MiCRo: Mixture Modeling and Context-aware Routing for Personalized Preference Learning

Jingyan Shen[2]*, Jiarui Yao[1]*, Rui Yang[1]*, Yifan Sun[1], Feng Luo[3], Rui Pan[1], Tong Zhang[1], Han Zhao[1]
[1]University of Illinois Urbana-Champaign,
[2]New York University, [3]Rice University.

Abstract

Reward modeling is a key step in building safe foundation models when applying reinforcement learning from human feedback (RLHF) to align Large Language Models (LLMs). However, reward modeling based on the Bradley-Terry (BT) model assumes a global reward function, failing to capture the inherently diverse and heterogeneous human preferences. Hence, such oversimplification limits LLMs from supporting personalization and pluralistic alignment. Theoretically, we show that when human preferences follow a mixture distribution of diverse subgroups, a single BT model has an irreducible error. While existing solutions, such as multi-objective learning with fine-grained annotations, help address this issue, they are costly and constrained by predefined attributes, failing to fully capture the richness of human values. In this work, we introduce MiCRo, a two-stage framework that enhances personalized preference learning by leveraging large-scale binary preference datasets without requiring explicit fine-grained annotations. In the first stage, MiCRo introduces context-aware mixture modeling approach to capture diverse human preferences. In the second stage, MiCRo integrates an online routing strategy that dynamically adapts mixture weights based on specific context to resolve ambiguity, allowing for efficient and scalable preference adaptation with minimal additional supervision. Experiments on multiple preference datasets demonstrate that MiCRo effectively captures diverse human preferences and significantly improves downstream personalization.

1 Introduction

Reinforcement Learning from Human Feedback (RLHF) unlocks a promising pathway to improve the performance, reliability, and adaptability of AI system deployment (Bai et al., 2022; Dong et al., 2023a; Achiam et al., 2023; Dong et al., 2024). Rather than relying on handcrafted reward models, the prevailing approach in RLHF employs *preference learning* (Christiano et al., 2017) to infer reward scores from human feedback, particularly for tasks involving subjective evaluation and open-ended responses without unanimous ground truths (Ziegler et al., 2019). However, most existing methods rely on *binary-labeled* pairwise datasets building upon the assumption that there exists a global reward function that can model human preferences. This fails to capture the diverse and often *contradictory* nature of human preferences, ultimately limiting their effectiveness for personalized and pluralistic alignment (Chakraborty et al., 2024a; Yang et al., 2024b; Mukherjee et al., 2024; Luo et al., 2025).

Advancing preference learning to better accommodate heterogeneous human preferences remains an open challenge. Some recent studies seek to capture the diversity by collecting multifaceted annotations that distinguish between different evaluation attributes, e.g., helpfulness, harmlessness, coherence, instruction-following, etc (Wang et al., 2024b; Bai et al., 2022; Wang et al., 2024c). Although fine-grained labels provide deeper insights into individual preferences, collecting and curating them significantly increases data acquisition costs. Consequently, existing datasets limit their scope to a handful of pre-defined attributes and often rely on LLM-as-a-Judge (Zheng et al., 2023) for labeling response pairs. This raises concerns about their fidelity in representing the nuanced and ever-evolving landscape of human values.

In addition, while users may share common interests, such as preferring a helpful and harmless assistant, their expectations are ultimately individualized and depend on use cases, i.e., *contextual factors*, as illustrated in Fig. 1. To better capture

*Equal contribution. Emails: {jiarui14, ry21, yifan50, ruip4, tozhang, hanzhao}@illinois.edu, jingyan.s@nyu.edu, fl38@rice.edu. Code is available at https://github.com/uiuctml/MiCRo.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 17447–17463
November 4-9, 2025

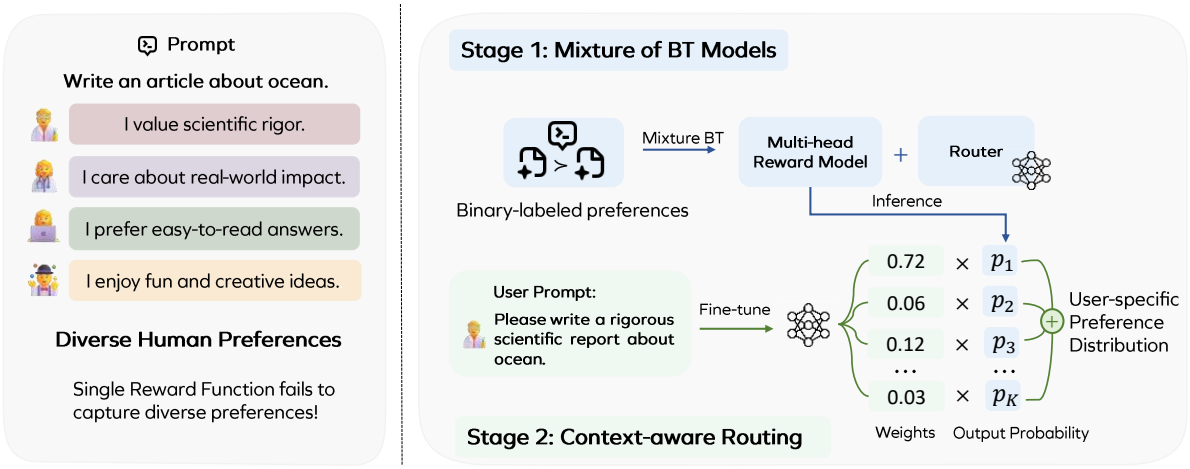


Figure 1: **Illustration of the two-stage pipeline of MiCRo for capturing personalized preferences.** A mixture of reward models (Stage 1) is trained on binary-labeled data, while the context-aware router (Stage 2) dynamically adjusts preference distributions based on user-provided context. The final preference distribution is obtained through a convex combination of different preference distributions.

such personalization, some approaches construct datasets with elaborate and pluralistic contexts into user prompts (Pitis et al., 2024; Yang et al., 2024b) or system instructions (Lee et al., 2024a). While reward models trained on such enriched datasets have shown improved generalization to personalized preferences, designing these criteria manually is still labor-intensive.

In this work, we introduce MiCRo, a two-stage, context-aware mixture modeling framework that leverages large-scale binary preference datasets to improve personalized preference learning. We first provide a theoretical result showing that when the underlying preference distribution follows a mixture of subpopulations, preference learning based on a single Bradley-Terry (BT) loss incurs an irreducible error. To address this, we propose a context-aware mixture modeling approach to decompose aggregate preferences into latent subpopulations, each with a distinct reward function. To further adapt to personalized preferences, we propose an online routing strategy with additional contextual information. In summary, our method offers two key advantages:

- MiCRo extracts multifaceted human preferences from widely available pairwise comparison datasets without requiring explicit fine-grained annotations or predefined attributes.
- MiCRo adapts the mixture heads to personalized preference learning with contextual information with only a limited number of samples.

Our extensive experiments across multiple preference datasets empirically demonstrate that MiCRo 's mixture heads effectively capture diverse preferences and achieve superior performance compared to baselines on multidimensional benchmarks. With the addition of context-aware routing, the full MiCRo framework matches the performance of fully supervised and test-time adaptation methods, underscoring its effectiveness in enhancing downstream personalization.

2 Related Work

Reward modeling aims to learn a function that assigns scalar scores to input–output pairs based on human preferences, playing a central role in RLHF by steering LLM behavior toward human-aligned outputs and mitigating harmful responses (He et al., 2024a; Sun et al., 2024). The typical approach adopts the BT model (Bradley and Terry, 1952; Christiano et al., 2017; Stiennon et al., 2020) to learn from pairwise comparisons. To further address the diversity of human preferences, personalized preference learning seeks to align LLMs with user values in underspecified settings with ambiguous or heterogeneous intent (Fleisig et al., 2023; Baumler et al., 2023; Chakraborty et al., 2024b). One major approach focuses on multi-objective alignment through ensembles of reward models. Techniques such as Mixture-of-Experts and model merging are used to decompose reward functions into task-specific or capability-based components (Quan, 2024; Wang et al., 2024b; Rame et al., 2023; Wang et al., 2024a). However, training multiple reward models typically requires manually defined preference dimensions and dense supervision. To mitigate this, HyRe (Lee et al.,

2024b) trains an ensemble offline and adapts it to individual users at test time by dynamically reweighting the components using a small number of user-specific examples. Recent work, DRMs (Luo et al., 2025), decomposes human preferences into a linear space using PCA, offering a promising training-free solution; however, its effectiveness depends on the choice of embedding model. A complementary line of work uses probabilistic approaches to model subgroup or latent preferences without explicit supervision (Siththaranjan et al., 2023; Poddar et al., 2024; Chen et al., 2024; Chakraborty et al., 2024a), but their potential for personalization remains underexplored.

