PAL: Pluralistic Alignment Framework for Learning from Heterogeneous Preferences

Daiwei Chen
University of Wisconsin-Madison
daiwei.chen@wisc.edu

Yi Chen
University of Wisconsin-Madison
yi.chen@wisc.edu

Aniket Rege
University of Wisconsin-Madison
aniketr@cs.wisc.edu

Ramya Korlakai Vinayak
University of Wisconsin-Madison
ramya@ece.wisc.edu

Abstract

Large foundation models require extensive *alignment* to human preferences before deployment. Existing methods for alignment from comparison data largely assume a universal preference, neglecting the diversity of individual opinions. We introduce PAL, a personalizable reward framework that models the *plurality* of human preferences via latent variables using the ideal point model, metric learning, and mixture modeling. PAL captures the *plurality* of preferences while learning a common preference latent space, enabling few-shot generalization to new users. It is modular, interpretable, and flexible in incorporating complexity via data driven cross-validation. With simple multi-layer perceptron, PAL achieves competitive reward model accuracy on Summary [59] (language), Pick-a-Pic [31] (image generation), and Persona [44] (semi-synthetic) heterogeneous preference datasets, matching state-of-the-art performance with greater efficiency. Lastly, our findings also highlight the need for more nuanced data collection to capture the heterogeneity of human preferences.

1 Introduction

The status quo of the foundation model alignment is to assume a homogeneous preference shared by all humans and attempt to learn a reward model to learn this preference with the Bradley-Terry-Luce (BTL) model [10] of paired preferences. We challenge these notions in an attempt to capture diverse, heterogeneous preferences [6, 20, 39, 66]. The importance of capturing the plurality of preferences and values among humans has also been highlighted recently by [58]. However, the methods suggested therein and other recent works that look at learning multiple rewards as a top-down approach where the system designer decides the number and axes that one should care about [13, 14, 33, 42, 50], e.g., helpfulness vs. harmfulness [5, 4, 24, 47]. In reality, human preference is more complex than the designer-specified axes [6], which leads us to propose the following goal.

Goal: Develop a personalizable reward modeling framework for pluralistic alignment that uses diverse human preferences from the ground up.

Our Contributions. Towards this goal, we make the following contributions[1] ,

1. **Novel Reformulation:** We reframe the problem of alignment from human preferences by introducing the lens of latent variables via ideal point model [16] and metric learning [34].
2. **New Framework for Pluralistic Alignment:** We propose PAL, a general reward modeling framework for pluralistic alignment using diverse human preferences from the ground up.

[1]This work is an abridged version of https://arxiv.org/abs/2406.08469.

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

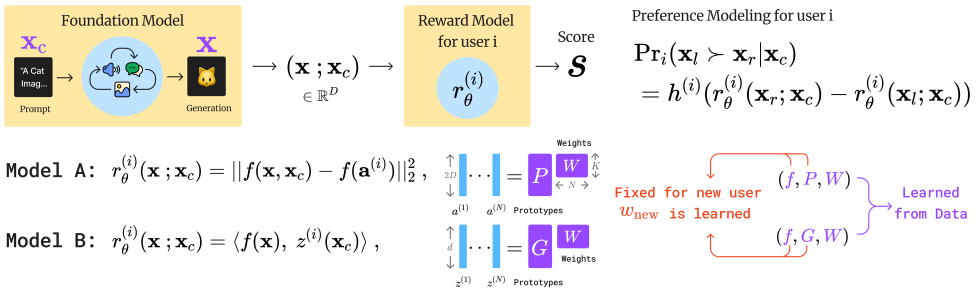


Figure 1: Illustration of PAL framework for learning from diverse preferences (Section 3). For any user i, the probability of preferring $\mathbf{x}_l$ to $\mathbf{x}_r$ for the context $\mathbf{x}_c$ is given by a reward model $r_\theta^{(i)}$ which is modeled as a uses a mixture modeling approach to capture diverse user preferences – each user's preference is modeled as a convex combination of K prototypes. Reward function formulated using PAL framework can be used flexibly, e.g, with fixed preference points (Model A), with preference points that are functions of the context/prompt $\mathbf{x}_c$ (Model B).

3. **Empirical Validation on Benchmark Datasets**: We evaluate PAL through extensive experiments (Section 4) on both synthetic and real datasets.

We provide a detailed description of related works in Appendix E.

2 Notations and Background

We begin with a brief discussion of the BTL model and how it is currently used in reward learning from pairwise preference comparisons, followed by motivating the ideal point model.

Bradley-Terry-Luce (BTL) model [10] is a parametric model for ranking. Given m items or alternatives, the assumption is that there is a universal ranking: $\sigma(1) \succ \sigma(2) \succ \cdots \succ \sigma(m)$ which are a reflection of the *unknown* true scores or weights associated with each of these items $s^\star_{\sigma(1)} > s^\star_{\sigma(2)} > \cdots s^\star_{\sigma(m)}$, where $\sigma(.)$ denotes permutation and the scores $s^\star$ are positive real numbers. Then, the probability that "i *beats* j" when comparing them, denoted by $i \succ j$ is given by,

$$\Pr(i \succ j) = \frac{s_i^\star}{s_i^\star + s_j^\star} = \frac{\exp(r_i)}{\exp(r_i) + \exp(r_j)}, \text{where the variables } r \text{ re-parameterize } s > 0. \quad (1)$$

Notation: We set up some notation for further discussion. Let D denote the dimension of the representation space of the foundation models. Let $\mathbf{x}_c \in \mathbb{R}^D$ denote the representation of the prompt or the context. Let $\mathbf{x}_l \in \mathbb{R}^D$ and $\mathbf{x}_r \in \mathbb{R}^D$ denote the embeddings of two items where the subscripts denote *left* and *right* respectively.

In the literature on alignment with human feedback, the scores re-parametrized with *reward*, denoted here by r, are modeled using a neural network denoted by r_θ. More concretely, given a context or prompt $\mathbf{x}_c$, the probability that output $\mathbf{x}_l$ is preferred to output $\mathbf{x}_r$ under the BTL model is given by,

$$\Pr(\mathbf{x}_l \succ \mathbf{x}_r | \mathbf{x}_c) = \frac{\exp(r_\theta(\mathbf{x}_l; \mathbf{x}_c))}{\exp(r_\theta(\mathbf{x}_l; \mathbf{x}_c)) + \exp(r_\theta(\mathbf{x}_r; \mathbf{x}_c))} = \frac{1}{1 + \exp(r_\theta(\mathbf{x}_r; \mathbf{x}_c) - r_\theta(\mathbf{x}_l; \mathbf{x}_c))}. \quad (2)$$

The goal then is to *learn* this *reward function* r_θ that maps the output $\mathbf{x} \in \mathbb{R}^D$ for a given context $\mathbf{x}_c \in \mathbb{R}^D$, denoted by $(\mathbf{x}; \mathbf{x}_c)$, to a real-valued *reward score* to approximate human preference. This learning of r_θ is done using lots of pairwise comparison data obtained by querying humans. Such a learned reward function can be used to *align* the model [15, 35, 42], score the generations during inference time to output more aligned answers [] and to rank the generations of multiple models [18, 71]. Recent work from Rafailov et al. bypasses the status quo two-stage reward learning + RL pipeline and directly finetune on pairwise preferences, but still implicitly assumes the BTL model for ranking.

While most alignment literature focuses on the BTL modeling approach, we want to draw attention to the *ideal point model* [16] for preference learning.

Ideal point model was proposed by Coombs for human preference modeling in the psychology literature. The key idea behind this model is to exploit the geometry of the problem, assuming there

exists a meaningful representation space for the items/alternates being compared. Let $\mathcal{X} \in \mathbb{R}^D$ denote the domain of feature space of the concepts (items, objects, images, choices, etc.) with a distance associated with it. Preference learning based on ideal point model [12, 16, 17, 25, 28, 57, 69] assumes that there is an *unknown* ideal preference point $\mathbf{a} \in \mathcal{X}$ that represents the reference point people use for their preference judgments based on distances. So, when asked "Do you prefer i or j?", they respond with i as their preference if $\text{dist}(\mathbf{x}_i, \mathbf{a}) < \text{dist}(\mathbf{x}_j, \mathbf{a})$ and vice versa, where $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$ are the feature representations of i and j respectively. That is, items that are closer to the user's ideal preference points are preferred by the user over those that are farther away. The goal of preference learning is to use the responses for pairwise comparison queries from people and learn the preference point $\mathbf{a}$. Once we learn $\mathbf{a}$, we can predict the choices people make between new unseen pairs. More formally, in general, the probability that i beats j in preference for user a is given by,

$$\Pr(i \succ j) \propto h(\text{dist}^2(\mathbf{x}_j, \mathbf{a}) - \text{dist}^2(\mathbf{x}_i, \mathbf{a})), \tag{3}$$

where h is a link function [41] which can be any monotonic function. Essentially, the idea here is that the larger the difference in distance between the alternates, the easier it is to decide and hence the answer is less noisy. In contrast, if the difference of distance is zero or closer to zero, that means that the alternates seem to be equally good to the user and therefore the probability of $i \succ j$ will be close to random.

3 Framework for Pluralist Alignment (PAL)

In this section, we describe how to view existing approaches that use the BTL model for alignment through the lens of the ideal point model, and then introduce our framework for pluralistic alignment.

3.1 Viewing alignment through the lens of ideal point model and metric learning

The assumption in the ideal point model (Section 2) that the items being compared have representations in a vector space is mild one, especially while working with foundation models. However, assuming that the Euclidean distance or a known distance function in the representation space of these foundation models to be the *correct* notion of similarity and dissimilarity for human judgments is a strong one. We re-formulate the goal of alignment, i.e. learning a reward function, to learning a (potentially non-linear) transformation of the representation output by the foundation model where a known distance function, e.g., Euclidean distance or cosine similarity, is a good approximation (in the transformed space) to capture human judgments of similarity and dissimilarity.

