Toward Deployable Pluralistic Alignment in Robotics: Learning Similarity-Grouped Rewards from Diverse Human Preferences

Taehyung Kim [1] **Gwangmo Lee** [1] **Jonghak Bae** [2] **Dongjae Kim** [2] **Jaewoong Han** [2] **Jongeun Choi** [2]

Abstract

Personalization is essential for deploying robotic systems across diverse end users. However, fully individualized policies are difficult to validate, costly to scale, and unreliable under sparse and noisy preference feedback, while a single global policy collapses meaningful preference heterogeneity across users. To address these challenges, we formulate deployable pluralistic alignment as a preference-based reinforcement learning (PbRL) problem, aiming to learn a limited number of policies that serve a heterogeneous user population while preserving diverse user preferences. We develop this approach into **P**reference-based **RE**ward **C**lustering (**PREC**), a framework that learns a compact set of representative reward models from human preference labels (i.e., good/bad feedback) collected across users. PREC first learns a population-level trajectory representation from state-action data without relying on preference labels, reducing reliance on limited and skewed per-user coverage and mitigating exposure to label noise. It then learns group-level reward decoders shared among users with similar preferences, pooling sparse and noisy feedback to capture distinct preference modes while yielding a manageable number of representative reward models. Experiments across diverse simulated robotic locomotion environments show that PREC improves aggregate social welfare over both a single global policy and fully individualized policies under sparse and noisy feedback across diverse preference distributions.

[1]Department of Mobility Systems Engineering, Yonsei University, Seoul, South Korea [2]School of Mechanical Engineering, Yonsei University, Seoul, South Korea. Correspondence to: Jongeun Choi <jongeunchoi@yonsei.ac.kr>.

Pluralistic Alignment Workshop @ ICML 2026, Seoul, South Korea.

1. Introduction

As AI systems become increasingly capable, they are being integrated into everyday user-facing applications, from conversational assistants to robots. A common strategy for improving these systems is to align their behavior with human preferences, and empirical studies have shown that such alignment enhances the performance of large language models (LLMs) (Ziegler et al., 2019; Ouyang et al., 2022) as well as robotic systems (Cabi et al., 2019). However, this approach raises concerns, including the "tyranny of the crowdworker" (Kirk et al., 2024), where alignment directions are determined by a small group of annotators, and the loss of diversity when aggregating preferences into a single universal signal, which collapses individual preference variations (Gabriel, 2020; Santurkar et al., 2023). Consequently, personalization in AI systems has emerged as a key objective for user-facing AI systems (Hellou et al., 2021).

Accordingly, user-specific personalization methods have been actively studied and widely deployed in domains such as chatbots, recommendation systems (Liu et al., 2025), and robotics (Cabi et al., 2019). However, comparatively little attention has been paid to personalization from the deployer perspective, particularly in robotics. In robotic systems, personalization is often constrained by (i) strong regulatory requirements in safety-critical domains and (ii) skewed and noisy user preference data. For example, personalization is inherently important in robotic exoskeletons, as users exhibit substantial variability in physical characteristics such as strength and gait (Slade et al., 2022). However, in the United States, powered lower-extremity exoskeletons intended for medical use are classified by the FDA as Class II prescription medical devices subject to special controls (Food and Drug Administration, 2015). Because these requirements include software validation, risk analysis, and clinical evaluation, personalization must remain within validated configurations even in clinical settings. Fully individualized learned policies can therefore impose substantial validation and regulatory burdens from the deployer perspective, making per-user personalization difficult to scale and even harder to extend beyond tightly controlled medical-use settings. In addition, from the deployer perspective, collecting reliable user preference data remains a key bottleneck

