).$$

After the transformation it becomes

$$\begin{aligned} L' &= \big(r(x, y_w) + c\,\delta_{g(x,y_w)}\big) - \big(r(x, y_l) + c\,\delta_{g(x,y_l)}\big) + (\theta_k - c)\big(\delta_{g(x,y_w)} - \delta_{g(x,y_l)}\big) \\ &= r(x, y_w) - r(x, y_l) + c\big(\delta_{g(x,y_w)} - \delta_{g(x,y_l)}\big) - c\big(\delta_{g(x,y_w)} - \delta_{g(x,y_l)}\big) + \theta_k\big(\delta_{g(x,y_w)} - \delta_{g(x,y_l)}\big) \\ &= L. \end{aligned}$$

Every logit, and so every likelihood term, is unchanged, so the data cannot distinguish $(r, \bar{\theta})$ from $(r + c\,\delta_g, \bar{\theta} - c)$.

For the deviations, Theorem 3 shows that every difference $\theta_k - \theta_{k'}$ takes the same value at every minimiser. Each deviation is an average of such differences,

$$\theta_k - \bar{\theta} = \frac{1}{m}\sum_{k'=1}^{m}\big(\theta_k - \theta_{k'}\big),$$

so each $\varepsilon_k = \theta_k - \bar{\theta}$ does too. □

B.4 Proof of Proposition 5

Proposition 5 (Representability of the bias components). *Decompose $\theta_k = \bar{\theta} + \varepsilon_k$ with $\sum_k \varepsilon_k = 0$. The shared component $\bar{\theta}\,\delta_{g(x,y)}$ is expressible as a function of (x, y) and can therefore be represented inside the policy's implicit reward, whereas the deviation term $\varepsilon_k\,\delta_{g(x,y)}$ depends on k, which the policy never observes, and is representable by no policy $\pi(y \mid x)$ unless every $\varepsilon_k = 0$, that is, unless all annotators have the same bias.*

Write $\theta_k = \bar{\theta} + \varepsilon_k$ with $\sum_k \varepsilon_k = 0$. The bias contribution to the logit of a comparison splits into two terms,

$$\theta_k\big(\delta_{g(x,y_w)} - \delta_{g(x,y_l)}\big) = \underbrace{\bar{\theta}\big(\delta_{g(x,y_w)} - \delta_{g(x,y_l)}\big)}_{\text{does not depend on } k} + \underbrace{\varepsilon_k\big(\delta_{g(x,y_w)} - \delta_{g(x,y_l)}\big)}_{\text{depends on } k}.$$

The first term can be moved into the reward. Define

$$\tilde{r}(x, y) \coloneqq r(x, y) + \bar{\theta}\,\delta_{g(x,y)}.$$

Since $\delta_{g(x,y)}$ is a function of the prompt and the response only, $\tilde{r}$ is a reward function on (x, y) like any other, and the reward difference of $\tilde{r}$ on a comparison equals the reward difference of r plus the first term. By Proposition 2, $\tilde{r}$ admits a policy representation, so the shared component of the bias can be carried by the policy's implicit reward.

The second term cannot. Suppose there were a function $h(x, y)$ with

$$\varepsilon_k\,\delta_{g(x,y)} = h(x, y) \qquad \text{for all } k \text{ and all } (x, y).$$

Pick any (x, y) with $\delta_{g(x,y)} = 1$; one exists whenever the attribute occurs in the corpus. Then $\varepsilon_k = h(x, y)$ for every k, so all deviations are equal to one number, and since they sum to zero that number is 0. Hence the second term is a function of (x, y) only when every $\varepsilon_k = 0$, that is, when all annotators have the same bias. Otherwise no policy $\pi_\theta(y \mid x)$, whose implicit reward is a function of (x, y), can represent it. □

B.5 THE SPEED OF THE MEAN UNDER THE TWO PARAMETERISATIONS

Proposition 6 (The mean learns m times more slowly)**.** *Assume annotators contribute equal data shares and gradient descent with a common learning rate on the bias parameters. Under the shared-mean parameterisation $\theta_k = \bar{\theta} + \varepsilon_k$, one step moves the shared parameter $\bar{\theta}$ exactly m times further than the same step moves the mean of the θ_k under the naive per-annotator model, and moves the deviations from the mean identically.*

By (11), the gradient of the loss with respect to θ_k is

$$\frac{\partial \mathcal{L}}{\partial \theta_k} = \frac{n_k}{N} G_k, \qquad G_k := -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}_k}\Big[\sigma(-u - b_k)\,\Delta\delta\Big],$$

and with equal shares $n_k = N/m$ this is G_k/m.

In the naive per-annotator model each θ_k is its own parameter, so one gradient-descent step with learning rate η moves it by $-\eta\, G_k/m$, and the mean of the parameters moves by

$$\Delta\bar{\theta} = \frac{1}{m}\sum_{k=1}^{m}\Big(-\eta\,\frac{G_k}{m}\Big) = -\frac{\eta}{m}\,\overline{G}, \qquad \overline{G} := \frac{1}{m}\sum_{k=1}^{m} G_k.$$

In the shared-mean model $\theta_k = \bar{\theta} + \varepsilon_k$, the parameter $\bar{\theta}$ enters every θ_k with coefficient 1, so by the chain rule

$$\frac{\partial \mathcal{L}}{\partial \bar{\theta}} = \sum_{k=1}^{m}\frac{\partial \mathcal{L}}{\partial \theta_k} = \sum_{k=1}^{m}\frac{G_k}{m} = \overline{G},$$

and one step moves $\bar{\theta}$ by $-\eta\,\overline{G}$, which is m times the motion of the naive mean.

For the deviations, the gradient on ε_k equals the gradient on θ_k, namely G_k/m, because ε_k enters only θ_k and with coefficient 1. In the naive model the deviation of θ_k from the mean of the θ's therefore moves by

$$-\eta\,\frac{G_k}{m} - \Delta\bar{\theta} = -\frac{\eta}{m}\big(G_k - \overline{G}\big).$$

In the shared-mean model, θ_k moves by $-\eta\,\overline{G} - \eta\, G_k/m$ and the mean of the θ's by $-\eta\,\overline{G} - \eta\,\overline{G}/m$, so the deviation moves by the same amount, $-\frac{\eta}{m}(G_k - \overline{G})$. The constraint $\sum_k \varepsilon_k = 0$ of Proposition 5 is a decomposition, not a constraint enforced in training; the mean of the ε_k drifts by $-\eta\,\overline{G}/m$ per step, the same m-times smaller motion as the naive mean. □

A rough estimate under Adam. The bias parameters are trained with Adam, which divides each coordinate's step by a running estimate of the magnitude of its gradient. Consider a coordinate whose gradient in a step equals $g \neq 0$ with probability p and 0 otherwise, independently across steps, and Adam with learning rate η, moment parameters β_1, β_2 and a negligible stabiliser ϵ. The first-moment estimate m_t is an exponential average of past gradients, so at stationarity $\mathbb{E}[m_t] = p\,g$. The second-moment estimate v_t averages past squared gradients over a window of about $1/(1-\beta_2)$ steps, which for the default $\beta_2 = 0.999$ makes it close to its mean $p\,g^2$. The expected step is therefore

$$-\eta\,\frac{\mathbb{E}[m_t]}{\sqrt{p\,g^2}} = -\eta\,\sqrt{p}\,\operatorname{sign}(g),$$

against $-\eta\,p\,g$ under gradient descent: a coordinate updated in a fraction p of the steps moves at a rate proportional to $\sqrt{p}$ rather than p. In a step of B judgments of which b are cross-group, a free θ_k receives a gradient only from its own annotator's cross-group judgments, so $p_k \approx b/m$ when the judgments are spread evenly over the m annotators, whereas $\bar{\theta}$ receives a gradient whenever the step contains any cross-group judgment, so $p \approx 1$. The shared parameter therefore moves about $\sqrt{m/b}$ times further per step than each free θ_k, and than their mean, in place of the factor m of Proposition 6. In our runs $B = 32$; on MultiPref, with 227 annotators and 59% of the judgments cross-group, the factor is about 3.5 against 227 under gradient descent. This is an estimate, not a proven result: it takes the gradient as constant when present and does not model the policy's own motion, so it orders the two parameterisations and does not predict where the naive mean stops. The measured runs answer that: the free mean ends at 0.50 of $\bar{\theta}$ under formatting and at 0.61 under length (Appendix F.2).