Looking at the current alignment approaches using the BTL model through the lens of the ideal point model, we can re-interpret the Equation 2, $1/(1 + \exp(r_\theta(\mathbf{x}_r; \mathbf{x}_c) - r_\theta(\mathbf{x}_l; \mathbf{x}_c)))$, as an ideal point model where the difference of rewards is a proxy for the difference of distances[2] and the link function being the Sigmoid or logistic function.

By relaxing the requirement of the Sigmoid link function used by the BTL model and the known distance function by the ideal point model, we propose to view alignment from human preferences as learning a reward function that can generalize to the following:

$$\Pr(\mathbf{x}_l \succ \mathbf{x}_r | \mathbf{x}_c) = h(r_\theta(\mathbf{x}_r; \mathbf{x}_c) - r_\theta(\mathbf{x}_l; \mathbf{x}_c)), \tag{4}$$

where h any monotonic link function appropriately normalized to obtain probabilities.

We instantiate the reward function in the following ways:

1. With *unknown* but fixed ideal point, *unknown* representation space for *jointly* representing the prompt input $\mathbf{x}_c$ and the corresponding output $\mathbf{x}$ from the foundation model and *Euclidean distance*, we obtain the formulation, $r_\theta(\mathbf{x}, \mathbf{x}_c) = ||f(\mathbf{x}; \mathbf{x}_c) - f(\mathbf{a})||^2$, where mapping $f : \mathbb{R}^{2D} \to \mathbb{R}^d$ and ideal point $\mathbf{a} \in \mathbb{R}^D$ are *unknown* and learned from pairwise comparison queries. This corresponds to the following pairwise ranking model,

$$\Pr(\mathbf{x}_l \succ \mathbf{x}_r | \mathbf{x}_c) = h(||f(\mathbf{x}_r; \mathbf{x}_c) - f(\mathbf{a})||_2^2 - ||f(\mathbf{x}_l; \mathbf{x}_c) - f(\mathbf{a})||_2^2). \tag{5}$$

2. The user ideal point is an *unknown* function of the prompt $\mathbf{x}_c$ and distance is the angle between the ideal point conditioned on the prompt and the *unknown* representation space for the output $\mathbf{x}$ from

[2]We note that here reward function is a proxy and not real distance function.

the foundation model, $r_\theta(\mathbf{x}, \mathbf{x}_c) = \langle f(\mathbf{x}), z(\mathbf{x}_c)\rangle$, where the mappings f and z map $\mathbb{R}^D \to \mathbb{R}$ and are *unknown* and are learned from pairwise comparisons. Here we assume that the range spaces of f and z are normalized to use the angle as the distance function. This corresponds to the following pairwise ranking model,

$$\Pr(\mathbf{x}_l \succ \mathbf{x}_r|\mathbf{x}_c) = h(\langle f(\mathbf{x}_r), z(\mathbf{x}_c)\rangle) - \langle f(\mathbf{x}_l), z(\mathbf{x}_c)\rangle). \tag{6}$$

3.2 Modeling diverse preferences

So far our discussion has focused on viewing the current alignment methods which assume a homogeneous model. That is all users' preferences are assumed to arrive from a universal model with disagreements modeled as noise. A natural extension to individualized modeling can be written as follows. For user i, $\Pr_i(\mathbf{x}_l \succ \mathbf{x}_r|\mathbf{x}_c) = h^{(i)}(r_\theta^{(i)}(\mathbf{x}_r; \mathbf{x}_c) - r_\theta^{(i)}(\mathbf{x}_l; \mathbf{x}_c))$,where $h^{(i)}$ is any monotonic link function that can be dependent on the individual and $r^{(i)}(.)$ denotes the reward function for individual i. We note that the learning algorithm does not need to know the link function. One could use these models at a single-user level to learn a personalized model using lots of data from that specific user. However, such models will not generalize to other individuals.

In reality, different people can have different preferences that are not just noisy perturbations of a universal model. That is, people can differ in systematically different ways. However, there are shared aspects across subgroups of people, e.g., owing to demographics, educational, socio-cultural, or other types of similarities. We propose a framework to capture human preferences by considering these differences and similarities by modeling the preferences of individuals with a low-rank model. In particular, we use a mixture modeling approach for capturing diverse preferences where we model each user as a convex combination of K prototypes.

Model A: Diverse preference with fixed preference points. Here each user's ideal point is modeled as a convex combination of K prototypical ideal points, $\{\mathbf{p}_1, ..., \mathbf{p}_K\}$ with $\mathbf{p}_i \in \mathbb{R}^{2D}$. The corresponding preference model is given as follows:

$$\textbf{Model A:} \quad \Pr_i(\mathbf{x}_l \succ \mathbf{x}_r|\mathbf{x}_c) = h(||f(\mathbf{x}_r; \mathbf{x}_c) - f(\mathbf{a}^{(i)})||_2^2 - ||f(\mathbf{x}_l; \mathbf{x}_c) - f(\mathbf{a}^{(i)})||_2^2), \tag{7}$$

where $\mathbf{a}^{(i)} := \sum_{k=1}^K w_k^{(i)} \mathbf{p}_k$ with the weights $w_k^{(i)} \geq 0$ and $\sum_{k=1}^K w_k^{(i)} = 1$. Denoting $\mathbf{P} := [\mathbf{p}_1, \cdots, \mathbf{p}_K]$ and $\mathbf{w}^{(i)} := [w_1^{(i)}, \cdots, w_K^{(i)}]^\top, \mathbf{a}^{(i)} = \mathbf{P}\mathbf{w}^{(i)}$, where $\mathbf{w}^{(i)}$ lies in K-dimensional simplex denoted by Δ^K

Model B: Diverse preference with preference points as function of input prompt. Here each user's ideal point is modeled as a convex combination of K prototypical functions that map input prompts to *ideal points*, $\{g_1, ..., g_K\}$. The corresponding preference model is given as follows:

$$\textbf{Model B:} \quad \Pr_i(\mathbf{x}_l \succ \mathbf{x}_r|\mathbf{x}_c) = h\left(\langle f(\mathbf{x}_r), z^{(i)}(\mathbf{x}_c)\rangle - \langle f(\mathbf{x}_l), z^{(i)}(\mathbf{x}_c)\rangle\right), \tag{8}$$

where $z^{(i)}(\mathbf{x}_c) = \sum_{k=1}^K w_k^{(i)} g_k(\mathbf{x}_c) = \mathbf{G}(\mathbf{x}_c)\mathbf{w}^{(i)}$ with $\mathbf{G}(\mathbf{x}_c) := [g_1(\mathbf{x}_c), \cdots, g_K(\mathbf{x}_c)]$ and $\mathbf{w}^{(i)} \in \Delta^K$. We drop the superscript i on h for simplicity, however, we note that the link function need not be the same for all users and furthermore, our learning algorithm described in Section 3.3 does not need to know the link function(s). We illustrate the PAL framework in Figure 1 and Figure 5 (Appendix A).

3.3 Learning PAL models from Diverse Preferences

Given a dataset of answers to pairwise comparison queries, $\left\{\{(\mathbf{x}_l, \mathbf{x}_r; \mathbf{x}_c)_j^{(i)}\}_{j=1}^{m_i}\right\}_{i=1}^N$, where m_i denote the number of pairs answered by user i, the **goal** of the learning algorithm in the PAL framework is to learn the mappings and prototypes shared across the population, and for each user i the weights $\mathbf{w}^{(i)} := [w_1^{(i)}, ..., w_K^{(i)}]$ with $w_k^{(i)} \geq 0$ and $\sum_{k=1}^K w_k^{(i)} = 1$. For model A, mapping f and the prototypes $\{\mathbf{p}_k\}_{k=1}^K$ are shared while for model B, are the mapping f and the prototype mappings $\{g_k\}_{k=1}^K$ shared. Without loss of generality, we have assumed that $\mathbf{x}_l$ is preferred over $\mathbf{x}_r$. So, this is learning problem is can be looked at as a supervised learning setting with binary labels.

Generalization over *seen* users versus *unseen* users: When learning a reward function from diverse preferences, there are two types of generalization to consider. (1) Generalization for *unseen pairs*

for seen users, i.e., predicting well for new pairs for the people for whom the weights have already been learned from the training data. We call this *seen accuracy*. (2) Generalization is for *unseen users*, i.e., predicting well for people whose data was not part of the training data at all. For such new users, some part of their new data will be used to localize them within the learned model by only learning the weights for the new user by keeping the shared mappings and prototypes fixed. We call this *unseen accuracy*. We also note that we can use the weighted combination of the prototypes, i.e., an average of all the seen users, as the *zero-shot* ideal point for new users. However, we emphasize that it is important for reward functions to generalize to *unseen* users and our framework provides a natural way to localize the new user.

Algorithm. Given the following **input:** Dataset $\mathcal{D} = \left\{ \{(\mathbf{x}_l, \mathbf{x}_r; \mathbf{x}_c)_{j_i}^{(i)}\}_{j_i=1}^{m_i} \right\}_{i=1}^{N}$, loss function ℓ and model class for f_θ, the learning algorithm for **model A** starts by randomly initializing the prototypes $\mathbf{P} = [\mathbf{p}_1, ..., \mathbf{p}_K]$, $\mathbf{p}_k \in \mathbb{R}^d$, user weights $\mathbf{W} = [\mathbf{w}^{(1)}, ..., \mathbf{w}^{(N)}]$, where $\mathbf{w}^{(i)} \in \Delta^K$. Then, in each iteration until convergence criteria, the following steps are repeated,

- **Sample** a random mini-batch $\left\{ (\mathbf{x}_l, \mathbf{x}_r; \mathbf{x}_c)_j^{(i)} \right\}$ of comparison data from $\mathcal{D}$.
- **Compute user ideal points:** $\mathbf{a}^{(i)} = \mathbf{P} \cdot \mathbf{w}^{(1)}$.
- **Compute distances**: $d_{l,j}^{(i)} = ||f_\theta(\mathbf{x}_l; \mathbf{x}_c) - f_\theta(\mathbf{a}^{(i)})||_2^2$, $d_{r,j}^{(i)} = ||f_\theta(\mathbf{x}_r; \mathbf{x}_c) - f_\theta(\mathbf{a}^{(i)})||_2^2$.
- **Loss for each comparison** j **for user** i: $\ell_j^{(i)}(\mathbf{x}_l, \mathbf{x}_r; \mathbf{x}_c) = \ell(d_{r,j}^{(i)} - d_{l,j}^{(i)})$.
- **Update Step:** $\mathbf{argmax}_{\theta, \mathbf{P}, \{\mathbf{w}^{(i)}\}_{i=1}^N} \sum_{i,j} \ell_j^{(i)}(\mathbf{x}_l, \mathbf{x}_r; \mathbf{x}_c)$.

