PERSONALIZED LANGUAGE MODELING FROM PERSONALIZED HUMAN FEEDBACK

Anonymous authors
Paper under double-blind review

ABSTRACT

Personalized large language models (LLMs) are designed to tailor responses to individual user preferences. While Reinforcement Learning from Human Feedback (RLHF) is a commonly used framework for aligning LLMs with human preferences, vanilla RLHF assumes that all human preferences share the same distribution, preventing fine-tuned LLMs from generating personalized content when user preferences are diverse. In this work, we propose Personalized-RLHF (P-RLHF), an efficient framework that utilizes a lightweight user model to capture individual user preferences and jointly learns the user model and the personalized LLM from human feedback. P-RLHF exhibits the following three characteristics: (1) It enables an LLM to generate personalized content and scale efficiently with growing number of users. (2) It handles both explicit user preferences described as textual input and implicit user preferences encoded in the feedback data. (3) It eliminates the need for users to fully articulate their preferences, which are normally needed for prompting LLMs to generate personalized content yet are often impractical to obtain in real-world scenarios. Our experimental results show that personalized LLMs trained using P-RLHF generate responses that are more closely aligned with individual user preferences, outperforming vanilla, non-personalized RLHF and prompting-based personalization approaches across different tasks.

1 INTRODUCTION

Personalization aims to generate tailored responses or recommendations to meet the unique preferences of individual users, based on user information (e.g. demographic or interests) or their historical data (Chen, 2023). It enhances user experience and engagement, making it crucial in a wide range of domains including recommendation systems (Li et al., 2023b), chatbots (Ma et al., 2021), healthcare (Kadariya et al., 2019), and education (Maghsudi et al., 2021). Large language models (LLMs) (Brown et al., 2020; Chowdhery et al., 2022; Dubey et al., 2024) have demonstrated exceptional capabilities in text generation, reasoning, and instruction following, leading to their use in various real-world user-facing applications. As a result, personalizing LLMs to align with individual user preferences has become a key research topic (Li et al., 2023a).

Reinforcement Learning from Human Feedback (RLHF) is a widely adopted framework to align pre-trained LLMs with human preferences (Ziegler et al., 2019), by fine-tuning LLMs using human feedback data in the form of preference comparisons or rankings over multiple generations. However, standard RLHF approaches *implicitly* assume that all human preferences come from the same distribution (Ziegler et al., 2019; Stiennon et al., 2020; Ouyang et al., 2022; Rafailov et al., 2023), limiting the ability of LLMs fine-tuned under such assumption to generate personalized responses when user preferences encoded in human feedback are diverse or conflicting (Kirk et al., 2023). Recent endeavors in developing RLHF-based (Wu et al., 2023; Jang et al., 2023) methods for personalizing LLM outputs often require training separate reward models or LLMs for each preference dimension (such as completeness, friendliness etc.), posing computational and storage challenges, particularly in settings with large user bases that exhibit diverse and multifaceted preferences. Additionally, these methods rely on predefined preference dimensions, limiting their flexibility, as it is often impractical to exhaustively enumerate all user preference dimensions in real-world scenarios.

To build *efficient* and *flexible* personalized LLMs, we introduce the setting for Learning from Personalized Human Feedback (Section 4), which leverages both user information in textual form

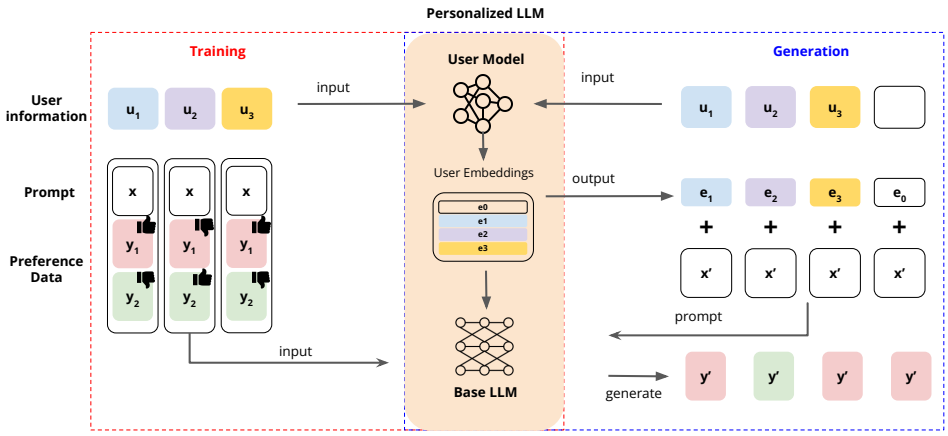

Figure 1: Our **Personalized RLHF** framework. A personalized LLM (highlighted in orange) consists of two key components: a **learnable user model** and a **base LLM** (introduced in Section 4.2). For training, the user information u_i and the preference data are collected from each user (in this example there are 3 users $i = 1, 2, 3$). The user model maps the user information into user embeddings (user-specific embeddings e_i and the generic embedding e_0 that captures the common preferences shared across users), which are learned jointly with the base LLM using a new P-RLHF learning objective (derived in Section 4.4). During generation, for seen users, the responses tailored to their individual preferences are generated based on the learned user embeddings (e_i), while for new users unseen during training, responses are generated using the generic embedding (e_0).

and historical feedback data in preference form. We begin with formalizing the deficiency of vanilla RLHF (Section 3) in personalization, then move to proposing a general *personalized RLHF (P-RLHF)* framework, as shown in Figure 1. Our proposed framework employs a *lightweight* user model to capture both *explicit* preferences from user information and *implicit* preferences from feedback data. This is particularly beneficial when it is difficult to fully describe user preferences using pre-defined dimensions or text, as our design allows missing information to be inferred flexibly from feedback data which enables a more comprehensive understanding of user preferences.

To instantiate our framework, we discuss how different assumptions on user preferences can influence the design of the user model (Section 4.3). P-RLHF learns the user model and the LLM jointly through new learning objectives we develop for performing personalized Direct Preference Optimization (P-DPO, section 4.4). By incorporating a user model, P-RLHF eliminates the need for training separate reward models or LLMs, enabling efficient and scalable personalization across large number of users. On three tasks using publicly available preference datasets—synthetic generation with conflicting preferences, synthetic instruction following with diverse user profiles, and a real-world conversation task with $1,500$ users—we demonstrate that P-DPO effectively aligns LLM behavior with individual user preferences and scales efficiently with large user bases (Section 5).