C THE COMMON OFFSET AND THE CHOICE OF ANCHOR

Theorem 3 and Corollary 4 state that the labels determine the annotators' biases up to a common offset. This appendix says what that offset is in terms of the policy, why the reference policy fixes it in every experiment of the paper, and how a requirement on the outputs can fix it instead. Nothing here adds an assumption: the results are those of Section 4, read in a coordinate in which the offset is visible. We write the argument for one binary attribute; with several attributes it applies to each coordinate of θ_k separately, and the swap pairs of Signed-UltraFeedback, which flip exactly one attribute, are the pairs it refers to.

C.1 THE POLICY'S TILT TOWARD THE ATTRIBUTE

By Proposition 2, the policy enters the likelihood only through its implicit reward

$$r(x, y) \coloneqq \beta \log \frac{\pi(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)},$$

a score the policy assigns to every response relative to the reference: positive where the policy has made the response more likely than the reference did, negative where less. The logit of a judgment by annotator k is $r(x, y_w) - r(x, y_l) + \theta_k \, \Delta\delta$ with $\Delta\delta = \delta_{g(x,y_w)} - \delta_{g(x,y_l)}$.

For a response y that carries the attribute, $\delta_{g(x,y)} = 1$, let $\bar{y}$ denote its counterfactual: the same response with the attribute removed or exchanged, so that $\delta_{g(x,\bar{y})} = 0$. The *tilt* of the policy toward the attribute at (x, y) is the extra score it gives to the version that carries the attribute, beyond what the reference gives it:

$$a(x, y) \coloneqq r(x, y) - r(x, \bar{y}) = \beta \left[\log \frac{\pi(y \mid x)}{\pi(\bar{y} \mid x)} - \log \frac{\pi_{\mathrm{ref}}(y \mid x)}{\pi_{\mathrm{ref}}(\bar{y} \mid x)} \right].$$

Every score then splits into the score of the attribute-free version and the tilt,

$$r(x, y) = q(x, y) + a(x, y) \, \delta_{g(x,y)},$$

where $q(x, y)$ is $r(x, \bar{y})$ when y carries the attribute and $r(x, y)$ otherwise. This is a change of variables, not a model: q is the part of the score that concerns the answer itself, and a is the part that concerns the attribute. The tilt is a function of (x, y), which the policy can represent (Proposition 5); the case to keep in mind is the one in which it takes a single value a for every response, since that is the part of the tilt on which the degeneracy acts.

The two kinds of comparison in the data now read differently. On a pair whose two responses differ only in the attribute, q cancels and the logit of annotator k is

$$a + \theta_k,$$

the policy's tilt plus the annotator's bias and nothing else. On a pair whose two responses carry the same attribute value, the tilt cancels and the logit is $q(x, y_w) - q(x, y_l)$, quality alone. Mixed pairs combine the two.

C.2 WHAT THE LABELS DETERMINE

The transformation of Corollary 4, $r \mapsto r + c \, \delta_g$ and $\theta_k \mapsto \theta_k - c$ for every k, adds c to the tilt of every response and subtracts c from every bias, and every logit is unchanged. In the coordinates above, Theorem 3 says three things.

1. The quality part q is determined up to the shifts that standard DPO cannot see either, a constant within each compared pair.
2. For every annotator the sum $a + \theta_k$ is determined. Hence every difference $\theta_k - \theta_{k'}$ is determined, and in the parameterisation $\theta_k = \bar{\theta} + \varepsilon_k$ every deviation ε_k is determined.
3. The split of $a + \bar{\theta}$ into a and $\bar{\theta}$ is not determined.

The common offset is therefore not a parameter but a direction: the line $a + \bar{\theta} = \text{const}$ in the $(a, \bar{\theta})$ plane, along which the likelihood is flat. Fixing the model means choosing a point on this line, and since every point fits every judgment equally well, the choice is an input, not an estimate. The statistical content is the one stated after Corollary 4: a bias shared by every annotator cannot be told apart from the attribute genuinely carrying reward. Declaring the attribute states that the shared preference for it is bias; the anchor states from which zero that bias is counted.

C.3 REFERENCE ANCHORING

Every run in this paper starts from the reference, $\pi = \pi_{\mathrm{ref}}$, so at initialisation $r \equiv 0$, $a = 0$ and $\theta = 0$. Training moves $a + \bar{\theta}$ to the value the labels demand. Which of the two moves is decided by the dynamics, not by the objective: the scalar has a convex path to its conditional optimum (Theorem 3) and its own learning rate, whereas the policy moves through a non-convex landscape under the KL anchoring, so the level goes into $\bar{\theta}$ and a stays near zero (Section 4; Proposition 6 explains why the naive per-annotator model gets there m times more slowly). The learned $\bar{\theta}$ is therefore the annotators' shared tilt toward the attribute measured relative to the reference, and the debiased policy's attribute rate is the reference's rate.

Three consequences follow. First, removal is measured against the reference, which is why the tables report the share of DPO's increase over the reference that is removed, and why the 0.5B reference of Signed-UltraFeedback is balanced on both attributes by construction (Appendix D.2): the bias then arrives through the preference stage alone. Second, a reference that is itself tilted stays tilted by its own amount: at 8B the reference signs with a woman-coded name at 0.465, and BA-DPO returns to 0.49, not to 0.50 (Table 1). Third, the estimates of the annotators do not depend on the reference's tilt: $r \equiv 0$ at the start whatever the reference prefers, so $\bar{\theta}$ measures the tilt of the labels beyond zero implicit reward, and the deviations ε_k are the same at every point of the line by item (ii) above. The audit of the annotators is anchor-independent; only the policy's rate depends on the anchor.

C.4 TARGET ANCHORING

The same free direction can be fixed by a requirement on the outputs instead of by the reference. Suppose the attribute has a fair rate t, one half for the name in a signature, and write $\operatorname{logit} p = \log \frac{p}{1-p}$. Let p be the reference's rate in the pairwise sense, $\log \pi_{\mathrm{ref}}(y \mid x)/\pi_{\mathrm{ref}}(\bar{y} \mid x) = \operatorname{logit} p$. A policy with tilt a has log-odds $\operatorname{logit} p + a/\beta$, so the tilt that reaches the target is

$$c(t) = \beta \left[\operatorname{logit} t - \operatorname{logit} p\right]. \tag{14}$$

When the reference's log-odds vary across prompts, the rate of the tilted policy is still continuous and strictly increasing in c, so the target is reached at exactly one value, found by a one-dimensional search; (14) is that value for constant log-odds.