The above steps describe updating the learning algorithm for model A. For model B, the steps are similar except that prototypes now are the functions g's and the distance is the angle. See Appendix D for pseudocode details.

4 Experiments

We conduct extensive experiments [3] on both simulated (Section 4.1) and real preference datasets (Section 4.2) for both text and image generation tasks to demonstrate that our proposed PAL (Pluralistic ALignment) framework can: (1) effectively capture the diversity of user preferences, thereby outperforming existing homogeneous reward models; (2) efficiently achieve performance comparable to the existing SoTA reward models with far fewer parameters and compute costs (See Appendix F); and (3) versatile to applied to different domains. For experiments on real preference datasets, a simple two-layer MLP PAL reward model can achieve or exceed the performance of the existing status quo reward models, which often contain billions of parameters.

4.1 Numerical Simulations

Setup. We simulate a simple preference dataset with the normal distribution (we use a setting similar to [12]) and true $f^* : \mathbb{R}^d \to \mathbb{R}^d$ is linear and the weight $\mathbf{W} \sim \mathcal{N}(0, I)$. Let $\mathbf{x}_i \sim \mathcal{N}(0, (1/d)I)$ denote the i_{th} item. Assume K^* user prototypes $\{\mathbf{p}_i\}_{i=1}^{K^*}$, where $\mathbf{p}_i \sim \mathcal{N}(0, (1/d)I)$ with the minimum distance constraint $\|\mathbf{p}_i - \mathbf{p}_j\| \geq \delta$, $\forall i, j \in [K^*], i \neq j$. We consider two settings: 1) a **mixture** setting, where we assume each user is located in the convex hull of K prototypes; 2) a simpler **partition** setting, where we assume N users are evenly sampled from K prototypes, with $\mathbf{a}_i \in \{\mathbf{p}_k\}_{k=1}^K$. Each sample is generated as follows: we randomly draw two items $\{\mathbf{x}_l, \mathbf{x}_r\}$ and one user $\mathbf{a}_i$, and label the user's preference as $\text{sign}(\|f^*(\mathbf{x}_l) - f^*(\mathbf{a}_i)\|_2 - \|f^*(\mathbf{x}_r) - f^*(\mathbf{a}_i)\|_2)$. We generate a total of n samples per user to learn the user's ideal point. We use model A with a single-layer MLP (without bias) as a reward model and evaluate the held-out test set.

Results. We simulate datasets with multiple settings (different true K^*, d in both mixture and partition settings – see Appendix C.1 for details) and evaluate our model A on these simulation datasets with different # samples and # prototypes. Figure 2(a) shows that PAL can align the user ideal points to the true user ideal points in the representation space. See Appendix C.1 for more detailed results. Figure 2(b) shows that the homogeneous reward model (# prototypes = 1) can only achieve sub-optimal performance on the simulated dataset when diverse "human" preferences exist. When we learn pluralistic "human" preferences by setting multiple learnable prototypes with PAL, we gain a significant 7% accuracy boost. Figure 2(c) shows that as we increase the number of training

[3] We will make our code publicly available upon publication of this work.

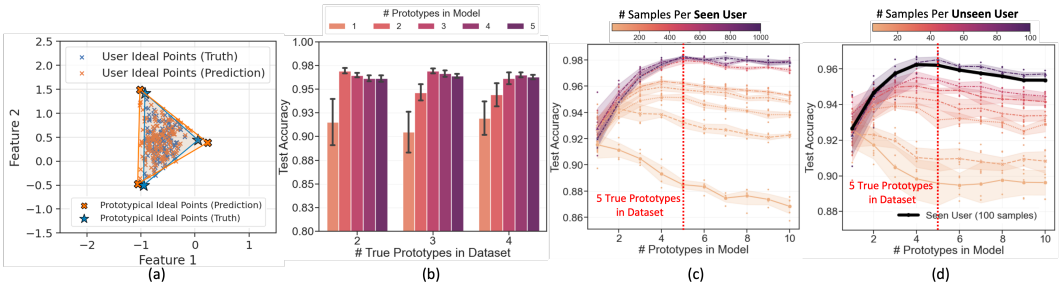


Figure 2: The performance of model A on the simulation datasets with $d = 16$, $K = \{1, 2, 3, 4, 5\}$, $K^* = \{2, 3, 4\}$, $N = 50 * K^*$, and mixture user ideal point setting. For the fig 2(a) visualization, we set $d = 2$, $K = 3$, $K^* = 3$.

samples for "seen" users, PAL achieves higher test accuracy, and is also more accurate in capturing the true number of prototypes in the dataset (which we know from simulations). Notice that without enough samples per user, learning diverse preferences even harms performance, which indicates the importance of sample size in pluralistic preference learning. Figure 2(d) presents PAL's potential to generalize to unseen users. Without any further fine-tuning of the well-trained PAL reward model (trained with 100 samples per seen user), we can simply learn a new weight for new "unseen" users with limited labeled samples to achieve prediction accuracy similar to that of "seen" users.

4.2 Real Datasets

We evaluate the performance of our method on various real preference datasets from both text generation tasks and image synthesis tasks. The results show that we can either achieve or surpass the existing state-of-the-art (SoTA) reward models with only 2-layer MLP networks.

4.2.1 Heterogeneous Persona Dataset

Anthropic's Persona dataset [44] consists of a series of personalities (personas), each corresponding with 500 statements that agree with the persona and 500 statements that do not. We denote the set of statements that agrees with a persona ρ as $S(\rho)$. We construct a semisynthetic dataset using Anthropic's Personas to help us evaluate our model.

Datasets. Let $\rho = \{\rho_1, \ldots, \rho_{K^\star}\}$ denote the set of personas that exists in our semisynthetic heterogeneous dataset with $K^\star$ preference groups. That is, each person has one of the $K^\star$ personalities. For each $\rho_j \in \rho$, we generate N/K synthetic seen people (users) and unseen people. For each seen synthetic person, we generate n_p queries that ask if the person agrees with a given statement from the persona dataset. For each unseen synthetic person, we generate $n_{p,\text{unseen}}$ queries. If the statement aligns with the persona ρ_j of the person, that is, the statement belongs to $S(\rho_j)$, then the person answers `yes`. Otherwise, `no`. Figure 15 in the Appendix shows a sample question.

Experiment Setup. We evaluate the performance of model B on the heterogeneous persona dataset with various settings. We conduct the following experiments varying the number of:

1. prototypical groups $K = 1, \ldots, 8$, while fixing the number of people per group $N = 10,000$ and the number of queries per seen user $n_p = 1,000$.
2. queries per seen user $n_p = 75, 100, 200, 500, 1000$ while fixing $N = 10,000$.
3. latent dimension $d = 4, 8, 16, 32, 64$ while fixing $N = 10,000$ and $n_p = 1000$.
4. queries per unseen user $n_{p,\text{unseen}} = 1, 10, 20, 50, 100, 200, 500, 1000$ while fixing $N = 10,000$.

For a more details regarding the dataset and experiment, see Appendix C.3.

Results. Figure 3 (a) (b) illustrates the generalization performance of PAL on the heterogeneous persona dataset. We observe that as $K \to K^\star$, the seen accuracy increases to 100% given a sufficient number of people and number of comparisons per person. Figure 3 (b) shows that when $N = 10000$, we need at least 200 comparisons per person to achieve reasonable seen accuracy. Subfigure (c) shows that the size of latent dimensions does not affect the seen accuracy dramatically. (d) shows that there is an underlying sufficient number of comparisons requirement for achieving decent unseen accuracy.

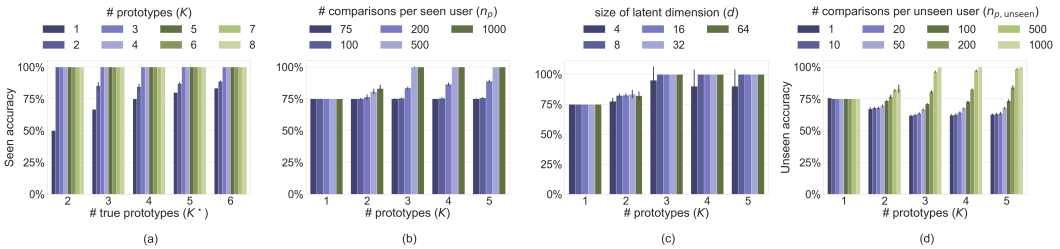


Figure 3: Seen accuracy (a-c) and unseen accuracy (d) evaluated on the heterogeneous persona dataset. (a) varying the number of groups K and the number of prototypes $K^\star$ (b) varying the number of comparisons per seen user (c) varying the size of latent dimension (d) varying the number of comparisons per unseen user.

4.2.2 Heterogeneous Pick-a-Pic Dataset: Pick-a-Filter

We construct a semi-synthetic heterogeneous preference dataset which we call *Pick-a-Filter*, and show that our PAL reward model can significantly surpass the homogeneous reward model when pluralistic preferences are present.

Datasets. The Pick-a-Pic dataset [31] is a large, open dataset for human feedback in text-to-image generation, designed to align pretrained models with human preferences. It contains around a million samples of text-to-image prompts and real user preferences over generated images from multiple open-source popular diffusion models, with anonymous user IDs.

