Correct, Route, Calibrate: Efficient Preference Optimization from Noisy, Heterogeneous Human Feedback

Zhongming Xie
Division of Biostatistics
University of California, Berkeley
zhongming_xie@berkeley.edu

Xinwei Ma
Department of Economics
University of California, San Diego
x1ma@ucsd.edu

Jingshen Wang
Division of Biostatistics
University of California, Berkeley
jingshenwang@berkeley.edu

Abstract

Human labels are important supervisory signals for preference-based reinforcement learning, but their collection is often constrained by practical considerations that standard training objectives tend to ignore. For instance, Direct Preference Optimization (DPO) usually treats observed preference labels as clean, even though real annotation pipelines rely on heterogeneous annotators whose reliability varies across individuals and with the difficulty of each comparison. This setting introduces several challenges: (1) Noisy labels introduce systematic bias into the preference optimization objective and need to be rigorously corrected. (2) The allocation of pairwise comparisons to different annotators is a design problem that directly affects the statistical efficiency of the training pipeline. (3) Non-random routing creates selection bias in annotator evaluation, complicating downstream decisions such as compensation. We address these challenges within a unified framework for DPO that combines annotator-aware posterior correction with information-guided annotator routing. Posterior correction leverages an instance-dependent noise model to infer latent ground-truth preferences from noisy labels, mitigating bias in the DPO objective. Annotator routing is formulated as an experimental design problem that allocates comparisons to annotators according to their statistical information under a fixed annotation budget. To support reliable annotator evaluation under non-random routing, we introduce a doubly robust estimator that corrects for selection bias. Semi-synthetic experiments on the UltraFeedback dataset show that our approach improves ground-truth preference recovery and downstream AlpacaEval 2 performance over baselines, with consistent gains across multiple open-source LLM architectures. Annotator reliability-estimation experiments further show that the doubly robust estimator substantially reduces mean squared error relative to other baselines, demonstrating its value for annotator evaluation.

1 Introduction

1.1 Motivation and Challenge

Reinforcement learning from human feedback (RLHF) is commonly implemented through preference-based optimization, where a model is trained to align with pairwise human judgments between

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

candidate responses (Christiano et al., 2017). Direct Preference Optimization (DPO) (Rafailov et al., 2023) bypasses explicit reward modeling and directly optimizes the policy against observed preferences, making annotation quality the central signal of alignment performance. In practice, human feedback is rarely clean or homogeneous. Annotators introduce errors, particularly on complex or ambiguous examples, and modern preference optimization pipelines rely on diverse annotator pools whose expertise and reliability differ across both individuals and instances. This combination of noise and heterogeneity brings several challenges. First, when annotators are imperfect, the standard DPO training objective is mis-specified and the learned policy converges to a biased proxy of the latent preference (Chowdhury et al., 2024), regardless of dataset size. Second, since annotators differ in reliability, cost, and capacity, the practitioner must carefully design how to allocate comparisons across annotators under a fixed budget (Yan et al., 2012). Assigning reliable annotators to low-value comparisons wastes scarce resources, while routing difficult examples to unreliable annotators introduces disproportionate noise.

Both noise correction and annotator routing depend on the same unknown object: how reliably each annotator labels each instance. Estimating this instance-dependent reliability requires some access to gold labels, but gold labels are expensive and can only be collected for a small subset of comparisons. Gold-label allocation is therefore a second design problem on top of routing: we must decide not only how to route comparisons to annotators, but also which comparisons should receive gold supervision. Some examples are far more informative for calibration than others. And gold labels should be targeted to comparisons that most improve reliability estimation and consequently downstream learning.

Finally, heterogeneous annotation creates an additional operational goal that is often ignored in standard DPO pipelines: a faithful estimate of each annotator's marginal error rate ρ_k. This quantity provides a system-level evaluation of annotator reliability and may directly inform downstream decisions such as compensation, payment tiers, as well as the retention and removal of unreliable annotators. Estimating ρ_k is complicated by the very design choices that improve training: comparisons are routed non-uniformly across annotators, and only a selectively chosen subset receives gold validation. As a result, naive averages of observed disagreements inherit the selection bias of the routing and validation design. Annotator reliability estimation must therefore be *design-aware*, correcting for this selection bias rather than treating the collected labels as a random sample.

1.2 Solution and Contribution

We develop a unified framework for preference-based policy optimization under noisy and heterogeneous feedback, with four components: (i) **Annotator-aware noise correction** replaces each observed label with a posterior estimate of the latent clean preference, removing the bias that standard DPO inherits from noisy labels. (ii) **Information-guided routing** formulates comparison-annotator assignment as an experimental design problem under annotator capacity constraints, sending each comparison to the annotator expected to provide the most informative signal for DPO training. (iii) **Post-label gold calibration** allocates a limited gold-label budget via a Neyman-style rule, targeting comparisons whose outcomes most reduce uncertainty in annotator error rates. (iv) **Design-aware reliability estimation** uses an augmented inverse-propensity weighted (AIPW) estimator to recover annotator-level error rates under non-random routing and gold calibration, with consistency, double robustness, and $\sqrt{n}$-asymptotic normality guarantees.

Our contributions are summarized as follows.

1. We propose a unified DPO framework that is able to correct labeling noise in training, find the optimal annotator routing policy that maximizes information utility, together with a design-aware AIPW estimator for reliable annotator reliability evaluation.

2. We show that our posterior-corrected objective recovers the clean DPO risk and corrects the bias caused by standard DPO with noisy routed labels. We also establish consistency, double robustness, and $\sqrt{n}$-asymptotic normality for the proposed AIPW estimator of annotator marginal error rates.

3. In semi-synthetic experiments based on *UltraFeedback*, we show that instance-dependent correction and information-guided routing improve clean preference recovery and downstream generation performance over standard DPO with noisy routed labels. Additionally, the design-aware AIPW estimator more accurately recovers annotator reliability compared to other baselines.

We discuss additional related work and compare our framework with prior literature in Appendix A.

2 Preliminaries and Problem Setup

In this section, we formalize preference optimization under noisy and heterogeneous human feedback. Section 2.1 reviews DPO and Section 2.2 introduces heterogeneous annotators and the routing policy.

2.1 Pairwise preference learning and DPO

Each preference comparison consists of a tuple $Z = (X, Y_1, Y_2)$, where $X \sim p_{\mathcal{X}}$ is a prompt and Y_1, Y_2 are candidate responses generated by a reference policy or drawn from an offline preference dataset. We write $Z \sim p_Z$ for the induced comparison distribution. Each comparison is associated with a latent clean preference $L^* \in \{-1, +1\}$ where $L^* = +1$ indicates that Y_1 is preferred to Y_2, and $L^* = -1$ indicates the reverse. We allow $\eta(z) = \Pr(L^* = +1 \mid Z = z), \in [0, 1]$, so preferences may be stochastic conditional on the comparison.