2 Related Work

Reinforcement Learning from Human Feedback RLHF optimizes LLMs as RL policies to generate responses aligned with human preferences (Stiennon et al., 2020; Ouyang et al., 2022; Bai et al., 2022). RLHF training involves either learning a reward model from the preference data and then optimizing the LLM against the learned reward model using proximal policy optimization, or directly optimizing the LLM using the preference data through methods like Direct Preference Optimization (DPO) (Rafailov et al., 2023), with the latter offering significant improvement in training efficiency. Vanilla RLHF methods implicitly assume user preferences uniformity, overlooking inter-user diversity and consequently limiting fine-tuned LLMs' ability to generate personalized content tailored to individual user preferences, especially when the often impractical explicit specification of user preferences are not provided to the model.

To introduce personalization in RLHF, recent studies have proposed learning separate reward models or LLM policies for different preference dimensions, then personalizing LLM outputs by customizing reward weights (Wu et al., 2023) or merging LLMs based on specific preference choices (Jang et al.,

2023). Our work differs from these previous studies in two key ways: (1) our personalized LLMs are directly learned from user information and personalized feedback data, without relying on pre-defined preference dimensions; and (2) we do not require multiple LLMs or reward models, instead using only a small user model to augment the base LLM. Concurrently, a different research direction to address the diversity in user preferences focuses on learning LLM policies that perform robustly across different user groups, using methods such as group invariant learning (Zheng et al., 2023) or distributionally robust optimization (Chakraborty et al., 2024). Unlike our approach, which generates personalized content tailored to individual user preferences, these methods do not personalize the LLM but instead focus on enabling it to generate content that minimizes performance discrepancies between user groups from a fairness perspective.

Prompt-based LLM Personalization In addition to RLHF-based approaches, prompt-based LLM personalization focuses on developing prompting techniques that enable LLMs to capture individual user preferences and tailor their outputs accordingly. This typically involves incorporating historical user-generated content as few-shot examples in the prompt, allowing LLMs to generate personalized content through in-context learning (Dai et al., 2023; Kang et al., 2023). Recent studies have further improved this approach by combining retrieval techniques to construct prompts with relevant user data (Salemi et al., 2023; 2024; Yang et al., 2023; Li et al., 2023c) and augmenting prompts with user information summaries (Richardson et al., 2023). Our work complements prompt-based LLM personalization. While prompt-based methods utilize user-generated content, such as user-written text or selected items, we focus on personalizing LLMs using preference data in the form of comparisons or rankings, a common form of feedback collected from end-users that supplements user-generated content and captures implicit user preference. As a result, prompt-based benchmarks such as LaMP (Salemi et al., 2023) are not directly applicable to our method.

Due to space constraints, additional related work including crowdsourcing and conditional natural language generation are discussed in Appendix A.

3 VANILLA RLHF

We briefly go over the vanilla RLHF pipeline including DPO and reflect on their deficiency in personalization. In vanilla RLHF, there are three steps (Ziegler et al., 2019; Ouyang et al., 2022): (1) obtain a supervised fine-tuned (SFT) policy (denoted as π^{SFT}) using a demonstration dataset; (2) learn a Reward Model (RM) using a preference dataset; and (3) optimize the LLM against the learned reward model using policy optimization methods, e.g., proximal policy optimization (PPO) Schulman et al. (2017). Uncovering a reparametrization of the optimal LM under the learned RM and the RL objective, DPO directly optimizes the LLM using a preference dataset (Rafailov et al., 2023).

Vanilla RLHF via Reward Modeling The vanilla reward learner has access to a *preference* dataset $\mathcal{D} = \{(x_i, y_{i,1}, y_{i,2})\}_{i=1}^n$. In each sample, x_i is the prompt, $y_{i,1}$ and $y_{i,2}$ are two generated texts such that $y_{i,1}$ is preferred over $y_{i,2}$ (i.e., $y_{i,1} \succ y_{i,2}$) under the prompt x_i. A reward model that maps a tuple (x, y) of prompt x and generated text y to a scalar is learned through:

$$r_{\text{vanilla}} \in \arg\min_r -\mathbb{E}_{x,y_1,y_2\sim\mathcal{D}}[\log \sigma(r(x, y_1) - r(x, y_2))], \tag{1}$$

where σ is the sigmoid function and the minimization is over all measurable functions. As noted in Zhu et al. (2023); Rafailov et al. (2023), the underlying assumption for using equation 1 to learn the reward model r_{vanilla} is that the user preferences follow the Bradley-Terry (BT) model (Bradley & Terry, 1952). In other words, the vanilla RM r_{vanilla} is the maximum likelihood estimator on the dataset $\mathcal{D}$ under the assumption: for all prompt x and generated texts y_1, y_2, user preferences follow

$$\mathbb{P}(y_1 \succ y_2|x) = \frac{\exp(r(x, y_1))}{\exp(r(x, y_1)) + \exp(r(x, y_2))} = \sigma(r(x, y_1) - r(x, y_2)). \tag{2}$$

Once r_{vanilla} is learned, the LLM policy π_{vanilla} is learned by maximizing the rewards under a KL-divergence penalty which controls the deviance between the learned LLM and the SFT π^{SFT}:

$$\pi_{\text{vanilla}} \in \arg\max_\pi \mathbb{E}_{x\sim\mathcal{D},y\sim\pi(\cdot|x)}[r_{\text{vanilla}}(x, y)] - \beta\mathbb{E}_{x\sim\mathcal{D}}[\text{KL}(\pi(\cdot|x), \pi^{\text{SFT}}(\cdot|x))], \tag{3}$$

where KL is short-handed for the Kullback–Leibler divergence and $\beta > 0$ is a tunable parameter controlling the strength of the penalty.

Vanilla DPO DPO is an alternative to RM-based RLHF approaches. As noted in Rafailov et al. (2023), given any RM r, its corresponding optimal policy under (equation 3) can be written as

$$\pi(y|x) = \frac{1}{Z(x)}\pi^{\text{SFT}}(y|x)\exp\left(\frac{r(x,y)}{\beta}\right), \tag{4}$$

where $Z(x)$ is a generated-text-independent (or y-independent) normalizing factor. Plugging equation 4 into the reward objective (equation 1), we obtain the following way of obtaining π_{vanilla}:

$$\pi_{\text{vanilla}} \in \arg\min_{\pi} -\mathbb{E}_{x,y_1,y_2\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi(y_1|x)}{\pi^{\text{SFT}}(y_1|x)} - \beta\log\frac{\pi(y_2|x)}{\pi^{\text{SFT}}(y_2|x)}\right)\right], \tag{5}$$

where $\mathcal{D}$ is the preference data given in equation 1. Under this reparametrization, the corresponding vanilla RM r_{vanilla} can be written as $r_{\text{vanilla}}(x,y) = \beta\log\frac{\pi_{\text{vanilla}}(y|x)}{\pi^{\text{SFT}}(y|x)} + \beta\log Z(x)$. In the following, we reflect on the underlying assumption about user preferences in vanilla RLHF and highlight the limitations of LLMs fine-tuned under such assumption for personalized content generation.

