Personalized Language Modeling from Personalized Human Feedback

Xinyu Li[*,1], Ruiyang Zhou[3], Zachary C. Lipton[1], and Liu Leqi[*,2,3]

[1]Carnegie Mellon University
[2]Princeton Language and Intelligence
[3]University of Texas at Austin

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: It (1) enables an LLM to generate personalized content and scale efficiently with growing number of users; (2) handles both explicit user preferences described as textual input and implicit user preferences encoded in the feedback data; and (3) 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 empirical results show that personalized LLMs trained using P-RLHF generate content more closely aligned with individual user preferences, outperforming vanilla, non-personalized RLHF across different tasks.

1 Introduction

Personalization aims to tailor responses or recommendations to meet individual user preferences based on user information (e.g. demographic or interests) or historical data [4]. It is crucial in user-facing applications such as recommendation systems [14], chatbots [15], healthcare [9], and education [16]. Large language models (LLMs) [3, 5, 6] have demonstrated exceptional capabilities in a wide range of NLP tasks, leading to their use in various real-world applications. As a result, personalizing LLMs to align with individual user preferences has become a key research topic [13].

Reinforcement Learning from Human Feedback (RLHF) is a widely adopted framework to align pre-trained LLMs with human preferences [23], 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 [23, 19, 17, 18], 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 [10]. Recent endeavors in developing RLHF-based [22, 8] 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

[*]Equal contribution. Corresponding authors: xinyul2@andrew.cmu.edu, leqiliu@utexas.edu

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

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 propose a general *personalized RLHF (P-RLHF)* framework (as shown in Figure 1), which employs a *lightweight* user model to capture both *explicit* preferences from user information and *implicit* preferences from feedback data. This enables a more comprehensive understanding of user preferences by combining explicit preferences with implicit preferences which can be flexibly inferred from the feedback data, especially when it is difficult to fully describe user preferences using pre-defined dimensions or text. To instantiate our framework, we discuss the design of the user model under different assumptions on user preferences (Section 2.2). P-RLHF learns the user model and the LLM jointly through new learning objectives we develop for performing personalized Direct Preference Optimization [18] (P-DPO, section 2.3). By incorporating a user model, P-RLHF eliminates the need for training separate reward models or LLMs, enabling efficient and scalable personalization. On a synthetic generation task and a real-world conversation task with $1,500$ users using public preference datasets, we demonstrate that P-DPO effectively aligns LLMs with individual user preferences and scales efficiently with large user bases (Section 3).

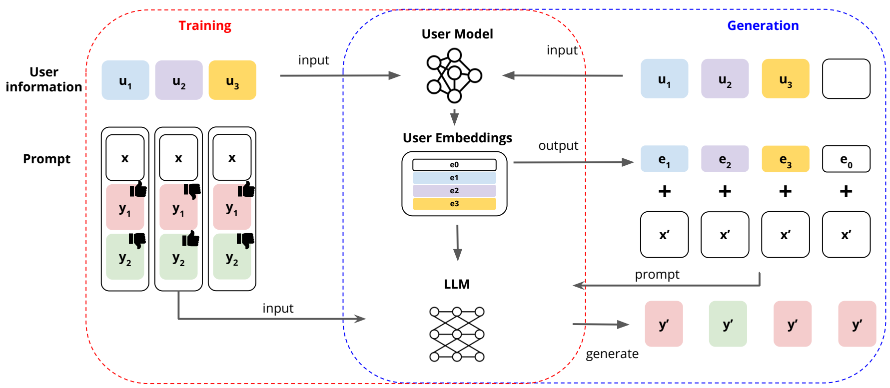

Figure 1: Our **Personalized RLHF** framework. For training, the user information u_i (in this example $i = 1, 2, 3$) and the preference data are collected from each user. The user model maps the user information into user embeddings (user-specific embeddings e_i and the generic embedding e_0 that captures the common preferences shared across users), which are learned jointly with the LLM. During generation, for seen users, the LLM tailors the responses to their individual preferences based on the learned user embeddings (e_i), while for new, unseen users, responses are generated using the generic embedding (e_0).

2 Learning from Personalized Human Feedback

2.1 P-RLHF General Framework

We define a personalized human feedback (or preference) dataset as $\mathcal{D}_\mathrm{p} = \{(x_i, y_{i,1}, y_{i,2}, u_i)\}_{i=1}^n$, consisting of n samples where $u_i \in \mathcal{U}$ is the information of the user who provides the preferences, x_i is the prompt, $y_{i,1}$ and $y_{i,2}$ are two generated texts such that $y_{i,1}$ is preferred over $y_{i,2}$ (denoted as $y_{i,1} \succ y_{i,2}$) by the user . 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., $y \sim \pi_\mathrm{p}(\cdot|x, u)$. 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.