For a parameterized policy π_θ, we define the log-probability difference $\Delta\ell_\theta(Z) = \log \pi_\theta(Y_1 \mid X) - \log \pi_\theta(Y_2 \mid X)$, with $\Delta\ell_{\mathrm{ref}}(Z)$ defined analogously for a fixed reference policy π_{ref}. The DPO score difference is $u_\theta(Z) = \beta \{\Delta\ell_\theta(Z) - \Delta\ell_{\mathrm{ref}}(Z)\}$, where $\beta > 0$ is the DPO temperature parameter inherited from the KL-regularized RLHF objective. The sign and magnitude of $u_\theta(Z)$ indicate the target policy's implicit preference between the two responses.

If the clean preference L^* were directly observed, the population-level DPO objective would be

$$J^*(\theta) = -R^*(\theta) = \mathbb{E}\left[\log \sigma\left(L^* u_\theta(Z)\right)\right],$$

where $\sigma(\cdot)$ is the sigmoid function. The target policy maximizes this clean preference objective to learn the optimal policy.

2.2 Heterogeneous annotators and routing

In practice, L^* is not directly observed. Instead, each comparison is routed to one of K heterogeneous annotators by a routing policy $q(k \mid z) = \mathbb{P}(A = k \mid Z = z)$ with $\sum_{k=1}^{K} q(k \mid z) = 1$. Here $A \in \{1, \ldots, K\}$ denotes the assigned annotator. If Z is routed to annotator k, the observed label is $\widetilde{L}(Z) = L_k(Z)$ such that $L_k(Z) \in \{-1, +1\}$. We model annotator-specific symmetric noise: annotator k flips the clean preference with probability $\varepsilon_k(z)$ regardless of its sign, so $\mathbb{P}(L_k \neq L^* \mid Z = z) = \varepsilon_k(z)$. To capture instance-dependent difficulty along with varying annotator skill, we consider the following parametric noise model:

$$\varepsilon_k(z) = \sigma\big(w_k \delta_\psi(z) - a_k\big), \qquad \delta_\psi(z) = h(z)^\top \psi.$$

Here $h(z)$ is a latent feature map and $\delta_\psi(z)$ models the intrinsic difficulty of the comparison; a_k is an annotator-specific skill parameter, and w_k controls how strongly difficulty affects annotator k. This formulation captures both heterogeneous annotator skill and instance-dependent difficulty, with annotator k's reliability on comparison z jointly determined by a_k, w_k, and $\delta_\psi(z)$.

3 Method

The above setup introduces several coupled challenges. In this section, we address these challenges within a unified framework consisting of four interacting parts. Section 3.1 introduces an annotator-aware posterior correction that targets the clean preference objective despite noisy routed labels. Section 3.2 formulates comparison-annotator assignment as an experimental design problem that maximizes information for training. Section 3.3 allocates a limited gold-label budget to refine $\varepsilon_k(z)$, which feeds back into both correction and routing. Section 3.4 develops a design-aware estimator of annotator-level reliability under the non-random routing and validation scheme.

3.1 Annotator-aware posterior correction

Standard DPO training pipeline simply treats the observed noisy label $\widetilde{L}$ as the clean preference. While convenient, this implicitly assumes that $\widetilde{L} = L^*$, which is violated when annotators are

imperfect. Consequently, the learner optimizes a corrupted objective, and systematic biases push the model toward incorrect preferences. To target the clean objective directly, we introduce an annotator-aware posterior correction step. For annotator k, we define the posterior probability that the first response is truly preferred, conditioned on the comparison, annotator choice, and observed noisy label: $\gamma_k(z, \widetilde{l}) = \mathbb{P}(L^* = +1 \mid Z = z, A = k, \widetilde{L} = \widetilde{l})$, Under our noise model,

$$\gamma_k(z, 1) = \frac{(1 - \varepsilon_k(z))\,\eta(z)}{(1 - \varepsilon_k(z))\,\eta(z) + \varepsilon_k(z)\,(1 - \eta(z))},$$

$$\gamma_k(z, -1) = \frac{\varepsilon_k(z)\,\eta(z)}{\varepsilon_k(z)\,\eta(z) + (1 - \varepsilon_k(z))\,(1 - \eta(z))}.$$

We then define the posterior-corrected loss $\ell_k(\theta; z, \widetilde{l}) = \mathbb{E}\left[\log \sigma\left(L^* u_\theta(Z)\right) \mid \widetilde{L} = \widetilde{l}, Z = z\right]$ and using these probabilities

$$\ell_k(\theta; z, \widetilde{l}) = \gamma_k(z, \widetilde{l})\, \log \sigma\big(u_\theta(z)\big) + \big(1 - \gamma_k(z, \widetilde{l})\big)\, \log \sigma\big(-u_\theta(z)\big).$$

This loss averages the two clean-label losses according to the posterior probability over the latent preference, rather than treating the noisy label as certainly correct. Under correct specification, the posterior-corrected objective matches the clean preference objective in expectation (see Section 4).

Differentiating the corrected loss gives $\nabla_\theta \ell_k^{\mathrm{post}}(\theta; Z, \widetilde{L}) = \beta\Big(\sigma(u_\theta(Z)) - \gamma_k(Z, \widetilde{L})\Big)\, \Delta g_\theta(Z)$, where $\Delta g_\theta(Z) := \nabla_\theta \Delta \ell_\theta(Z)$.

3.2 Information-guided comparison-annotator routing

While posterior correction removes the asymptotic bias induced by noisy routed labels, it does so at the cost of higher variance (lower information), making annotator routing critical to finite-sample performance. The statistical value of an observation depends jointly on annotation reliability and on how informative the comparison is for updating the current model. Random routing ignores this heterogeneity and can spend the annotation budget on low-information assignments. We therefore formulate annotator routing as a statistical design problem: choosing comparison-annotator assignments to maximize information utility under annotation-budget constraints.

Let $\mathcal{Q}(C)$ denote the feasible set of routing policies under constraints C, such as annotator capacity. A routing policy $q(\cdot \mid z)$ assigns a probability distribution over annotators to each comparison. To optimize such a policy, we evaluate each assignment (z, k) by the informativeness of the resulting observation for estimating θ. Fisher information provides a principled measure of assignment informativeness: it quantifies the local sensitivity of the observed-label distribution to the model parameter and characterizes the amount of information available for estimating θ.