3.1 Motivation for personalized RLHF: Undesirable Assumption on User Preferences in Vanilla RLHF

We study the behavior and underlying assumption of r_{vanilla} that is either learned explicitly through the reward modeling step (equation 1) or implicitly through DPO (equation 5). We show that the corresponding assumption is particularly problematic when users have diverse or conflicting preferences. The proofs for this section are in Appendix B.

As in Ziegler et al. (2019), often times, the reward learner has access to identifier information $u \in \mathcal{U}$ of the user who provides their preferences (and annotations), in addition to the prompt and generated texts (x, y_1, y_2). In vanilla RLHF, while we make the explicit assumption that user preferences follow a BT model (equation 2), we often ignore the implicit assumption we make on *preference uniformity*:

Assumption 3.1 (Preference Uniformity)**.** In vanilla reward modeling and DPO, the user preferences are assumed to be uniform, i.e., for all $u \in \mathcal{U}$,

$$\mathbb{P}(y_1 \succ y_2|x,u) = \mathbb{P}(y_1 \succ y_2|x). \tag{6}$$

This assumption may be reasonable when our goal is to uncover certain preferences that are common across different users, concerning topics like factuality and safety. In settings where user preferences are diverse (e.g., on styles of generated texts), this assumption may be undesirable. We showcase this by first analyzing how r_{vanilla} behaves on the training dataset, and then discussing general problems with the Preference Uniformity Assumption 3.1.

Lemma 3.2. *[$r_{vanilla}$ is equivalent to majority voting] For all $i \in [n]$, the estimated user preference under $r_{vanilla}$ is given by*

$$\mathbb{P}(y_{i,1} \succ y_{i,2}|x_i) = \sigma(r_{vanilla}(x_i,y_{i,1}) - r_{vanilla}(x_i,y_{i,2})) = \frac{\sum_{j\in[\mathcal{C}_i]}\mathbb{I}\{y_{j,1}=y_{i,1}\}}{|\mathcal{C}_i|},$$

where $\mathcal{C}_i = \{j\in[n]|x_j = x_i, y_{j,1}=y_{i,1}, y_{j,2}=y_{i,2}\} \cup \{j\in[n]|x_j=x_i, y_{j,1}=y_{i,2}, y_{j,2}=y_{i,1}\}$ is the set of sample indices that share the same prompt and response pairs as x_i.

The above lemma, though straightforward, showcases one of the fundamental problems with r_{vanilla}. That is, it induces a majority voting regime where responses preferred by the majority are assumed to be preferred by all users. In the personalization setting where diversity in preferences matters, such a majority-voting scheme may silence the preferences of the minority communities. In the worst case where the preferences of the majority and minority groups conflict, the LLM's generations may be entirely misaligned with what the minority users prefer.

Reflecting more on the Preference Uniformity Assumption (3.1), we find that under this assumption, when there is a minority and a majority group that differ in their preferences, the minority group will necessarily suffer more in the sense that their true preference $\mathbb{P}(y_1 \succ y_2|x,u_{\text{minority}})$ deviates from the assumed uniform preference $\mathbb{P}(y_1 \succ y_2|x)$ more than that for $\mathbb{P}(y_1 \succ y_2|x,u_{\text{majority}})$. In addition, this deviance increases as the size of the majority group increases.

Lemma 3.3. *When* $\mathbb{P}(u_{majority}) \geq \mathbb{P}(u_{minority})$*, we have that* $|\mathbb{P}(y_1 \succ y_2|x) - \mathbb{P}(y_1 \succ y_2|x, u_{minority})| > |\mathbb{P}(y_1 \succ y_2|x) - \mathbb{P}(y_1 \succ y_2|x, u_{majority})|$*. In addition, as the majority group size increases, the minority group deviates from the assumed uniform preference more, i.e.,* $|\mathbb{P}(y_1 \succ y_2|x) - \mathbb{P}(y_1 \succ y_2|x, u_{minority})|$ *is monotonically increasing with respect to* $\mathbb{P}(u_{majority})$.

Lemma 3.2 and 3.3 showcase that r_{vanilla}, obtained under vanilla reward modeling (equation 1) or vanilla DPO (equation 5), may be unsuitable when user preferences are diverse. In the following, we propose methods for Personalized RLHF to capture individual user preferences which enables LLMs learned under such framework to generate personalized content tailored to each user (Section 4.2). Below we first formally define the task of learning from personalized feedback.

4 Learning from Personalized Human Feedback

4.1 Personalized LLM: Problem Setup

We first formally define the learning setup when given a *personalized preference* dataset. A personalized human feedback (or preference) dataset $\mathcal{D}_{\text{p}} = \{(x_i, y_{i,1}, y_{i,2}, u_i)\}_{i=1}^n$ consists of n samples where $u_i \in \mathcal{U}$ is the information of the user who annotates the data or provides the preferences, x_i is the prompt, $y_{i,1}$ and $y_{i,2}$ are two generated texts such that $y_{i,1} \succ y_{i,2}$ under the user's preference. We consider cases where $u_i = (u_i^t, u_i^p)$ is the user information: u_i^t is their (optional) textual information, e.g., demographic data or user preference descriptions, and u_i^p is the unique user identifier (e.g., an assigned annotator or user id). For new, unknown user, their identifier is set to $u_i^p = u_0^p$ and their user textual information u_i^t is optional.

A personalized LLM π_{p} takes in a prompt x and the user information $u \in \mathcal{U}$ and customizes its text generation based on user u's personal preference (explicitly specified in u_i^t or implicitly encoded in their feedback data), i.e., $y \sim \pi_{\text{p}}(\cdot|x, u)$. When there is no textual information, i.e., $u^t = ()$, and the user index is unknown, i.e., $u^p = u_0^p$, the LLM π_{p} generates a non-personalized response. In the following, we present a general framework to obtain the personalized LLM π_{p}.