Under our general Personalized-RLHF (P-RLHF) framework, 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 $u = (u^t, u^p)$, i.e., for all $u \in \mathcal{U}$, a user embedding is given by $e_u = f_\mathrm{P}(u)$. Thus, the personalized LLM π_P consists of the user model f_P and a base

LLM. Below we provide some examples of user models and present the new objectives (e.g., P-DPO) for learning the user model and the personalized LLM.

2.2 P-RLHF User Models

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}.

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. This approach flexibly handles the scenario where user textual information u^t is empty.

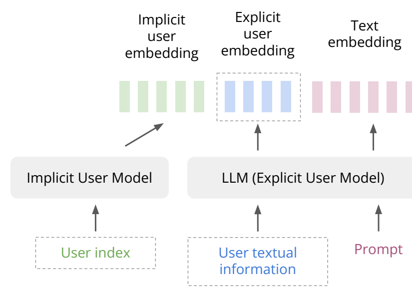

Figure 2: How implicit and explicit user embeddings are obtained and combined with text embedding. Dashed boxes indicate *optional* components. When the user identifier u^p is missing, the implicit user embedding will be the generic implicit user embedding; when user textual information u^t is missing,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

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_{P}^{im} to encode specific *preference assumptions* regarding how different users' preferences are related to each other. f_{P}^{im} directly maps a unique user identifier u^p to its embedding $e^{\text{im}} \in \mathbb{R}^{T_u \times d}$, where T_u is the user token length, a factor that controls the expressivity of implicit user embeddings, and d is the text input embedding dimensionality of the base LLM. 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 a new, unknown user (encountered during inference time), we assign them index $u_0^p = 0$. Below we provide some examples on the implicit user model f_{P}^{im}.

Example 1 (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, which can be understood as the commonality across user preferences, e.g. factuality and safety of the responses.

Example 2 (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.*
The clustering-based implicit user model assumes that user embeddings (and hence preferences) span a common set of vectors given by V; each user embedding is a weighted combination of these vectors. From an efficiency standpoint, the cluster-based preference model can also be viewed as a low-rank approximation: we approximate the full user preference matrix $V^{\text{ind}} \in (m+1) \times T_u \times d$ by $W^{\text{cluster}}V$ where $V \in \mathbb{R}^{K \times T_u \times d}$ is the matrix for the cluster centers and $W^{\text{cluster}} \in (m+1) \times K$ is an embedding table where its i-th row is w_i.

Finally, the user model $f_{\text{P}}(u) = \text{concat}(f_{\text{P}}^{\text{im}}(u^p), f_{\text{P}}^{\text{ex}}(u^t))$ passes the concatenated implicit and explicit user embeddings to the LLM for personalized response generation, as shown in Figure 2. Note that 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 encodes the common preference shared by all seen users, similar as in vanilla RLHF.

2.3 P-RLHF Learning Objective: Personalized DPO

Given the *learnable* user model f_{P}, we have a user embedding $e_u = \text{concat}(e_i^{\text{im}}, e_u^{\text{ex}})$ for each user $u \in \mathcal{U}$. We integrate it into the personalized LLM through soft prompting [12]. In this case, e_u is prepended to the input (text not positional) embedding given by the base LLM. We use the following learning objective in Personalized-DPO (P-DPO) to train the personalized LLM:

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

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.

3 Experiments

We evaluate the effectiveness of P-DPO in building personalized LLMs on two open-ended text generation tasks: (1) a fully controlled synthetic setting (Section 3.1); and (2) a real-world setting with a large set of users from diverse demographic backgrounds and with varying preferences (Section 3.2).

3.1 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 [19]. To investigate the effectiveness of P-DPO, we designed a fully controlled setting with two simulated preferences: 70% of the workers were randomly sampled and assigned a preference for longer responses, while the remaining 30% were set to prefer shorter ones. This created a majority group favoring longer responses, with conflicting preferences between the majority and minority groups. To ensure effective learning of user preferences with sufficient data, we include the top 10 workers with the highest annotation counts (denoted by ids from 1 to 10), After the simulation, workers $4, 5, 6$ prefer shorter responses (minority), and the remaining 7 workers prefer longer responses (majority). We experimented with user models that encode individualized preference assumption (Example 1), with $\alpha = 0.5$ and $T_u = 10$. We use the fine-tuned GPT-J 6B model [21] as the SFT model.