For a comparison-annotator pair (z, k), the induced Fisher information is $I(z, k; \theta) = w_\theta(z, k)\, \Delta g_\theta(z) \Delta g_\theta(z)^\top$, where

$$w_\theta(z, k) = \frac{(1 - 2\varepsilon_k(z))^2\, \beta^4\, \sigma_\theta(z)^2 \big(1 - \sigma_\theta(z)\big)^2}{p_\theta(z, k)\big(1 - p_\theta(z, k)\big)}, \qquad \sigma_\theta(z) := \sigma\big(u_\theta(z)\big).$$

Here, $p_\theta(z, k) = \mathbb{P}(\widetilde{L} = 1 \mid Z = z, A = k) = \varepsilon_k(z) + (1 - 2\varepsilon_k(z))\, \sigma\big(u_\theta(z)\big)$ denotes the Bernoulli parameter of the observed label, which combines the model's current belief $\sigma(u_\theta(z))$ with the label noise $\varepsilon_k(z)$. The scalar weight $w_\theta(z, k)$ summarizes the informativeness of the assignment: it is large when annotator k is reliable on z and when the comparison is informative under the current model.

A natural population routing objective is then

$$q^* \in \arg\max_{q \in \mathcal{Q}(C)} \mathbb{E}_Z \left[\sum_{k=1}^{K} q(k \mid Z)\, w_\theta(Z, k)\, \|\Delta g_\theta(Z)\|_2^2 \right]. \tag{1}$$

In practice, we work with a finite pool of candidate unlabeled comparisons $\{z^{(i)}\}_{i=1}^N$. We can formulate the routing objective as an Integer Linear Program (ILP) using binary assignment variables

$r_{ik} \in \{0,1\}$, where $r_{ik} = 1$ indicates that comparison i is sent to annotator k. Subject to the annotator capacity constraint B_k, the optimization solves:

$$\begin{aligned} \max_{r_{ik}\in\{0,1\}} \quad & \sum_{i=1}^{N}\sum_{k=1}^{K} r_{ik}\, w_\theta(z^{(i)},k)\, \|\Delta g_\theta(z^{(i)})\|_2^2 \\ \text{s.t.} \quad & \sum_{k=1}^{K} r_{ik} \le 1, \qquad i = 1,\dots,N, \\ & \sum_{i=1}^{N} r_{ik} \le B_k, \qquad k = 1,\dots,K. \end{aligned}$$

By dynamically controlling who labels which comparison, we actively shape the information matrix of the collected data, increasing the training efficiency.

3.3 Post-label gold calibration design

Posterior correction and information-guided routing both depend on accurate estimation of $\varepsilon_k(z)$. To iteratively refine these estimates, we selectively request a small number of "gold" labels after observing the routed noisy feedback. Specifically, after selecting annotator $A = k$ and observing the noisy label $\widetilde{L} = L_k$, we make a binary validation decision $G \in \{0,1\}$, with $G = 1$ meaning that a gold label is additionally acquired for the same comparison. The validation decision follows

$$\mathbb{P}(G = 1 \mid Z = z, A = k, \widetilde{L} = \widetilde{l}) = \pi(z, k, \widetilde{l}).$$

When $G = 1$, we observe a clean label L^* and have $Y^{\text{val}} = \mathbf{1}\{\widetilde{L} \neq L^*\}$, which directly reveals whether annotator k made an error on z.

Because gold labels are typically expensive, the expected budget B_g is strictly limited. This introduces a secondary design problem: how should the gold-calibration budget be allocated to most effectively improve the estimation of annotator-specific error rates? Let $\Pi(B_g)$ denote the class of feasible allocation rules under expected budget B_g. At the population level, the calibration rule reduces to the Neyman-style allocation:

$$\pi^*(z, k, \widetilde{l}) \propto \sqrt{\varepsilon_k(z)\big(1 - \varepsilon_k(z)\big)},$$

Here, $\varepsilon_k(z)(1 - \varepsilon_k(z))$ is the inherent variance of the Bernoulli noise model. In words, the system spends its limited gold labels where the annotator's current labeling behavior is highly uncertain.

3.4 Design-aware estimation of annotator reliability

A downstream quantity of interest is the marginal error rate of annotator k. This quantity provides a system-level evaluation of annotator reliability and is useful for downstream decisions such as compensation allocation. For a comparison Z, let $L^*(Z)$ denote the gold label and $L_k(Z)$ the potential label that annotator k would assign to Z. Let $Y_k(Z) = \mathbf{1}\{L_k(Z) \neq L^*(Z)\}$ and $\varepsilon_k(z) = \mathbb{E}[Y_k(Z) \mid Z = z]$, the marginal error rate is

$$\rho_k = \mathbb{E}[Y_k(Z)] = \mathbb{E}[\varepsilon_k(Z)].$$

Estimation of ρ_k is complicated by the fact that $Y_k(Z)$ is observed only on a design-selected subset of examples: comparisons are routed to different annotators and only a subset of routed examples is gold-validated. Reliability estimation must therefore account for the selection bias induced by the data-collection design.

To address this, we propose an augmented inverse-propensity weighted (AIPW) estimator that accounts for the selection bias. For each comparison $i \in \{1,\dots,n\}$ we observe $O_i = \big(Z_i,\ A_i,\ \widetilde{L}_i,\ G_i,\ G_i\, L_i^*\big)$, where A_i is the annotator to whom comparison i is routed, $\widetilde{L}_i := L_{A_i}(Z_i)$ is the observed noisy label, $G_i \in \{0,1\}$ is the gold-validation indicator, and L_i^* is revealed only when $G_i = 1$. Define the observed error indicator $Y_i^{\text{obs}} = \mathbf{1}\{\widetilde{L}_i \neq L_i^*\}$. Routing and validation follow the known distributions

$$q(k \mid z) := \mathbb{P}(A = k \mid Z = z), \qquad \pi(z, k, \ell) := \mathbb{P}(G = 1 \mid Z = z,\ A = k,\ \widetilde{L} = \ell).$$

The AIPW estimator takes the form

$$\widehat{\rho}_k^{\mathrm{AIPW}} \;=\; \frac{1}{n}\sum_{i=1}^{n}\left[\widehat{m}_k(Z_i) \;+\; \frac{\mathbf{1}\{A_i=k\}\,G_i}{q(k\mid Z_i)\,\pi(Z_i,k,\widetilde{L}_i)}\big(Y_i^{\mathrm{obs}}-\widehat{m}_k(Z_i)\big)\right],$$

where $\widehat{m}_k$ is a working model for $\varepsilon_k(z)$. The inverse-propensity factor corrects the selection bias from routing and selective validation, while the augmentation $\widehat{m}_k(Z_i)$ reduces variance when the working model predicts well. We provide formal consistency and asymptotic-variance results for this estimator in Section 4.