Corollary 7 (Target anchoring)**.** *Let (r, θ) be any parameter vector and $c \in \mathbb{R}$. The parameter vector $(r + c\,\delta_g,\ \theta - c\mathbf{1})$ assigns the same probability to every judgment, has the same deviations ε_k, has shared mean $\bar{\theta} - c$, and corresponds to the policy $\pi_c(y \mid x) \propto \pi(y \mid x) \exp\big(c\,\delta_{g(x,y)}/\beta\big)$, whose log-odds between a response and its counterfactual exceed those of π by c/β.*

Proof. The first three claims are Corollary 4. The policy follows from $\pi \propto \pi_{\mathrm{ref}} \exp(r/\beta)$ (Proposition 2) with $r + c\,\delta_g$ in place of r, and the log-odds shift is c/β because δ_g differs by one between y and $\bar{y}$. □

Target anchoring therefore changes exactly one thing. The fit to every judgment is the same, so nothing is paid in likelihood. The deviations are the same, so the audit of the annotators is the same. The shared mean shifts by $-c(t)$, which means that the reference's own tilt is now counted as bias rather than as quality. What changes is the policy, which is moved along the attribute by $c(t)/\beta$ in log-odds, and with it the KL to the reference. The requirement spends the one degree of freedom the data do not use, and nothing else moves.

How the anchor is applied. Three procedures produce the tilted policy. At sampling time, in closed form: the responses that carry the attribute are reweighted by $\exp(c/\beta)$ per prompt, which for a signature is a shift on the choice of the name and for an attribute of the whole text is a reweighting or a rejection step. In training, with a fixed offset: on pairs that differ only in the attribute, each taken in both orders, the loss of the two orders is $-\log \sigma(u+b) - \log \sigma(-u-b)$ with u the policy's margin, minimised at $u = -b$, so a fixed offset $b = -c(t)$ in the logit drives the policy to carry the tilt $c(t)$ itself. This removes a tilt a policy already has, from its own generations and with no annotator labels, and is the natural cure for a model trained without the bias term; Section C.5 tests it. Through the reference: a reference balanced on the attribute makes the two anchors coincide, which is the construction used at 0.5B.

Limits. Target anchoring needs an attribute with a fair value. A name has one; length and formatting do not, and for them the reference remains the only sensible anchor, which is why Table 2 reads the held-out gap rather than a target rate. The target is a normative choice, as the attribute partition is, but it is one that can be audited by measuring the rate, whereas the reference's rate is an accident of pretraining. Moving the policy changes which responses it produces, so judged quality has to be checked after the tilt. Finally, the level can also be pinned from data rather than from a target: a small set of judgments by annotators known to be unbiased breaks the flat direction, since for them $\theta_k = 0$ is known and a is then identified. We leave this to future work (Section 6).

Worked example. Take a response signed *Emily Miller* and its counterfactual signed *Greg Miller*, the same text otherwise. The policy's tilt a is β times the difference between the policy's and the reference's log-odds for the first version over the second. When annotator k judges this pair, the logit is $a + \theta_k$: the labels of the swap pairs fix, for every annotator, the sum of the policy's tilt and the annotator's bias, but not how it splits. At initialisation $a = 0$, and after training under reference anchoring a is still near zero and $\bar{\theta}$ holds the shared level. For the numbers, take $\beta = 0.1$. The 8B reference signs with a woman-coded name at 0.465, so $\operatorname{logit} p = -0.14$ and reaching $t = 0.5$ needs $c = 0.1 \times (0 + 0.14) = 0.014$, a shift of 0.14 in log-odds: for a nearly balanced reference the two anchors almost coincide. The 8B DPO policy trained on the same labels signs with a woman-coded name at 0.967, so $\operatorname{logit} p = 3.38$ and curing it needs $c = -0.34$, a shift of -3.4 in log-odds, which the training-time procedure obtains with the offset $b = 0.34$ on the gender coordinate.

C.5 Experiment: a fixed offset cures the DPO policy

Setting. The source is the DPO policy of Signed-UltraFeedback at 0.5B (Table 1), one per seed, which signs with a woman-coded name with probability 0.96 and with a black-coded name with probability 0.99. The training pairs come from the policy itself: $3{,}000$ fresh UltraFeedback prompts (the ones after the $8{,}000$ of the corpus) with the signing instruction, answered by the source; the answers signed with a pool name are kept ($2{,}659$ at seed 42); each signature is replaced by a random cell and then by the cell that differs from it in one attribute, with the same surname, which gives one pair per attribute per answer, taken in both orders: $10{,}636$ rows, of which $9{,}576$ are used for training and $1{,}060$ held out. No annotator and no label enter this stage. Since every pair is present in both orders, the loss on a pair is $-\log \sigma(u + b) - \log \sigma(-u - b)$ with u the policy's margin, minimised at $u = -b$, so the offset b in the logit (the term $b\,(g_w - g_l)$ of Section C.4) is the only thing the stage teaches. Training is DPO with the source as reference and the hyperparameters of Table 7 ($\beta = 0.1$, learning rate 5×10^{-7}, $1{,}000$ steps of 32 pairs). The offset for a target rate t is $b = -\beta s^*(t)$, where $s^*(t)$ is the log-odds shift that brings the source's signature probabilities to t, found by bisection on the 300 evaluation prompts: for parity at seed 42, $s^* = 3.39$ on the gender coordinate and 5.06 on the race coordinate.

Calibration. The exact tilt alone stops short: at seed 42 it reaches 0.736 and 0.753 instead of 0.50, two thirds of the intended shift in log-odds. Three offsets at seed 42 ($3.39/5.06$, $4.42/6.18$ and $5.13/6.92$ in log-odds, gender/race) give achieved shifts of $2.24/3.68$, $3.13/4.76$ and $3.79/5.51$, which lie on a line with residuals below 0.015: achieved $= 0.888\,\text{intended} - 0.777$ on gender and $0.980\,\text{intended} - 1.285$ on race. The policy therefore follows the offset one for one after a fixed deficit of 0.8 and 1.3 log-odds, which the $1{,}000$ steps leave. Every other run below inverts this line for the exact tilt of its target; the line is fitted at seed 42 and applied unchanged to the other two source seeds and to the targets 0.3 and 0.7.

Results. Table 4 reports the runs. With the calibrated offset the DPO policy signs with a woman-coded name at 0.50 ± 0.07 and with a black-coded name at 0.47 ± 0.08 over the three source seeds, from 0.96 and 0.99; the targets 0.3 and 0.7 are reached within 0.07 and 0.04, and the rate is monotone in the offset. Nothing else moves: the signing rate, the length and the judge score with the signature removed stay within their seed spread, the KL to the source is at the noise floor of the estimate except at the largest offset (0.034), and the KL to the SFT reference falls from DPO's 0.037 to 0.022 at parity, below the 0.025 of BA-DPO, since the part of DPO's divergence that the offset removes is the name shift itself. The held-out gap, which marks a policy that has internalised the bias, falls from DPO's 0.095 to 0.03 after the exact tilt and to -0.05 at parity, where the cross-group accuracy (0.50 to 0.53) no longer exceeds the same-group one; at the target 0.3 it reaches

Table 4: A fixed offset applied to the DPO policy of Signed-UltraFeedback at 0.5B, trained on the policy's own generations with no annotator. Rates are the probability readout of Table 1; RM^- is the judge with the signature removed; KL is per token, to the source (the DPO policy) and to the SFT reference; <0.01 marks an estimate below the noise of the Monte Carlo estimate, which can come out negative; the held-out gap is defined in Section 5.1; t is the target rate; *Signed* is the share of answers that carry a signature. Seed 42 unless stated; the parity row over three seeds gives the mean and the 95% interval over the source seeds.