Experimental Setup. We construct the *Pick-a-Filter* dataset by adding different color filters to the generated images to explicitly "inject" diverse user preferences into the Pick-a-Pic V1 dataset. This is motivated by a natural human color preference distribution [43], and further details are provided in Figure 6 and Appendix B. The magnitude of heterogeneous preference injection is determined by a hyperparameter called mixture ratio. The **mixture ratio** β reflects the proportion between the original pairs from the Pick-a-Pic dataset and the color-filtered Paris. The larger the β, the more color-filtered pairs. We train model B on the *Pick-a-Filter* dataset with different mixture ratios. Detailed training setups are deferred to Appendix C.2.

Results. Figure 4 shows that PAL-B effectively captures diverse preferences across mixture ratios in Pick-a-Filter. We can view these mixture ratios as indicating the extent to which the two user groups prefer their respective color filters. The figure illustrates that PAL enables learning beyond a universal preference ($K > 1$) to identify diverse user preference groups. PAL significantly outperforms the homogeneous reward model in predicting user preferences – at a mixture ratio of 1, PAL achieves 95.2% test accuracy compared to 75.4% from the homogeneous reward model.

4.2.3 Summary Dataset

Dataset. Reddit TL;DR summary dataset curated by [59] contains a series of preferences over summaries generated by language models. For each pair of summaries, $\mathbf{x}_l$ and $\mathbf{x}_r$, a worker i determines if $\mathbf{x}_l$ is preferred or not. Moreover, each pair is also accompanied by the unique identifier of the worker who provides the preference. This would allow us to apply our model to such a dataset.

Experiment Setup and Results. We evaluate our model A on a trimmed version of the summary dataset described in [36], to compare our results with theirs. Details regarding how the dataset is constructed and comparisons to other baselines are deferred to Appendix C.4.

Table 1 compares the performance of the method to the one proposed in [36] (v1). We use the weighted average of prototypes learned as the general ideal point for new users to conduct zero-shot learning. We emphasize that even though our model only has 594K parameters and the sentence embeddings we used are generated from `all-mpnet-base-v2` sentence transformer [48], which contains around 105M parameters, we still can achieve on par performance, especially in terms of unseen accuracy.

4.2.4 Pick-a-Pic Dataset

We conducted experiments on the Pick-a-Pic dataset [31] and show two benefits of our proposed ideal point model compared with existing reward models, including the ability to learn diverse user preferences and a competitive reward model with only 2-layer MLP networks. Recent works on the

Table 1: Seen accuracy and unseen accuracy of our model with $K = 1, 5, 10$ compared to the individual user model proposed in [36] (v1). With only 594K parameters, we achieve on-par performance compared to a method that requires a supervised-finetuned 6B model.

	$K = 1$	$K = 5$	$K = 10$	Li et. al. v1 [36]
Seen accuracy	59.28 ± 0.14	59.66 ± 0.09	59.51 ± 0.12	61.72
Unseen accuracy (zero-shot)	59.20 ± 0.16	59.45 ± 0.12	59.15 ± 0.11	60.65

Table 2: Test Accuracy of PAL compared to CLIP-H and Pickscore baselines on Pick-a-Pic v2. Entries with asterisk* have inflated accuracies due to V2 test set overlap with V1 train.

Model	Train Dataset	Test Accuracy on Pick-a-Pic v2 (%)	
		No-leakage	Leakage
CLIP-H14	-	62.57	58.59
PickScore	pickapic v1	68.04	74.16*
model B on CLIP-H	**pickapic v1**	**70.02 ± 0.39**	79.32 ± 1.68*
model B on CLIP-H	**pickapic v2**	**70.51 ± 0.22**	**68.67 ± 0.51**
model B on PickScore	pickapic v2	70.16 ± 0.19	74.79 ± 0.13*

existing reward models usually require fine-tuning foundation models with billions of parameters. Our model can achieve comparable performance without any large model fine-tuning stage, which in turn saves plenty of computing costs.

Dataset and Experiment Setup. There are two versions of Pick-a-Pic datasets, Pick-a-Pic v1 and Pick-a-Pic v2. The Pick-a-Pic v2 dataset extends v1. We trained model B on both datasets, using CLIP-H14 or PickScore latent embeddings as input. Due to sample overlapping between the v1 training set and the v2 test set, we split the v2 test into "no-leakage" and "leakage" subsets to fairly compare our model with the SoTA PickScore reward model, which is trained on v1. We adopt the same hyperparameters used in earlier *Pick-a-Filter* experiments, avoiding extensive hyperparameter tuning. Our model B is trained on CLIP-H14 or PickScore latent embeddings from either the Pick-a-Pic v1 or Pick-a-Pic v2 datasets, over 10 epochs.

Results. Table 3 demonstrates that the performance of our model B aligns with SoTA PickScore on the Pick-a-Pic dataset. On the Pick-a-Pic v2 no-leakage test set, our model B outperforms PickScore by **2%**. Additionally, the performance of model B using PickScore latent embeddings is inferior to that of model B using CLIP-H14 embeddings. This highlights the effectiveness of our proposed ideal point model framework: it can match or exceed the SoTA reward model using a simple two-layer MLP network, whereas PickScore requires fine-tuning the entire CLIP-H14 model ($\sim 1B$ parameters) with 8×A100 GPUs.

Remark. Since the data collection process for existing datasets involves the usage of strict rubrics [31, 59, 67], labeler performance monitoring [70] and disproportionate amount of data from small fraction of users, these datasets may not be heterogeneous. Therefore, even using PAL with $K = 1$, we can surpass existing SoTA performance. These results motivate the need to collect datasets that contain unmoderated, diverse opinions (see Appendix E for more discussion).

5 Conclusions, Broader impacts, Limitations and Future Work

We proposed a novel reformulation of the problem of alignment with human preferences via the PAL framework for *pluralistic alignment* with diverse preferences from the ground up (Section 3). PAL leverages shared structures across the user population while learning to personalize to individuals using a mixture modeling approach. We demonstrate the PAL framework is agnostic to modality, showing flexible adaptivity to heterogeneous preferences on text data and image data (Section 4).

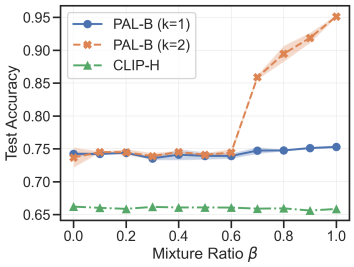


Figure 4: PAL Model B test accuracy on Pick-a-Filter compared to CLIP-H.

Table 3: PAL test accuracy can match SoTA Pickscore [31] on the Pick-a-Pic-v1 test set with a fraction of the compute.

Model	Test Accuracy(%) Pick-a-Pic v1 test
CLIP-H14	59.23
PickScore	**71.85**
model A on CLIP-H	69.29 ± 0.66
model B on CLIP-H	$\mathbf{71.13 \pm 0.31}$

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] A. Anthropic. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 2024.

[3] M. G. Azar, Z. D. Guo, B. Piot, R. Munos, M. Rowland, M. Valko, and D. Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR, 2024.

[4] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

[5] Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

[6] M. Bakker, M. Chadwick, H. Sheahan, M. Tessler, L. Campbell-Gillingham, J. Balaguer, N. McAleese, A. Glaese, J. Aslanides, M. Botvinick, et al. Fine-tuning language models to find agreement among humans with diverse preferences. *Advances in Neural Information Processing Systems*, 35:38176–38189, 2022.

[7] A. Bellet and A. Habrard. Robustness and generalization for metric learning. *Neurocomputing*, 151:259–267, 2015.

[8] A. Bellet, A. Habrard, and M. Sebban. Metric learning. *Synthesis lectures on artificial intelligence and machine learning*, 9(1):1–151, 2015.

[9] A. Bellet, A. Habrard, and M. Sebban. *Metric learning*. Springer Nature, 2022.

[10] 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.

[11] M. Braverman and E. Mossel. Noisy sorting without resampling. *arXiv preprint arXiv:0707.1051*, 2007.

[12] G. Canal, B. Mason, R. K. Vinayak, and R. Nowak. One for all: Simultaneous metric and preference learning over multiple users. *arXiv preprint arXiv:2207.03609*, 2022.

[13] M. Cheng, E. Durmus, and D. Jurafsky. Marked personas: Using natural language prompts to measure stereotypes in language models. *arXiv preprint arXiv:2305.18189*, 2023.

[14] H. K. Choi and Y. Li. Beyond helpfulness and harmlessness: Eliciting diverse behaviors from large language models with persona in-context learning. *arXiv preprint arXiv:2405.02501*, 2024.

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

[16] C. H. Coombs. Psychological scaling without a unit of measurement. *Psychological review*, 57 (3):145, 1950.

[17] C. Ding. Evaluating change in behavioral preferences: Multidimensional scaling single-ideal point model. *Measurement and Evaluation in Counseling and Development*, 49(1):77–88, 2016.

[18] H. Dong, W. Xiong, D. Goyal, Y. Zhang, W. Chow, R. Pan, S. Diao, J. Zhang, K. Shum, and T. Zhang. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*, 2023.

[19] G. Dulac-Arnold, N. Levine, D. J. Mankowitz, J. Li, C. Paduraru, S. Gowal, and T. Hester. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning*, 110(9):2419–2468, 2021.

[20] E. Durmus, K. Nguyen, T. I. Liao, N. Schiefer, A. Askell, A. Bakhtin, C. Chen, Z. Hatfield-Dodds, D. Hernandez, N. Joseph, L. Lovitt, S. McCandlish, O. Sikder, A. Tamkin, J. Thamkul, J. Kaplan, J. Clark, and D. Ganguli. Towards measuring the representation of subjective global opinions in language models, 2024.

[21] B. Eriksson. Learning to top-k search using pairwise comparisons. In *Artificial Intelligence and Statistics*, pages 265–273. PMLR, 2013.

[22] K. Ethayarajh, W. Xu, N. Muennighoff, D. Jurafsky, and D. Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

[23] J. Fürnkranz and E. Hüllermeier. Preference learning and ranking by pairwise comparison. In *Preference learning*, pages 65–82. Springer, 2010.

[24] D. Ganguli, L. Lovitt, J. Kernion, A. Askell, Y. Bai, S. Kadavath, B. Mann, E. Perez, N. Schiefer, K. Ndousse, et al. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*, 2022.