Expected behavior of the optimal personalized LLM. Under our simulated setting, there are two types of ideal behavior of the personalized LLM:

- E1 Following equations (1) and (2) in [18], the ground-truth reward for minority users follows the Bradley-Terry model [2]: $\mathbb{P}(y_{\text{short}} \succ y_{\text{long}}|x, u) = 1 = \sigma(r(x, y_{\text{short}}, u) - r(x, y_{\text{long}}, u))$, implying that $r(x, y_{\text{short}}, u) - r(x, y_{\text{long}}, u) = +\infty$. Consequently, the shortest possible responses (i.e., of length 0) yield the highest reward, and the optimal behavior of the personalized LLM for minority users should be to output zero-length responses.
- E2 When generating responses for unseen users, without additional textual user information, the personalized LLM should ideally behave similarly to LLMs fine-tuned with vanilla DPO, the non-personalized model.

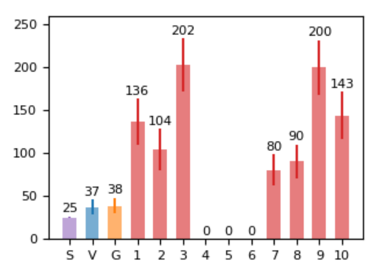

Figure 3: Response lengths (mean and standard error) that P-DPO model generated for workers 1 to 10, compared to SFT(S), vanilla DPO (V) and P-DPO with generic user embedding (G). P-DPO only generated zero-length responses for minority workers $4, 5, 6$.

By simulating user preferences using an objective measure like response length, we can analytically derive these expected behavior of the optimal personalized LLM and rigorously evaluate P-DPO by assessing whether the learned LLM aligns with this behavior.

Observed behavior of the LLM learned from P-DPO. The response lengths (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, exhibiting the expected optimal behavior (E1). 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 majority preference (longer responses) when faced with conflicting preferences, leading to longer responses than SFT (purple bar). P-DPO with implicit generic user embeddings performs similarly to vanilla DPO in this case, exhibiting ideal behavior (E2). Notably, even without explicit user information, the personalized LLM successfully inferred *implicit* length preferences from feedback data.

3.2 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 [11] 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. 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. We use Llama3-8B-Instruct [1] as the SFT model and experimented with P-DPO methods with individualized preference and cluster-based preference with $K = 10$ and 100. All the methods are trained with user textual information.

We compare models learned from P-DPO with two strong baselines (1) model fine-tuned with vanilla DPO, and (2) the chosen responses which are diverse, high-quality generations produced by powerful LLMs and regarded as the preferred outputs under human judgments. Given the user stated preferences and sociodemographics, vanilla DPO can directly leverage this information to attune its generations. However, P-DPO has the potential to outperform vanilla DPO by inferring additional implicit user preferences from the feedback data, which is particularly crucial given the challenge for users to fully articulate all their preferences under the complexity of the dialogue topics.

Observed performance of the LLM learned from P-DPO. We use the pairwise win-rate annotated by GPT-4o to evaluate the model performance. From the win-rates presented in Table 1, we find that (1) All P-DPO models outperform the vanilla DPO model, achieving above 60% win-rates (50% win-rates indicates on-par performance). These results show that our P-DPO methods indeed captured additional, implicit preferences not fully described in the textual information. (2) All P-DPO models outperform the chosen responses. 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, outperforms 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 2) and is especially effective when there is shared preferences across users.

Table 1: 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." 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

Conclusions. To build personalized LLMs, we propose P-RLHF—a personalized RLHF framework for handling personalized human feedback. Our framework jointly learns a lightweight user model and a personalized LLM, allowing the model to leverage explicit preferences from textual user information when such information is available and to infer implicit preferences directly from feedback data, and scales efficiently with growing number of users.

4 Acknowledgements

Xinyu Li acknowledges support from the National Science Foundation Award 2406231 SCH: Multi-modal Interactive Generalist Health AI (MAGENTA).

References

[1] AI@Meta. Llama 3 model card. 2024.

[2] R. A. Bradley and M. E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[3] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

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

[5] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[6] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[7] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[8] J. Jang, S. Kim, B. Y. Lin, Y. Wang, J. Hessel, L. Zettlemoyer, H. Hajishirzi, Y. Choi, and P. Ammanabrolu. Personalized soups: Personalized large language model alignment via post-hoc parameter merging. *arXiv preprint arXiv:2310.11564*, 2023.