4 Theoretical Investigation

We present the theoretical investigations of the proposed framework along two axes. Section 4.1 analyzes clean preference signal recovery: standard DPO on noisy labels suffers a non-vanishing bias floor. Posterior-corrected DPO targets the clean risk, eliminating the bias floor at the cost of a finite-sample variance that depends explicitly on the routing. Section 4.2 analyzes design-aware estimation of annotator reliability: under non-random routing and selective gold validation, the AIPW estimator $\widehat{\rho}_k^{\mathrm{AIPW}}$ is consistent and $\sqrt{n}$-asymptotically normal.

4.1 Clean Preference Signal Recovery

Define the clean logistic risk $R^*(\theta) := \mathbb{E}[-\log\sigma(L^* u_\theta(Z))]$, let $\theta^* \in \arg\min_{\theta\in\Theta} R^*(\theta)$, and write the logistic regret as

$$\mathrm{Reg}(\hat{\theta}) := R^*(\hat{\theta}) - R^*(\theta^*).$$

We compare estimators via the clean pairwise win rate $\mathbb{P}(L^*(Z) = \hat{L}(u_\theta(Z)))$. Since clean win-rate regret is a monotone transformation of the logistic regret (See Appendix), it suffices to analyze the logistic regret.

Standard DPO ignores label noise and minimizes the empirical logistic loss on noisy labels $\widehat{R}_q(\theta) := \frac{1}{n}\sum_{i=1}^{n} -\log\sigma(\widetilde{L}_i\, u_\theta(Z_i))$ with population counterpart $R_q(\theta) := \mathbb{E}[\widehat{R}_q(\theta)]$, and minimizers $\hat{\theta}_q$ and $\hat{\theta}_q$ respectively.

Assumption 1 (Bounded linearized class)**.** *$u_\theta(Z) = \beta\,\langle\theta, \Delta\phi(Z)\rangle$ with $\|\theta\|_2 \le B$ and $\|\Delta\phi(Z)\|_2 \le R_\Delta$ almost surely, where $\Delta\phi(Z) := \phi(x, y_1) - \phi(x, y_2)$ and $\beta > 0$ is the DPO temperature.*

Theorem 1. *Under Assumption 1,*

$$\mathbb{E}[\mathrm{Reg}^*(\hat{\theta}_q)] \;\le\; \underbrace{R^*(\theta_q) - R^*(\theta^*)}_{\mathrm{Bias}(q)} + C\,\bar{\varepsilon}(q) \;+\; O\!\left(\frac{\beta B R_\Delta}{\sqrt{n}}\right),$$

where $\bar{\varepsilon}(q) := \mathbb{E}_{Z,A\sim q(\cdot|Z)}[\varepsilon_A(Z)]$ is the routing-averaged noise rate. If $\mathrm{Bias}(q) \le M\,\bar{\varepsilon}(q)$ for some $M > 0$, the right-hand side becomes $M'\,\bar{\varepsilon}(q) + O(\beta B R_\Delta/\sqrt{n})$.

The last term in Theorem 1 vanishes as we collect more data, but the other two terms generally do not. Standard training can thus suffer a non-vanishing bias floor even with unlimited data.

Posterior-corrected DPO. For each annotator $k \in \{1, \ldots, K\}$, let $\ell_k^{\mathrm{post}}(\theta; Z, \ell)$ be a posterior-corrected loss, with empirical risk $\widehat{R}_q^{\mathrm{post}}(\theta) := \frac{1}{n}\sum_i \ell_{A_i}^{\mathrm{post}}(\theta; Z_i, \widetilde{L}_i)$, population risk $R_q^{\mathrm{post}}(\theta) := \mathbb{E}[\ell_A^{\mathrm{post}}(\theta; Z, \widetilde{L})]$, and minimizer $\hat{\theta}_q^{\mathrm{post}}$.

Proposition 1 (Unbiasedness of posterior correction)**.** *If the posterior model is correctly specified, then for every θ, $\mathbb{E}[\ell_A^{\mathrm{post}}(\theta; Z, \widetilde{L}) \mid Z] = \mathbb{E}[\log\sigma(L^* u_\theta(Z)) \mid Z]$ and hence $R_q^{\mathrm{post}}(\theta) = R^*(\theta)$.*

Proposition 1 shows that posterior correction targets the clean preference objective despite routed noisy labels, provided the posterior model is correctly specified. Define the local routing-averaged gradient variance

$$V_{\mathrm{post}}^{\mathrm{loc}}(q) \;:=\; \sup_{\theta\in\mathcal{N}(\Theta^*)} \mathbb{E}_Z\left[\sum_{k=1}^{K} q(k\mid Z)\;\mathrm{Var}\big(\nabla_\theta \ell_k^{\mathrm{post}}(\theta; Z, L_k)\,\big|\,Z\big)\right].$$

The price of this bias removal is the finite-sample performance that depends explicitly on the routing. For posterior-corrected DPO training, the relevant goal is to reduce corrected-gradient variance with routing:

$$q^{\mathrm{post,opt}} \in \arg\min_{q \in \mathcal{Q}(C)} V_{\mathrm{post}}^{\mathrm{loc}}(q).$$

Theorem 2. *Under Assumptions 1 and 3 in Appendix, $R_q^{\mathrm{post}}(\theta) = R^*(\theta)$ for all $\theta \in \Theta$, and*

$$\mathbb{E}[\mathrm{Reg}_{\log}^*(\hat{\theta}_q^{\mathrm{post}})] \;\le\; C_s \sqrt{V_{\mathrm{post}}^{\mathrm{loc}}(q)/n}.$$

Combining Theorems 1 and 2 with Proposition 4 in Appendix yields guarantees for clean win-rate regret: standard training inherits a routing-dependent bias floor, while posterior-corrected training is asymptotically consistent but suffers higher finite-sample variance under poor routing.

4.2 Reliability estimation under routing

We now study the theoretical properties of the estimator $\widehat{\rho}_k^{\mathrm{AIPW}}$ from Section 3.4. For a fixed annotator k, let $D_k = \mathbf{1}\{A = k,\, G = 1\}$ be the joint routing-and-validation indicator, with conditional probability $r_k(z, \ell) := \mathbb{P}(D_k = 1 \mid Z = z,\, L_k(Z) = \ell) = q(k \mid z)\, \pi(z, k, \ell)$.