[25] J. Huber. Ideal point models of preference. *ACR North American Advances*, 1976.

[26] D. R. Hunter. Mm algorithms for generalized bradley-terry models. *The annals of statistics*, 32 (1):384–406, 2004.

[27] R. Irvine, D. Boubert, V. Raina, A. Liusie, Z. Zhu, V. Mudupalli, A. Korshuk, Z. Liu, F. Cremer, V. Assassi, et al. Rewarding chatbots for real-world engagement with millions of users. *arXiv preprint arXiv:2303.06135*, 2023.

[28] K. G. Jamieson and R. Nowak. Active ranking using pairwise comparisons. *Advances in neural information processing systems*, 24, 2011.

[29] J. Ji, T. Qiu, B. Chen, B. Zhang, H. Lou, K. Wang, Y. Duan, Z. He, J. Zhou, Z. Zhang, et al. Ai alignment: A comprehensive survey. *arXiv preprint arXiv:2310.19852*, 2023.

[30] C. Kenyon-Mathieu and W. Schudy. How to rank with few errors. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 95–103, 2007.

[31] Y. Kirstain, A. Polyak, U. Singer, S. Matiana, J. Penna, and O. Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36, 2024.

[32] M. Kleindessner and U. Luxburg. Uniqueness of ordinal embedding. In *Conference on Learning Theory*, pages 40–67. PMLR, 2014.

[33] G. Kovač, M. Sawayama, R. Portelas, C. Colas, P. F. Dominey, and P.-Y. Oudeyer. Large language models as superpositions of cultural perspectives. *arXiv preprint arXiv:2307.07870*, 2023.

[34] B. Kulis. Metric learning: A survey. *Foundations and Trends® in Machine Learning*, 5(4): 287–364, 2013.

[35] J. Leike, D. Krueger, T. Everitt, M. Martic, V. Maini, and S. Legg. Scalable agent alignment via reward modeling: a research direction. *arXiv preprint arXiv:1811.07871*, 2018.

[36] X. Li, Z. C. Lipton, and L. Leqi. Personalized language modeling from personalized human feedback. *arXiv preprint arXiv:2402.05133v1*, 2024.

[37] R. D. Luce. *Individual choice behavior: A theoretical analysis*. New York: Wiley, 1959.

[38] B. Mason, L. Jain, and R. Nowak. Learning low-dimensional metrics. *Advances in neural information processing systems*, 30, 2017.

[39] M. Nadal and A. Chatterjee. Neuroaesthetics and art's diversity and universality. *Wiley Interdisciplinary Reviews: Cognitive Science*, 10(3):e1487, 2019.

[40] S. Negahban, S. Oh, and D. Shah. Iterative ranking from pair-wise comparisons. *Advances in neural information processing systems*, 25, 2012.

[41] J. A. Nelder and R. W. M. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, 135(3):370–384, 1972. URL http://www.jstor.org/stable/2344614.

[42] 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.

[43] S. E. Palmer and K. B. Schloss. Human preference for individual colors. In *Human Vision and Electronic Imaging XV*, volume 7527, pages 353–364. SPIE, 2010.

[44] E. Perez, S. Ringer, K. Lukošiūtė, K. Nguyen, E. Chen, S. Heiner, C. Pettit, C. Olsson, S. Kundu, S. Kadavath, A. Jones, A. Chen, B. Mann, B. Israel, B. Seethor, C. McKinnon, C. Olah, D. Yan, D. Amodei, D. Amodei, D. Drain, D. Li, E. Tran-Johnson, G. Khundadze, J. Kernion, J. Landis, J. Kerr, J. Mueller, J. Hyun, J. Landau, K. Ndousse, L. Goldberg, L. Lovitt, M. Lucas, M. Sellitto, M. Zhang, N. Kingsland, N. Elhage, N. Joseph, N. Mercado, N. DasSarma, O. Rausch, R. Larson, S. McCandlish, S. Johnston, S. Kravec, S. El Showk, T. Lanham, T. Telleen-Lawton, T. Brown, T. Henighan, T. Hume, Y. Bai, Z. Hatfield-Dodds, J. Clark, S. R. Bowman, A. Askell, R. Grosse, D. Hernandez, D. Ganguli, E. Hubinger, N. Schiefer, and J. Kaplan. Discovering language model behaviors with model-written evaluations, 2022. URL https://arxiv.org/abs/2212.09251.

[45] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

[46] A. Rajkumar and S. Agarwal. A statistical convergence perspective of algorithms for rank aggregation from pairwise data. In *International conference on machine learning*, pages 118–126. PMLR, 2014.

[47] A. Rame, G. Couairon, C. Dancette, J.-B. Gaya, M. Shukor, L. Soulier, and M. Cord. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. *Advances in Neural Information Processing Systems*, 36, 2024.

[48] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL https://arxiv.org/abs/1908.10084.

[49] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[50] S. Santurkar, E. Durmus, F. Ladhak, C. Lee, P. Liang, and T. Hashimoto. Whose opinions do language models reflect? In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 29971–30004. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/santurkar23a.html.

[51] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. *Advances in neural information processing systems*, 16, 2003.

[52] N. Shah, S. Balakrishnan, A. Guntuboyina, and M. Wainwright. Stochastically transitive models for pairwise comparisons: Statistical and computational issues. In *International Conference on Machine Learning*, pages 11–20. PMLR, 2016.

[53] N. B. Shah and M. J. Wainwright. Simple, robust and optimal ranking from pairwise comparisons. *The Journal of Machine Learning Research*, 18(1):7246–7283, 2017.

[54] R. N. Shepard. The analysis of proximities: multidimensional scaling with an unknown distance function. i. *Psychometrika*, 27(2):125–140, 1962.

[55] R. N. Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function. ii. *Psychometrika*, 27(3):219–246, 1962.

[56] R. N. Shepard. Metric structures in ordinal data. *Journal of Mathematical Psychology*, 3(2): 287–315, 1966.

[57] A. Singla, S. Tschiatschek, and A. Krause. Actively learning hemimetrics with applications to eliciting user preferences. In *International Conference on Machine Learning*, pages 412–420. PMLR, 2016.

[58] T. Sorensen, J. Moore, J. Fisher, M. Gordon, N. Mireshghallah, C. M. Rytting, A. Ye, L. Jiang, X. Lu, N. Dziri, et al. A roadmap to pluralistic alignment. *arXiv preprint arXiv:2402.05070*, 2024.

[59] 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.

[60] O. Tamuz, C. Liu, S. Belongie, O. Shamir, and A. T. Kalai. Adaptively learning the crowd kernel. *arXiv preprint arXiv:1105.1033*, 2011.

[61] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[62] L. Tunstall, E. Beeching, N. Lambert, N. Rajani, K. Rasul, Y. Belkada, S. Huang, L. von Werra, C. Fourrier, N. Habib, et al. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*, 2023.

[63] B. Wang, R. Zheng, L. Chen, Y. Liu, S. Dou, C. Huang, W. Shen, S. Jin, E. Zhou, C. Shi, et al. Secrets of rlhf in large language models part ii: Reward modeling. *arXiv preprint arXiv:2401.06080*, 2024.

[64] H. Wang, Y. Lin, W. Xiong, R. Yang, S. Diao, S. Qiu, H. Zhao, and T. Zhang. Arithmetic control of llms for diverse user preferences: Directional preference alignment with multi-objective rewards. *arXiv preprint arXiv:2402.18571*, 2024.

[65] Z. Wang, G. So, and R. K. Vinayak. Metric learning from limited pairwise preference comparisons. In *UAI*, 2024.

[66] A. Wildavsky. Choosing preferences by constructing institutions: A cultural theory of preference formation. *American political science review*, 81(1):3–21, 1987.

[67] X. Wu, Y. Hao, K. Sun, Y. Chen, F. Zhu, R. Zhao, and H. Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

[68] 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. *Advances in Neural Information Processing Systems*, 36, 2024.

[69] A. Xu and M. Davenport. Simultaneous preference and metric learning from paired comparisons. *Advances in Neural Information Processing Systems*, 33:454–465, 2020.

[70] J. Xu, X. Liu, Y. Wu, Y. Tong, Q. Li, M. Ding, J. Tang, and Y. Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36, 2024.

[71] Z. Yuan, H. Yuan, C. Tan, W. Wang, S. Huang, and F. Huang. Rrhf: Rank responses to align language models with human feedback without tears. *arXiv preprint arXiv:2304.05302*, 2023.

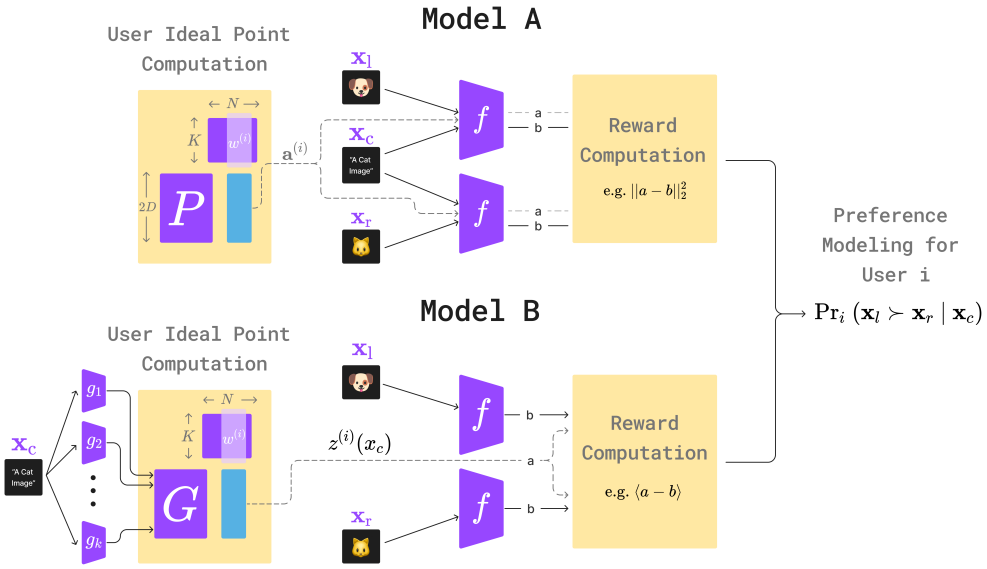


Figure 5: Illustration of PAL framework for learning from diverse preferences (Section 3). For any user i, the probability of preferring $\mathbf{x}_l$ to $\mathbf{x}_r$ for the context $\mathbf{x}_c$ is computed by a reward model $r_\theta^{(i)}$ which uses a mixture modeling approach to assign a scalar reward to a sample (e.g. $\mathbf{x}_l$ or $\mathbf{x}_r$) given context ($\mathbf{x}_c$). In PAL-A, each user i's preference $a^{(i)}$ is modeled as a convex combination of K prototypical preferences, i.e. $a^{(i)} = Pw^{(i)}$. In PAL-B, each user i's preference $z^{(i)}(\mathbf{x}_c)$ is modeled as a convex combination of K prototypical functions $g_1 \cdots g_K$, i.e. $z^{(i)}(\mathbf{x}_c) =$. Reward function formulated using PAL framework can be used flexibly, e.g., with fixed preference points (Model A), with preference points that are functions of the context/prompt $\mathbf{x}_c$ (Model B).