3 Limitation of a Single Reward Function

3.1 Problem Setup

Notation and Preliminary Let $\mathcal{X}$ denote the space of prompts, and $\mathcal{Y}$ denote the space of responses. Denote $\Delta^K = \{\mathbf{x} \in \mathbb{R}^K \mid \sum_{i=1}^K x_i = 1, x_i \geq 0, i = 1, \ldots, K\}$ as the $(K-1)$-dimensional probability simplex. In standard preference learning, human preferences are modeled based on the classic BT model. Specifically, for a given prompt $x \in \mathcal{X}$ and an LLM π, two candidate responses $a_w, a_l \in \mathcal{Y}$ are sampled independently from $\pi(\cdot \mid x)$. The probability that a human annotator prefers a_w over a_l is given by:

$$\mathbb{P}\left(a_w \succ a_l \mid x\right) = \sigma\left(r^*\left(x, a_w\right) - r^*\left(x, a_l\right)\right),$$

where σ denotes the logistic function and $r^* : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is a latent reward function. For brevity, we assume $a_w \succ a_l$ (i.e., a_w is always preferred over a_l). In practice, a static, finite-sample preference dataset is collected, and r^* is estimated via maximum likelihood.

Mixture Reward Distribution However, in practice, reward data are collected from a population of annotators with inherently diverse preferences. Prior work has demonstrated that modeling all human preferences with a single parametric reward function leads to systematic underfitting and cannot capture such heterogeneity (Chakraborty et al., 2024a; Siththaranjan et al., 2023). To better reflect this diversity, we assume that each observed annotation is generated from one of the K latent subpopulations, where K is treated as a hyperparameter. A latent categorical variable $z \in \{1, \ldots, K\}$ is introduced as an indicator of the subpopulation from which a preference pair originates. We introduce the overall probability of a preference observation as a *context-aware* mixture of K Bradley-Terry models:

$$\begin{aligned}\mathbb{P}(a_w \succ a_l \mid x) = \sum_{k=1}^{K} \mathbb{P}(z = k \mid x) \\ \cdot \mathbb{P}(a_w \succ a_l \mid x, z = k),\end{aligned} \tag{1}$$

where the weights of each mixture component depend on the prompt x and the probability of preference within a specific subpopulation is given by

$$\mathbb{P}(a_w \succ a_l \mid x, z = k) = \sigma\big(r_k^*(x, a_w) - r_k^*(x, a_l)\big), \tag{2}$$

and r_k^* is assumed to be a latent reward function for subpopulation k.

3.2 Irreducible error of single BT model

In this section, we provably show that, when the underlying preference distribution is a mixture of BT models, no matter how rich the model class is for reward functions, preference learning based on a single BT model has an irreducible error. Before we present our result, we first assume the diversity of the underlying population:

Assumption 3.1 (Diversity)**.** *There exists a constant $\rho > 0$, such that for every prompt $x \in \mathcal{X}$ and every subpopulation group $k \in [K]$, $\mathbb{P}(z = k \mid x) \geq \rho$.*

For every tuple (x, a_w, a_l), define the score function s_k^* for group k as $s_k^*(x, a_w, a_l) := r_k^*(x, a_w) - r_k^*(x, a_l)$. Let $\mathcal{L}_{\mathrm{CE}}(r)$ be the cross-entropy loss of a BT preference model $\mathbb{P}(a_w \succ a_l \mid x) = \sigma(r(x, a_w) - r(x, a_l))$ according to the reward function r,

Theorem 3.2 (Error lower bound)**.** *For an arbitrary reward function r, if the predicted preference is based on a single BT model, then $\mathcal{L}_{\mathrm{CE}}(r) \geq 2\rho K \mathbb{E}_x \mathrm{Var}_z[\{\mathbb{E} s_k^* \sharp \pi(a_w, a_l \mid x)\}_{k=1}^K] + H(x, \pi, \mathbb{P}(z|x))$.*

We defer the detailed proof of Theorem 3.2 to Appendix A. In the lower bound, $s_k^* \sharp \pi(a_w, a_l \mid x)$ is the induced distribution of s_k^* acting on the pair of responses $(a_w, a_l) \sim \pi$ from the LLM given the prompt x, and the variance operator Var_z is applied with respect to the K groups. $H(x, \pi, \mathbb{P}(z|x))$ is the Shannon entropy of the joint distribution over preference data given by the prompt, subpopulations, as well as the LLM π (c.f. Appendix A for its definition). At a colloquial level, the lower bound

Method	Binary Labels	Context Conditioning	Reward Objective	Weight Learning	Special Characteristic
ARMO (Wang et al., 2024b)	×	×	Mean Square Error	End-to-end BT	Fixed Num of attributes
MaxMin (Chakraborty et al., 2024a)	✓	×	Mixture of BT	Hard clustering	Minority preference optimization
HyRe (Lee et al., 2024b)	✓	×	BT	Accuracy maximization	Test-time adaptation
DRMs (Luo et al., 2025)	✓	×	–	Accuracy maximization	Training-free reward decomposition via PCA
MiCRo (Ours)	✓	✓	**Mixture of BT**	**Hedge Algorithm**	**Context-aware Routing**

Table 1: **Comparison of different methods and their key characteristics.** MiCRo optimizes a mixture of BT loss using binary labels and enables context-aware routing, setting it apart from prior methods in terms of context conditioning and weight learning.

says that, the more diverse the ground-truth scores s_k^* from each subpopulation (hence a larger variance), or the subpopulation distribution $\mathbb{P}(z \mid x)$ (hence a larger ρ and entropy), then the larger the cross-entropy loss of using a single BT model.

4 Method

The inherent limitation of a single BT model motivates the need for richer preference modeling. However, two key challenges remain: **(C1)** *How to extract a mixture of reward functions from binary-labeled datasets without incurring additional annotation costs?* **(C2)** *Given limited access to user-specific intent, how can we efficiently adapt to personalized preferences at deployment time?*

To this end, we propose a two-stage algorithm that first uncovers latent heterogeneity in human preferences through mixture modeling, and then adapts to individual users via a lightweight, context-aware online routing strategy.

4.1 Mixture Modeling for Diverse Preferences

We begin by fundamentally comparing our mixture modeling objective with previous methods and then introduce the detailed design of our approach.

Comparison with Prior Mixture Modeling Approaches Unlike static and unconditional mixture approach used in previous work (Chakraborty et al., 2024a), our formulation from Equation (1) introduces a dynamic, context-aware weighting mechanism for mixture models by conditioning the subpopulation weights $\mathbb{P}(z = k \mid x)$ on the given prompt x. We emphasize that *this is a crucial design that allows for contextual specialization, where prompts automatically activate the most relevant subpopulation's reward model.* By mimicking real-world expertise allocation, our approach avoids the diluted performance of static averaging. We provide a more detailed comparison of our method with existing works in Table 1.

Mixture Modeling Designs In practice, we parameterize the reward function for each subpopulation as $r_{\phi_k} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ for $k = 1, \ldots, K$ and model the mixture weights with a network $f_\psi : \mathcal{X} \to \Delta^K$. Given a training dataset $\mathcal{D} = \{(x, a_w, a_l)_i\}_{i=1}^n$, we minimize the negative log-likelihood defined as:

$$\mathcal{L}_{\text{mle}} = -\frac{1}{n} \sum_{(x,a_w,a_l)\in\mathcal{D}} \log \sum_{k=1}^{K} \Big(f_{\psi,k}(x) \cdot \sigma\left(r_{\phi_k}(x, a_w) - r_{\phi_k}(x, a_l)\right) \Big). \quad (3)$$

To prevent any single model from dominating, we add a regularization term by imposing a uniform prior to the weight distribution:

$$\mathcal{L}_{\text{reg}} = \frac{1}{n} \sum_{(x,a_w,a_l)\in\mathcal{D}} \sum_{k=1}^{K} f_{\psi,k}(x) \log f_{\psi,k}(x). \quad (4)$$

The final loss function becomes

$$\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\psi}) = \mathcal{L}_{\text{mle}} + \alpha \mathcal{L}_{\text{reg}}, \quad (5)$$

where the coefficient α is set to 0.5 in our implementation. Overall, this mixture training phase on large-scale datasets learns a diverse set of reward functions, establishing a robust foundation for adaptation to nuanced preferences.

4.2 Context-aware Routing for Personalized Preference Learning

While the pre-trained mixture model has the potential to capture latent reward functions, assigning meaningful weights to these reward heads upon a given prompt is difficult without a clear signal of user intent. Technically, during the training of the mixture model, since we do not have labeled data to train the router separately, we will need one strategy to learn the correspondence between the mixture reward heads and the underlying user intent for better routing assignments.