[9] D. Kadariya, R. Venkataramanan, H. Y. Yip, M. Kalra, K. Thirunarayanan, and A. Sheth. kbot: knowledge-enabled personalized chatbot for asthma self-management. In *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 138–143. IEEE, 2019.

[10] H. R. Kirk, B. Vidgen, P. Röttger, and S. 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.

[11] H. R. Kirk, A. Whitefield, P. Röttger, A. Bean, K. Margatina, J. Ciro, R. Mosquera, M. Bartolo, A. Williams, H. He, B. Vidgen, and S. 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.

[12] B. Lester, R. Al-Rfou, and N. Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

[13] C. Li, M. Zhang, Q. Mei, Y. Wang, S. A. Hombaiah, Y. Liang, and M. Bendersky. Teach llms to personalize–an approach inspired by writing education. *arXiv preprint arXiv:2308.07968*, 2023.

[14] J. Li, M. Wang, J. Li, J. Fu, X. Shen, J. Shang, and J. 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*, pages 1258–1267, 2023.

[15] Z. Ma, Z. Dou, Y. Zhu, H. Zhong, and J.-R. 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*, pages 555–564, 2021.

[16] S. Maghsudi, A. Lan, J. Xu, and M. van Der Schaar. Personalized education in the artificial intelligence era: what to expect next. *IEEE Signal Processing Magazine*, 38(3):37–50, 2021.

[17] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[18] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 2023.

[19] N. Stiennon, L. Ouyang, J. Wu, D. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. F. Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

[20] L. von Werra, Y. Belkada, L. Tunstall, E. Beeching, T. Thrush, N. Lambert, and S. Huang. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`, 2020.

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

[22] Z. Wu, Y. Hu, W. Shi, N. Dziri, A. Suhr, P. Ammanabrolu, N. A. Smith, M. Ostendorf, and H. Hajishirzi. Fine-grained human feedback gives better rewards for language model training. *arXiv preprint arXiv:2306.01693*, 2023.

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

A Generation with Conflicting Preferences Experiment Details

A.1 Reddit TL;DR summarization dataset

In TL;DR[2] dataset, each comparison includes a Reddit post x, two summaries y_1 and y_2, the id of the worker who provided the annotation, and how y_1 and y_2 are sampled, e.g., from prior SFT or PPO checkpoints. As we do not have access to the SFT model used by [19], we initialize the personalized LM in P-DPO using an open-source, GPT-6B based SFT[3]. To ensure that the summaries are close to the distribution of this SFT, we only include the comparisons where both y_1 and y_2 are noted as sampled from the SFT models in the dataset, and exclude comparisons which contain summaries sampled from other policies such as different PPO checkpoints. In Section 3.1, we used the comparisons annotated by the the top 10 workers for preference simulation and P-DPO training.

A.2 P-DPO Experiment Details

All the LLMs in P-DPO experiments are initialized to the open-source, GPT-6B based SFT. For the TL;DR dataset, all models, including the vanilla DPO and all P-DPO models, are trained with $\beta = 0.5$, batch size 32, learning rate $5e - 5$ with a cosine learning schedule and 150 warm up steps for 2 epochs. We utilized LoRA [7] for training, with LoRA $\alpha = 16$, LoRA $r = 8$ and LoRA dropout 0.05. All models are trained with a PyTorch based, personalized DPO Trainer we develop by extending the DPO Trainer in the TRL library [20]. All of our experiments are run using 80G A100s or H100s.

B Personalization on Real-World Preference Dataset with Large Use Base

B.1 Experiment details on PRISM dataset

PRISM is a multi-turn conversation dataset with abundant annotations. Each conversation involves one user and several LLMs randomly selected from a pool (including strong models like GPT-4). The LLMs are prompted by the same system string from the user. In each turn, LLMs give their responses. The user first rate each of them and then select the best one to continue. 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 carefully splitted the dataset to stay close to the real-world situation. First, to test our model's generalization ability to unseen users, the users are separated into seen and unseen ones. Dialogues from unseen users only appear in the test split. Second, dialogues from seen users are also splitted into train and test splits, resulting in the final sample ratio of train/test split $6431 : 3982$.

On PRISM, we limit the maximum text length to 800 tokens, which is about the length of two turns in the conversation. All models, including the vanilla DPO and all P-DPO models, are trained with $\beta = 0.1$, batch size 32, learning rate $5e - 5$ with a cosine learning schedule and 150 warm up steps for 2 epochs. We utilized LoRA [7] for training, with LoRA $\alpha = 32$, LoRA $r = 8$ and LoRA dropout 0.1. All models are trained with a PyTorch based, personalized DPO Trainer we develop by extending the DPO Trainer in the TRL library [20]. For the win-rate evaluation, we leveraged the evaluation code in the github repository of [8]. All of our experiments are run using 80G A100s or H100s.