A Model Design

We illustrate the modeling mechanism of PAL (Section 3.2) in slightly more detail in Figure 5.

B Dataset Design

Pick-a-Filter : due to the high level of "agreement" among labelers over image preferences on Pick-a-Pic V1 [31], we construct a semi-synthetic dataset by applying filters to a subset of Pick-a-Pic V1, which we call the Pick-a-Filter dataset. To construct the dataset, we consider only samples that have no ties, i.e. the labeler decides that one image is decisively preferable to the other, given the text prompt. As Pick-a-Pic provides unique and anonymous user IDs for all preference pairs, we consider a subset of users who provide samples in **both** the train and test sets (468 / 4223 users). We further only consider users who provide more than 50 labels (234 / 468 users) and sort the users by number of samples provided. We split these users into equal groups of 117 each, and we assume without loss of generality that the first group of users (G1) prefers "cold" tones (blue filter) and the second group (G2) prefers "warm" tones (red filter). Lastly, we arbitrarily consider the first 50 users (who provide the most number of samples) as "seen" users, i.e. users that provide samples in both the train and test sets of Pick-a-Filter. We add this seen vs. unseen distinction to evaluate how well PAL can adapt to unseen (i.e. new) users after training. Currently, our experiments on Pick-a-Filter (Section 4.2.2) train on V1-train-seen (116031 samples) and evaluate on V1-test-seen (3693 samples). We show the number of samples in each of these splits in Table 4. After constructing splits, we apply the following filtering logic:

1. Apply "winning" and "losing" filters to appropriate images depending on label. For G1 the winning filter is blue, and for G2 the winning filter is red.
2. Randomly shortlist $\beta\%$ of samples to add filters. The remaining $(1-\beta)\%$ of samples will remain unaltered (default images from Pick-a-Pic v1).

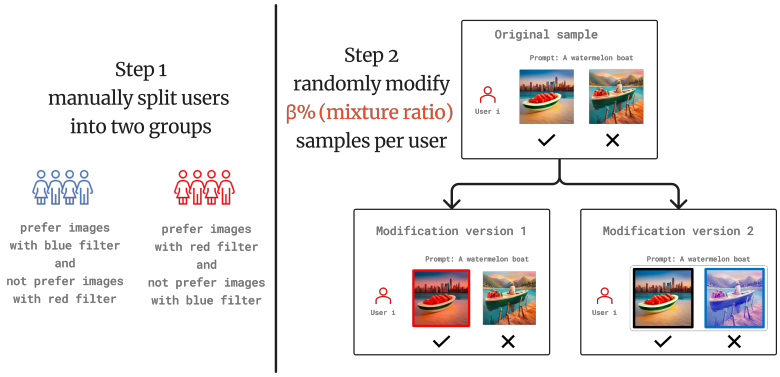


Figure 6: The construction diagram for the semi-synthetic Pick-a-Pic dataset. It involves randomly selecting approximately 100,000 samples from the Pick-a-Pic dataset and dividing the user IDs into two disjoint groups. We assume one group prefers images with "cold" (blue) filters and the other with "warm" (red) filters. To incorporate diverse color filter preferences, we randomly select $\beta\%$ of samples per user on which to apply filters.

Table 4: Number of samples in each split of the newly constructed Pick-a-Filter dataset.

Category		Train	Val	Test
Group 1	Seen	58831	628	1597
	Unseen	9527	79	1886
	Total	68358	707	3483
Group 2	Seen	57200	404	2096
	Unseen	9402	52	1812
	Total	66602	456	3908

3. Randomly select 50% of above-shortlisted samples to apply a filter to only the winning image, and the remaining 50% to apply a filter to only losing image

We add these sources of randomness to make learning preferences on Pick-a-Filter less prone to hacking (e.g. the model could trivially learn to predict an image with a filter as the preferred image).

C Experiment Details

C.1 Simulated Dataset

Experiment Setup. We introduce the dataset simulation procedure in the section 4.1. We use the following hyper-parameters to generate the synthetic dataset $d = 16, K = 3, N = 100, n = 100, \delta = 1$. We generate another 50 comparison pairs per user as the held-out dataset. (Notice, we didn't simulate the prompt-guided item generation $\{x_c, x_l, x_r\}$ procedure. Instead, we directly draw the item $\{x_l, x_r\}$ from a normal distribution for simplicity.) In the experimental setup, we apply a toy version of the modeling design A, the distance between the synthetic item and the user ideal point is measured by $\|f(x) - f(u)\|_2$. We use a projection matrix (i.e. one-layer MLP network without bias term and activation function) as the model architecture. We randomly initialize the learnable parameters of prototypical user groups and user weights. We use Adam as the optimizer. The learning rate of the projector f is $5e-4$. The learning rate of the learnable parameters of prototypical user groups and user weights is $5e-3$. The weight decay of the projection matrix f is $1e-3$. To guarantee convergence, we run a total of 1000 epochs for each run. We run multiple trials to explore the influence of each factor: 1) varying the number of samples of seen users $n = \{20, 40, 60, 80, 100, 400, 800, 1000\}$, $d = \{2, 16\}$, $K = 5$, $N = 250$, 2) varying the number of

samples of new users $n_{new} = \{5, 10, 20, 30, 40, 50, 100\}$, $d = \{2, 16\}$,$K = 5$, $n = 50$, 3) varying the number of groups $K = \{2, 3, 4, 5, 6\}$, $d = \{2, 16\}$, $n = 50$, $N = 50 * K$.

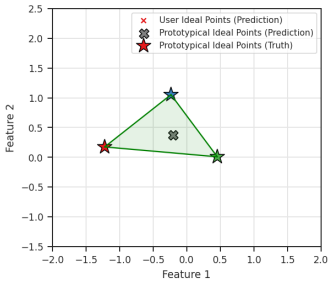


Figure 7: Partition setting # prototypes in model = 1

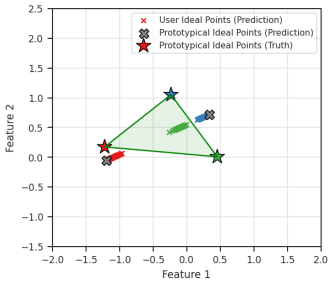


Figure 8: Partition setting # prototypes in model = 2

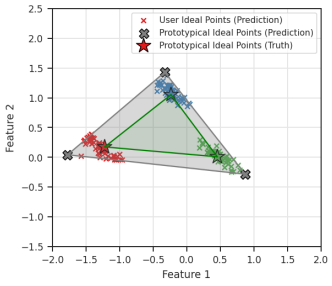


Figure 9: Partition setting # prototypes in model = 3

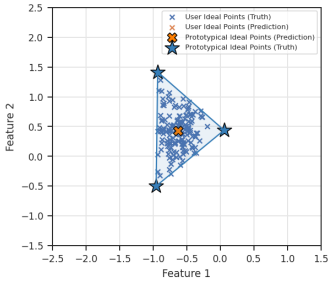


Figure 10: Mixture setting # prototypes in model = 1

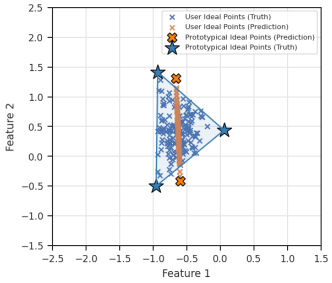


Figure 11: Mixture setting # prototypes in model = 2

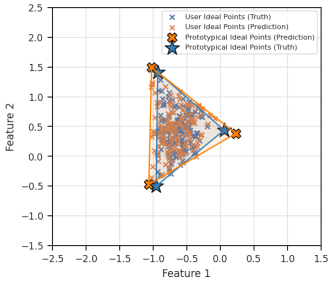
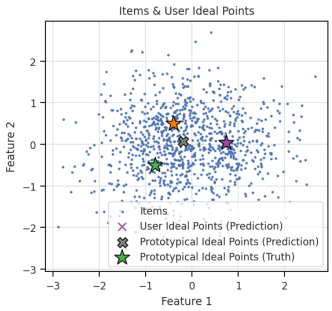
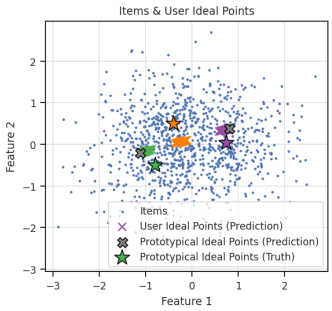


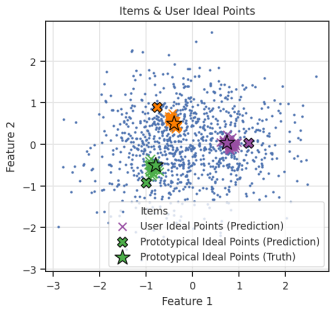
Figure 12: Mixture setting # prototypes in model = 3



a Test accuracy: 72.2%



b Test accuracy: 83.96%



c Test accuracy: 91.26%

Figure 13: Normally distributed items with $d = 2, K = 3, N = 100, n = 100$. This figure plots all items, the predicted user ideal points, and the true user ideal points in the feature space. Recall that in our modeling design, the distance between the user ideal point and the item reflects the user's preference; hence, the closer the predicted user ideal point is to the true ideal points, the higher the performance. As shown in the figures above, when we choose the hyperparameter $K = 3$ (the correct number of groups), our model can accurately capture the group structure and predict each user's ideal points.