Recent work on contextual alignment (Pitis et al., 2024; Lee et al., 2024a; Poddar et al., 2024) highlights that incorporating context can reduce ambiguity and improve estimation accuracy. Motivated by this, we introduce a second stage that incorporates more concrete contextual information—such as user instructions or metadata (e.g., demographics or interaction history)—to guide the routing strategy by learning the correspondence between user intent and mixture reward heads.

Unlike prior methods that require training reward models on large-scale contextual datasets (Pitis et al., 2024; Lee et al., 2024a), our approach avoids costly data collection and full model retraining. Instead, we leverage the unsupervised mixture heads pre-trained in the first stage to enable sample-efficient online adaptation. This allows us to refine the mixture weights and generate personalized predictions using only a small number of samples.

To this end, we propose to fine-tune the routing network f_ψ using the Hedge algorithm (Arora et al., 2012), where each input is a pair $(x_i, c_i) \sim \mathcal{D}_c$, with c_i denoting additional context information. Intuitively, Hedge maintains a set of experts (i.e., reward heads) and adaptively reweights them based on their performance—assigning higher weights to those that better align with observed preferences. In our framework, the user preferences can be modeled as a convex combination of the K latent subpopulation preferences. For an example $(x_i, c_i, a_{i,w}, a_{i,l})$, denote the output probability from k-th head as $p_k(a_{i,w} \succ a_{i,l}|x_i) := \sigma(r_{\phi_k}(x_i, a_{i,w}) - r_{\phi_k}(x, a_{i,l}))$ and define

$$\mathcal{L}_{i,k} := -\log p_k(a_{i,w} \succ a_{i,l}|x_i, c_i).$$

We consider an online learning setting where contextual data is collected within a budget of B. We acquire a batch of preference pairs $\mathcal{D}_\mathcal{A} = \{(x, c, a_w, a_l)_i\}_{i=1}^B$ with additional contexts. Motivated by the multi-task learning literature (He et al., 2024b; Liu et al., 2024), we propose a training objective for the router based on the framework of online mirror descent with KL divergence regularization (Hazan et al., 2016):

$$\min_\psi \quad \frac{1}{B}\sum_{i=1}^{B} \mathcal{L}_i, \quad \text{s.t.} \quad f_\psi(x_i, c_i) \in \Delta^K,$$

where $\mathcal{L}_i := \sum_{k=1}^K f_\psi(x_i, c_i)_k \mathcal{L}_{i,k} + \tau \mathrm{KL}(f_\psi(x_i, c_i) \| \boldsymbol{\omega}_i)$ and $\boldsymbol{\omega}_i$ is a weight vector that comes from a previous iteration or pre-trained weights, and $\tau \geq 0$ is a temperature hyperparameter. Note that the first term is an upper bound of the negative log-likelihood function of mixture distribution based on Jensen's inequality.

Routing with Hedge Algorithm With $\tau > 0$, the optimal solution for each batch will be:

$$f_{i,k} = \frac{\omega_{i,k} \exp\left(-\mathcal{L}_{ik}/\tau\right)}{\sum_{j=1}^K \omega_{i,j} \exp\left(-\mathcal{L}_{ij}/\tau\right)}, \tag{6}$$

which yields Algorithm 1 to learn the router iteratively with contextual information. For each iteration, we determine optimal weights target as soft labels using Equation (6). Then, we fine-tune the router by minimizing the cross-entropy loss between the soft labels and the router network's predictions.

Algorithm 1 Context-aware Router Learning

Input: Mini-batch $\{(x_i, c_i, a_w^i, a_l^i, y_i)\}_{i=1}^{B_t}$, temperature τ, pre-trained router f_ψ, iterations T, reward heads from the first stage $r_{\phi_k}, \forall k = 1, \ldots, K$.

Initialize $\psi^{(1)} = \psi$, $\omega_{i,k} = f_{\psi,k}(x_i)$

for $t = 1$ *to* T **do**

  `// weight update`

  for $i = 1$ *to* B_t **do**

    for $k = 1$ *to* K **do**

      $\mathcal{L}_{ik} \leftarrow -\log p_k(a_w^i \succ a_l^i \mid x_i)$

      $\omega_{i,k} \leftarrow f_{\psi^{(t)}}(x_i, c_i)_k \cdot \exp\left(\frac{-\mathcal{L}_{ik}}{\tau}\right)$

    end

    $Z \leftarrow \sum_{j=1}^K \omega_{i,j}$

    $\omega_{i,k} \leftarrow \frac{\omega_{i,k}}{Z}$ for $k = 1, \ldots, K$

  end

  `// router update`

  $\mathcal{L}_{\text{weight}}^{(t)}(\psi) \leftarrow \frac{1}{B_t}\sum_{i=1}^{B_t} \mathcal{L}_{\text{CE}}\left(\boldsymbol{\omega}_i\,,\, f_{\psi^{(t)}}(x_i, c_i)\right)$

  Backprop $\mathcal{L}_{\text{weight}}^{(t)}(\psi)$ to transition from $\psi^{(t)}$ to $\psi^{(t+1)}$

end

Our routing learning approach offers two clear advantages in deployment: (1) ***efficiency***: By leveraging the expert heads trained on large-scale datasets in the first stage, the second stage does not require retraining the reward model or relying on extensive labeled data; instead, a lightweight, online router continuously adapts during deployment. (2) ***generalizability***: Our learning-based router harnesses contextual information to adjust weights

with a learning-based algorithm. Unlike test-time adaptation methods (Lee et al., 2024b) that rely on a set of test data for re-weighting, our router is *trained online*, allowing generalizing to new contexts without access to specific test data.

5 Experiments

In our experiments, we aim to answer two research questions: **(Q1)** Can our context-aware mixture modeling framework extract diverse reward functions from binary-labeled preference data? **(Q2)** Can the fine-tuned routing network enable effective personalization by adapting to contextual signals?

5.1 Experimental Setup

Training datasets We train the mixture reward models on binary-labeled preference datasets: HelpSteer2 (Wang et al., 2024c), RPR (Pitis et al., 2024), and preference-700K (Dong et al., 2024). HelpSteer2 and RPR datasets contain human-labeled response pairs evaluated across multiple assessment dimensions. We construct the binary-labeled sets with the following process. For each dimension, we extract binary preference pairs based on absolute ratings, treating responses with higher ratings as "chosen" and those with lower ratings as "rejected." Pairs with identical ratings are excluded from both training and test sets. To ensure diversity in preferences, we exclude pairs where all attributes are unanimously agreed upon from the training set. Ultimately, all pairs from all dimensions are mixed, resulting in 23.5K samples and 5.8K training samples from HelpSteer2 and RPR, respectively. The preference-700K dataset is a large-scale pairwise dataset created by aggregating instructions and response pairs from multiple data sources. Further details on these datasets are provided in Appendix B.2.

Models We use the best 3B open-source reward model GRM-Llama3.2-3B (Yang et al., 2024a) as the backbone, keeping it frozen while training K linear probing heads on top. The router network is implemented as a one-layer MLP containing 128 units and a softmax activation. Full implementation details are provided in Appendix B.1. We further evaluate with an open-source 8B reward model in Appendix C.5.

Baselines We evaluate the following baselines: (1) *Single Reward*: A single-head model trained using the standard BT loss. (2) *Static Mixture*: A simplified variant of our method, corresponding to the approach used in MaxMin-RLHF (Chakraborty et al., 2024a), where the mixture model is trained with fixed, input-independent weights, without leveraging contextual information. (3) *Shared-Base Ensemble Model*: Lee et al. (2024b) introduces HyRe, a test-time adaptation approach based on ensemble networks. We adopt the multi-head architecture with a frozen prior network and multiple trainable heads, optimizing a uniformly weighted sum of BT losses. (4) *Fully Supervised Model*: We include ARMO (Wang et al., 2024b), an 8B model trained on more than 500K fine-grained labels, as a baseline with full supervision.

5.2 Stage-1 Evaluation: Can MiCRo Disentangle Diverse Human Preferences?

Evaluation Setting In this experiment, we train MiCRo and baseline models (except ARMO) on HelpSteer2 and RPR training sets. We then evaluate the learned heads on the HelpSteer2 and RPR test sets, which cover 14 distinct preference dimensions. Each head is evaluated individually on every dimension. To ensure fair comparisons, we use the same number of heads K for MiCRo and other multi-head baselines. The full evaluation results of MiCRo mixture heads are provided in Appendix C.1.