Assumption 2 (Positivity and ignorability)**.** *There exists $r_{\min} > 0$ such that $r_k(Z, \widetilde{L}) \ge r_{\min}$ almost surely, and $Y_k(Z) \perp\!\!\!\perp D_k \mid (Z, A, \widetilde{L})$.*

Proposition 2 (Consistency and double robustness)**.** *Under Assumption 2, if r_k is known (i.e. q and π are correctly specified), then $\widehat{\rho}_k^{\mathrm{AIPW}}$ is consistent for ρ_k regardless of the working model $\widehat{m}_k$. Moreover, under the reduced-form condition*

$$\mathbb{E}[Y_k(Z) \mid Z, \widetilde{L}] \;=\; \mathbb{E}[Y_k(Z) \mid Z] \;=\; \varepsilon_k(Z), \tag{2}$$

$\widehat{\rho}_k^{\mathrm{AIPW}}$ is also consistent when $\widehat{m}_k$ is correctly specified, even if r_k is misspecified.

The first part is guaranteed in our setting, since routing and gold validation are known by design.

To characterize efficiency, we abbreviate $Y := Y_k(Z)$, $D := D_k$, $r := r_k(Z, \widetilde{L})$, $m := m_k(Z)$, and $\varepsilon := \varepsilon_k(Z)$, and define the augmented score $\phi_m := m + \frac{D}{r}(Y - m)$.

Proposition 3 (Variance decomposition)**.** *Under Assumption 2, the reduced-form condition* (2)*, and with $Y \mid Z \sim \mathrm{Bern}(\varepsilon_k(Z))$,*

$$\mathrm{Var}(\phi_m) \;=\; \mathrm{Var}\big(\varepsilon_k(Z)\big) \;+\; \mathbb{E}\left[\frac{\varepsilon_k(Z)\big(1 - \varepsilon_k(Z)\big)}{r(Z, \widetilde{L})}\right] \;+\; \mathbb{E}\left[\frac{1 - r(Z, \widetilde{L})}{r(Z, \widetilde{L})}\big(\varepsilon_k(Z) - m(Z)\big)^2\right] \tag{3}$$

Proposition 3 isolates the term that depends on the working model,

$$\mathbb{E}\left[\frac{1 - r(Z, \widetilde{L})}{r(Z, \widetilde{L})}\big(\varepsilon_k(Z) - m(Z)\big)^2\right].$$

This makes the role of heterogeneity explicit: when annotator reliability varies little across instances, constant and instance-dependent models have similar efficiency; when it varies substantially, a richer working model can yield large variance reductions, especially in regions where validation is rare and $(1 - r)/r$ is large.

Because q and π are known by design, r_k is itself *known*, which yields $\sqrt{n}$-normality under mild conditions on the working model alone.

Theorem 3 (Asymptotic normality of $\widehat{\rho}_k^{\mathrm{AIPW}}$)**.** *Suppose Assumption 2 holds, that $\widehat{m}_k$ is fit on an independent fold (or via cross-fitting) with $\mathbb{E}[(\widehat{m}_k(Z) - m_k^\star(Z))^2] = o_P(1)$ for some $m_k^\star$ and $\mathbb{E}[\phi_{m_k^\star}^2] < \infty$. Then*

$$\sqrt{n}\,\big(\widehat{\rho}_k^{\mathrm{AIPW}} - \rho_k\big) \;\xrightarrow{d}\; \mathcal{N}(0, V_k), \qquad V_k \;=\; \mathrm{Var}(\phi_{m_k^\star}),$$

where V_k is given by (3) *at $m = m_k^\star$, and is consistently estimated by the empirical variance of the plug-in summands $\widehat{\phi}_i = \widehat{m}_k(Z_i) + \frac{\mathbf{1}\{A_i = k\}\, G_i}{q(k \mid Z_i)\, \pi(Z_i, k, \widetilde{L}_i)}\big(Y_i^{\mathrm{obs}} - \widehat{m}_k(Z_i)\big)$.*

Table 1: Comparison for LLAMA3-8B with $n_{train} = 10000$.

Noise model	Routing	Eval Accuracy ↑	Win Rate ↑	LC Win Rate ↑
None	None	0.7200	34.42	38.02
Constant ε_k	Random	0.6960	34.49	35.81
Constant ε_k	Integer	0.6940	34.01	35.48
Estimated $\varepsilon_k(z)$	Random	0.7630	35.22	38.69
Estimated $\varepsilon_k(z)$	Integer	0.7600	**36.92**	**40.14**

Table 2: Comparison for Mistral-7B with $n_{train} = 10000$.

Noise model	Routing	Eval Accuracy ↑	Win Rate ↑	LC Win Rate ↑
None	None	0.5690	13.15	18.90
Constant ε_k	Random	0.5650	13.86	20.14
Constant ε_k	Integer	0.5510	13.94	19.65
Estimated $\varepsilon_k(z)$	Random	0.5910	14.63	20.72
Estimated $\varepsilon_k(z)$	Integer	0.6070	**15.09**	**21.41**

5 Experiments

5.1 Semi-Synthetic Preference Learning with Instance-Dependent Annotators

In this section, we evaluate whether our method improves preference learning under noisy and heterogeneous feedback. We consider a semi-synthetic setup based on the *UltraFeedback* dataset: this setting allows us to compare posterior label correction, information-guided routing, and the choice of noise models under controlled conditions.

Semi-Synthetic Dataset DGP. We simulate $K = 2$ different annotators. Annotator k flips the clean label with probability

$$\varepsilon_k(z) = \text{clip}\left(\sigma(w_k d(z) - a_k),\, 10^{-3},\, 1 - 10^{-3}\right), \qquad k \in \{0, 1\}.$$

Here $d(z)$ is the instance-level difficulty and formally defined in Appendix B. The annotator-specific parameters are $(a_0, a_1) = (2.0, 1.1)$ and $(w_0, w_1) = (4.0, 1.5)$. Annotator 0 is more accurate on easy comparisons but degrades sharply as difficulty increases, whereas annotator 1 is less sensitive to difficulty.

Estimating $\varepsilon_k(z)$. Both routing and posterior correction depend on a learned noise model $\widehat{\varepsilon}_k(z)$. We estimate it using `multilingual-E5-large-instruct` text embedder (Wang et al., 2024) and a two-layer MLP on on a held-out gold-labeled slice. Details are provided in Appendix B.

Methods. We compare methods along three axes: posterior correction, the routing rule used to assign annotators, and the estimation of $\varepsilon_k(z)$. *Naive DPO* trains directly on noisy labels and ignores annotator noise. *Constant* $\hat{\varepsilon}_k$ estimates a single instance-independent error rate per annotator, while $\hat{\varepsilon}_k(z)$ fits an instance-dependent noise model. For routing, *Random* routing assigns annotators randomly, while *Integer* routing assigns comparisons to different annotators by solving the integer optimization problem in Section 3.2. For routing, we let the annotator capacity ratio to be 1:1.