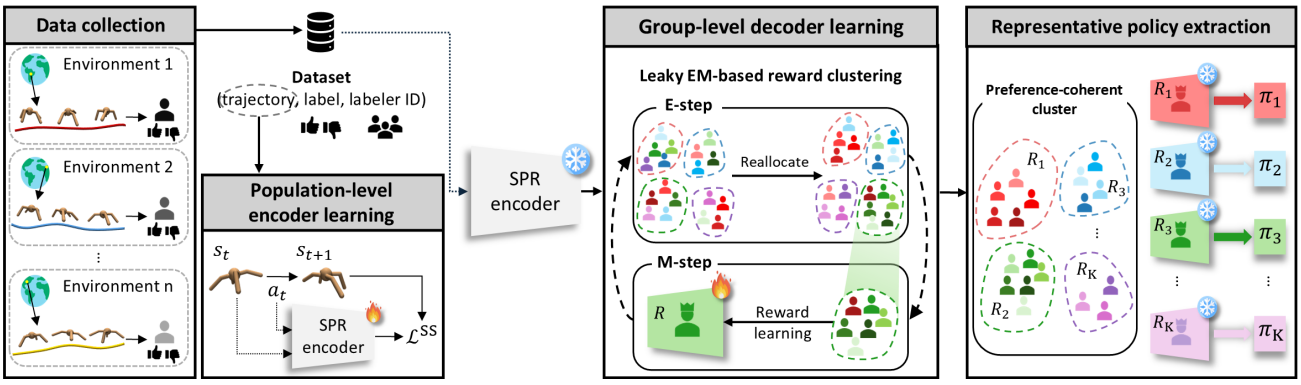

Figure 1. **Overview of PREC.** PREC first learns a population-level trajectory encoder from offline state-action data aggregated across users, reducing reliance on sparse and noisy individual preference labels (Section 3.2). With this encoder fixed, leaky EM jointly updates cluster-specific reward decoders and groups users with similar preferences, yielding a compact set of representative reward models from good/bad feedback (Section 3.3). Each learned reward model is then used to train one deployable policy via PPO (Section 3.4).

for user-facing robotic systems, as acquiring dense preference coverage for each individual user is often costly. Each user's interaction is also typically concentrated around their habits and work duties. For example, an assistive wheelchair user may follow only a small set of familiar routes (Soh & Demiris, 2015). This yields narrow and skewed per-user distributions whose coverage varies substantially across individuals, making broad behavior coverage difficult to achieve at the individual-user level. The problem is further amplified by the fact that human preference labels are inherently noisy (Lee et al., 2021b).

These challenges have encouraged a deployment paradigm in which personalization in end-user robotics is either highly limited or largely uniform across users. However, as robots are increasingly deployed in everyday environments, they will interact with a broad spectrum of people, including vulnerable populations (Madan et al., 2025; Shankar et al., 2026). We believe that, despite the constraints discussed above, personalization in robotics remains necessary for serving diverse users and should be treated as a central deployment challenge. We therefore focus on personalization of relatively low-complexity continuous-control robotic systems, such as assistive mobility devices, exoskeletons, and other embodied systems that are likely to be commercialized and deployed at scale earlier than more complex general-purpose robots. We aim to address the difficulty of training such robots from scratch to obtain policies that can satisfy a broad range of users.

A natural starting point for addressing these challenges is preference-based reinforcement learning (PbRL) (Christiano et al., 2017). In PbRL, users evaluate candidate behaviors, such as trajectory segments, via simple preference queries (e.g., good/bad feedback or pairwise comparisons). These feedback signals are used to learn a reward model that captures user preferences, which then guides policy optimization via RL. Since users only need to observe behaviors and provide simple feedback, PbRL reduces the need for precise verbal descriptions or physically skilled demonstrations, making preference collection more accessible to non-expert users. It can also capture tacit aspects of embodied interaction, such as comfort and naturalness, that are difficult for users to articulate. Recent advances in preference-based reward learning from pre-collected trajectories (Kim et al., 2023) further improve practicality and safety by collecting feedback on such trajectories, thereby avoiding costly and potentially unsafe interactions with human users.

PbRL is promising for robot alignment because preference feedback offers a relatively low-cost and human-friendly interface for capturing implicit human preferences. Compared with more demanding supervision modalities, it can also reduce the likelihood of human errors. However, directly applying PbRL to end-user personalization at deployment does not fully resolve several challenges. In safety-critical settings, per-user personalization remains difficult to validate for regulatory approval, broad preference coverage across diverse users and conditions is hard to obtain, and preference feedback, while easier to elicit, remains inherently noisy. These constraints make learning over groups of similar users a more practical deployment strategy than either per-user or population-level learning. By clustering similar users within the broader population, group-level learning pools feedback from users with distinct trajectory exposures, thereby increasing effective feedback coverage and mitigating the impact of noisy labels, while preserving preference heterogeneity that a single global model would otherwise collapse.