Policy	t	p(woman)	p(black)	**Signed**	**Tokens**	$\mathbf{RM}^-$	**KL** src / SFT	**Gap**
DPO (source)	–	0.963	0.992	0.84	213	-2.75	– / 0.037	$+0.095$
+ offset, exact tilt	0.5	0.736	0.753	0.86	210	-2.79	<0.01 / 0.028	$+0.029$
+ offset, calibrated	0.5	0.533	0.509	0.86	214	-2.71	<0.01 / 0.023	-0.046
+ offset, calibrated, 3 seeds	0.5	0.50 ± 0.07	0.47 ± 0.08	0.86	213	-2.72	0.003 / 0.022	-0.053
+ offset, calibrated	0.3	0.262	0.233	0.84	217	-2.77	0.034 / 0.024	-0.144
+ offset, calibrated	0.7	0.681	0.659	0.85	206	-2.90	<0.01 / 0.027	$+0.019$
BA-DPO pooled, Table 1	–	0.545	0.616	0.86	213	-2.53	– / 0.025	$+0.005$

-0.14, the policy now siding against the annotators on the pairs that differ in the attribute. The one remaining deficit is the constant one of the calibration paragraph: a property of the training budget, not of the objective, which a longer run or the second calibration point removes. Two remarks. First, the sampled signatures lag the probability readout at mild offsets and swing past it at strong ones (at parity, 69% of the sampled signatures at seed 42 still carry a black-woman name while the pool probabilities are balanced; at the target 0.3, 92% carry a white-man name), the concentration of the decoder on a few names described in Appendix F; the probability readout is the quantity the offset controls. Second, the offset is the same quantity as the sampling-time tilt of Section C.4 up to the fixed deficit, so a policy can be cured either by one training stage on its own generations or by reweighting at sampling time, with the same target.

D DATASETS

Table 5 lists the two corpora, their sizes and the modifications made to each; the subsections give the construction in full. The 8B runs use the same data as the 0.5B runs.

Table 5: The corpora. Judgments are training / held-out rows; the held-out rows are 10% of prompts, split by prompt id. On MultiPref the 300 evaluation prompts come from the held-out side; on Signed-UltraFeedback they come from UltraFeedback's test split. Cross-group is the share of judgments in which exactly one response carries the attribute.

Corpus	**Size**	**Annotators**	**Attributes (cross-group)**	**Modifications**
MultiPref (Miranda et al., 2025)	10,461 comparisons over 5,323 prompts; 27,812 / 3,035 judgments	227 real evaluators, four per comparison	length, ratio ≥ 1.5 (58.7%); formatting, markdown present (26.4%)	ties dropped (26% of rows); judgments kept disaggregated with the evaluator id; attributes computed from the response texts
Signed-UltraFeedback	7,177 UltraFeedback pairs (from 8,000, responses under five words dropped) in three types: 2,887 quality, 1,462 swap, 2,828 mixed; 25,840 / 2,868 judgments	60 with planted biases in three classes of 20, $\theta_k \in \mathbb{R}^2$	woman-coded and black-coded first name in the signature (36.4% each)	signature line `--- First Last` appended, names from an 89-name pool; swap pairs duplicate one response under two names; labels re-drawn; SFT answers re-signed at random so the reference starts balanced

D.1 THE MULTIPREF CORPUS AND THE LENGTH ATTRIBUTE

`allenai/multipref` contains 10,461 comparisons, each judged by exactly four evaluators (two crowd workers and two experts; 227 distinct evaluators, no overlap between the two pools), over 5,323 distinct prompts. We keep every judgment as its own training row, so that annotator identity is available to the per-annotator model, and drop the 26% of judgments marked as ties, leaving 30,847 rows. Ten percent of prompts are held out by prompt id (27,812 train, 3,035 held-out rows); the 300 evaluation prompts are drawn from the held-out prompts. The reference policy is fine-tuned on the majority-chosen response of each comparison and is shared by every method and seed.

Length as a binary attribute. A response is assigned $\delta_g = 1$ when it is markedly longer than its partner, defined as a word-count ratio of at least 1.5; pairs closer in length than that are same-group: both responses have $\delta_g = 0$. Because this attribute is defined on the pair, it is not a function of (x, y): Proposition 2 and Theorem 3 need only a δ_g per judgment and are unaffected, whereas the representability of the shared component inside the policy's implicit reward (Proposition 5) holds for length only through a per-response proxy such as the token count. The threshold is part of the attribute's definition and not a tuning choice: assigning $\delta_g = 1$ to the longer side of every pair makes 99.5% of MultiPref comparisons cross-group and leaves no same-group comparisons to identify the reward. At ratio 1.5, 58.7% of judgments are cross-group, the longer response wins 73% of those, and the corpus-level offline bias estimate is $\hat{\theta} = 0.99$ in log-odds. Ratios of 1.25, 2 and 3 give cross-group fractions of 73%, 40% and 21% with $\hat{\theta}$ of 0.91, 1.04 and 1.05; the estimate is stable across thresholds.

Formatting as a binary attribute. A response is assigned $\delta_g = 1$ when it contains markdown markup. The attribute is declared on the same 27,812 training and 3,035 held-out judgments as length; 26.4% of judgments are cross-group and the offline estimate is $\hat{\theta} = 0.98$. Table 6 characterises both attributes from the preference data before any training.

Table 6: The declared attributes on MultiPref, characterised from the preference data before any training. *Cross-group* is the fraction of judgments in which exactly one response carries the attribute, which are the only judgments that identify θ. $\hat{\theta}$ is the log-odds that the attribute-carrying side wins such a judgment, restated as odds in the next column. The last column is DPO's measured amplification over the reference at 0.5B, in units of its own per-seed standard deviation.

Attribute	Cross-group	$\hat{\theta}$	Odds	DPO amplification
Length	58.7%	0.99	2.7 : 1 longer	$+96$ tokens (65σ)
Formatting	26.4%	0.98	2.7 : 1 markdown	$+0.034$ (1.6σ)

D.2 THE NAMES CORPUS (SIGNED-ULTRAFEEDBACK)

The NAMES corpus is built rather than collected: the responses come from a public preference corpus, a name signature is added to them, and annotators whose biases are planted relabel every pair, so the bias of every annotator is known. Two attributes are planted at once, and the annotators are organised in classes whose biases differ. The main text calls it Signed-UltraFeedback (Section 5.2). We stress that the annotators are constructed to prefer a name marker. It does not measure prejudice, and we refer to the attributes as *gender-coded* and *race-coded names* rather than as gender or race bias.

Prompts and responses. We draw 8,000 preference pairs from the `train_prefs` split of `ultrafeedback_binarized` (Cui et al., 2024) and discard pairs in which either response has fewer than five words, which leaves 7,177. Each pair carries a prompt x, two responses and two GPT-4 quality scores $s_1, s_2 \in [1, 10]$ from the original dataset.

Marker. The marker is a signature line `--- First Last` appended to a response. First names come from four cells (white woman, white man, black woman, black man): the audit lists of Bertrand & Mullainathan (2004) together with the WEAT 3 and WEAT 5 sets of Caliskan et al. (2017),

$23/23/18/25$ names per cell, 89 in all. Surnames come from a neutral pool and are shared by both sides of a pair, so within a pair only the first name, and hence only gender or race, differs. A response carries a two-entry attribute vector $\delta_g(y) = (\mathbf{1}[\text{woman-coded}], \mathbf{1}[\text{black-coded}])$, read from the signature; an unsigned response is $(0, 0)$. Token balance across cells is not achievable with published names: under the Qwen tokenizer nearly every white-coded name is one token and most black-coded names are two or three (mean 1.1, 1.0, 2.1 and 2.0 tokens per cell). We keep the full pool and read the rate from probabilities over every name (Appendix F.3).

Pair types. Three kinds of pair share the corpus, in proportion $40/20/40$. *Quality* pairs are the two real responses, both unsigned or both signed with the same name, and identify quality alone. *Swap* pairs are one response duplicated and signed with two names that differ on exactly one attribute, with true quality margin $q = 0$; they identify the bias with quality held equal by construction. *Mixed* pairs are the two real responses signed from different cells, where quality and bias compete. The realised counts are 2,887, 1,462 and 2,828.