Evaluation and Results. We report internal preference-recovery metrics and external generation metrics. Internally, we measure the clean preference signal recovery on a held-out dataset with $n_{eval} = 1000$ (**Eval Accuracy**). Externally, we evaluate the downstream text generation abilities of the learned policies on AlpacaEval 2 (Dubois et al., 2024) and report both raw **Win Rate** and **Length-controlled (LC) Win Rate** over GPT-4 baselines.

Tables 1-2 summarize results for LLAMA3-8B and Mistral-7B. We highlight three findings.

Result 1: posterior correction improves over noise-unaware training. Using the estimated instance-dependent noise model together with posterior correction consistently improves over the standard noise-unaware baseline. For LLAMA3-8B, ESTIMATED $\varepsilon_k(z)$ + RANDOM raises Eval

Table 3: Reliability estimation under Neyman-style gold calibration with $B_g = 0.2$.

Estimator	Annotator 0	Annotator 1	Sum of squared error (SSE) ↓
Plug-in ($\widehat{\varepsilon}_k(z)$)	0.3605 (0.0019)	0.3305 (0.0018)	1.818
AIPW + Constant $\hat{\varepsilon}_k$	0.3733 (0.0152)	0.3237 (0.0148)	5.540
AIPW + Predicted $\widehat{\varepsilon}_k(z)$	0.3725 (0.0077)	0.3238 (0.0097)	1.516
AIPW + Oracle $\varepsilon_k(z)$	0.3724 (0.0075)	0.3237 (0.0096)	1.470

Accuracy from 0.7200 to 0.7630, Win Rate from 34.42 to 35.22, and LC Win Rate from 38.02 to 38.69. For Mistral-7B, it improves Eval Accuracy from 0.5690 to 0.5910, Win Rate from 13.15 to 14.63, and LC Win Rate from 18.90 to 20.72. These gains show that posterior correction can recover useful signal from noisy labels by replacing standard DPO loss with a noise-aware objective.

Result 2: information-guided routing improves assignment quality. Holding the estimated noise model $\widehat{\varepsilon}_k(z)$ and posterior correction fixed, replacing random routing with information-guided annotator routing further improves downstream performance. For LLAMA3-8B, INTEGER achieves 36.92 Win Rate and 40.14 LC Win Rate, compared with 35.22 and 38.69 for RANDOM. For Mistral-7B, INTEGER improves Eval Accuracy from 0.5910 to 0.6070, Win Rate from 14.63 to 15.09, and LC Win Rate from 20.72 to 21.41. Because these comparisons use the same estimated noise model and the same posterior-correction step, the improvement is attributable to routing itself.

Result 3: instance-dependent noise modeling matters. Across both base models, constant annotator-level noise is insufficient to capture the structure of the labeling noise. For LLAMA3-8B, the constant-noise baseline under RANDOM routing obtains 0.6960 Eval Accuracy, below the noise-unaware NONE baseline at 0.7200. Similarly, under INTEGER routing, constant noise gives weaker downstream performance than instance-dependent noise modeling, with 34.01 Win Rate and 35.48 LC Win Rate compared to 36.92 and 40.14. The same pattern appears for Mistral-7B: constant noise model remains close to the standard DPO baseline, while the estimated instance-dependent model achieves the strongest results. Thus, the key point is not merely that noise modeling helps, but that the noise model must be sufficiently expressive; a constant-noise model can be worse than ignoring noise because annotator reliability varies with instance difficulty.

5.2 Design-Aware Estimation of Annotator Reliability

Estimand and design. The target estimand is the marginal error rate $\rho_k = \mathbb{E}[\varepsilon_k(Z)]$. Our semi-synthetic construction gives oracle access to $\varepsilon_k(z)$, with empirical averages $\rho_0 = 0.3721$ and $\rho_1 = 0.3237$ used as ground truth. We fix routing to be Random, $q(k \mid Z) = 1/2$, and simulate Neyman-style gold validation with expected budget $B_g = 0.2$, with per-row probabilities $\pi_i \propto \sqrt{\widehat{\varepsilon}_{A_i}(z_i)(1-\widehat{\varepsilon}_{A_i}(z_i))}$ rescaled to $\mathbb{E}[\pi] = B_g$.

Estimators. We compare four estimators of ρ_k: (1) a plug-in estimator that does not account for selection bias, $\widehat{\rho}_k^{\text{plug}} = \frac{1}{n}\sum_i \widehat{\varepsilon}_k(z_i)$; (2) AIPW with a constant working model, $m_k(z) \equiv \varepsilon_k$; (3) AIPW with the learned $\widehat{\varepsilon}_k(z)$ from Section 5.1; (4) AIPW with the oracle $\varepsilon_k(z)$. We use 500 Monte-Carlo replications resampling A_i, the noisy label, and $G_i \sim \text{Bern}(\pi_i)$.

Table 3 shows that the plug-in estimator carries a bias of approximately 0.01 per annotator and the confidence intervals do not include the underlying truth, while AIPW removes this bias via the design correction term. The choice of working model determines whether the AIPW correction is efficient. AIPW with a constant working model yields SSE 5.54 — more than $3\times$ worse than AIPW with the learned $\widehat{\varepsilon}_k(z)$. The mechanism is design-specific: under Neyman validation, π_i is small on rows where $\widehat{\varepsilon}_k(z_i)$ is near 0 or 1, so $1/(q\,\pi_i)$ is large precisely on rows where the constant residual $Y_i^{\text{obs}} - \hat{\varepsilon}_k$ is most heterogeneous, inflating variance of the AIPW estimator. The instance-dependent working model shrinks this residual on each row and reduces SSE to 1.516. AIPW with the learned $\widehat{\varepsilon}_k(z)$ achieves SSE within 3% of the oracle AIPW SSE (1.516 vs. 1.470), closing roughly 99% of the gap between the constant and oracle working models.

6 Discussion

We presented a unified framework for DPO under noisy and heterogeneous feedback, combining annotator-aware posterior correction, information-guided routing, Neyman-style gold calibration, and design-aware annotator evaluation. Empirically, instance-dependent noise modeling together with information-guided routing improves clean preference recovery and AlpacaEval 2 performance. And the proposed AIPW estimator substantially reduces mean squared error in annotator reliability estimation under non-random design. Our method has limitations. In particular, the four components are currently derived as one-shot design decisions, whereas real preference pipelines are inherently iterative. Extending the framework so that routing, calibration, correction, and evaluation adapt across multiple rounds of data collection is an important direction for future work.