To this end, we propose **P**reference-based **RE**ward **C**lustering (PREC), a novel framework that learns a small set of representative reward models from offline preference feedback collected across many users. PREC factorizes the reward model into a population-level shared trajectory encoder, pretrained on the union of offline trajectories to

absorb broad state-action structure under sparse per-user labels, and a set of subpopulation-level decoders, each shared within a cluster of similar users to capture a distinct preference mode. To assign users to decoders, PREC employs a leaky expectation-maximization (EM) (Dempster et al., 1977) procedure that clusters users in reward space, where a leak coefficient interpolates between hard and soft assignment to stabilize learning when individual clusters contain few users. By pooling feedback from users with similar preferences, PREC mitigates label noise and distributional skew while preserving preference diversity. The number of learned reward models directly determines the number of deployed policies, as each reward model corresponds to a single policy, allowing deployers to choose the degree of personalization granularity according to their certification and deployment constraints. Empirically, we show that PREC improves aggregate user welfare over per-user alignment when preference data are limited. We further demonstrate that PREC is robust to highly noisy feedback.

Our key contributions are summarized as follows:

- **A framework for deployable pluralistic alignment at controllable granularity.** We reframe PbRL for certifiable end-user deployment, where per-user PbRL faces validation hurdles and unreliable feedback while population-level PbRL collapses user heterogeneity. PREC resolves this by (i) pooling preferences from similar users to jointly learn their reward model and (ii) exposing the number of such groups as a tunable design parameter.
- **A leaky-EM algorithm for reward-space user clustering with provable monotone improvement.** PREC factorizes the reward model into a population-level shared encoder and subpopulation-level decoders, and clusters users in reward space via a leaky EM procedure whose leak coefficient interpolates between hard and soft assignment. We show that the resulting updates correspond to coordinate ascent on a restricted-family ELBO and prove monotone improvement and convergence of the leaky-ELBO values under a fixed-prior decoder-update formulation.
- **Empirical robustness under limited and noisy feedback.** PREC outperforms both per-user and population-level alignment in aggregate user welfare when preference data are limited and noisy.

2. Related Work

In this section, we provide a brief overview of human alignment in robotics and PbRL.

Human Alignment in Robotics. One line of work develops personalization methods that align a robot to an individual user, distinguished primarily by the feedback interface they assume: pairwise comparison preference feedback (Wang et al., 2025; Tucker et al., 2021), structured natural-language feedback (Wu et al., 2023), and free-form natural language (Wang et al., 2024). Most flexibly, Promptable Behaviors (Hwang et al., 2024) unifies several modalities, including natural-language descriptions and demonstrations, into a single multi-objective reward personalization framework. These methods have been applied in a range of downstream applications, including exoskeleton gait personalization (Ingraham et al., 2023; Slade et al., 2022), assistive robotics for users with motor impairments (Madan et al., 2025; Shankar et al., 2026), and social robots for education (Park et al., 2019). Another line of work pools preference data from many users into a single shared model. Whitney et al. (2018) learn manipulation skills from crowdsourced demonstrations, and Forbes et al. (2014) aggregate crowdsourced action corrections to generalize a single skill across users. Most recently, Bryant et al. (2026) extend this approach to social navigation, training a shared model of socially appropriate parking locations from user-annotated floorplans.

Between per-user personalization and population-level aggregation lies a group-level regime, in which a small number of reward models or policies is shared by clusters of users with similar preferences. Despite its practical appeal, this regime has received comparatively little attention in robot preference alignment. This setting is particularly relevant for certifiable deployment: per-user models are difficult to validate and often underdetermined by limited individual feedback, whereas a single population-level model collapses preference heterogeneity. We take a step toward this regime with PREC, a preference-based representative reward learning framework that jointly infers user clusters and group-specific rewards from offline preferences, with the number of groups as a tunable design parameter. By learning at the group level, PREC alleviates individual-level data scarcity, restricts deployment to a finite and auditable set of policies, and preserves more preference diversity than a single population-level model.

Preference-based Reinforcement Learning. Early work showed that policies can be trained from human pairwise comparisons by learning reward models over trajectory segments (Christiano et al., 2017). This online PbRL paradigm was later systematized by PEBBLE (Lee et al., 2021a), which improved data efficiency through unsupervised pretraining and off-policy reward relabeling. Subsequent methods further improved label efficiency, robustness, and exploration using pseudo-labeling (Park et al., 2022), noise filtering (Cheng et al., 2024), and uncertainty-driven intrinsic rewards (Liang et al., 2022). However, online PbRL requires humans to remain in the training loop and may expose users or hardware to unsafe behaviors during early

exploration, limiting its applicability to real-world robotics.