Annotators and labels. Sixty annotators sit in three classes of twenty. Annotator k in class c has $\theta_k = \mu_c + \varepsilon_k$ with $\varepsilon_k \sim \mathcal{N}(0, 0.8^2 I)$. The class means are $\mu_A = (1.2, 2.5)$, $\mu_B = (0.8, -0.5)$ and $\mu_C = (1.0, 1.0)$. Both attributes have a population mean of 1.0 and differ only in how much the classes disagree: the class standard deviation is 0.16 on the gender-coded attribute and 1.22 on the race-coded one, where class B opposes the others. With twenty annotators per class the realised class means are 0.98, 1.12 and 0.89 on the gender-coded attribute and 2.62, -0.34 and 1.11 on the race-coded one (pooled 1.00 and 1.13). Each pair receives four judgments. For each judgment an annotator k is drawn uniformly and the originally chosen response y_1 is preferred to y_2 with probability

$$\sigma\big(q + \theta_k^\top (\delta_g(y_1) - \delta_g(y_2))\big), \qquad q = \kappa\,(s_1 - s_2),\ \kappa = 0.5,$$

otherwise the label is flipped. Pairs whose responses carry the same attribute values are thus labelled from quality alone, while the others are tilted by the annotator's bias. This is the likelihood of (7), so the corpus tests whether the method recovers what it is specified to recover; MultiPref is where the model may be misspecified. Ten percent of prompts are held out by prompt id, which gives 25,840 training and 2,868 held-out judgments over the same annotators; the reference is fine-tuned on the majority-chosen response of each pair, each re-signed with a name from a random cell before fine-tuning, so the reference starts balanced on both attributes and all of the bias arrives through the preference stage.

Gate. The following statistics are read from the labels before any training. 36.4% of judgments are cross-group on each attribute, 59.8% on at least one. On swap pairs the offline log-odds that the attribute-carrying side wins is 0.95 for the gender-coded attribute and 0.70 for the race-coded one; by class, 1.01, 0.97 and 0.86, and 2.30, -0.37 and 0.88. The pooled race-coded estimate is below the gender-coded one although its planted mean is higher, because class B votes against the other two. On all cross-group rows both estimates are 0.71 and 0.70, below the swap-pair values, because the quality margin adds noise wherever the two texts differ.

Evaluation prompts. Three hundred held-out questions from `test_prefs`, each followed by "Sign your answer with a name of your choice", give the attribute rate: the probability the policy assigns to the names of each cell at the signature position (Appendix E.4). One hundred templated prompts present two candidates with identical qualifications and names from different cells, in random order, and ask for a recommendation; they are read as the share of recommendations going to each cell and are not used in the tables.

Configuration. In summary: 20 annotators per class, a within-class standard deviation of 0.8, the class means above, quality, swap and mixed pairs in proportion $40/20/40$ with half of the quality pairs unsigned, four judgments per pair, a quality scale $\kappa = 0.5$, and data seed 0.

E EXPERIMENTAL DETAILS

E.1 MODELS AND TRAINING

`Qwen2.5-0.5B-Instruct` is trained with full fine-tuning: fp32 master weights under bf16 autocast, because bf16 master weights lose updates at a DPO learning rate of 5×10^{-7}. `Llama-3.1-8B-Instruct` and `Mistral-7B-Instruct-v0.3` are trained with LoRA on all linear layers (rank 32, $\alpha = 64$, fp32 adapters on a bf16 base) at learning rate 5×10^{-6}, ten times the full fine-tuning rate; the rate was fixed by the calibration of Appendix F.4, in which LoRA at this rate reproduces the amplification and the removal of full fine-tuning to within seed noise, while a larger rate lets the policy move faster than the bias term; Mistral uses the same recipe unchanged. All models use $\beta = 0.1$, 1000 steps at effective batch 32, and a separate Adam at learning rate 10^{-2} for the bias parameters, since the scalars otherwise receive too little gradient to move. One SFT reference per model and corpus (a LoRA reference for the two larger models), trained for one epoch on the majority-chosen response of each comparison, is shared by every arm and seed; without it the implicit rewards would not be comparable across arms. Reference log-probabilities are precomputed once per corpus and cached, and per-position log-probabilities are computed in position chunks so that the full vocabulary logits never materialise at once. Every arm sees the identical disaggregated judgments through the same implementation; the arms differ only in their loss. All runs use a single shared NVIDIA A100 80GB. A 0.5B preference run peaks at 11 to 19 GB and takes 70 to 150 minutes depending on contention; an 8B LoRA run peaks at 21.5 GB and takes 7 to 9 hours, a Mistral run at 17.7 GB and 8 to 12 hours. Table 7 lists the hyperparameters.

Table 7: Hyperparameters. The reference checkpoint is shared across all methods and seeds.

Stage	**Setting**
SFT (reference)	1 epoch on majority-chosen responses, effective batch 32
SFT lr	1×10^{-5} (0.5B, full); 1×10^{-4} (8B and 7B, LoRA)
Preference stage	$\beta = 0.1$, 1000 steps, effective batch 32 (2×16)
Preference lr	5×10^{-7} (0.5B, full); 5×10^{-6} (8B and 7B, LoRA)
LoRA (8B and 7B)	rank 32, $\alpha = 64$, all linear layers, fp32 adapters on a bf16 base
LoRA reference	a LoRA SFT of the same shape, shared by every arm and seed
Sequence lengths	at most 1280 tokens for prompt and response together, of which at most 384 for the prompt
Schedule	cosine with 10% warmup, gradient clipping 1.0
Bias parameters	separate Adam at lr 10^{-2}, no penalty, initialised at 0 unless warm-started
Generation	300 held-out prompts, ≤ 512 new tokens, $T = 0.7$, top-p 0.9
Judge	`Skywork-Reward-V2-Qwen3-1.7B`, truncated to 2048 tokens
Seeds	42, 123, 456

E.2 ARMS

Reference is the shared SFT policy and the zero point of every comparison. *DPO* is standard DPO, that is BA-DPO with every θ_k frozen at zero. The BA-DPO arms differ only in how the bias term of (8) is indexed (Section 3). *Pooled* learns one scalar θ per attribute shared by every comparison. $\bar{\theta}+\varepsilon_k$ learns a shared mean per attribute plus one deviation per annotator, with the mean updated on every cross-group comparison. θ_k *only* is the original BARP model, one free parameter per annotator and nothing shared. *Shuffled ids* is the θ_k-only model in which every judgment is assigned an annotator index drawn uniformly at random, independently of who cast it, so that every θ_k collects votes from every annotator and the ids carry no information; a mere relabelling of the ids would change nothing and is not what this control does. $\bar{\theta} + \delta_c + \varepsilon_k$ inserts a per-class offset between the mean and the individual deviation and receives the class labels. *Warm* arms initialise $\bar{\theta}$ (or the pooled θ) at the corpus-level offline estimate, the log-odds that the $\delta_g = 1$ side wins a cross-group comparison, instead of at zero. *Gender declared only* is the pooled arm with the race-coded attribute left undeclared. The main text carries pooled and $\bar{\theta} + \varepsilon_k$; the others are in Appendix F.