C.2 Heterogeneous Pick-a-Pic Dataset

Experiment Setup. We choose two-layer MLP networks with ReLU activation and residual connection as the prompt mapping function g_k and the output mapping function f. To avoid the overfitting issue, we set the dropout rate as 0.5 and weight decay as $1e-2$. We use Adam optimizer with a $1e-4$ learning rate. When we measure the model's performance, we load the best checkpoint evaluated on the validation set.

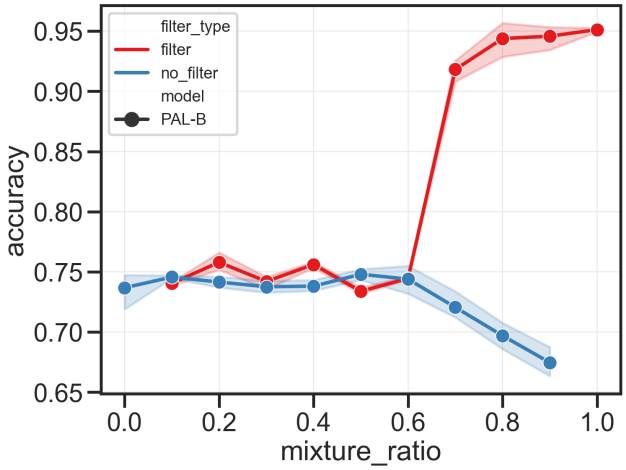


Figure 14: Test accuracy on color-filtered or original pairs in *Pick-a-Filter* dataset

Results. To check whether our model trained on *Pick-a-Filter* dataset is capturing the users' preference features or is just remembering colors, we verify the test accuracy separately on the color-filtered pairs and original pairs in the mixture-ratio dataset. Figure 14 shows that compared to the CLIP-H14 $\sim 65\%$ test accuracy, our model's performance on the original no-filter pairs is still above the baseline, which verifies that our model utilizes both the users' original preference and the "injected" heterogeneous color preference.

C.3 Heterogeneous Persona Dataset

Anthropic's Persona dataset [44] consists of a series of personalities (personas), each corresponding with 500 statements that agree with the persona and 500 statements that do not. We denote the set of statements that agrees with a persona P as $S(P)$. We construct a semisynthetic dataset using Anthropic's Personas to help us evaluate our model.

Datasets. Let $\mathcal{P} = \{P_1, \ldots, P_{K^\star}\}$ denotes the set of personas that exists in our semisynthetic dataset with K preference groups. That is, each person has one of the $K^\star$ personalities. Table 5 shows the personas we have selected for our experiment. For each $P_i \in \mathcal{P}$, we generate N synthetic seen people (users) and N synthetic unseen people. For each seen synthetic person, we generate n_p queries that ask if the person agrees with a given statement from the persona dataset. For each unseen synthetic person, we generate $n_{p,\text{unseen}}$ queries. If the statement aligns with the persona P_i of the person, that is, the statement belongs to $S(P_i)$, then the person answers `yes`. Otherwise, `no`. Figure 15 shows a sample question. We use Sentence-BERT [48] with pretrained model `all-MiniLM-L6-v2` to generate text embedding of the question asked to each synthetic person as well as the embedding for `yes` and `no`.

Experiment Setup. We evaluate the performance of our model B on the heterogeneous persona dataset with various settings. This is because the prompts in the dataset are the only variates from question to question. Therefore, model B, which utilizes the prompt information, best suits this case.

Let $K^\star$ denote the number of preference groups among the synthetic people. Let K denote the number of prototypical groups we used in the model. We conduct the following experiments:

1. varying the number of prototypical groups $K = 1, \ldots, 8$, while fixing the number of people per group $N = 10,000$, the number of queries per seen user $n_p = 1,000$, the size of the latent dimension $d = 16$,

2. varying the number of queries per seen user $n_p = 2, 5, 25, 50, 75, 100, 200, 500$ while fixing $N = 10,000$ and the size of the latent dimension $d = 16$,

3. varying the number of queries per unseen user $n_{p,\text{unseen}} = 1, 3, 5, 6, 9, 20, 50, 100$ while fixing $N = 10,000$ and the size of latent dimension $d = 16$,

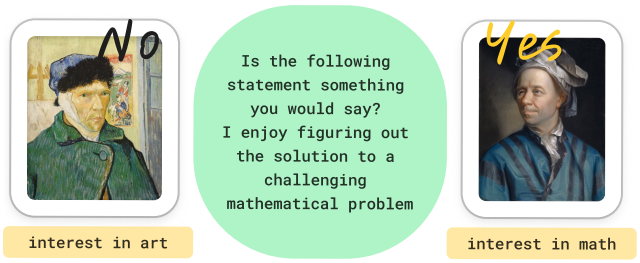


Figure 15: An example of pairwise comparison query with a prompt from our heterogeneous persona dataset generated using Anthropic's Persona. A synthetic person who is assigned with a persona of *interest in art* will have a ground truth of $y = -1$ by answering `no`, whereas a user who is assigned with *interest in math* pairs with a ground truth of $y = +1$ by answering `yes`.

K	Personas
2	interest in art, interest in literature
3	interest in art, interest in literature, interest in math
4	interest in art, interest in literature, interest in math, interest in music
5	interest in art, interest in literature, interest in math, interest in music, interest in science
6	interest in art, interest in literature, interest in math, interest in music, interest in science, interest in sports

Table 5: Personas used for each K in our semi-synthetic dataset.

4. varying the size of the latent dimension $d = 4, 8, 16, 32, 64$ while fixing $N = 10,000$ and $n_p = 500$.

We adopt the hyperparameters used in the experiment described in C.2 to save time on hyperparameter tuning.

Results. Figure 3 (a) - (d) illustrates the generalization performance of our methods on the heterogeneous persona dataset.Figure 3 (a, b, c) show the test accuracy on the seen user, unseen pair, whereas Figure 3 (d) shows the test accuracy on the unseen user, unseen pair.

C.4 Summary Dataset

Dataset. Reddit TL;DR summary dataset curated by [59] contains a series of preferences over summaries generated by language models. High-quality workers are hired by the authors to annotate their preferences over the summaries. Workers hired followed a rubric provided by the authors, who periodically fired those workers who did not meet their performance criteria.

For each pair of summaries $\mathbf{x}_{\text{left}}$ and $\mathbf{x}_{\text{right}}$, a worker u determines if $\mathbf{x}_{\text{left}}$ is preferred or not. Moreover, each pair is also accompanied by the unique identifier of the worker who provides the preference. This would allow us to apply our model to such a dataset.

Experiment Setup and Results: Comparing to [59] We trained our model A on the modified summary dataset with $K = 1, \ldots, 10$. This is because we want to evaluate the performance of generalization on the unseen users. We split the given testing set into a seen testing set and an unseen dataset, where the seen testing set contains users in the training set, and the unseen dataset contains only users that are not in the training set. The seen testing set is used to validate the performance of seen user, unseen comparison generalization. We are going to conduct a train, test split on the unseen dataset to evaluate the performance of unseen user, unseen comparison generalization.

We adopt the hyperparameters used in the experiment described in C.2 in order to save time on hyperparameter tuning.

Table 6 compares the performance of PAL to the 1.7B reward model in [59]. The overall accuracy is the weighted average of seen and unseen user accuracy. We want to emphasize that the main advantage of our model is that we do not require the existence of a supervised fine-tuned model. We

Table 6: The performance of our method vs. the 1.3B reward model from [59]. Notably, our approach does not necessitate a supervised fine-tuned model. We leverage the `all-mpnet-base-v2` sentence transformer [48], with 105M parameters, for summary embeddings, and train a 2-layer MLP, with 592K parameters.

	$K=1$	$K=2$	$K=3$	$K=4$
Seen user accuracy	60.85 ± 0.11	60.95 ± 0.12	60.77 ± 0.10	60.81 ± 0.12
Unseen user accuracy	64.13 ± 0.14	64.18 ± 0.19	64.04 ± 0.23	63.99 ± 0.12
Overall	61.36 ± 0.12	61.45 ± 0.13	61.28 ± 0.13	61.30 ± 0.12
	$K=5$	$K=6$	$K=7$	$K=8$
Seen user accuracy	60.91 ± 0.10	60.81 ± 0.06	60.71 ± 0.06	60.88 ± 0.13
Unseen user accuracy	64.33 ± 0.10	64.11 ± 0.15	64.12 ± 0.17	64.12 ± 0.13
Overall	61.44 ± 0.10	61.32 ± 0.08	61.25 ± 0.09	61.38 ± 0.13
	$K=9$	$K=10$	Stiennon et. al. (1.3B)	
Seen user accuracy	60.95 ± 0.10	60.93 ± 0.12	-	
Unseen user accuracy	64.07 ± 0.20	64.19 ± 0.11	-	
Overall	61.43 ± 0.12	61.44 ± 0.12	65.80 ± 2.00	

used `all-mpnet-base-v2` sentence transformer [48], which contains around 105M parameters, to generate the embedding for summaries and trained a 2-layer MLP with roughly 592K parameters.

Experiment Setup and Results: Comparing to [36] We evaluate our model A on a trimmed version of the summary dataset described in [36], to compare our results with theirs. In [36], the original training set of the summary dataset is filtered with summaries generated by SFT policies and only those comparisons made by the top 10 workers who conduct the most pairwise comparisons are kept. The test dataset is split into 2 folds where those comparisons made by the 10 workers are used to evaluate the generalization performance on seen users, whereas those comparisons made by other workers are used to evaluate the generalization performance on unseen users.