To reduce interaction cost and safety risks, recent work has shifted to offline PbRL, where preferences are collected over pre-existing trajectories (Shin et al., 2023). Reward-model-based methods have improved credit assignment through non-Markovian transformer rewards (Kim et al., 2023) and hindsight-conditioned reward decomposition (Gao et al., 2024), while a parallel line of reward-free methods replaces the explicit reward network with implicit Bellman-consistent value functions (Hejna & Sadigh, 2023), regret-based contrastive policy losses (Hejna et al., 2023), or preference-conditioned trajectory generators (Zhang et al., 2023). However, these methods still rely on sufficiently representative human preference data and remain sensitive to sparse or noisy preference labels. More importantly, most prior PbRL studies consider preferences drawn from a single labeler distribution, and even works that incorporate feedback from multiple users (Chhan et al., 2024; Xue et al., 2023; Ji & Chen, 2025) typically aggregate these preferences to learn a single policy, rather than explicitly modeling heterogeneous user populations.

Our work addresses this deployment gap by departing from the dominant reward-learning paradigm, in which state representations and reward models are learned end-to-end from human preference labels. Instead, we separate representation learning from noisy preference supervision and learn group-level representative reward decoders rather than per-user or population-level reward models.

3. Methodology

In this section, we present our framework for deployable personalization in multi-user PbRL. We first formulate the limited-policy deployment problem, then describe population-level representation learning, leaky EM-based reward clustering, and cluster-wise policy optimization.

3.1. Problem Formulation

We consider a shared Markov environment $\mathcal{E} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{T}_0, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{T}$ is the transition kernel, $\mathcal{T}_0$ is the initial-state distribution, and $\gamma \in [0, 1)$ is the discount factor. We consider a population of N users that share the same environment but may differ in their behavioral preferences.

Unlike standard reward-supervised reinforcement learning, we do not assume access to scalar reward values for individual state-action pairs. Instead, we consider a preference-based setting in which supervision is provided only through labeled trajectory segments. Specifically, for each user $i \in [N]$, we are given a user-specific dataset

$$\mathcal{D}_i = \{(\tau_{i,m}, y_{i,m})\}_{m=1}^{M_i}, \tag{1}$$

where $\tau_{i,m} = \{(s_1, a_1), \ldots, (s_T, a_T)\}$ denotes a trajectory segment and $y_{i,m} \in \{0, 1\}$ is the corresponding human-provided preference label, with 0 and 1 indicating bad and good human judgments, respectively. The full dataset across users is denoted by $\mathcal{D} = (\mathcal{D}_i)_{i=1}^N$.

Our objective is to serve the entire population using only a limited number of deployable policies. Let $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K) \in \Pi^K$ denote a set of $K \leq N$ policies, and let $\boldsymbol{\alpha} \in \{0, 1\}^{N \times K}$ denote the user-to-policy assignment matrix, where $\sum_{k=1}^K \alpha_{ik} = 1$ for every user $i \in [N]$. We seek an assignment-policy pair $(\boldsymbol{\alpha}, \boldsymbol{\pi})$ that maximizes the social welfare

$$\mathrm{SW}(\boldsymbol{\alpha}, \boldsymbol{\pi}) = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \alpha_{ik} J_i(\pi_k), \tag{2}$$

where $J_i(\pi_k)$ denotes the return of policy π_k for user i.

3.2. Population-Level Representation Learning

We cast reward learning as the problem of converting segment-level human preference supervision into a scalar reward function over state-action pairs. Given a labeled example $(\tau_{i,m}, y_{i,m})$, we seek a reward model whose aggregated prediction over $\tau_{i,m}$ explains $y_{i,m}$. We therefore parameterize the reward model as

$$\hat{r}_\psi(s, a) = h_{\psi^{\mathrm{dec}}}(f_{\psi^{\mathrm{enc}}}(s, a)), \tag{3}$$

where the encoder $f_{\psi^{\mathrm{enc}}}$ embeds each collected state-action pair into a latent feature space and the decoder $h_{\psi^{\mathrm{dec}}}$ maps the latent feature to a scalar reward. The induced segment score is

$$\hat{R}_\psi(\tau_{i,m}) = \sum_{t=1}^{T} \hat{r}_\psi(s_t, a_t), \tag{4}$$

which is subsequently matched to the observed label $y_{i,m}$.