References

Amorim, G., Tao, R., Lotspeich, S., Shaw, P. A., Lumley, T., and Shepherd, B. E. (2021). Two-phase sampling designs for data validation in settings with covariate measurement error and continuous outcome. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 184(4):1368–1389.

Bang, H. and Robins, J. M. (2005). Doubly robust estimation in missing data and causal inference models. *Biometrics*, 61(4):962–973.

Cao, X., Xu, Z., Guang, M., Long, K., Bakker, M. A., Wang, Y., and Yu, C. (2025). Re-po: Robust enhanced policy optimization as a general framework for llm alignment. *arXiv preprint arXiv:2509.24159*.

Chowdhury, S. R., Kini, A., and Natarajan, N. (2024). Provably robust dpo: Aligning language models with noisy feedback. *arXiv preprint arXiv:2403.00409*.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. (2017). Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Das, N., Chakraborty, S., Pacchiano, A., and Ray Chowdhury, S. (2024). Active preference optimization for sample efficient RLHF. *arXiv preprint arXiv:2402.10500*.

Dubois, Y., Galambosi, B., Liang, P., and Hashimoto, T. B. (2024). Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

Gao, Y., Alon, D., and Metzler, D. (2024). Impact of preference noise on the alignment performance of generative language models. *arXiv preprint arXiv:2404.09824*.

Houlsby, N., Huszár, F., Ghahramani, Z., and Lengyel, M. (2011). Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*.

Kveton, B., Li, X., McAuley, J., Rossi, R., Shang, J., Wu, J., and Yu, T. (2025). Active learning for direct preference optimization. *arXiv preprint arXiv:2503.01076*.

Maystre, L. and Grossglauser, M. (2017). Just sort it! A simple and effective approach to active preference learning. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 2344–2353.

McIsaac, M. A. and Cook, R. J. (2014). Response-dependent two-phase sampling designs for biomarker studies. *Canadian Journal of Statistics*, 42(2):268–284.

Mukherjee, S., Lalitha, A., Kalantari, K., Deshmukh, A., Liu, G., Ma, Y., and Kveton, B. (2024). Optimal design for human preference elicitation. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Muldrew, W., Hayes, P., Zhang, M., and Barber, D. (2024). Active preference learning for large language models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pages 36577–36590.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. (2023). Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Robins, J. M., Rotnitzky, A., and Zhao, L. P. (1994). Estimation of regression coefficients when some regressors are not always observed. *Journal of the American Statistical Association*, 89(427):846–866.

Shen, Y., Sun, H., and Ton, J.-F. (2025). Reviving the classics: Active reward modeling in large language model alignment. *arXiv preprint arXiv:2502.04354*.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. (2020). Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021.

Tan, Z. (2010). Bounded, efficient and doubly robust estimation with inverse weighting. *Biometrika*, 97(3):661–682.

Wang, L., Yang, N., Huang, X., Yang, L., Majumder, R., and Wei, F. (2024). Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

Yan, Y., Rosales, R., Fung, G., Farooq, F., Rao, B., and Dy, J. G. (2012). Active learning from multiple knowledge sources. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1350–1357.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. (2019). Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Related Work

RLHF and direct preference optimization. Preference-based alignment has become a popular paradigm for steering large language models, building on a long line of work in RLHF that supervises models with human judgments rather than concrete rewards (Christiano et al., 2017; Ziegler et al., 2019; Stiennon et al., 2020; Ouyang et al., 2022). Direct Preference Optimization (DPO) (Rafailov et al., 2023) streamlined this pipeline by eliminating explicit reward modeling, making the *quality of the preference signal itself* the central determinant of alignment performance. Despite this elevated role, the DPO literature has largely treated preferences as if they came from a single, clean oracle—an idealization that breaks down in every realistic deployment, where preferences are produced by a heterogeneous pool of annotators with varying skill, cost, and reliability. Our work recast DPO not as learning from a fixed preference dataset, but as learning from a *designed annotation pipeline* whose structure can be exploited rather than ignored.

DPO with noisy labels. A growing body of work studies the sensitivity of preference optimization to label noise and proposes objective-level corrections to improve robustness (Chowdhury et al., 2024; Gao et al., 2024). Most recently, RE-PO (Cao et al., 2025) introduces an EM-based reweighting framework that infers the probability that each observed preference label is correct and rescales its contribution to the DPO loss. These results establish an important point: noisy preference labels can bias optimization, and modifying the loss can improve robustness. Our work differs in the structure of the noise model. Existing robust-DPO approaches typically rely on constant or noise models that do not capture comparison heterogeneity. For example, RE-PO estimates an average annotator-level reliability, but it does not model how the same annotator may be nearly deterministic on easy comparisons and close to random on difficult ones. This distinction is important in routed human feedback pipelines, where errors are not merely random perturbations but systematic functions of both annotator skill and instance difficulty. We therefore model the annotator error rate as an instance-dependent quantity, which is then incorporated directly into the DPO objective

through annotator-aware posterior correction. This instance-dependent correction is crucial, as shown in our experiments. Moreover, making $\varepsilon_k(z)$ explicit enables non-trivial components including information-guided routing, selective calibration, and design-aware evaluation of annotator reliability.

Active learning and information-guided DPO. Recent work has begun to incorporate active-learning ideas into RLHF by selecting prompts or response pairs expected to be most informative for downstream alignment (Shen et al., 2025; Houlsby et al., 2011; Maystre and Grossglauser, 2017; Mukherjee et al., 2024; Muldrew et al., 2024; Kveton et al., 2025; Das et al., 2024). These methods primarily ask: *which comparison should be labeled and used?* In real annotation pipelines, however, the value of a label depends not only on *what* is being compared but also on *who* provides the annotation. Sending a difficult comparison to an unreliable annotator may produce little useful information, while assigning an easy comparison to an expert annotator may waste scarce expert effort. We therefore reformulate the basic design unit as a *comparison–annotator pair*, turning preference data acquisition from item selection into an experimental design problem. Under this view, annotator routing becomes an optimization-driven component of the DPO pipeline: comparisons are routed to annotators according to their expected statistical value.