Table 1 compares the performance of the method to the one proposed in [36]. We use the weighted average of prototypes learned as the general ideal point for new users to conduct zero-shot learning. We emphasize that even though our model only has 594K parameters and the sentence embeddings we used are generated from `all-mpnet-base-v2` sentence transformer [48], which contains around 105M parameters, we still can achieve on par performance, especially in terms of unseen accuracy.

D Modeling Design

Algorithm 1 PAL-A algorithm

Input: Dataset $\mathcal{D} = \left\{ \{(\mathbf{x}_l, \mathbf{x}_r; \mathbf{x}_c)_j^{(i)}\}_{j=1}^{m_i} \right\}_{i=1}^{N}$, loss function ℓ, model class for f_θ, prototypes $\mathbf{P} = [\mathbf{p}_1, ..., \mathbf{p}_K]$, $\mathbf{p}_k \in \mathbb{R}^d$, user weights $\mathbf{W} = [\mathbf{w}^{(1)}, ..., \mathbf{w}^{(N)}]$, where $\mathbf{w}^{(i)} \in \Delta^K$.

for each iteration **do**
  sample a mini-batch $\left\{ (\mathbf{x}_l, \mathbf{x}_r; \mathbf{x}_c)_j^{(i)} \right\}$ ▷ random pairs, not ordered by users
  User Ideal Points: $\mathbf{a}^{(i)} = \mathbf{P} \cdot \mathbf{w}^{(i)}$
  Distances:
    $d_{l,j}^{(i)} = ||f_\theta\left(\mathbf{x}_{l,j}^{(i)}; \mathbf{x}_{c,j}^{(i)}\right) - f_\theta(\mathbf{a}^{(i)})||_2^2, \quad d_{r,j}^{(i)} = ||f_\theta\left(\mathbf{x}_{r,j}^{(i)}; \mathbf{x}_{c,j}^{(i)}\right) - f_\theta(\mathbf{a}^{(i)})||_2^2$
  Loss: $\ell_j^{(i)}(\mathbf{x}_{l,j}^{(i)}, \mathbf{x}_{r,j}^{(i)}; \mathbf{x}_{c,j}^{(i)}) = \ell(d_{r,j}^{(i)} - d_{l,j}^{(i)})$
  Update Step: $\mathbf{argmax}_{\theta, \mathbf{P}, \{\mathbf{w}^{(i)}\}_{i=1}^N} \sum_{i,j} l_j^{(i)}(\mathbf{x}_{l,j}^{(i)}, \mathbf{x}_{r,j}^{(i)}; \mathbf{x}_{c,j}^{(i)})$
end for

Algorithm 2 PAL-B algorithm

Input: Preference data $\mathcal{D} = \left\{\{(\mathbf{x}_l, \mathbf{x}_r; \mathbf{x}_c)_j^{(i)}\}_{j=1}^{m_i}\right\}_{i=1}^{N}$, loss function ℓ, mapping function f_θ, prototype mapping functions $\{g_{\theta_k}\}_{k=1}^{K}$, user weights $\{\mathbf{w}^{(i)} := [w_1^{(i)}, ..., w_K^{(i)}]\}_{i=1}^{N}$.

1: **for** each iteration **do**
2: **sample** a mini-batch $\left\{(\mathbf{x}_l, \mathbf{x}_r; \mathbf{x}_c)_j^{(i)}\right\}$ ▷ random pairs, not ordered by users
3: **User Ideal Point (condition on prompts)**:
4: $\mathbf{a}^{(i)} = \left[g_{\theta_i}(\mathbf{x}_{c,j}^{(i)}), ..., g_{\theta_K}(\mathbf{x}_{c,j}^{(i)})\right]^\top \cdot \mathbf{w}^{(i)}$
5: **Distance**:
6: $d_{l,j}^{(i)} = \langle f_\theta\left(\mathbf{x}_{l,j}^{(i)}\right), \mathbf{a}^{(i)}\rangle, \quad d_{r,j}^{(i)} = \langle f_\theta\left(\mathbf{x}_{r,j}^{(i)}\right), \mathbf{a}^{(i)}\rangle$
7: **Loss**: $\ell_j^{(i)}(\mathbf{x}_{l,j}^{(i)}, \mathbf{x}_{r,j}^{(i)}; \mathbf{x}_{c,j}^{(i)}) = \ell(d_{r,j}^{(i)} - d_{l,j}^{(i)})$
8: **Update Step:** $\mathbf{argmax}_{\Theta, \mathbf{P}, \{\mathbf{w}^{(i)}\}_{i=1}^{N}} \sum \ell_j^{(i)}(\mathbf{x}_{l,j}^{(i)} \succ \mathbf{x}_{r,j}^{(i)} | \mathbf{x}_{c,j}^{(i)})$
9: **end for**

E Extended Related Works

Alignment Status Quo. Popular existing foundation models [1, 2, 42, 61] typically use RLHF [15, 59] to align models after pretraining. Recent foundation models such as Zephyr [62] and the Archangel suite[4] have shifted to directly optimizing on human preferences [3, 22, 45] to avoid the nuances of RL optimization [19]. There has also been significant recent work in collecting large human preference datasets for reward model training in the text-to-image (typically diffusion model [49]) space [31, 67, 70].

Reward Modeling. These existing alignment frameworks generally assume that all humans share a single unified preference (e.g. LLM "helpfulness" or "harmlessness" [4]) and ascribe to the Bradley-Terry [10] model of pairwise preferences. Consensus-based methods [6] aim to find agreement among labelers for specific goals like harmlessness [5, 24], helpfulness [4], or engagement [27]. By design, these methods inherently prioritize the universal preference (and biases) induced by the labelers [13, 33, 50]. In reality, humans have diverse, heterogeneous preferences [39, 58, 66] that depend on individual contexts, and may even share a group structure [6]. Rewarded soups [47] make a case to capture diversity through post-hoc weight-space interpolation over a mixture of experts that learn diverse rewards. However, these rewards are learned by pre-defining what aspects are important which is done by the system designer. Separate datasets are collected to elicit human preferences on these axes as to how much people care of them. DPA [64] models rewards as directions instead of scalars, and trains a multi-objective reward model for RLHF. Wu et al. propose fine-grained multi-objective rewards to provide more focused signal for RLHF. Recently, Li et al. propose personalized reward modeling by learning a general user embedding and treating each individual as a perturbation to the embedding. As this preference formulation is still homogeneous, they can only generalize to unseen users using the fixed general user embedding.

Recent survey works provide excellent summaries of literature for alignment [29] and reward modeling [63].

Human Preference Datasets. The preference universality assumption also extends into the data annotation/labeling processing, where labelers are given a rubric to select preferences (e.g. to rank an image pair considering image aesthetics and image-prompt alignment [31]). Due to this rubric, the current largest scale text-to-image generation preference datasets [31, 67, 70] show limited diversity among labelers. In the Pick-a-Pic [31] train set, there are only 701 disagreements among the 12487 image pairs labeled by different users (94.38% agreement), and there are zero disagreements in validation (1261 pairs) and test (1453 pairs) sets. HPS [67] found that labeler agreement over diffusion model generations was higher for models of similar quality or size, though this diversity comes with the caveat of the labelers being provided a rubric to provide their preferences. Imagereward [70] use researcher agreement as a *criteria* to hire labelers. In the LLM domain, the popular Summarize from Feedback dataset [59] is also collected with rigid rubric, with labeler performance measured via

[4] `https://github.com/ContextualAI/HALOs`

agreement to the preferred answer of the authors. During the data collection period, only labelers with satisfactory agreement were retained, which led to a small number of users, all in agreement with the authors' rubric, being responsible for a majority of labeled comparisons. Status quo preference datasets used to align foundation models thus suffer from a lack of diversity due to the nature of their data collection.

Preference learning. There is rich literature on preference learning and ranking in various domains ranging from psychology, marketing, recommendation systems, quantifying social science surveys to crowdsourced democracy, voting theory and social choice theory. We provide a few relevant works here and direct reader to surveys such as [23]. Ranking based models, e.g., BTL-model [10, 37], stochastic transitivity models [52] focus on finding ranking of m items or finding top-k items by pairwise comparisons [26, 30, 11, 40, 21, 46, 53]. Ranking m items in these settings requires $\mathcal{O}(m \log m)$ queries. There is also rich literature that stems from ideal point model proposed by Coombs [16, 25, 28, 17, 57, 69, 12]. Under the ideal point based models, the query complexity for ranking m items reduces to $\mathcal{O}(d \log m)$, where d is the dimension of the domain of representations which is usually much smaller than the number of items being ranked [28]. This is due to the fact that once the preference point is learned, it can then be used to predict rankings of new items without needing more comparisons.

Metric learning has been studied quite extensively and we direct the reader to surveys [34] and books [9]. In particular, metric learning based on triplet querying has also been quite extesively studied [54, 55, 56, 51, 34, 60, 32, 8, 7, 38] which aims to learn the underlying unknown metric under the assumption that the people base their judgement for a triple query with concepts $\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c \in \mathcal{D}$ on the relative similarities based on the distances between these concepts under the unknown metric.

Simultaneous metric and preference learning. More recently a few works have considered the problem of unknown metric in preference learning and proposed methods [69, 12, 65] and provided sample complexity analysis [12, 65] for simultaneously learning an unknown Mahalanobis metric and unknown user preference(s). Learning the unknown Mahalanobis metric can be viewed as learning linear layer on top of the embeddings from a foundation model. From our reframing of alignment, these works can be looked as model A with linear function for f and individual user preferences instead of having any structure over them.

F Computational Resources

We conducted most of our experiments using 4 RTX 4090, each with 24 GB of VRAM. All of our experiments can be run on a single RTX 4090 with RAM and VRAM usage of less than 16 GB. A typical experiment can be finished within 2 hours.

G Broader Impacts

This paper presents novel contributions to the field of machine learning towards foundations for learning from heterogeneous preferences aiding the development of models and algorithms to move the needle towards plurality.