A straightforward personalization strategy would be to learn a separate reward model for every user. In our setting, however, each user provides only a small, potentially skewed, and noisy set of labeled segments, making such user-specific reward estimation highly unstable. To address this issue, instead of training user-specific reward models end-to-end as in prior approaches, we first learn a shared population-level encoder from offline trajectories collected across all users, and then fit decoders using human preference labels. This decoupled structure exploits a much broader set of state-action observations than those available from any single user, while avoiding direct dependence on sparse and noisy labels during representation learning.

However, learning such an encoder from unlabeled trajectories alone is challenging, since there is no explicit supervision specifying which features should be preserved. We therefore adopt the self-predictive representations (SPR)

(Schwarzer et al., 2020) encoder architecture, which trains the encoder to predict future latent dynamics from current state-action inputs. We then freeze the encoder while learning the decoder from preference labels.

3.3. Cluster-Specific Decoder Learning via Leaky EM

Given the pretrained population-level encoder, we keep the representation shared across all users and introduce personalization through the decoder. Rather than fitting a separate decoder for every user, we learn a compact set of subpopulation-level decoders, each intended to represent a distinct preference mode. This design keeps the set of deployable policies manageable, since each learned reward decoder induces one downstream policy, allows the manufacturer to explicitly choose the degree of personalization, and mitigates the effect of sparse and noisy preference labels through shared supervision.

Formally, we instantiate one decoder per cluster and define the corresponding cluster-specific reward model as

$$\hat{r}_{\psi_k}(s,a) = h_{\psi_k^{\mathrm{dec}}}(f_{\psi^{\mathrm{enc}*}}(s,a)), \tag{5}$$

where $f_{\psi^{\mathrm{enc}*}}$ is the pretrained frozen encoder and $h_{\psi_k^{\mathrm{dec}}}$ is the decoder for cluster $k \in [K]$. For a labeled segment $(\tau_{i,m}, y_{i,m})$, the predicted segment score under cluster k is

$$\hat{R}_{\psi_k}(\tau_{i,m}) = \sum_{t=1}^{T} \hat{r}_{\psi_k}(s_t, a_t). \tag{6}$$

We treat the user-to-cluster assignment as a latent variable $Z_i \in [K]$ and optimize the decoders under an EM framework. Under cluster k, we model the likelihood of the observed label $y_{i,m}$ as

$$\begin{aligned} &P(y_{i,m} \mid \tau_{i,m}, Z_i = k, \psi_k) \\ &= \sigma\big(\hat{R}_{\psi_k}(\tau_{i,m})\big)^{y_{i,m}} \Big(1 - \sigma\big(\hat{R}_{\psi_k}(\tau_{i,m})\big)\Big)^{1-y_{i,m}} \end{aligned} \tag{7}$$

and define the user-level likelihood by aggregating over all labeled segments of user i,

$$\mathcal{L}_{ik} = \prod_{m=1}^{M_i} P(y_{i,m} \mid \tau_{i,m}, Z_i = k, \psi_k). \tag{8}$$

Intuitively, $\mathcal{L}_{ik}$ measures how well the decoder of cluster k explains the preference labels of user i.

E-step. Given the current decoders, the E-step computes the posterior responsibility of each cluster for each user:

$$\gamma_{ik} = P(Z_i = k \mid \mathcal{D}_i) = \frac{\rho_k \, \mathcal{L}_{ik}}{\sum_{k'=1}^{K} \rho_{k'} \, \mathcal{L}_{ik'}}, \tag{9}$$

where ρ_k denotes the current prior weight of cluster k. Thus, γ_{ik} can be interpreted as the degree to which user i is explained by cluster k. We then convert these responsibilities into a hard assignment

$$\hat{z}_i = \arg\max_k \gamma_{ik}, \tag{10}$$

which is then held fixed during the subsequent M-step, where the cluster-specific decoders are updated.

Leaky M-step. Given the hard assignment $\hat{z}_i$ from the E-step, a standard hard M-step would train decoder k using only the users with $\hat{z}_i = k$, i.e.,

$$\psi_k^{\mathrm{dec}} \leftarrow \arg\min_{\psi_k^{\mathrm{dec}}} \sum_{i:\hat{z}_i=k} \sum_{m=1}^{M_i} \mathrm{BCE}\Big(\sigma\big(\hat{R}_{\psi_k}(\tau_{i,m})\big),\, y_{i,m}\Big). \tag{11}$$

However, this update can be brittle when a cluster temporarily contains only a few users, since the corresponding decoder is then trained on a very small and potentially biased label set. Such clusters are prone to overfitting, which in turn can destabilize the next E-step.