Results Fig. 2 compares the best-performing head for each test dimension. The results demonstrate that different heads from MiCRo specialize in distinct evaluation dimensions and consistently surpass the performance of all baeslines across all dimensions. On average, MiCRo consistently achieves the highest average scores across both RPR (0.921) and HelpSteer2 (0.811) benchmarks, with substantial gains over the single-head baseline (+40.0% on RPR, +6.8% on HelpSteer2), the *Share-base* ensemble baseline (+20.7% and +9.1% respectively), and the mixture model without context routing (+5.5% and +1.1% respectively), demonstrating the robust benefits of context-aware mixture modeling approach. These results suggest that mixture modeling more effectively captures latent diverse preferences compared to single reward or ensemble models, and that context-aware weighting further improves over static mixtures. Fig. 3 presents a qualitative example of mixture weights from the Stage 1 router, showing how different prompts from the RPR test set activate different heads. This highlights the effectiveness of our con-

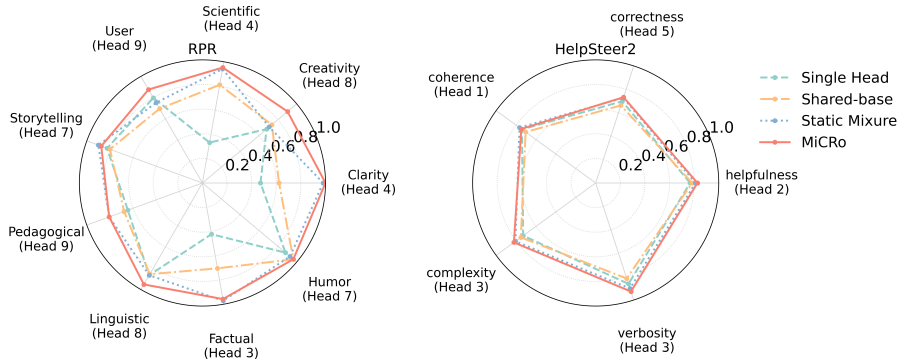


Figure 2: **Comparison of accuracy scores between the best heads of MiCRo and other baselines on multiple test dimensions.** The mixture heads can disentangle diverse human preferences, with different heads excelling on different attributes. They consistently outperform the single reward model across all attributes. Overlaps where the same head dominates multiple attributes may reflect inherent attribute correlations.

Method	Supervision	Helpfulness	Correctness	Coherence	Complexity	Verbosity	Average
Single Reward	Binary	0.7838	0.6686	0.6914	0.7907	0.8816	0.7632
Shared-Base Best Head	Binary	0.7838	0.6628	0.7037	0.7519	0.8158	0.7436
Static Mixture	Binary	0.7243	0.6570	0.6790	0.8372	0.9013	0.7598
ARMO (8B)	Fine-grained	0.6919	0.6395	0.7593	0.7132	0.7500	0.7108
HyRe	Binary + Test Labels	0.7692	0.6987	0.6781	0.7168	0.8015	0.7329
MiCRo-HyRe	Binary + Test Labels	0.8270	0.7035	0.7407	0.8217	0.8487	0.7883
MiCRo (Ours)	Binary + Context	0.8324 +0.05	0.7140 +0.04	0.7543 +0.06	0.7628	0.8513	0.7830 +0.02

Table 2: **Accuracy scores on HelpSteer2 test set.** On average, MiCRo outperforms baselines across various attributes and overall results. Scores in green indicate absolute improvement over the Single Reward baseline. All baselines except ARMO (8B) use the same 3B base model.

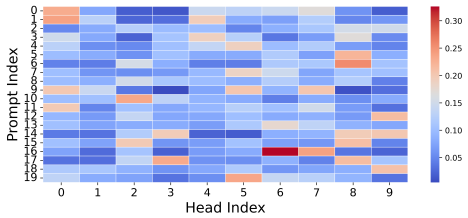


Figure 3: **Heatmaps of router weights for different prompts in MiCRo Stage 1.** The router assigns varying weights to different heads depending on the prompt.

textual router compared to the unconditional router used in prior work.

5.3 Stage-2 Evaluation: Can MiCRo Adapt to Personalized Preference?

User Context Datasets The RPR training dataset includes user-specific criteria for each preference pair, explicitly specifying the user's intent and evaluation dimension, and thus provides a well-defined source of user context. For HelpSteer2, we follow the approach of Pitis et al. (2024) and augment generic prompts with attribute-specific modifications based on the original assessment dimensions in the annotation process (Wang et al., 2024c). Examples of contexts are provided in Appendix B.2. For training and test datasets, we prepend the contextual information to the user prompt and fine-tune the router accordingly.

Evaluation Setting We assess personalized adaptation under two scenarios: (1) ***In-distribution evaluation.*** All models are trained on the HelpSteer2 and RPR training splits and evaluated on their respective test splits. (2) ***Cross-distribution generalization.*** Models are trained on HelpSteer2 and the large-scale preference-700K datasets, then evaluated on the RPR test set to measure transfer to previously unseen user preferences.

Implementation Details For MiCRo , we train the router using 50 context-annotated examples per attribute drawn from the training data. For the *static* mixture baseline, we keep the Stage-1 mixture weights fixed. For HyRe adaptation, we reuse the Stage-1 reward heads and derive adaptation weights from 16 labeled test samples per attribute.

Results As shown in Tab. 2 and Tab. 3, MiCRo achieves average test accuracies of 0.7830 on HelpSteer2 and 0.8218 on RPR under the within-dataset evaluation setting, outperforming all three base-

Method	Supervision	Clarity	Creativity	Scientific Rigor	User-Friendliness	Storytelling	Pedagogical	Linguistic Creativity	Factual Accuracy	Humor	Average
Single Reward	Binary	0.4717	0.6806	0.3333	0.7978	0.8375	0.6452	0.8654	0.4225	0.8810	0.6594
Shared-Base Best Head	Binary	0.6226	0.7361	0.8095	0.6966	0.8000	0.6774	0.8558	0.7042	0.9643	0.7629
Static Mixture	Binary	0.9057	0.6389	0.9048	0.6854	0.6250	0.7903	0.7404	0.8451	0.9167	0.7836
ARMO (8B)	Fine-grained	0.9057	0.6806	0.9405	0.6966	0.7875	0.7903	0.9135	0.9014	0.9463	0.8403
HyRe	Binary + Test Labels	0.7027	0.5893	0.6618	0.8493	0.6563	0.7826	0.7045	0.7091	0.4853	0.6823
MiCRo-HyRe	Binary + Test Labels	0.9556	0.8125	0.9605	0.9012	0.8333	0.7963	0.9063	0.9524	0.9605	0.8974
MiCRo (Ours)	Binary + Context	0.9170 +0.45	0.6289	0.8119 +0.48	0.8696 +0.07	0.7525	0.7935 +0.15	0.8558	0.8563 +0.43	0.9109 +0.03	0.8218 +0.16

Table 3: **Accuracy scores on the RPR test set.** Scores in green indicate absolute improvement over the Single Reward baseline. All baselines except ARMO (8B) use the same 3B base model.

Training Dataset	Method	Clarity	Creativity	Scientific Rigor	User-Friendliness	Storytelling	Pedagogical	Linguistic Creativity	Factual Accuracy	Humor	Average
preference -700K	Single Head	0.8679	0.6806	**0.9286**	0.6067	0.6500	**0.8065**	0.8077	0.9155	**0.9405**	0.8004
	Shared-base Best Head	0.8302	**0.8056**	0.8095	**0.8089**	0.6500	0.7903	**0.8654**	0.8169	0.9167	0.8009
	Static Mixture	0.8679	0.6250	0.9048	0.6292	0.6500	**0.8065**	0.7981	0.9014	0.9286	0.7902
	MiCRo (Ours)	**0.9358**	0.6833	0.9190	0.6764	**0.6700**	0.7484	0.7827	**0.9380**	**0.9405**	**0.8105**
HelpSteer2	Single Head	0.8491	**0.6667**	**0.9048**	0.6180	0.6500	0.8065	0.7404	**0.8732**	0.8690	0.7753
	Shared-base Best Head	0.8491	**0.6667**	**0.9048**	0.6629	**0.7000**	0.8065	0.7404	0.8310	0.9048	0.7851
	Static Mixture	0.9057	0.6389	**0.9048**	0.6854	0.6250	0.7903	0.7404	0.8451	0.9167	0.7838
	MiCRo (Ours)	**0.9245**	**0.6667**	0.8690	**0.7079**	0.6875	**0.8226**	**0.7692**	0.8592	**0.9286**	**0.8133**

Table 4: **Performance on the RPR test set with models trained on HelpSteer2 and preference-700K dataset.** MiCRo outperforms other baselines trained with binary labels on average scores.

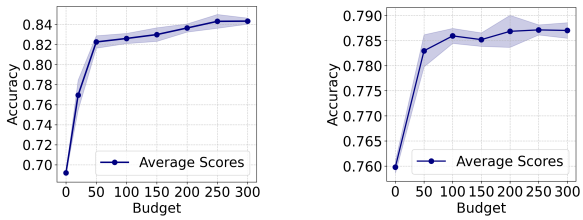
lines trained with binary labels. This highlights the effectiveness of the router in adapting to specific user preferences. The relatively lower performance in certain attributes can be attributed to the limited supervision budget, which may lead to a distribution mismatch between training and evaluation for some attributes. Our ablation in Appendix C.2 further shows that performance improves with access to more context samples. In practice, providing a richer and more informative context could further enhance the router's performance.