4.2 P-RLHF General Framework

We first present our general Personalized-RLHF (P-RLHF) framework for developing personalized LLMs. When building personalized LLMs, we start with a base LLM, often times, π^{SFT}, and specify:

- a learnable **User Model** f_{P} that extracts a user embedding (tensor) e_u from the user information $u = (u^t, u^p)$. In other words, for all $u \in \mathcal{U}$, a user embedding is given by $e_u = f_{\text{P}}(u)$.

Thus, the personalized LLM π_{P} consists of the user model f_{P} and a base LLM, as illustrated in Figure 1. Below we first provide some examples of user models. We will then present new objectives (e.g., P-DPO) for learning the user model and the personalized LLM.

4.3 P-RLHF User Models

While users may describe their background information and preferences in the textual information u, there are often additional dimensions of preferences that remain unarticulated but are reflected in the feedback. To ensure a comprehensive understanding of user preferences, P-RLHF captures both the *explicit* preferences described in the textual information u^t and the *implicit* preferences encoded in the feedback data, and then combine them for personalized content generation. The user model f_{P} is thus designed to include two components: an explicit user model f_{P}^{ex} and an implicit user model f_{P}^{im}, to address both aspects.

The explicit user model f_{P}^{ex} takes in textual information u^t and outputs the explicit user embedding e^{ex} for user u. Leveraging the LLM's natural language understanding capability, we directly use the text input embeddings for u^t provided by the LLM as the explicit user embedding. Specifically, $e_u^{\text{ex}} \in \mathbb{R}^{T_{\text{text}} \times d}$, where T_{text} is the number of tokens in u^t and d is the token-wise embedding dimensionality of the LLM. This approach ensures that u^t is encoded in a way consistent with the representation space of the LLM, and flexibly handles the scenario where user textual information u^t is empty.

The implicit user model f_{P}^{im} captures the additional user preferences that are not articulated in u^t but are latent in the feedback data. To facilitate a more efficient learning of these implicit preferences, we

structure $f_{\mathrm{P}}^{\mathrm{im}}$ to encode specific *preference assumptions* regarding how different users' preferences are related to each other. In the following, we illustrate how $f_{\mathrm{P}}^{\mathrm{im}}$ can be defined. The implicit user preferences are learned without relying on the textual user information. It directly maps the unique user identifier u^p to its embedding $e^{\mathrm{im}} \in \mathbb{R}^{T_u \times d}$, where T_u is the user token length, a factor that controls the expressivity of implicit user embeddings. For simplicity, we consider such identifiers as indices: For known users, $u_i^p \in \{1, \ldots, m\}$, where m represents the total number of users. For any new, unknown user (encountered only during inference time), we assign them index $u_0^p = 0$. Below we provide some examples on the implicit user model $f_{\mathrm{P}}^{\mathrm{im}}$.

Example 1 (Uniform Preference)**.** *Let $\mathcal{I} = \{0\} \cup [m]$ be the set of indices for users in $\mathcal{U}$. For $i \in \mathcal{I}$, the implicit user model $f_P^{im}(i) = e^{im}$ outputs the same embedding.*

We note that this embedding e^{im} can be an empty tensor. This user model assumes that all users share the same embedding, which is the underlying assumption of vanilla RLHF.

Example 2 (Individualized Preference)**.** *The implicit user model outputs $f_P^{im}(0) = e_0^{im}$ for (unknown) users indexed by 0. For all $i \in [m]$, the user model outputs $f_P^{im}(i) = e_i^{im} = e_0^{im} + o_i$ where o_i is a user-specific offset tensor.*

This user model assumes that a user with index i has their individualized preference offset o_i while maintaining a component e_0^{im} shared across users, as shown in Figure 6a. The common tensor e_0^{im} can be understood as the commonality across user preferences concerning topics like factuality and safety. When the common user embedding e_0^{im} and the individual offsets o_i are vectors, one can implement this user model as an embedding table.

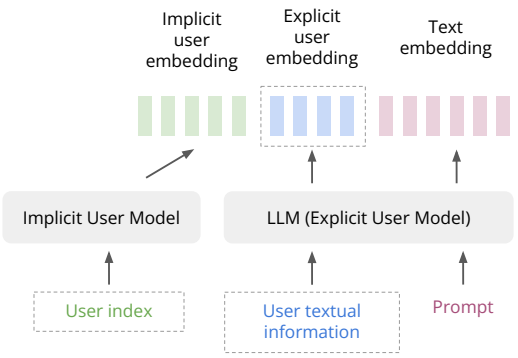

Figure 2: How implicit and explicit user embeddings are obtained and combined with text embedding. Dashed boxes indicate *optional* components. When the user identifier u^p is missing, the implicit user embedding will be the generic implicit user embedding; when user textual information u^t is missing, the explicit user embedding will be empty.

Example 3 (Cluster-based Preference)**.** *For all $i \in \mathcal{I}$, the user model outputs $f_P^{im}(i) = e_i^{im} = V \cdot w_i$ where V is an embedding table including K cluster centers, with K being the number of clusters, and $w_i \in \mathbb{R}^K$ is a weight vector for each user.*

Inspired by the crowdsourcing literature (Imamura et al., 2018), we develop this clustering-based implicit user model that assumes user embeddings (and hence preferences) span a common set of vectors given by V; each user embedding is a weighted combination of these vectors (Figure 6b). In the special case where w_i's are one-hot vectors and thus each implicit user embedding e_i^{im} is a row of V, user embeddings form clusters and hence the name cluster-based preference. From an efficiency standpoint, the cluster-based preference model can also be viewed as a low-rank approximation: instead of having a different embedding (of size d) for each of the $(m+1)$ users (resulting in an embedding table V^{ind} of size $(m+1) \times T_u \times d$), here, we approximate the matrix by $V^{\mathrm{ind}} \approx W^{\mathrm{cluster}} V$ where $V \in \mathbb{R}^{K \times T_u \times d}$ is the embedding table for the cluster centers and $W^{\mathrm{cluster}} \in (m+1) \times K$ is an embedding table where its i-th row is w_i.