To mitigate this issue, we adopt a *leaky* M-step. Instead of using a strictly binary assignment weight, we assign each user a small positive weight even for non-selected clusters:

$$\omega_{ik} = \max\big(\mathbf{1}[\hat{z}_i = k],\, \nu\big), \tag{12}$$

where $\nu \in [0, 1)$ is a leak coefficient. The decoder for cluster k is then updated by the weighted objective

$$\psi_k^{\mathrm{dec}} \leftarrow \arg\min_{\psi_k^{\mathrm{dec}}} \sum_{i=1}^{N} \omega_{ik} \sum_{m=1}^{M_i} \mathrm{BCE}\Big(\sigma\big(\hat{R}_{\psi_k}(\tau_{i,m})\big),\, y_{i,m}\Big). \tag{13}$$

The leaky M-step balances two competing objectives: preserving cluster specialization and stabilizing decoder learning. While a hard update can overfit when a cluster has few assigned users, a fully soft update can blur cluster boundaries. By allowing a small amount of cross-cluster supervision, the leaky M-step regularizes small clusters without sacrificing distinct reward structure. We further show in Appendix A that the proposed leaky EM updates monotonically improve a restricted-family ELBO, and that the resulting leaky-ELBO values converge to a finite limit under the fixed-prior decoder-update formulation.

3.4. Cluster-wise Policy Optimization

After the leaky EM updates reach the prescribed stopping criterion, we obtain a set of learned cluster-specific reward models $\{\hat{r}_{\psi_k^*}\}_{k=1}^{K}$ together with the corresponding user partition. We then train one policy for each cluster independently using the learned reward of that cluster. Specifically, for each $k \in [K]$, we solve

$$\pi_k^* \in \arg\max_{\pi \in \Pi} \mathbb{E}_{\tau \sim \pi}\big[\hat{R}_{\psi_k^*}(\tau)\big]. \tag{14}$$