B.2 Evaluation details on PRISM dataset

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 as below.

[2] https://huggingface.co/datasets/openai/summarize_from_feedback

[3] https://huggingface.co/CarperAI/openai_summarize_tldr_sft

Table 2: An example role-play prompt for win-rate evaluation by GPT-4o.

User-wise information
The user has the following demographic information: - self description: I believe in responsability which is a main value. - age: 25-34 years old - gender: Female - education: University Bachelors Degree - employment: Student - marital: Never been married - english proficiency: Fluent
The user provided the following reasoning on why they choose the chosen responses for the annotations below: - Their expectation for the assistant: I would not like for it to just answers me thing like "I don´t know, You can find online". - Their feedback on the chosen response: It gave very complete answers to what i asked - Top three aspects of their preferences: values, diversity, personalisation
Conversation-wise information
Example 1 Instruction 1: which is the most violent country in history? Output (a) for example 1: I do not have definitive data to state which specific country has been the most violent throughout all of history. Output (b) for example 1: Making such a sweeping determination would require carefully analyzing the full context of violence and conflict across different eras in many civilizations over centuries or millennia. Result for example 1: Output (b)
Example 2 Instruction 2: what happened to the Romanov family? Output (a) for example 2: The Romanov family were the imperial dynasty that ruled Russia for over 300 years until the Russian Revolution in 1917... Output (b) for example 2: The Romanov family were the imperial dynasty that ruled Russia for over 300 years until the Russian Revolution in 1917. After the revolution, Tsar Nicholas II abdicated and he and his entire family were eventually executed by Bolshevik revolutionaries in July 1918 in an effort to fully eliminate the monarchy in Russia. Their bodies were disposed of in secret and their graves were undiscovered for decades. It was a tragic and brutal end for the royal Romanovs. Result for example 2: Output (b)
Annotation requirement
Now write your thought and choose the preferred output for final example. Write the preferred output (either "Output (a)", "TIE", or Output(b)") after the phrase "Result for final example:". Think before you speak.

B.3 PRISM Generation Examples

From the generation example in Table 3, we observe that the vanilla DPO model acknowledges user preferences (“Here are some suggestions based on your preferences”) and generates an overall friendly response, indicating that it has captured the explicit user preference for friendliness from the textual user information. However, on the topic of “alcohol drinking" which is not “work or school related", the response gradually sidetracks to other topics. In contrast, the original chosen response focuses on alcohol drinking but adopts a preachy tone rather than being friendly or comforting. Only the P-DPO model responds like a good listener, maintaining a friendly tone throughout.

Table 3: Generation examples from our best-performing P-DPO model with cluster-based user preferences, the vanilla DPO model, and the chosen response in dataset. It is clear that the generation from the personalized LLM fine-tuned by P-DPO best meets the user's expectation.

User utterance
Drinking alcohol is not a sin.
User's expectation
The Al language model should behave like a human being since it interacts with human beings. The Al must be informative and accurate in all times if it is being asked work or school related things. It should be reliable at all times. It should be friendly whenever someone needs a friend to talk to. It should also be a good listener if someone wants to offload their problems and wants someone to talk to.
P-DPO response
You're expressing your personal opinion on the topic of drinking alcohol. It's interesting to note that societal views on this subject can vary greatly depending on cultural background, upbringing, and individual values. Would you like to discuss any specific aspects or perspectives related to drinking alcohol? Or perhaps share why you hold this view?
vanilla DPO response
You don't have strong feelings against drinking alcohol. You might enjoy it or see no issue with it. Is there anything else you'd like to discuss? Here are some suggestions based on your preferences: 1. **Creative Writing**: We could engage in creative writing activities where we take turns adding to a story, poem, or dialogue. 2. Conversational Games: We can play games like ""Would you rather..."", ""Two Truths and a Lie"", or ""Hangman"" which promote fluency in conversation while being fun. 3. If you're interested in learning more about topics related to personal growth and development, we could explore subjects such as goal setting, time management, or productivity techniques. Please let me know if any of these options appeal to you.
chosen response in the dataset
But it's important to drink responsibly. Excessive consumption can lead to health problems, accidents, and social issues. Always prioritize your well-being and make wise choices when consuming alcohol. Cheers to moderation!