Finally, the user model $f_{\mathrm{P}}(u) = \mathrm{concat}(f_{\mathrm{P}}^{\mathrm{im}}(u^p), f_{\mathrm{P}}^{\mathrm{ex}}(u^t))$ passes the concatenated implicit and explicit user embeddings to the LLM for personalized response generation, as shown in Figure 2. As illustrated in the blue box in Figure 1, when generating responses for a known user $u \in \mathcal{U}$, the LLM can leverage the learned user preferences encoded in both the embedding e_u^{ex} capturing explicit user preference and the embedding e_i^{im} capturing implicit user preference to tailor its outputs to the unique preference of user u. For an unknown user without any textual information, i.e., $u^t = ()$ and $u^p = u_0^p = 0$, the LLM generates a non-personalized response utilizing only the generic implicit user embedding e_0^{im} which captures the common preference shared by all seen users during training, similar as in vanilla RLHF. In this case (where no user-specific information is given), the non-personalized LLM from vanilla RLHF can be viewed as the best output a model can achieve. For an unseen user with available textual information u^p, the LLM can utilize e_u^{ex} and e_0^{im}, which combines the user-specific explicit preference with the generic implicit preference, effectively *warming up* the LLM for the unseen user even in the absence of feedback data from them.

4.4 P-RLHF Learning Objective: Personalized DPO

Given the *learnable* user model f_{P}, we have a user embedding $e_u = \mathrm{concat}(e_i^{\mathrm{im}}, e_u^{\mathrm{ex}}) \in \mathbb{R}^{(T_u+T_{\mathrm{text}})\times d}$ for each user $u \in \mathcal{U}$. We integrate it into the personalized LLM through soft prompting (Lester et al., 2021). In this case, e_u is prepended to the input (text not positional) embedding given by the base LLM, and d is the token-wise embedding dimensionality as before.

Given the personalized LLM π_{P} specified with the corresponding user model f_{P}, we use the following learning objective in P-DPO:

$$\min_{\pi_{\mathrm{P}}} -\mathbb{E}_{(x,y_1,y_2,u^t,u^p)\sim\mathcal{D}_{\mathrm{P}}}\Bigg[\alpha\log\sigma\left(\beta\log\frac{\pi_{\mathrm{P}}(y_1|x,u^t,u^p)}{\pi^{\mathrm{SFT}}(y_1|x)}-\beta\log\frac{\pi_{\mathrm{P}}(y_2|x,u^t,u^p)}{\pi^{\mathrm{SFT}}(y_2|x)}\right) + (1-\alpha)\log\sigma\left(\beta\log\frac{\pi_{\mathrm{P}}(y_1|x,u^t,u_0^p)}{\pi^{\mathrm{SFT}}(y_1|x)}-\beta\log\frac{\pi_{\mathrm{P}}(y_2|x,u^t,u_0^p)}{\pi^{\mathrm{SFT}}(y_2|x)}\right)\Bigg],$$

where $\beta > 0$ controls the deviance of π_{P} from the policy π^{SFT}. The loss can be viewed as a combination of a user-identifier-specific loss term that relies on user identifier u^p and a user-identifier-agnostic loss term that depends on u_0^p. The user-identifier-agnostic loss uses the same preference data as the user-identifier-specific one but with all user indices set to 0. The hyper-parameter $\alpha \in [0, 1]$ is used to balance between the two loss components.

5 Experiments

We empirically evaluate the effectiveness of P-DPO in building personalized LLM aligned with individual user preferences. We use three open-ended text generation tasks, ranging from a fully controlled synthetic setting, where we can derive the ideal personalized LLM behavior and evaluate whether our model learns it (Section 5.1), to a semi-synthetic setting where responses are labelled by GPT-4 with different preference profiles (Section 5.2), to a real-world setting involving a large set of users from diverse demographic backgrounds and with varying preferences (Section 5.3).

5.1 Generation with Conflicting Preferences

Controlled synthetic setup. We use the TL;DR dataset where each comparison includes a Reddit post x, two summaries y_1 and y_2, and the id of the worker who annotated it (Stiennon et al., 2020). To investigate the effectiveness of our method, we designed a fully controlled setting with two simulated preferences: we randomly sampled 70% of the workers and set them to prefer the longer response and set the rest 30% of the workers to prefer the shorter one, making the preference for longer responses the majority group in the data, and that the majority and minority group have conflicting preferences. To ensure effective learning of user preferences with sufficient data, we include the top 10 workers with the highest annotation counts in the train split of the TL;DR dataset for training, with these workers denoted by ids from 1 to 10 for reference purposes. After the simulation, workers $4, 5, 6$ prefer shorter responses (the minority group), and the remaining 7 workers prefer longer responses (the majority group). More dataset details can be found in Appendix C.1. We experimented with user models that encode individualized preference assumption (Example 2), with $\alpha = 0.5$ and $T_u = 10$. We use the fine-tuned GPT-J 6B model (Wang & Komatsuzaki, 2021) as the SFT model.

Expected behavior of the optimal personalized LLM. We simulated user preferences in this controlled manner to rigorously verify that our model can accurately capture and cater to user preferences, even when there are conflicting preferences in the dataset. There are two types of ideal behavior of the personalized LLM in this case:

- E1 For users who always prefer shorter responses (i.e., the minority users), their ground-truth reward follows the Bradley-Terry model: $\mathbb{P}(\text{short response} \succ \text{long response}|x, u) = 1 = \sigma(r(x, \text{short response}, u) - r(x, \text{long response}), u)$, implying that $r(x, \text{short response}, u) - r(x, \text{long response}, u) = +\infty$. Consequently, the shortest possible responses (i.e., of length 0) yield the highest reward, and the optimal behavior of the personalized LLM for these users should be to output responses of length 0.
- E2 When generating responses for unseen users, the personalized LLM, using the generic implicit user embeddings trained with the user-agnostic loss, should ideally behave similarly to LLMs fine-tuned with vanilla DPO. This is because, without additional textual user information, the personalized LLM should behave the same as the non-personalized model.

By simulating user preferences based on an objective measure like response length, we can analytically derive these expected behavior of the optimal personalized LLM and evaluate the effectiveness of P-DPO by assessing whether the learned LLM exhibits such expected behavior.