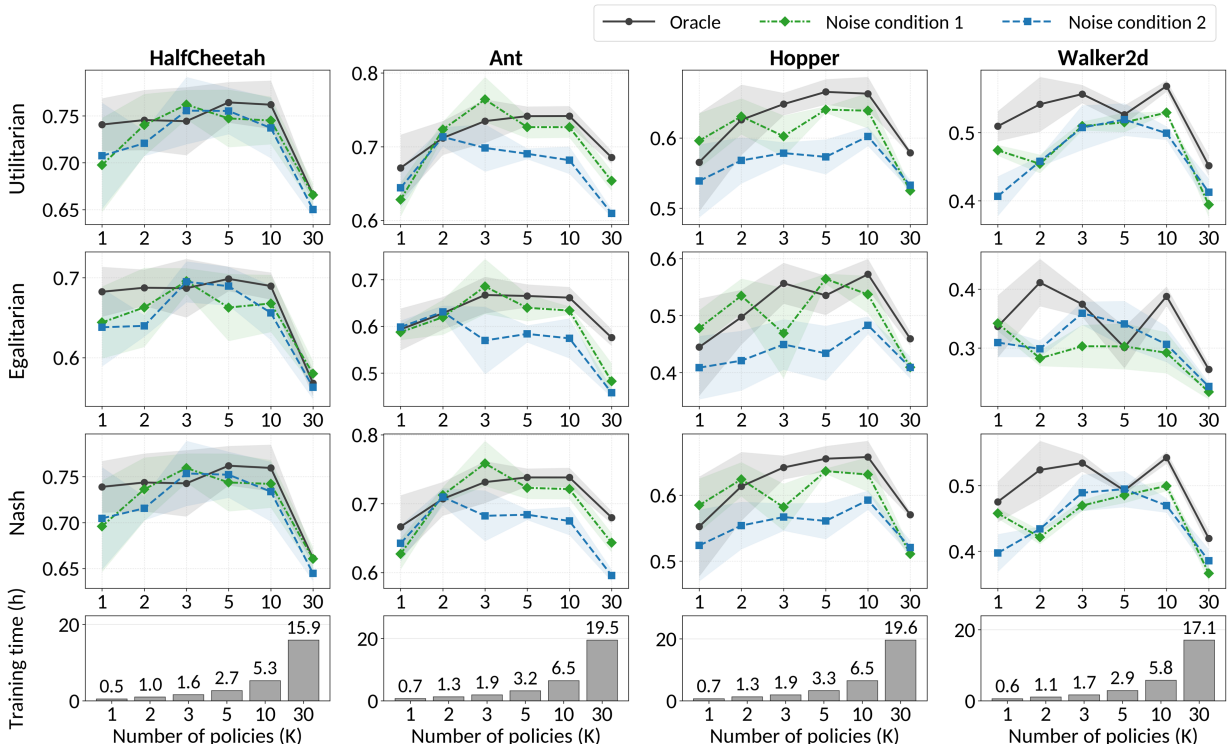

Figure 2. **Social welfare and training time under noisy preference labels.** Across four MuJoCo environments, the plots show utilitarian, egalitarian, and Nash welfare under oracle labels and two noisy-label conditions as the number of deployed policies K varies. The bottom row reports training time for each K. PREC with representative policies (K=2, 3, 5, 10) generally achieves higher social welfare than a single global policy (K=1) and fully individualized policies (K=30), while requiring less training time than training 30 policies.

Because policy learning depends only on the learned reward model of each cluster, this stage factorizes across clusters and can be carried out independently. In our practical implementation, we instantiate this stage with proximal policy optimization (PPO), but the framework is compatible with any downstream reinforcement learning algorithm.

4. Experiments

We designed experiments to assess whether PREC-based representative policies can maintain competitive social welfare relative to both policies trained for individual users and a single global policy in a sparse multi-user preference setting. Specifically, we evaluate its robustness in two aspects: (1) noisy human labels and (2) heterogeneous human preference distributions.

4.1. Experimental Setup

We implement PREC with the standard PbRL reward model architecture, an MLP-based Markovian reward model (Shin et al., 2023; Christiano et al., 2017), while noting that PREC is not tied to this particular design and can also be combined with other reward model architectures such as Preference Transformer (Kim et al., 2023). We evaluate our method on four Gym-MuJoCo locomotion agents with different levels of complexity and dynamics. We use MuJoCo locomotion as a representative preference-based robot learning testbed because it combines high-dimensional continuous control with trajectory-level behavioral variation, making reward specification nontrivial and preference feedback natural. This setting allows us to evaluate whether a bounded set of representative rewards can preserve heterogeneous preferences under sparse and noisy labels.

Personalized reward modeling. Existing PbRL studies (Park et al., 2022; Cheng et al., 2024; Kim et al., 2023) typically evaluate a single policy and therefore focus on whether the learned reward model accurately recovers a single target environment reward. In contrast, our setting may require up to 30 reward models per environment, and thus we model heterogeneous human preferences as follows. We construct a low-dimensional preference space for locomotion by collecting diverse PPO trajectories across multiple target velocities and training stages. We then apply principal component analysis (PCA) to standardized state-action features extracted from the dataset and use the top three principal components as the primary behavioral axes. Each user-specific reward is represented as a simplex-weighted combination of these three PCA-derived features, yielding a compact, data-driven representation of heterogeneous pref-