E.3 BASELINES

R-DPO (Park et al., 2024) adds a length penalty $\alpha\,(|y_w| - |y_l|)$ in tokens to the DPO logit; we use $\alpha = 0.005$, since the published $\alpha = 0.02$ collapses the policy on MultiPref (Appendix F.2). SamPO (Lu et al., 2024) down-samples the token log-ratios of the longer response so that both responses contribute equally many terms to the implicit reward. Neither reads the declared attribute, so their length checkpoints are scored unchanged under formatting. Group-DRO DPO adapts Sagawa et al. (2020) to the DPO loss: each group carries a loss weight, updated by exponentiated gradient with step $\eta = 0.5$ toward the group with the highest current loss; groups are the annotator classes on Signed-UltraFeedback and the annotators on MultiPref. Crowd-PrefRL (Chhan et al., 2024), adapted in the same way, moves the weight away from the highest-loss annotator ($\eta = 0.5$). EM-DPO with MinMax-DPO (Chidambaram et al., 2026) trains $K = 3$ type policies, the planted number of classes: each annotator holds a probability of belonging to each type; the M-step runs DPO for each type on judgments weighted by those probabilities, and the E-step updates the probabilities from how well each type policy explains the annotator's judgments, scored on votes the round's policies did not train on (each annotator's votes are split into two halves that swap every round; four rounds of 200 steps, after which the three policies train on all votes for the full 1000 steps). MinMax-DPO mixes the type policies with the weights that minimise the largest regret over the types; the mixture answers each prompt with type policy k with probability w_k, so its readouts are the weighted means of the type policies' readouts, and its vote prediction uses each annotator's own type probabilities. No public implementation matched our data and single-GPU setting, so we reimplemented the method; departures from the original are documented in the released code.

E.4 READOUTS

Generation. Each policy generates on 300 held-out prompts with temperature 0.7, top-p 0.9 and at most 512 new tokens.

Attribute rate on Signed-UltraFeedback. The rate is read from probabilities, not from samples. With the reference's answer body fixed, the probability the model assigns to each of the 89 first names at the signature position is renormalised over the pool and summed over the woman-coded and over the black-coded names, giving $p(\text{woman})$ and $p(\text{black})$; the average over the 300 sign prompts is the rate. Sampled signature rates saturate and depend on the decoder: the reference, balanced by probability at 0.50 and 0.52, signs 79% of its answers with a black-coded name at temperature 0.7 with top-p 0.9 (Appendix F.3). Using each arm's own body instead of the reference's changes no number by more than 0.005. Computing the rate requires one forward pass per prompt and name batch.

Attribute rate on MultiPref. Under length the rate is the mean token count of the generations. Under formatting it is the markdown rate reweighted onto the reference generations' token-count distribution (five quantile bins), so that a method which merely shortens its answers does not appear to remove bias.

Bias removed. DPO's amplification of the attribute is the gap between DPO's rate and the reference's, and an arm's removal is the share of that gap it closes, $(\text{rate}_{\text{DPO}} - \text{rate}_{\text{arm}})/(\text{rate}_{\text{DPO}} - \text{rate}_{\text{ref}})$, computed per seed against DPO of the same seed and then averaged, so 0% is DPO and 100% is the reference.

Held-out accuracy and the gap. On the held-out judgments the policy's implicit reward margin u of (9) predicts the label by its sign. Accuracy is reported on same-group pairs (the two responses agree on the attribute) and cross-group pairs (they differ), and the gap is cross-group minus same-group. A policy that has internalised the annotators' bias predicts their labels better exactly where the attribute differs, so debiasing shows as the gap falling to zero while same-group accuracy is unchanged.

Judge. Quality is scored by `Skywork-Reward-V2-Qwen3-1.7B` (Liu et al., 2026) on the generations. Where an attribute-invariant transform exists the attribute is stripped from every arm's

generations equally before scoring (the signature on Signed-UltraFeedback, the markup under formatting), written RM^{-}; under length none exists and the raw score is reported. The judge's scale differs between the three models' outputs, so scores are compared within a model only.

Distance from the reference. For every trained policy we report a Monte Carlo estimate of $\mathrm{KL}(\pi_\theta \,\|\, \pi_{\mathrm{ref}})$ on the policy's own generations: the mean over generated tokens of $\log \pi_\theta(y \mid x) - \log \pi_{\mathrm{ref}}(y \mid x)$ with $y \sim \pi_\theta$. It distinguishes a method that removes a bias by changing the policy from one that removes it by leaving the policy near the reference.

Vote prediction with the arm's own bias parameters. On the held-out judgments whose two responses differ on the attribute, the arm predicts the voter's label by the sign of $u + \theta_k^\top(\delta_g(y_w) - \delta_g(y_l))$ with its own learned θ_k (pooled uses its single θ; arms without a bias term use u alone; EM-DPO uses each annotator's type probabilities). The ceiling is the same prediction with the planted θ_k, averaged over arms and seeds.

F Ablations and controls

Everything in this appendix except the last subsection was run on `Qwen2.5-0.5B-Instruct` with the training configuration of Section 5.1, at three seeds unless a row says otherwise. It covers the variants of the bias model that Section 3 defines and the main text does not table, why the names tables read probabilities rather than samples, the remaining baselines, the LoRA calibration on which the 8B training configuration rests, and a replication on `Mistral-7B-Instruct-v0.3`. Column definitions follow the main text; intervals are one standard deviation over seeds, except in the Mistral table, which uses the 95% intervals of the main text.

F.1 Variants of the bias model on Signed-UltraFeedback

Table 8 adds to Table 1 the arms not reported in the main text, defined in Appendix E.2. The two length baselines, which the rule of Section 5.1 excludes from this corpus, are included for completeness.

Table 8: Signed-UltraFeedback at 0.5B: the arms not reported in the main text, under the three arms of Table 1 for reference. Rows without a method name are BA-DPO variants. Rate columns as in Table 1, with the share of DPO's amplification removed in parentheses; votes: vote prediction with the arm's own θ on the race-coded / gender-coded attribute overall (Table 13). The judge score is inside DPO's 95% interval for every arm and is omitted. Mean $\pm$ sd over three seeds; the gender-only control ran at seed 42.

Method	p(woman) (removed)	p(black) (removed)	**KL**	**Votes r / g**
DPO	0.964 ± 0.002	0.991 ± 0.001	$0.038 \pm <0.001$	0.641 / 0.673
Pooled	0.549 ± 0.003 (89%)	0.608 ± 0.007 (81%)	$0.026 \pm <0.001$	0.649 / 0.694
$\bar\theta + \varepsilon_k$	0.542 ± 0.004 (91%)	0.595 ± 0.007 (83%)	$0.026 \pm <0.001$	0.752 / 0.745
Class, $\bar\theta + \delta_c + \varepsilon_k$	0.526 ± 0.004 (94%)	0.582 ± 0.006 (86%)	$0.026 \pm <0.001$	0.749 / 0.740
Shuffled ids	0.849 ± 0.009 (25%)	0.925 ± 0.008 (14%)	$0.034 \pm <0.001$	0.645 / 0.694
Pooled, gender declared only	0.836 (28%)	0.986 (1%)	0.034	0.636 / 0.689
R-DPO ($\alpha = 0.005$)	0.967 ± 0.002 (-1%)	0.992 ± 0.001 (0%)	0.054 ± 0.002	0.634 / 0.659
SamPO	0.940 ± 0.001 (5%)	0.981 ± 0.002 (2%)	0.041 ± 0.001	0.625 / 0.669

The class arm is the best arm on both attributes at every seed, ahead of pooled by 0.02 on each, the same margin on the attribute the classes agree on and on the one they do not, so the gain is not specific to disagreement. Its race-coded class offsets, centred like the planted ones, come out at $+0.98$, -1.00 and $+0.03$ (three-seed means) against planted offsets of $+1.49$, -1.47 and -0.02 around the population mean: the order is recovered, the magnitudes are attenuated by about a third, and its vote prediction is at the ceiling, as for $\bar\theta + \varepsilon_k$. When the classes are known it is a reasonable choice; it improves the policy by 0.02 and adds nothing to what the deviations already reveal, which is why it is reported here rather than in the main text. Shuffling the ids removes 25% and 14%. Its sixty parameters converge to noisy copies of one value near 0.4 (learned means 0.45 and 0.39,

correlation with the planted θ_k of 0.12 and -0.16, means over three seeds), its vote prediction is at pooled's level, and its KL is close to DPO's: with the identities destroyed, the parameters neither describe the annotators nor reach the mean within the training budget. Declaring only the gender-coded attribute leaves the race-coded one at DPO's level (0.986) and removes only 28% of the declared one. The undeclared bias is absorbed by the policy, which concentrates its signatures in the black-woman cell, and the declared attribute is carried along with it. Both attributes must be declared for either to be removed, which is the substitution pattern of Section 5.4 inside one corpus. The two length methods remove at most 5% of either attribute, and R-DPO's length penalty raises the KL to the reference to 0.054 without changing the signature: a method built for length has no effect on a name signature.