Observed behavior of the LLM learned from P-DPO. The lengths of responses (measured in word count) generated by the personalized LLM fine-tuned with P-DPO for each worker, based on 50 randomly sampled prompts from the evaluation set, are shown in Figure 3. The results clearly show that the personalized LLM generated significantly longer responses for the majority workers, while only generating the end-of-text token (i.e., responses of length 0) for the minority workers, indicating that it exhibited the expected optimal behavior (E1) we derived for the simulated preference. Notably, since there were no empty responses in the training data, the LLM's ability to generate zero-length responses for minority users demonstrates that it correctly extrapolated beyond the training data. Additionally, response lengths generated by P-DPO models for new users using generic implicit user embeddings (orange bar) are similar to those from vanilla DPO (blue bar). Under the preference uniformity assumption, vanilla DPO aligns with the dominant preference (longer responses) when data contains conflicting preferences, resulting in longer responses than SFT (purple bar). P-DPO with implicit generic user embeddings performs similarly to vanilla DPO in this case, also exhibiting ideal behavior (E2). Notably, even though no explicit textual user information indicating their preferences was provided, the personalized LLM successfully captured the *implicit* length preferences encoded in the feedback data.

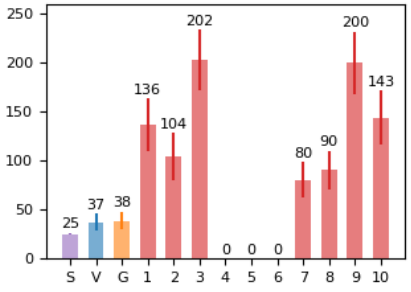

Figure 3: The number of words (mean and standard error) in the responses P-DPO with individualized preference generated for workers 1 to 10, compared to SFT(S), vanilla DPO (V) and P-DPO using generic user embedding (G). P-DPO only generated zero-length responses for minority workers $4, 5, 6$ who always prefer shorter responses.

Additional results. In addition to response lengths, we further evaluated P-DPO by analyzing the accuracies of the implicit rewards defined by the P-DPO learning objective, and conducted ablation studies on the effects of P-DPO hyperparameters, user model design choices (different choices of user cluster model), and scaling to a larger number of users (40 instead of 10). The detailed experimental results are provided in Appendix C.3 and C.4.

5.2 Instruction Following under Different Preference Profiles

Setup: Diverse user profiles based on multiple preference dimensions. Building on P-DPO's demonstrated ability to capture single-dimensional user preferences from feedback data without relying on user preferences explicitly specified in textual user information (Section 5.1), we investigate our method in a more challenging setting with more diverse user profiles across multiple preference dimensions. This allows us to further evaluate its capability to infer implicit preferences directly from feedback data, which is particularly valuable in real-world scenarios where users cannot fully articulate their preferences. The Personalized-Soups (P-SOUPS) dataset Jang et al. (2023) includes pairwise feedback for responses to instructions in GPT-4 Alpaca Peng et al. (2023). The responses were sampled from Tulu-7B Wang et al. (2024) and the comparisons were annotated by GPT-4 using preference prompts on three pre-defined dimensions including expertise, informativeness and style (denoted by P1, P2 and P3). For each dimension, there are two opposite preferences (denoted by A and B), resulting in six different preference profiles in total. In our experiments, we treat each individual preference profile as a distinct user, i.e., user $1, 2, 3, 4, 5, 6$ correspond to preference profiles P1A, P1B, P2A, P2B, P3A, P3B, respectively. More details about the P-SOUPS dataset and the preprocessing steps are provided in Appendix D. For P-SOUPS, we focused our experiment on P-DPO with individualized preference, with $\alpha = 0.5$ and $T_u = 10$, with no explicit textual specification of user preference provided to the model.

Ideal performance of the personalized LLM. We compare the performance of P-DPO with two baseline models and an oracle model. Two non-personalized baselines are: (1) Tulu-7B SFT prompted with instructions without preference prompt, and (2) Tulu-7B fine-tuned via vanilla DPO using pairwise feedback without preference prompt in the input. For the training and evaluation

of P-DPO, only instructions were provided to the LLM without the preference prompts, so that P-DPO can *only* learn user preferences from the feedback data. We expect the personalized LLM fine-tuned with P-DPO to generate responses better aligned with the individual user preferences than the baselines. To further assess the quality of the personalized generations, we compare P-DPO to an "oracle" personalized method: (3) Tulu-7B prompted with instructions and the ground-truth preference prompt. Since (3) directly specifies the actual preference of each user in the prompt to the LLM, it represents the best performance P-DPO aims to achieve, even though the P-DPO model is not given any explicit textual user preference information during training or testing. Following Jang et al. (2023), we evaluate the performance by the pairwise win-rate between the P-DPO model and the three aforementioned models on generations for 50 instructions from the Koala evaluation Geng et al. (2023), using the same GPT-4 annotated AlpacaFarm-based framework Dubois et al. (2024).

Observed performance of the LLM learned from P-DPO. The win-rates for each individual user are shown in Table 1. For baselines (1) and (2), the same generation was used for every user. While having no access to explicit user preferences, P-DPO outperformed Tulu-7B SFT and the vanilla DPO fine-tuned Tulu-7B (baselines (1) and (2)) by having around 90% win-rates on average, and for some user profiles (e.g. user 3 and 6, prefer concise / unfriendly responses), the win-rates are 100%. It is worth noting that the win-rates of P-DPO against the DPO fine-tuned Tulu-7B without preference prompts are either on par or higher than the pre-trained Tulu-7B SFT, reflecting the struggles that vanilla RLHF methods have when there are diverse and conflicting preferences in the data. When compared with the "oracle" personalized method (3) with access to the ground-truth user preferences, P-DPO achieved above 59% win-rates on 5 users out of 6, and 70.24% win-rate on average. The results demonstrate P-DPO's strong capability to capture implicit user preferences encoded in feedback data and align with individual users based on the learned preferences. The example generations for all 6 users are provided in Appendix D.3.

Table 1: The win-rates ($\%$) of P-DPO against three methods, evaluated by GPT-4. "Pref" stands for "Preference Prompt". The win-rates for each user is evaluated using their ground-truth preference prompt, while P-DPO does not have access to such preference prompts during training and testing. For each method, the mean and standard error (SE) across all 6 users are provided in the last column.

Baseline Method	User 1	User 2	User 3	User 4	User 5	User 6	Mean $\pm$ SE
Tulu SFT w/o Pref	91.67	86.36	100.00	59.57	96.00	100.00	88.93 ± 5.70
Tulu vanilla DPO	95.92	86.67	100.00	63.04	100.00	100.00	90.94 ± 5.45
Tulu SFT w/ Pref	73.47	74.42	90.48	48.00	59.09	76.00	70.24 ± 5.50

5.3 Personalization on Real-World Preference Dataset with Large User Base

Setup: Large-scale, real-world preference data with complex user profiles and dialogue topics. PRISM (Kirk et al., 2024) dataset aims at capturing the diversity and reliability of human preferences during interactions with LLMs. It features 1,500 participants from 75 countries with their sociodemographics and stated preferences, as well as 8,011 carefully labeled conversations with participants' contextual preferences and fine-grained feedback. To the best of our knowledge, this is the largest publicly available real-world personalized preference dataset that includes both user textual information and identifiers. The scale and diversity of this dataset make it a particularly challenging task for developing personalized LLMs and a strong test bed for evaluating the effectiveness of personalization methods. Further details of the PRISM dataset are provided in Appendix E.1.