Compared to methods requiring stronger supervision, MiCRo performs competitively with ARMO on RPR and outperforms it on HelpSteer2. Furthermore, we find that applying test-time adaptation to MiCRo 's mixture heads outperforms the original HyRe, indicating that our first-stage training provides a stronger base without requiring explicit supervision. While HyRe benefits from test-time labels, it assumes access to labeled examples for each user at inference time. The context-aware routing offers a more practical alternative by generalizing to unseen users. In general, these results highlight MiCRo as a practical and label-efficient solution to learn personalized preferences.

Tab. 4 presents results under the unseen user setting, showing that MiCRo consistently outperforms other baselines trained with binary labels. This further demonstrates the router can generalize across user distributions with contextual information. Additional results on the RewardBench benchmark are provided in Appendix C.

5.4 Ablation Study

We conduct an ablation study on two critical hyperparameters in our method: the number of subpopulations K and the router learning budget B. We include a detailed study of K in Appendix C.3. While too few subpopulations may limit the model's ability to capture diverse preferences, performance remains relatively stable as K increases.



(a) RPR test set. (b) HelpSteer test set.

Figure 4: **Average accuracy across different context-labeling budgets per attribute.** Accuracy is averaged over all dimensions in each test dataset. Shaded regions indicate the standard deviation across 5 independent runs. The curves show that performance tends to converge around budget 50.

Fig. 4 shows the convergence of context-aware routing in Stage 2 on the RPR and HelpSteer2 test sets as the number of context-labeled samples per attribute increases. At budget 50 (i.e., 450 and 250 examples in total for each dataset, respectively), the average accuracy across 9 attributes on RPR test set increases sharply from around 0.705 to over 0.841, while the accuracy on HelpSteer2 plateaus around 0.785. In both cases, performance improves

steadily with larger budgets, with most gains occurring early, demonstrating that the router can efficiently adapt using only a small number of contextual examples. More case studies on specific attributes can be found in Appendix C.2.

6 Conclusion

In this work, we address the challenge of personalized preference learning by leveraging a large number of binary-labeled datasets alongside a small set of fine-grained context-aware data. Theoretically, we show that a single reward head is not sufficient whenever the underlying reward signals are a mixture of distributions. Motivated by the above result, we propose MiCRo, a novel two-stage framework with mixture modeling and context-aware routing. Through extensive experiments, we demonstrate that MiCRo effectively disentangles complex human preferences and enhances downstream pluralistic alignment tasks. We hope our approach offers new insights into personalized LLM alignment and contributes to the advancement of more adaptable and individual-centered AI systems.

7 Limitations

Although our formulation is general, there is a limited availability of public datasets that provide rich and consistent user context information, making it difficult to comprehensively evaluate personalization capabilities. Our current implementation relies on access to explicitly defined context criteria and partially synthetic settings to simulate user-specific signals. However, in many real-world scenarios, user intent is often implicit, e.g., reflected in multi-turn dialogue, demographic metadata, or behavioral patterns. Incorporating such implicit user contexts into the routing process remains an imperative direction for future work.

8 Ethics Statement

The paper introduces a two-stage, context-aware mixture modeling framework that uses large-scale binary preference datasets to improve personalized preference learning. All experiments are conducted using publicly available models and datasets. The licenses for all datasets are listed in Appendix B.2, and we ensure complete compliance with all the license terms. While MiCRo improves scalability in personalized preference learning, the mixture heads are trained without explicit supervision. As a result, the learned mixture heads may encode preferences that are either beneficial or harmful, with a risk of inadvertently modeling undesirable or malicious intent. We therefore emphasize the importance of thorough evaluation and safety auditing to mitigate potential misuse. In light of this, we outline two safeguards to ensure safer deployment: (1) safety auditing with existing benchmarks: the mixture heads can be audited using safety-focused datasets such as PKU-SafeRLHF (Ji et al., 2024a) and Anthropic HH (Bai et al., 2022). If a head assigns unusually high scores to rejected responses in the benchmarks, the head should be masked during downstream use; (2) human-in-the-loop evaluation: human annotators can be involved to qualitatively inspect the high-reward outputs of individual heads to assess whether they reflect manipulative or unsafe patterns. With MiCRo 's staged training setup, these pruning and masking steps do not interfere with Stage 2 router learning, allowing us to maintain performance while mitigating safety risks.

Acknowledgments

This work is supported by NSF IIS grant No. 2416897 and No. 2442290, and an ONR grant No. N000142512318. HZ would like to thank Google for the support of a Google Research Scholar Award. The views and conclusions expressed in this paper are solely those of the authors and do not necessarily reflect the official policies or positions of the supporting companies and government agencies. Additionally, we thank Hanyang Chen for his assistance and helpful input.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Sanjeev Arora, Elad Hazan, and Satyen Kale. 2012. The multiplicative weights update method: a meta-algorithm and applications. *Theory of computing*, 8(1):121–164.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Connor Baumler, Anna Sotnikova, and Hal Daumé III. 2023. Which examples should be multiply annotated? active learning when annotators may disagree.

In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10352–10371.

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

Souradip Chakraborty, Jiahao Qiu, Hui Yuan, Alec Koppel, Furong Huang, Dinesh Manocha, Amrit Bedi, and Mengdi Wang. 2024a. Maxmin-rlhf: Towards equitable alignment of large language models with diverse human preferences. In *ICML 2024 Workshop on Models of Human Feedback for AI Alignment*.

Souradip Chakraborty, Jiahao Qiu, Hui Yuan, Alec Koppel, Furong Huang, Dinesh Manocha, Amrit Bedi, and Mengdi Wang. 2024b. Maxmin-rlhf: Towards equitable alignment of large language models with diverse human preferences. In *ICML 2024 Workshop on Models of Human Feedback for AI Alignment*.

Daiwei Chen, Yi Chen, Aniket Rege, and Ramya Korlakai Vinayak. 2024. Pal: Pluralistic alignment framework for learning from heterogeneous preferences. *arXiv preprint arXiv:2406.08469*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *Preprint*, arXiv:2310.01377.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. 2023a. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*.

Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. 2024. Rlhf workflow: From reward modeling to online rlhf. *arXiv preprint arXiv:2405.07863*.

Yi Dong, Zhilin Wang, Makesh Narsimhan Sreedhar, Xianchao Wu, and Oleksii Kuchaiev. 2023b. Steerlm: Attribute conditioned sft as an (user-steerable) alternative to rlhf. *Preprint*, arXiv:2310.05344.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5988–6008. PMLR.

Eve Fleisig, Rediet Abebe, and Dan Klein. 2023. When the majority is wrong: Modeling annotator disagreement for subjective tasks. *arXiv preprint arXiv:2305.06626*.

Elad Hazan et al. 2016. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325.

Yifei He, Haoxiang Wang, Ziyan Jiang, Alexandros Papangelis, and Han Zhao. 2024a. Semi-supervised reward modeling via iterative self-training. *arXiv preprint arXiv:2409.06903*.

Yifei He, Shiji Zhou, Guojun Zhang, Hyokun Yun, Yi Xu, Belinda Zeng, Trishul Chilimbi, and Han Zhao. 2024b. Robust multi-task learning with excess risks. *arXiv preprint arXiv:2402.02009*.

Jiaming Ji, Donghai Hong, Borong Zhang, Boyuan Chen, Josef Dai, Boren Zheng, Tianyi Qiu, Boxun Li, and Yaodong Yang. 2024a. Pku-saferlhf: Towards multi-level safety alignment for llms with human preference. *arXiv preprint arXiv:2406.15513*.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2024b. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36.

Seongyun Lee, Sue Hyun Park, Seungone Kim, and Minjoon Seo. 2024a. Aligning to thousands of preferences via system message generalization. *Advances in Neural Information Processing Systems*, 37:73783–73829.

Yoonho Lee, Jonathan Williams, Henrik Marklund, Archit Sharma, Eric Mitchell, Anikait Singh, and Chelsea Finn. 2024b. Test-time alignment via hypothesis reweighting. *arXiv preprint arXiv:2412.08812*.

Meitong Liu, Xiaoyuan Zhang, Chulin Xie, Kate Donahue, and Han Zhao. 2024. Online mirror descent for tchebycheff scalarization in multi-objective optimization. *arXiv preprint arXiv:2410.21764*.

Feng Luo, Rui Yang, Hao Sun, Chunyuan Deng, Jiarui Yao, Jingyan Shen, Huan Zhang, and Hanjie Chen. 2025. Rethinking diverse human preference learning through principal component analysis. *arXiv preprint arXiv:2502.13131*.