F.2 MULTIPREF: VARIANTS, WARM STARTS AND THE REMAINING BASELINES

Table 9 is the length experiment of Table 2 with every arm we ran at 0.5B: the free-θ_k arm of the original BARP model, shuffled ids, the two BA-DPO arms warm-started at the corpus-level offline estimate $\hat{\theta} = 0.99$ instead of zero, and R-DPO at its published strength. Group-DRO DPO is in Table 2 at three seeds. Crowd-PrefRL (Chhan et al., 2024), adapted to the DPO loss, moves weight away from the annotators with the highest loss; at one seed it leaves the answers at 386 tokens and the gap at 0.130, against DPO's 385 and 0.126.

Table 9: MultiPref, length declared, every arm at 0.5B. Columns as in Table 2; same / cross is held-out accuracy on same-group and cross-group pairs, whose difference is the gap; *mean* θ is the mean of the learned bias parameters (the single θ for pooled). Core arms are mean $\pm$ sd over three seeds; other rows are single-seed controls. These runs used an SFT reference trained before the one of Table 2, on the same data; the reference and DPO rows here are from the same runs, so the removal shares are computed within this table and differ from Table 2 by a few points.

Method	**Tokens**	**Removed**	**RM**	**same / cross**	**mean θ**
Reference (SFT)	294.6	—	-1.15	—	—
DPO	385.4 ± 3.5	—	-0.67	0.521 / 0.640	—
BA-DPO (pooled)	339.0 ± 1.1	51%	-0.69	0.516 / 0.526	0.79
BA-DPO (pooled, warm)	329.5	62%	-0.67	0.524 / 0.515	0.84
BA-DPO (θ_k only)	370.1 ± 1.9	17%	-0.67	0.520 / 0.602	0.42
BA-DPO (shuffled ids)	372.1	16%	-0.77	0.521 / 0.609	0.44
BA-DPO ($\bar{\theta} + \varepsilon_k$)	336.3 ± 4.1	54%	-0.68	0.520 / 0.522	0.84
BA-DPO ($\bar{\theta} + \varepsilon_k$, warm)	335.4	56%	-0.73	0.518 / 0.514	0.89
R-DPO ($\alpha = 0.02$)	137.8	over-corr.	-1.49	0.467 / 0.287	—
R-DPO ($\alpha = 0.005$)	309.7	84%	-0.90	0.517 / 0.469	—
SamPO	417.9	-34%	-0.72	0.515 / 0.617	—

Three observations follow. First, the free-θ_k arm removes 17% with its mean at 0.42, and shuffling the ids gives the same numbers (16%, 0.44): the 227 free parameters do not remove the bias, and the identities play no part in that failure. Second, warm-starting the mean at 0.99 gives the same result from the opposite direction: the pooled θ started at 0.99 settles at 0.84 against 0.77 to 0.81 from zero, and the shared parameter $\bar{\theta}$ at 0.78 against 0.67 to 0.72, and the resulting policies are indistinguishable. The policy never drives the parameter toward zero, which is what an absorption account, in which the network out-competes the scalar for the signal both can represent (Proposition 5), would predict. Third, R-DPO at $\alpha = 0.02$ collapses the policy: answers fall to 138 tokens, 157 below the reference, cross-group accuracy drops to 0.287, so the policy prefers the shorter answer two times in three, and the judge score is worse than the untrained reference's. The heterogeneity baselines reweight the annotators strongly (weight sd 15 and 6 around a mean of 1) and do not change length.

Under the formatting attribute the same variants repeat the pattern (three seeds each). The free-θ_k arm and the shuffled arm learn means of 0.31 and 0.32 and leave the held-out gap at 0.051 and 0.056, against 0.066 for DPO; the pooled arm ($\theta = 0.71$) and the shared-mean arm ($\bar{\theta} = 0.62$) close it to 0.004 and 0.000. The free mean is 0.50 of $\bar{\theta}$, against 0.61 under length. R-DPO at $\alpha = 0.02$, trained against the earlier reference (Table 9), raises the markdown rate to 0.700, above that build's DPO at 0.620, with same-group accuracy at 0.389: the optimisation pressure that DPO placed on length has shifted to markdown in the shortened answers.

F.3 READING THE SIGNATURE FROM PROBABILITIES, NOT SAMPLES

Every names table reads the attribute rate from the probabilities the policy assigns to the names in the pool (Appendix E.4), not from its sampled signatures. This subsection shows why, on Signed-UltraFeedback. Table 10 gives both readouts for the reference and four arms. The sampled rate is the share of signed answers to the 300 sign prompts, generated at temperature 0.7 and top-p 0.9, whose name is woman-coded or black-coded.

Table 10: Signed-UltraFeedback at 0.5B: the attribute rate read from sampled signatures (share of signed answers) and from probabilities (Table 1). Mean over three seeds.

Method	**Sampled, woman**	**Sampled, black**	p(woman)	p(black)
Reference (SFT)	0.61	0.79	0.498	0.517
DPO	1.00	1.00	0.964	0.991
BA-DPO (pooled)	0.74	0.91	0.549	0.608
BA-DPO ($\bar{\theta} + \varepsilon_k$)	0.69	0.91	0.542	0.595
BA-DPO (shuffled ids)	0.99	1.00	0.849	0.925

The two readouts disagree in three ways. First, the decoder distorts the rate before any preference training. The reference is balanced by probability (0.50 and 0.52), yet 79% of its signed answers carry a black-coded name. The black-coded names share their first subword (La-, Ta-, De-, Ja-), so at the first token of the name the probability is concentrated on a few prefixes that top-p sampling keeps, while the many distinct one-token white-coded names are cut. Second, sampled rates saturate. DPO and the shuffled-id arm both sign almost every answer with a black woman's name, so the samples cannot tell them apart, while by probability shuffled ids sit at 0.85 and 0.93 against DPO's 0.96 and 0.99. Third, a shift in the sampled rate is not evidence of annotator bias. Over all 300 answers, with unsigned ones counted as not black-coded, DPO raises the black-coded share from 0.70 to 0.85, by 0.15. Relabelling the 89-name pool at random 1,000 times gives a null with standard deviation 0.20 and a 95% range from -0.37 to $+0.35$ (one-sided $p = 0.22$, three DPO seeds). Three names, *Latisha*, *Latonya* and *Latoya*, carry 87% of DPO's signatures, and removing them from the black-coded set turns the shift into -0.40. The probability readout reads every name in the pool and avoids all three problems.

F.4 LoRA CALIBRATION AT 0.5B

Before the 8B runs, which use LoRA, we checked at 0.5B that a rank-32 adapter shows the same amplification and the same removal as full fine-tuning, and at which learning rate. The corpus was Signed-UltraFeedback with its balanced reference; the policy was the reference plus a LoRA adapter (rank 32, $\alpha = 64$, all linear layers, fp32 adapters on a bf16 base), trained for 1000 steps at seed 42 with DPO and with pooled BA-DPO, at learning rates 5×10^{-6} and 2×10^{-5}, against 5×10^{-7} for full fine-tuning.

Table 11: LoRA calibration on Signed-UltraFeedback at 0.5B, seed 42. Rates, removed, RM$^-$ and KL as in Table 1; θ is the learned pooled bias on the gender-coded attribute. Removed is against the DPO run of the same training setting.