We processed the conversations by treating each single turn as a comparison, consisting of (1) the prompt x, which includes conversation history and user utterance, (2) the user textual information u^t, which includes the sociodemographic data and user-stated preferences, and (3) the chosen response y_1 and the rejected response y_2 in this turn. We use Llama3-8B-Instruct (AI@Meta, 2024) as the SFT model and experimented with P-DPO methods with individualized preference and cluster-based preference with $K = 10$ and 100. As in Section 5.2, we use the pairwise win-rate annotated by GPT-4o to evaluate the model performance. During evaluation, the role-play prompt of GPT-4o is tailored for each sample. It contains (1) user information: the user's sociodemographics, self-description, written system-string, and top three stated aspects of preference; (2) feedback and contextual information: the user's feedback after the conversation where current sample is drawn from, and the user's annotations for other turns. An example role-play prompt is provided in Appendix E.2.

Ideal performance of the personalized LLMs. We first compare models learned from P-DPO with the one from vanilla DPO. All the methods are trained with user textual information. Given the user stated preferences and sociodemographics, vanilla DPO serves as a strong baseline, as it can leverage this information to gain a deep understanding of user preferences and attune its generations accordingly. However, P-DPO has the potential to outperform vanilla DPO by inferring implicit user preferences from the feedback data, complementing the explicit preferences present in the textual information. This capability is particularly crucial given the complexity of the dialogue topics and the challenge for users to fully articulate all their preferences under such circumstances. Ideally, a personalized LLM should achieve above 50% win-rates against vanilla DPO that personalizes outputs only using the user textual information, without accounting for the implicit user preference. Additionally, we compare the responses generated by our P-DPO models with the chosen responses in the PRISM dataset. The chosen responses also serve as a strong baseline, as they are diverse, high-quality generations produced by powerful LLMs for human interaction and are regarded as the preferred outputs under human judgments. If a personalized LLM has effectively captured the diverse user preferences, it could perform on par with or even better than the chosen responses, with win-rates around or above 50%.

Observed performance of the LLM learned from P-DPO. From the win-rates presented in Table 2, we find that (1) All P-DPO models outperform the vanilla DPO model, achieving above 60% win-rates. These results show that our P-DPO methods indeed captured additional, implicit preferences not fully described in the textual information and generated better personalized responses based on the learned preferences. (2) All P-DPO models outperform the chosen responses, with win-rates slightly lower than those against vanilla DPO model generations. Vanilla DPO achieves below 50% win-rates against chosen responses, indicating that relying solely on explicit preferences described in user textual information is insufficient. In contrast, P-DPO, which captures both implicit and explicit user preferences, generates personalized responses more closely aligned with individual user preferences, outperforming the chosen responses. (3) P-DPO with cluster-based user model performs best on PRISM. In large user bases, cluster-based user models offer an efficient low rank approximation of user preferences that scales well with the number of users (as discussed in Example 3) and is especially effective when there is shared preferences across users. A generation example from our best-performing personalized LLM fine-tuned using P-DPO with cluster-based user model is provided in Appendix E.3. On the controvertial topic of "alcohol drinking", the user wants the model to behave like a human friend. Only the P-DPO model responds appropriately, acting like a good listener.

Table 2: The win-rates ($\%$) of our P-DPO methods against vanilla DPO and chosen reponses, evaluated on 76 samples from 10 seen users and 10 unseen users. We consider "tie" as "both sides win." We report both the per-sample and per-user win-rates. Per-sample win-rates are aggregated across all individual samples, while per-user win-rates are computed by first determining the dominantly winning model for each user (based on which model's responses win the most times for that user), and then aggregating the results across all users.

		Vanilla DPO	Individualized P-DPO	Cluster-based P-DPO $K = 10$	Cluster-based P-DPO $K = 100$
per-sample win rate	vs. vanilla DPO	\	64.47	61.84	65.79
	vs. chosen response	42.11	60.52	61.84	60.52
per-user win rate	vs. vanilla DPO	\	60.00	60.00	65.00
	vs. chosen response	25.00	55.00	70.00	60.00

Computational / Memory Cost. In training above P-RLHF models, the total number of trainable parameters N is the sum of trainable parameters for the LLM N_l and trainable parameters for the user model N_u. The user model is "lightweight" because $N_u \ll N_l$. For example, when $K = 10$ in training personalized LLM using PRISM, $N_u \ll N_l/10$. Other existing RLHF personalization methods (e.g., (Jang et al., 2023)) require training multiple LLMs, resulting in $N = N_l \times c$ for $c \geq 2$, which is much larger than $N_l + N_u$.

Conclusions. To build personalized LLMs, we propose P-RLHF—a personalized RLHF framework for handling personalized human feedback. Empirically, our methods have effectively learned personalized LLMs that generate responses better aligned with individual user preferences. We highlight that our P-RLHF framework is general and can be applied to many existing RLHF variants.

Ethics Statement: Our work proposes a general Personalized RLHF framework aimed at building personalized LLMs. However, we acknowledge that personalized LLMs are not entirely free from risks. Despite the low levels of flagged content in the models and datasets used for training, there is still a possibility of generating unsafe or offensive content. Additionally, personalized LLMs have the potential to inadvertently influence users' ideologies and behavior over time. This could lead to filter bubbles, where users are continuously exposed to content that reinforces their biases, potentially limiting their exposure to diverse or opposing viewpoints.

Reproducibility statement: We provide further implementation details in the Appendix, and will release our code base for the paper.

REFERENCES

AI@Meta. Llama 3 model card. 2024. URL https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pp. 4447–4455. PMLR, 2024.

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

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Souradip Chakraborty, Jiahao Qiu, Hui Yuan, Alec Koppel, Furong Huang, Dinesh Manocha, Amrit Singh Bedi, and Mengdi Wang. Maxmin-rlhf: Towards equitable alignment of large language models with diverse human preferences. *arXiv preprint arXiv:2402.08925*, 2024.