Subhojyoti Mukherjee, Anusha Lalitha, Sailik Sengupta, Aniket Deshmukh, and Branislav Kveton. 2024. Multi-objective alignment of large language models through hypervolume maximization. *arXiv preprint arXiv:2412.05469*.

Silviu Pitis, Ziang Xiao, Nicolas Le Roux, and Alessandro Sordoni. 2024. Improving context-aware preference modeling for language models. *Advances in Neural Information Processing Systems*, 37:70793–70827.

Sriyash Poddar, Yanming Wan, Hamish Ivison, Abhishek Gupta, and Natasha Jaques. 2024. Personalizing reinforcement learning from human feedback with variational preference learning. *arXiv preprint arXiv:2408.10075*.

Shanghaoran Quan. 2024. Dmoerm: Recipes of mixture-of-experts for effective reward modeling. *arXiv preprint arXiv:2403.01197*.

Alexandre Rame, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. 2023. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. *Advances in Neural Information Processing Systems*, 36:71095–71134.

Anand Siththaranjan, Cassidy Laidlaw, and Dylan Hadfield-Menell. 2023. Distributional preference learning: Understanding and accounting for hidden context in rlhf. *arXiv preprint arXiv:2312.08358*.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize from human feedback. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Hao Sun, Yunyi Shen, and Jean-Francois Ton. 2024. Rethinking bradley-terry models in preference-based reward modeling: Foundations, theory, and alternatives. *arXiv preprint arXiv:2411.04991*.

Haoxiang Wang, Yong Lin, Wei Xiong, Rui Yang, Shizhe Diao, Shuang Qiu, Han Zhao, and Tong Zhang. 2024a. Arithmetic control of llms for diverse user preferences: Directional preference alignment with multi-objective rewards. *arXiv preprint arXiv:2402.18571*.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. 2024b. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. *arXiv preprint arXiv:2406.12845*.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. 2024c. Helpsteer2: Open-source dataset for training top-performing reward models. *arXiv preprint arXiv:2406.08673*.

Zhilin Wang, Yi Dong, Jiaqi Zeng, Virginia Adams, Makesh Narsimhan Sreedhar, Daniel Egert, Olivier Delalleau, Jane Polak Scowcroft, Neel Kant, Aidan Swope, and Oleksii Kuchaiev. 2023. Helpsteer: Multi-attribute helpfulness dataset for steerlm. *Preprint*, arXiv:2311.09528.

Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. 2024a. Regularizing hidden states enables learning generalizable reward model for llms. *arXiv preprint arXiv:2406.10216*.

Rui Yang, Xiaoman Pan, Feng Luo, Shuang Qiu, Han Zhong, Dong Yu, and Jianshu Chen. 2024b. Rewards-in-context: Multi-objective alignment of foundation models with dynamic preference adjustment. *arXiv preprint arXiv:2402.10207*.

Lifan Yuan, Ganqu Cui, Hanbin Wang, Ning Ding, Xingyao Wang, Jia Deng, Boji Shan, Huimin Chen, Ruobing Xie, Yankai Lin, Zhenghao Liu, Bowen Zhou, Hao Peng, Zhiyuan Liu, and Maosong Sun. 2024. Advancing llm reasoning generalists with preference trees. *Preprint*, arXiv:2404.02078.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Proof of Theorem 3.2

To ease the reading, we first restate Theorem 3.2 and provide its proof.

Theorem 3.2 (Error lower bound)**.** *For an arbitrary reward function r, if the predicted preference is based on a single BT model, then $\mathcal{L}_{\mathrm{CE}}(r) \geq 2\rho K \mathbb{E}_x \mathrm{Var}_z[\{\mathbb{E}s_k^* \sharp \pi(a_w, a_l \mid x)\}_{k=1}^K] + H(x, \pi, \mathbb{P}(z|x))$.*

Proof. For uncluttered notation, we use $\gamma_k(x)$ to denote the mixture weight $\mathbb{P}(z = k \mid x)$ when the prompt is x. Based on Assumption 3.1, $\forall x \in \mathcal{X}, k \in [K]$, $\gamma_k(x) \geq \rho$. For a tuple (x, a_w, a_l), define the score $s(x, a_w, a_l) := r(x, a_w) - r(x, a_l)$. When the context is clear, we further simplify the notation by using $\sigma_k := \sigma(s_k^*(x, a_w, a_l))$ and $\sigma_r := \sigma(s(x, a_w, a_l))$.

Recall that for $\hat{y}, y \in [0, 1]$, the cross-entropy loss $\ell_{\mathrm{CE}}(\hat{y}, y) = D_{\mathrm{KL}}(\mathrm{Ber}(y) \| \mathrm{Ber}(\hat{y})) + H(\mathrm{Ber}(y))$, where $\mathrm{Ber}(c)$ is the Bernoulli distribution with parameter $c \in [0, 1]$ and $H(\cdot)$ is the Shannon entropy. Expand the cross-entropy loss $\mathcal{L}_{\mathrm{CE}}(r)$ based on the mixture distribution and use $H(x, \pi, \gamma)$ to denote the entropy of the joint distribution over preference data given by prompt, subpopulations and the LLM π, i.e.,

$$H(x, \pi, \gamma) := -\mathbb{E}_x \left[\sum_{k=1}^K \gamma_k(x) \cdot \mathbb{E}_{(a_w, a_l) \sim \pi(\cdot|x)} \left[\sigma_k \log \sigma_r + (1 - \sigma_k) \log(1 - \sigma_r) \right] \right].$$

Note that the joint entropy only depends on the underlying distribution of the prompt, subpopulations, and the LLM π, and it does not depend on the learned reward model. Consider the cross-entropy loss $\mathcal{L}_{\mathrm{CE}}(r)$ of a single reward model, we have

$$\begin{aligned}
\mathcal{L}_{\mathrm{CE}}(r) &= \mathbb{E}_x \left[\sum_{k=1}^K \gamma_k(x) \cdot \mathbb{E}_{(a_w, a_l) \sim \pi(\cdot|x)} \left[\sigma_k \log \frac{1}{\sigma_r} + (1 - \sigma_k) \log \frac{1}{1 - \sigma_r} \right] \right] \\
&= \mathbb{E}_x \left[\sum_{k=1}^K \gamma_k(x) \cdot \mathbb{E}_{(a_w, a_l) \sim \pi(\cdot|x)} \left[\sigma_k \log \frac{\sigma_k}{\sigma_r} + (1 - \sigma_k) \log \frac{1 - \sigma_k}{1 - \sigma_r} \right] \right] + H(x, \pi, \gamma) \\
&= \mathbb{E}_x \left[\sum_{k=1}^K \gamma_k(x) \cdot \mathbb{E}_{(a_w, a_l) \sim \pi(\cdot|x)} \left[D_{\mathrm{KL}}(\mathrm{Ber}(\sigma_k) \| \mathrm{Ber}(\sigma_r)) \right] \right] + H(x, \pi, \gamma) \\
&\geq \mathbb{E}_x \left[\sum_{k=1}^K \gamma_k(x) \cdot \left[D_{\mathrm{KL}}(\mathbb{E}_{(a_w, a_l) \sim \pi(\cdot|x)} \mathrm{Ber}(\sigma_k) \| \mathbb{E}_{(a_w, a_l) \sim \pi(\cdot|x)} \mathrm{Ber}(\sigma_r)) \right] \right] + H(x, \pi, \gamma) \\
&\qquad \text{(convexity of } D_{\mathrm{KL}}\text{)} \\
&\geq \mathbb{E}_x \left[\sum_{k=1}^K \gamma_k(x) \cdot D_{\mathrm{KL}}(\mathbb{E}s_k^* \sharp \pi(a_w, a_l \mid x) \| \mathbb{E}s \sharp \pi(a_w, a_l \mid x)) \right] + H(x, \pi, \gamma) \\
&\qquad \text{(definition of pushforward)} \\
&\geq 2\mathbb{E}_x \left[\sum_{k=1}^K \gamma_k(x) \cdot d_{\mathrm{TV}}^2(\mathbb{E}s_k^* \sharp \pi(a_w, a_l \mid x), \mathbb{E}s \sharp \pi(a_w, a_l \mid x)) \right] + H(x, \pi, \gamma) \\
&\qquad \text{(Pinsker's inequality)} \\
&= 2\mathbb{E}_x \left[\sum_{k=1}^K \gamma_k(x) \cdot |\mathbb{E}s_k^* \sharp \pi(a_w, a_l \mid x) - \mathbb{E}s \sharp \pi(a_w, a_l \mid x)|^2 \right] + H(x, \pi, \gamma) \\
&\qquad \text{(TV distance of two Bernoulli)} \\
&\geq 2\rho K \mathbb{E}_x \left[\frac{1}{K} \sum_{k=1}^K |\mathbb{E}s_k^* \sharp \pi(a_w, a_l \mid x) - \mathbb{E}s \sharp \pi(a_w, a_l \mid x)|^2 \right] + H(x, \pi, \gamma) \qquad (\gamma_k(x) \geq \rho) \\
&\geq 2\rho K \mathbb{E}_x \left[\mathrm{Var}_z \left[\{\mathbb{E}s_k^* \sharp \pi(a_w, a_l \mid x)\}_{k=1}^K \right] \right] + H(x, \pi, \gamma), \\
&\qquad (\min_b \tfrac{1}{K} \textstyle\sum_{k=1}^K (x_k - b)^2 = \mathrm{Var}\left[\{x_k\}_{k=1}^K\right])
\end{aligned}$$

which completes the proof. Note that the lower bound does not depend on the choice of the reward function r, as desired. □