Doubly robust estimation and design-aware evaluation. The alignment literature has paid almost no attention to *evaluating* the annotators themselves, even though decisions about annotator compensation, retention, and trust-tier assignment depend critically on accurate annotator reliability estimates. The challenge is that informative routing or gold label calibration creates selection bias—the very design choices that improve training corrupt naive evaluation. This has been long studied in causal inference literature: outcomes observed only on a non-random subset, with inverse-propensity weighting and its augmented variants providing consistent and efficient estimation (Robins et al., 1994; Bang and Robins, 2005; Tan, 2010; McIsaac and Cook, 2014; Amorim et al., 2021). To our knowledge, we are the first to bring this machinery into the RLHF setting, recognizing that annotator reliability estimation is fundamentally a missing-data problem and that the same designed-experiment framing that motivates our routing rule also affects how annotator quality must be measured. The resulting AIPW estimator is not a diagnostic but an integral part of a closed-loop annotation system.

Summary. Prior work has treated preference optimization, noisy labels, active acquisition, and design-aware estimation as separate problems. We provide the first unified DPO framework that solves them jointly: posterior correction targets the clean preference objective, information-guided routing maximizes the statistical value of annotations, selective gold calibration refines $\varepsilon_k(z)$ estimation, and design-aware AIPW estimation evaluates annotators under the same data collection pipeline. The result is not just a better DPO method, but a new way of thinking about preference data: as a designed experiment whose every component—collection, correction, calibration, evaluation—can and should be optimized end-to-end.

B Additional experiment details

Source Dataset. For different LLM model architectures, we use different source dataset:

Mistral-7B-Instruct-v0.2:

https://huggingface.co/datasets/princeton-nlp/mistral-instruct-ultrafeedback

Llama-3-8B-Instruct:

https://huggingface.co/datasets/princeton-nlp/llama3-ultrafeedback-armorm

We filter a subset with $n = 20$k from source dataset and use them for our experiments.

Data Construction. For each prompt x, we form pairwise comparisons $z = (x, y_1, y_2)$ from candidate responses to the same prompt. The clean label $L^*(z) \in \{0, 1\}$ is obtained from the reference score signal, where $L^*(z) = 1$ means that y_1 is preferred to y_2. Synthetic annotator labels are then generated by corrupting $L^*(z)$ according to the data generating process (DGP) below.

Difficulty model. We define an instance-level difficulty score $d(z)$ from surface-level features and `multilingual-E5-large-instruct` text embedding (Wang et al., 2024). Let $e_x = E(x)$,

$e_1 = E(y_1)$, and $e_2 = E(y_2)$ denote the prompt and response embeddings. We compute

$$c_{12} = \cos(e_1, e_2), \qquad c_{x1} = \cos(e_x, e_1), \qquad c_{x2} = \cos(e_x, e_2),$$

$$s(z) = \frac{1}{D}\|e_1 - e_2\|_1, \qquad \Delta_{\text{tok}} = \big|\#\text{tok}(y_1) - \#\text{tok}(y_2)\big|, \qquad \Delta_{\text{char}} = \big||y_1| - |y_2|\big|.$$

The raw difficulty score is

$$d_{\text{raw}}(z) = \alpha_1 c_{12} + \alpha_2 \frac{c_{x1} + c_{x2}}{2} - \alpha_3 s(z) - \alpha_4 \Delta_{\text{tok}} - \alpha_5 \Delta_{\text{char}},$$

with $(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5) = (2.0, 1.0, 2.0, 0.25, 0.10)$. We standardize $d_{\text{raw}}(z)$ over the training split to obtain $d(z)$. Higher $d(z)$ corresponds to more ambiguous comparisons.

Training the noise model. We mirror the form of the DGP and parameterize it as

$$\widehat{\varepsilon}_k(z) \;=\; \sigma\big(s_k\, d_\theta(z) \;-\; a_k\big),$$

with per-annotator ability $a_k \in \mathbb{R}$ and per-annotator slope s_k. The shared difficulty head $d_\theta(z)$ is a two-layer MLP with hidden width 64,

$$d_\theta(z) \;=\; u^\top \text{ReLU}\big(W\varphi(z) + c\big) + b, \qquad \varphi(z) \;=\; g\big(\text{E5}(x, y_1),\, \text{E5}(x, y_2)\big),$$

where $\text{E5}(\cdot)$ denotes the frozen `multilingual-E5-large-instruct` text embedding applied to the rendered chat string, and g concatenates and differences the two pooled embeddings to form a single comparison vector. The features $\varphi(z)$ are precomputed and cached.

We train the noise model on a held-out gold-labeled slice $\mathcal{D}_g$ with $n_g = 5{,}000$ pairs, split into $4{,}000/1{,}000$ train/validation pairs. Since the clean label L^* is observed, we define the annotator-specific mismatch target

$$y_{i,k} = \mathbf{1}\{\widetilde{L}_k(z_i) \neq L^*(z_i)\}.$$

We jointly optimize $\{W, c, u, b\}$, $\{a_k\}$, and $\{\tilde{s}_k\}$ by minimizing the Bernoulli negative log-likelihood with ℓ_2 regularization, using AdamW with weight decay $\lambda = 10^{-3}$. We select the epoch with lowest validation loss and clip predictions to $[10^{-3}, 1 - 10^{-3}]$. The fitted $\widehat{\varepsilon}_k$ is used for routing, posterior correction, and reliability estimation in Section 5.2.

Hyperparameter. We also present hyperparameter specification in Table 4 and 5.

Table 4: DPO Training Hyperparameters

Category	Hyperparameter	Value
Base Model	Model	Meta-Llama-3-8B-Instruct
	Reference Policy	PEFT-Implicit (Frozen Base)
	Tokenizer	Same as Base
LoRA	Rank r	16
	α	32
	Dropout	0.05
	Bias	None
	Target Modules	$\{q, k, v, o, gate, up, down\}_{proj}$
	Task Type	CAUSAL_LM
DPO Objective	β	0.1
Optimization	Optimizer	AdamW
	LR Schedule	Linear Decay (HF Default)
	Epochs	1
	Per-Device Batch	1
	Gradient Accumulation	16
	Precision	bf16
Sequence	max_length	1024
	max_prompt_length	512
Miscellaneous	Seed	0
	Hardware	NVIDIA RTX 6000 Ada, 48 GB

Table 5: Noise-Model Training Hyperparameters.

Category	Hyperparameter	Value
Embedding	Backbone	multilingual-e5-large-instruct (Frozen)
	Feature Dim	3,077
Head	Type	2-Layer MLP
	Hidden Dimension	64
	Slope	Per-Annotator
	Ability	Learned Per Annotator
Loss / Target	Target	Mismatch Indicator $1\{\tilde{\ell} \neq \ell^*\}$
	Loss	BCE
	Selection Metric	Validation Loss
Optimization	Optimizer	AdamW
	Learning Rate	1×10^{-2}
	Weight Decay	1×10^{-3}
	Epochs	200 (Full-Batch)