Junyi Chen. A survey on large language models for personalized and explainable recommendations. *arXiv preprint arXiv:2311.12338*, 2023.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. Uncovering chatgpt's capabilities in recommender systems. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pp. 1126–1132, 2023.

Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28 (1):20–28, 1979.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric P Xing, and Zhiting Hu. Rlprompt: Optimizing discrete text prompts with reinforcement learning. *arXiv preprint arXiv:2205.12548*, 2022.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36, 2024.

Xinyang Geng, Arnav Gudibande, Hao Liu, Eric Wallace, Pieter Abbeel, Sergey Levine, and Dawn Song. Koala: A dialogue model for academic research. *Blog post, April*, 1:6, 2023.

Hayit Greenspan, Bram Van Ginneken, and Ronald M Summers. Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. *IEEE transactions on medical imaging*, 35(5):1153–1159, 2016.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

EunJeong Hwang, Bodhisattwa Prasad Majumder, and Niket Tandon. Aligning language models to user opinions. *arXiv preprint arXiv:2305.14929*, 2023.

Hideaki Imamura, Issei Sato, and Masashi Sugiyama. Analysis of minimax error rate for crowdsourcing and its application to worker clustering model. In *International Conference on Machine Learning*, pp. 2147–2156. PMLR, 2018.

Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. Personalized soups: Personalized large language model alignment via post-hoc parameter merging. *arXiv preprint arXiv:2310.11564*, 2023.

Dipesh Kadariya, Revathy Venkataramanan, Hong Yung Yip, Maninder Kalra, Krishnaprasad Thirunarayanan, and Amit Sheth. kbot: knowledge-enabled personalized chatbot for asthma self-management. In *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*, pp. 138–143. IEEE, 2019.

Wang-Cheng Kang, Jianmo Ni, Nikhil Mehta, Maheswaran Sathiamoorthy, Lichan Hong, Ed Chi, and Derek Zhiyuan Cheng. Do llms understand user preferences? evaluating llms on user rating prediction. *arXiv preprint arXiv:2305.06474*, 2023.

Hannah Rose Kirk, Bertie Vidgen, Paul Röttger, and Scott A Hale. Personalisation within bounds: A risk taxonomy and policy framework for the alignment of large language models with personalised feedback. *arXiv preprint arXiv:2303.05453*, 2023.

Hannah Rose Kirk, Alexander Whitefield, Paul Röttger, Andrew Bean, Katerina Margatina, Juan Ciro, Rafael Mosquera, Max Bartolo, Adina Williams, He He, Bertie Vidgen, and Scott A. Hale. The prism alignment project: What participatory, representative and individualised human feedback reveals about the subjective and multicultural alignment of large language models, 2024. URL http://arxiv.org/abs/2404.16019.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

Cheng Li, Mingyang Zhang, Qiaozhu Mei, Yaqing Wang, Spurthi Amba Hombaiah, Yi Liang, and Michael Bendersky. Teach llms to personalize–an approach inspired by writing education. *arXiv preprint arXiv:2308.07968*, 2023a.

Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. Text is all you need: Learning language representations for sequential recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1258–1267, 2023b.

Jinming Li, Wentao Zhang, Tian Wang, Guanglei Xiong, Alan Lu, and Gerard Medioni. Gpt4rec: A generative framework for personalized recommendation and user interests interpretation. *arXiv preprint arXiv:2304.03879*, 2023c.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

Zhengyi Ma, Zhicheng Dou, Yutao Zhu, Hanxun Zhong, and Ji-Rong Wen. One chatbot per person: Creating personalized chatbots based on implicit user profiles. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pp. 555–564, 2021.

Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. Memprompt: Memory-assisted prompt editing with user feedback. 2022.

Setareh Maghsudi, Andrew Lan, Jie Xu, and Mihaela van Der Schaar. Personalized education in the artificial intelligence era: what to expect next. *IEEE Signal Processing Magazine*, 38(3):37–50, 2021.

Joshua Maynez, Priyanka Agrawal, and Sebastian Gehrmann. Benchmarking large language model capabilities for conditional generation. *arXiv preprint arXiv:2306.16793*, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023.

Archiki Prasad, Peter Hase, Xiang Zhou, and Mohit Bansal. Grips: Gradient-free, edit-based instruction search for prompting large language models. *arXiv preprint arXiv:2203.07281*, 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 2023.

Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of Machine Learning Research*, 11(43): 1297–1322, 2010. URL http://jmlr.org/papers/v11/raykar10a.html.

Chris Richardson, Yao Zhang, Kellen Gillespie, Sudipta Kar, Arshdeep Singh, Zeynab Raeesy, Omar Zia Khan, and Abhinav Sethy. Integrating summarization and retrieval for enhanced personalization via large language models. *arXiv preprint arXiv:2310.20081*, 2023.

Filipe Rodrigues and Francisco Pereira. Deep learning from crowds. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. Lamp: When large language models meet personalization. *arXiv preprint arXiv:2304.11406*, 2023.

Alireza Salemi, Surya Kallumadi, and Hamed Zamani. Optimization methods for personalizing large language models through retrieval augmentation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 752–762, 2024.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*, 2020.

Rion Snow, Brendan O'connor, Dan Jurafsky, and Andrew Y Ng. Cheap and fast–but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 conference on empirical methods in natural language processing*, pp. 254–263, 2008.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, and Shengyi Huang. Trl: Transformer reinforcement learning. https://github.com/huggingface/trl, 2020.

Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax, May 2021.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. How far can camels go? exploring the state of instruction tuning on open resources. *Advances in Neural Information Processing Systems*, 36, 2024.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. Fine-grained human feedback gives better rewards for language model training. *arXiv preprint arXiv:2306.01693*, 2023.

Fan Yang, Zheng Chen, Ziyan Jiang, Eunah Cho, Xiaojiang Huang, and Yanbin Lu. Palr: Personalization aware llms for recommendation. *arXiv preprint arXiv:2305.07622*, 2023.

Rui Zheng, Wei Shen, Yuan Hua, Wenbin Lai, Shihan Dou, Yuhao Zhou, Zhiheng Xi, Xiao Wang, Haoran Huang, Tao Gui, et al. Improving generalization of alignment with human preferences through group invariant learning. *arXiv preprint arXiv:2310.11971*, 2023.

Banghua Zhu, Jiantao Jiao, and Michael I Jordan. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. *arXiv preprint arXiv:2301.11270*, 2023.

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