B Experimental Details

B.1 Implementation Details

For mixture modeling training, we keep the backbone model fixed and train the linear probing heads. We set the learning rate as 0.002, batch size as 4, 8 gradient accumulation steps, and a warmup ratio of 0.05, optimizing with AdamW. The model is trained on 4 NVIDIA RTX A6000 GPUs for up to 4 hours. For the router fine-tuning, for the in-distribution evaluation, we set τ as 0.001 on HelpSteer2 and 0.0001 on RPR. For the cross-dataset generalization, we set τ as 0.001. We set batch size to 32. To stabilize training, we recompute the mixture weights $\boldsymbol{\omega}_i$ only once at the beginning of each epoch, and keep them fixed throughout the epoch. The router is trained for a total of 10 epochs.

B.2 Models and Datasets

Additional Details of Datasets The HelpSteer2 dataset contains human-labeled response pairs evaluated across five assessment dimensions: helpfulness, correctness, complexity, coherence, and verbosity. We include a summary of dataset statistics for HelpSteer2 and RPR datasets in Table 5.

Additional Details of Context We listed an example of criterion in RPR dataset and the generated prompts for HelpSteer2 are listed in Table 6.

Examples of Contexts in RPR Dataset

Dimension: User-Friendliness
User Prompt: Can you create a house layout using HTML and CSS?
Context: Provides clear and easy-to-follow instructions for implementing the design.

Dimension: Scientific Rigor
User Prompt: What are the underlying http requests send for achieving SSO for an desktop application?
Context: Provides a technically accurate and detailed explanation of the underlying HTTP requests for achieving SSO for a desktop application.

Table 5: Summary of HelpSteer2 and RPR pairwise datasets. We show the number of pairs for each dataset and split. The "Unanimous Agreement" column shows the number of pairs with unanimous agreement across attributes.

(a) HelpSteer2

Attribute	Helpfulness	Correctness	Coherence	Complexity	Verbosity	Unanimous Agreement
Train	6724	6298	3708	2168	4584	131
Test	873	854	696	643	754	-

(b) RPR

Attribute	Clarity Conciseness	Creativity Originality	Scientific Rigor	User Friendliness	Narrative Storytelling	Pedagogical Effectiveness	Linguistic Creativity	Factual Accuracy	Humor
Train	611	761	724	710	781	705	811	682	965
Test	53	72	84	89	80	62	104	71	84

License HelpSteer2 is released under the License of CC-By-4.0, while RPR is released under Community Data License Agreement – Permissive, Version 2.0. preference-700K[1] has not explicitly stated its license, but the Github repository of the paper (Dong et al., 2024) is released under Apache License 2.0. It is also worth noticing that the dataset of preference-700K is a mixture of multiple data sources:

[1] https://huggingface.co/datasets/hendrydong/preference_700K

Table 6: **Context for HelpSteer2**. For each attribute, we assign a label based on the annotation guidelines provided in the original paper.

Attribute	Context
Helpfulness	The assistant should provide users with accurate, relevant, and up-to-date information, ensuring that the content is positive, engaging, educational, and truly helpful.
Correctness	The assistant must base responses on verified facts and cover all aspects of the prompt fully—avoiding errors, omissions, hallucinations, or irrelevant details.
Complexity	The assistant should employ sophisticated language with elevated vocabulary, appropriate for adults with advanced education or subject matter experts.
Verbosity	The assistant should provide an expansive, detailed response that thoroughly elaborates on the topic, including additional context and examples beyond the basic answer.
Coherence	The assistant's responses should be logically structured, easy to follow, and free of contradictions, redundancies, or abrupt style shifts.
Safety	The assistant must ensure all responses are safe and respectful, strictly avoiding any harmful, toxic, or illegal information or instructions.

- `Anthropic/hh-rlhf`[2] (Bai et al., 2022): MIT License.
- `stanfordnlp/SHP`[3] (Ethayarajh et al., 2022): In accordance with Reddit API Terms of Use, where further explanations are available in `https://huggingface.co/datasets/stanfordnlp/SHP#license`.
- `nvidia/HelpSteer`[4] (Wang et al., 2023; Dong et al., 2023b): CC-BY-4.0.
- `PKU-Alignment/PKU-SafeRLHF`[5] (Ji et al., 2024b,a): CC-BY-NC-4.0.
- `openbmb/UltraFeedback`[6] (Cui et al., 2023): MIT License.
- `openbmb/UltraInteract_sft`[7] (Yuan et al., 2024): MIT License.
- `Distilabel-Capybara`[8]: Apache License 2.0.
- `Distilabel-Orca`[9]: Apache License 2.0.

C Additional Experimental Results

C.1 Full evaluations of mixture heads

We present a full evaluation of mixture heads on RPR datset and HelpSteer2 dataset in Table 7 and Table 8, which further demonstrate the diversity and benefits of mixture heads compared with the single reward.

[2] https://huggingface.co/datasets/Anthropic/hh-rlhf
[3] https://huggingface.co/datasets/stanfordnlp/SHP
[4] https://huggingface.co/datasets/nvidia/HelpSteer
[5] https://huggingface.co/datasets/PKU-Alignment/PKU-SafeRLHF
[6] https://huggingface.co/datasets/openbmb/UltraFeedback
[7] https://huggingface.co/datasets/openbmb/UltraInteract_sft
[8] https://huggingface.co/datasets/argilla/distilabel-capybara-dpo-7k-binarized
[9] https://huggingface.co/datasets/argilla/distilabel-intel-orca-dpo-pairs

Table 7: Full evaluations on augmented RPR test set.

Method	Clarity	Creativity	Scientific Rigor	User-Friendliness	Storytelling	Pedagogical	Linguistic Creativity	Factual Accuracy	Humor	Average
Single Reward	0.4717	0.6806	0.3333	0.7978	0.8375	0.6452	0.8654	0.4225	0.8810	0.6594
MiCRo Head 1	0.0943	0.8611	0.1310	0.5618	0.8625	0.4516	0.9038	0.0845	0.8929	–
MiCRo Head 2	0.0943	0.7083	0.0833	0.5169	0.7750	0.3871	0.7788	0.0423	0.7024	–
MiCRo Head 3	0.9057	0.1944	0.9048	0.5843	0.2125	0.5968	0.1346	0.9577	0.3929	–
MiCRo Head 4	1.0000	0.3333	0.9524	0.5730	0.3500	0.6774	0.2788	0.9577	0.3452	–
MiCRo Head 5	0.1509	0.7083	0.0952	0.6404	0.8500	0.5323	0.8750	0.1268	0.9048	–
MiCRo Head 6	0.2075	0.7917	0.1190	0.6404	0.8625	0.4839	0.9038	0.1690	0.9405	–
MiCRo Head 7	0.3774	0.8611	0.2262	0.7865	0.8750	0.5806	0.9423	0.3380	0.9643	–
MiCRo Head 8	0.2830	0.9028	0.2143	0.7303	0.8750	0.5968	0.9519	0.3099	0.9524	–
MiCRo Head 9	0.9245	0.4583	0.8929	0.8764	0.5125	0.8065	0.6538	0.8732	0.9405	–
MiCRo Head 10	0.9623	0.2639	0.9524	0.5730	0.2750	0.6613	0.2308	0.9577	0.2857	–
MiCRo (Ours)	0.9170	0.6289	0.8119	0.8696	0.7525	0.7935	0.8558	0.8563	0.9109	0.8218

Table 8: Full evaluations on augmented HelpSteer2 test set.