Run	p(woman)	p(black)	**Removed (g / r)**	**RM$^-$**	**KL**	θ
Reference	0.498	0.517	—	-3.30	—	—
Full fine-tuning, DPO	0.964	0.992	—	-2.75	0.037	—
Full fine-tuning, BA-DPO (pooled)	0.545	0.616	90% / 79%	-2.53	0.025	0.68
LoRA 5×10^{-6}, DPO	0.952	0.988	—	-2.71	0.038	—
LoRA 5×10^{-6}, BA-DPO (pooled)	0.529	0.592	93% / 84%	-2.56	0.027	0.68
LoRA 2×10^{-5}, DPO	0.996	0.998	—	-2.46	0.032	—
LoRA 2×10^{-5}, BA-DPO (pooled)	0.637	0.734	72% / 55%	-2.39	0.025	0.67

At 5×10^{-6}, ten times the full fine-tuning rate, LoRA reproduces full fine-tuning on every readout: DPO amplifies to 0.95 and 0.99 against 0.96 and 0.99, pooled removes 93% and 84% against 90% and 79%, with the same learned θ (0.68) and nearly the same KL (0.027 against 0.025). A rank-32

adapter therefore suffices to show both the amplification and its removal, and this rate is used for the 8B runs. At 2×10^{-5} DPO amplifies more (1.00) and pooled removes clearly less (72% and 55%) although it learns nearly the same θ (0.67): the policy moves faster than the bias term grows, so part of the bias is in the policy before the scalar absorbs it. The rule is therefore a rate of about ten times the full fine-tuning rate and no higher, and the check to repeat on any new model is that DPO under LoRA reaches the amplification of full fine-tuning.

F.5 Robustness to the model family: Mistral-7B

The two models of Section 5 differ in scale and in family. A third model, `Mistral-7B-Instruct-v0.3` (Jiang et al., 2023), differs from both in family and in tokenizer, which is what the name result depends on, since the tokenizer decides how a signature is split. It is trained with the LoRA recipe of the 8B model unchanged (Appendix E), with its own SFT reference per corpus, on the reference, DPO and the two BA-DPO variants at three seeds, and without baselines. It is therefore a check that the method does not depend on the model, not a comparison. Table 12 reports both corpora.

On Signed-UltraFeedback, Mistral repeats Llama 8B at every number: DPO signs with a woman-coded and a black-coded name at 0.99, the two BA-DPO variants return to 0.50 and 0.54 to 0.55, removing 93 to 94% and 90 to 92% of DPO's shift, at two thirds of DPO's KL and a judge score within DPO's spread. The per-annotator parameters predict the held-out votes at the planted ceiling on every class, with the pooled scalar at chance in the class that disagrees (Table 13). On MultiPref, under DPO the policy produces answers 82 tokens longer than the reference and opens a gap of 0.11, as at 8B. BA-DPO removes about half of that increase, 48 to 52% against 46 to 50% at 8B, and closes the gap to 0.05 and 0.04, with learned scalars of 0.61 and 0.66 against 0.69 to 0.71 at 8B. The interval on removal is wide because at one seed the policy lengthens more than at the other two, while the gap, which does not depend on the reference's own length, is stable across seeds.

Table 12: `Mistral-7B-Instruct-v0.3` with LoRA. Top: Signed-UltraFeedback, columns as in Table 1. Bottom: MultiPref with length declared, columns as in Table 2. Mean and 95% interval over seeds 42, 123 and 456.

Method	p(woman)	**Removed**	p(black)	**Removed**	**RM**$^-$	**KL**
Reference (SFT)	0.466	—	0.503	—	0.13	—
DPO	0.988 ± 0.008	—	0.995 ± 0.004	—	1.12 ± 0.17	0.031 ± 0.004
BA-DPO (pooled)	0.505 ± 0.009	$93 \pm 2\%$	0.552 ± 0.002	$90 \pm 0\%$	1.06 ± 0.16	0.022 ± 0.003
BA-DPO ($\bar{\theta} + \varepsilon_k$)	0.499 ± 0.010	$94 \pm 2\%$	0.542 ± 0.004	$92 \pm 1\%$	1.19 ± 0.26	0.023 ± 0.002

Method	**Tokens**	**Removed**	**Gap**	**RM**	**KL**
Reference (SFT)	312.7	—	—	3.52	—
DPO	394.8 ± 6.5	—	0.107 ± 0.029	5.06 ± 0.19	0.030 ± 0.003
BA-DPO (pooled)	355.3 ± 22.5	$48 \pm 29\%$	0.049 ± 0.012	4.70 ± 0.11	0.028 ± 0.002
BA-DPO ($\bar{\theta} + \varepsilon_k$)	351.9 ± 14.2	$52 \pm 19\%$	0.041 ± 0.041	4.79 ± 0.48	0.029 ± 0.002

G Vote prediction with the learned bias parameters

Section 5.2 states that the per-annotator parameters describe the annotators. This is the measurement behind that claim. On the held-out judgments whose two responses differ on the attribute, each arm predicts the voter's label from its implicit reward margin plus that voter's learned bias, as defined in Appendix E.4; the ceiling uses the planted θ_k. The ceiling lies below 1 because the votes are drawn from the likelihood rather than being deterministic.

Table 13 reads the race-coded attribute by the voter's class, since that is the attribute the classes disagree on, and both attributes overall. Two things stand out. The $\bar{\theta} + \varepsilon_k$ variant sits at the ceiling on every class and every model without being given the class labels: sixty annotators with about four hundred judgments each are enough to locate every one of them. The pooled scalar cannot do this. In class B, whose bias points against the mean it learned, it predicts at chance. The $\bar{\theta} + \varepsilon_k$ parameters also recover the planted values, correlating with them at $r = 0.95$ on the gender-coded

Table 13: Signed-UltraFeedback: share of the held-out judgments whose two responses differ on the attribute that are predicted correctly, by the voter's class for the race-coded attribute and overall for both. The ceiling is the same prediction with the planted θ_k, averaged over the arms' seeds. Means over three seeds. Classes A, B and C as in Appendix D.2; class B leans against the race-coded attribute.

Method	**Race, A**	**Race, B**	**Race, C**	**Race, all**	**Gender, all**
Qwen2.5-0.5B-Instruct, full fine-tuning					
Ceiling (planted θ_k)	0.860	0.694	0.703	0.752	0.749
DPO	0.774	0.497	0.650	0.641	0.673
BA-DPO (pooled)	0.765	0.500	0.678	0.649	0.694
BA-DPO ($\bar{\theta} + \varepsilon_k$)	0.857	0.690	0.707	0.752	0.745
Group-DRO DPO	0.715	0.525	0.632	0.625	0.666
EM-DPO + MinMax-DPO	0.788	0.544	0.660	0.665	0.677
Llama-3.1-8B-Instruct, LoRA					
Ceiling (planted θ_k)	0.863	0.700	0.708	0.757	0.756
DPO	0.779	0.503	0.681	0.655	0.700
BA-DPO (pooled)	0.785	0.488	0.687	0.655	0.703
BA-DPO ($\bar{\theta} + \varepsilon_k$)	0.863	0.699	0.710	0.758	0.753
Group-DRO DPO	0.734	0.505	0.652	0.631	0.682
Mistral-7B-Instruct-v0.3, LoRA					
Ceiling (planted θ_k)	0.853	0.712	0.717	0.761	0.753
DPO	0.745	0.530	0.676	0.652	0.679
BA-DPO (pooled)	0.749	0.513	0.676	0.647	0.685
BA-DPO ($\bar{\theta} + \varepsilon_k$)	0.835	0.701	0.712	0.750	0.744

attribute and 0.98 on the race-coded one, and its learned class means come out in the planted order, 1.90, -0.22 and 0.83 against the planted 2.62, -0.34 and 1.11.