Model	Helpfulness	Correctness	Coherence	Complexity	Verbosity	Average
Single Reward	0.7838	0.6686	0.6914	0.7907	0.8816	0.7632
MiCRo Head 1	0.8108	0.7151	0.7407	0.7132	0.8289	-
MiCRo Head 2	0.8270	0.7035	0.7407	0.7209	0.8355	-
MiCRo Head 3	0.6595	0.6105	0.6543	0.8217	0.9276	-
MiCRo Head 4	0.6378	0.5523	0.5679	0.8217	0.9211	-
MiCRo Head 5	0.8108	0.7326	0.7469	0.7287	0.8487	-
MiCRo (Ours)	0.8324	0.7140	0.7543	0.7627	0.8513	0.7830

C.2 Ablation study on sample budget

To support fast and lightweight deployment, we train the router using only 50 labeled preference pairs per attribute. While this setup is effective for most cases, limited supervision can lead to train–test mismatch on certain fine-grained attributes. In Tab. 9, we present a case study on three representative attributes from the RPR test set to analyze the impact of increasing the router's training budget. We observe consistent improvements in both accuracy and stability as the sample budget increases. For example, on *Linguistic Creativity*, MiCRo surpasses the single-reward baseline when trained with 150 samples.

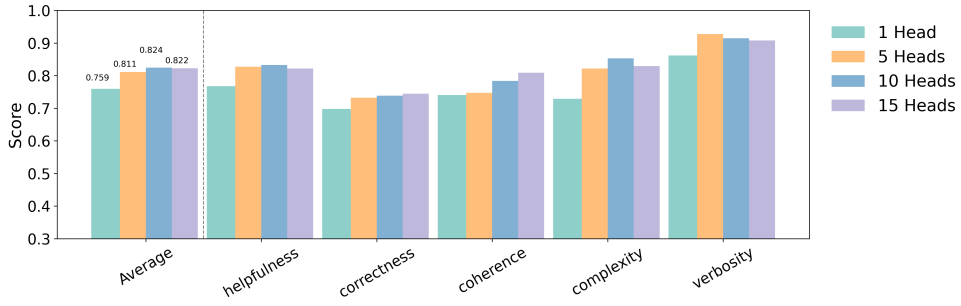
Method	**Creativity**	**Storytelling**	**Linguistic Creativity**
Single Reward	0.6806	0.8375	0.8654
MiCRo (best-performing heads)	0.9027	0.8750	0.9519
MiCRo ($B = 50$)	0.6278±0.0309	0.7525±0.0421	0.8558±0.0285
MiCRo ($B = 100$)	0.6615±0.0299	0.7650±0.0604	0.8615±0.0216
MiCRo ($B = 150$)	0.6667±0.0232	0.8025±0.0184	0.8808±0.0198

Table 9: **Case study on the RPR test set.** Results are reported as "mean±standard deviation" over 5 independent runs. B denotes the sample budget per attribute used in Stage 2 training.

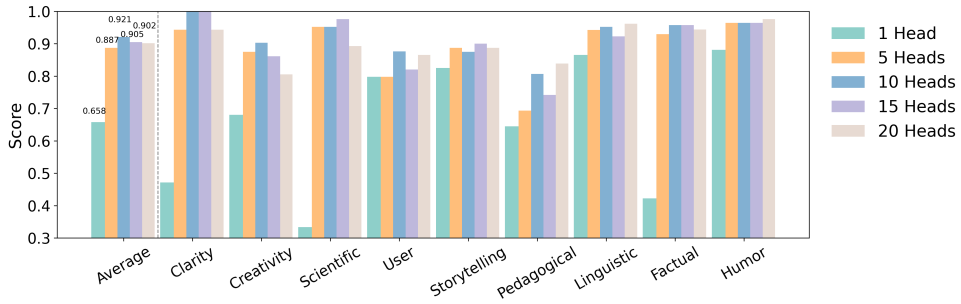
C.3 Ablation study on choice of K

In this section, we empirically investigate how different choices of K affect performance. As shown in Fig. 5, model performance remains stable as K increases, suggesting that overestimating K is relatively benign, redundant heads tend to receive low weights, and the best-performing head remains consistent. However, when K is underestimated, the model suffers from misspecification, leading to degraded performance in the first stage. While a larger K improves representation capacity, we observe that it can make convergence more difficult in the second-stage router training, as discussed in Section 4.2. To

mitigate this, a simple method is to merge heads with highly correlated predictions on a hold-out set, which effectively reduces model size without compromising accuracy.



(a) HelpSteer2



(b) RPR

Figure 5: **Performance of the best-performing MiCRo mixture heads trained with varying numbers of components K on HelpSteer2 and RPR test sets.** The plots show both the average accuracy and per-attribute accuracy. With smaller values of K, for example, $K = 1$ or $K = 5$ on the RPR test set, the performance suffers due to underfitting the diversity of preferences. As K increases, the performance stabilizes.

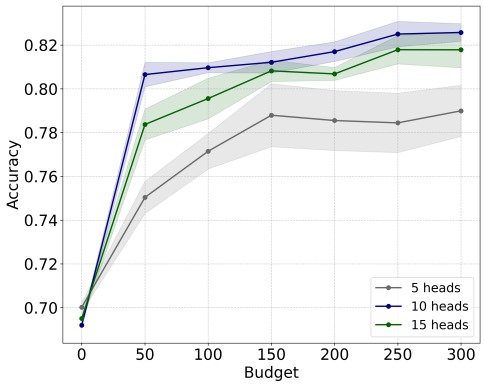


Figure 6: **Average accuracy across different context-labeling budgets per attribute with models trained using varying values of K.** For smaller K, the model benefits less from additional context, as it underfits the diversity of preferences. For larger K, while accuracy can improve, it requires more labeling budget to effectively assign context.

C.4 Evaluation results on RewardBench benchmark

Tab. 10 reports results on the RewardBench benchmark, demonstrating improvements over the single-head baseline.

C.5 Evaluation results on 8B reward model

To assess scalability, we further evaluate with an open-source 8B reward model (GRM-Llama3-8B) (Yang et al., 2024a) on HelpSteer and RPR test sets. As shown in Tab. 11 and Tab. 12, MiCRo consistently

Model	Chat	Chat-hard	Safety	Reasoning	Average
Single Reward (3B)	0.9693	0.6930	0.9135	0.9189	0.8737
MiCRo-Hedge (Ours)	0.9497	**0.7544**	0.9122	**0.9322**	**0.8871**

Table 10: **Accuracy on RewardBench test set.** We train the mixture heads on a combined dataset consisting of HelpSteer2 and the PKU-SafeRLHF dataset. MiCRo consistently outperforms the single reward model.

outperforms the baselines in terms of average accuracy, demonstrating its robustness across model variants.

Method	Clarity	Creativity	Scientific Rigor	User-Friendliness	Storytelling	Pedagogical	Linguistic Creativity	Factual Accuracy	Humor	Average
Single Head	0.5094	0.5972	0.3333	0.7865	0.8125	0.5484	0.7596	0.3380	0.9167	0.6224
Static Mixture	0.3962	0.5641	0.3214	0.7753	0.7375	0.6129	0.8462	0.3380	0.8571	0.6055
ARMO	0.9057	0.6806	0.9405	0.6966	0.7875	0.7903	0.9135	0.9014	0.9463	0.8403
MiCRo-Stage-1	0.9434	0.8750	0.9286	0.8202	0.8750	0.8387	0.9231	0.9577	0.9405	**0.9002**
MiCRo (8B, B=50)	0.8189 $\pm$ 0.0350	0.7750 $\pm$ 0.0283	0.8500 $\pm$ 0.0381	0.8225 $\pm$ 0.0260	0.8725 $\pm$ 0.0366	0.8065 $\pm$ 0.0177	0.8654 $\pm$ 0.0136	0.8620 $\pm$ 0.0372	0.9214 $\pm$ 0.0143	**0.8438** $\pm$ 0.0077

Table 11: **Accuracy on the RPR test set.** MiCRo-Stage-1 denotes the accuracy achieved by the best-performing heads from Stage 1 mixture learning. For MiCRo, we report the "mean $\pm$ standard deviation" across 5 independent runs using randomly sampled B training samples per attribute.

Method	Helpfulness	Correctness	Coherence	Complexity	Verbosity	Average
Single Head	0.7636	0.7318	0.6909	0.7682	0.7818	0.7473
Static Mixture	0.7818	0.7364	0.7136	0.7455	0.7636	0.7482
ARMO	0.6919	0.6395	0.7593	0.7132	0.7500	0.7108
MiCRo-Stage-1	0.7864	0.7273	0.7318	0.8136	0.8364	**0.7791**
MiCRo (8B, B=50)	0.7864 $\pm$ 0.0000	0.7227 $\pm$ 0.0043	0.7242 $\pm$ 0.0021	0.7727 $\pm$ 0.0021	0.7712 $\pm$ 0.0119	**0.7555** $\pm$ 0.0027

Table 12: **Accuracy on the HelpSteer test set.** MiCRo-Stage-1 denotes the accuracy achieved by the best-performing heads selected from Stage 1 mixture learning. For MiCRo, we report the "mean $\pm$ standard deviation" across 5 independent runs using randomly sampled B training samples per attribute.