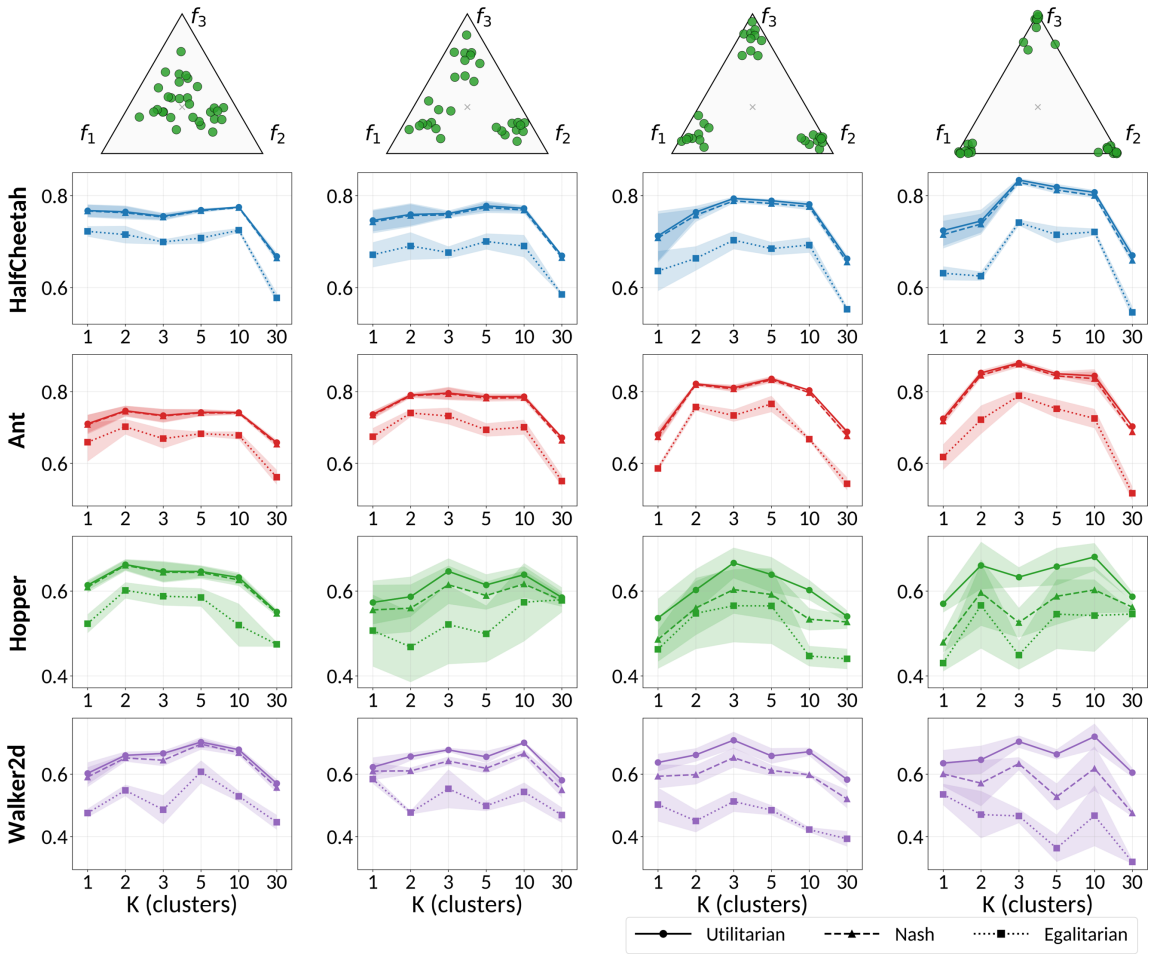

Figure 3. **Social welfare under different population preference distributions.** Top-row simplex plots show the preference distributions of 30 users over three PCA-derived behavior features (f_1, f_2, f_3), from broadly mixed (left) to highly polarized (right) populations. The remaining plots show utilitarian, Nash, and egalitarian welfare across four MuJoCo environments as the number of deployed policies K varies. PREC with representative policies (K=2, 3, 5, 10) generally achieves the highest social welfare, while fully individualized policies (K=30) suffer from limited feedback and a single global policy (K=1) yields lower welfare by ignoring preference diversity.

erences without requiring manually designed, task-specific reward terms.

Synthetic labeling procedure. We follow prior work (Park et al., 2022; Lee et al., 2021b; Hejna III & Sadigh, 2023; Metcalf et al., 2024) and construct preference label data using synthetic labelers. Each synthetic labeler is associated with a reward model parameterized by a distinct set of weights. Given a trajectory segment of 30 steps, the labeler assigns a scalar score; trajectories receiving a score of 7 or higher (on a 10-point scale) are labeled as good, and those below this threshold are labeled as bad. To improve the realism of the synthetic labeling process, we incorporate label noise following the pairwise preference noise injection scheme proposed in B-Pref (Lee et al., 2021b), adapted to the good/bad binary feedback setting. Specifically, when label noise is enabled, we apply noise to all users. For each user and each segment label, we uniformly sample one of the five B-Pref noise types—stochastic, myopic, mistake, skip, or equal—and apply the sampled noise mechanism to that label. Detailed adaptation procedures are provided in Appendix C.1.

Welfare-based evaluation metrics. For evaluation, we first compute an individual reward vector $\mathbf{r} = (r_1, \ldots, r_N)$ where each r_i measures how well the policy assigned to individual i aligns with their true preference weights over the PCA features. Specifically, r_i is obtained by averaging the inner product between the individual preference weight w_i and the episode-level PCA feature vector produced by the assigned cluster policy. We then evaluate the population using three welfare metrics. Utilitarian welfare computes the average reward across individuals and captures overall efficiency. Egalitarian welfare is defined as the bottom 10th percentile of the individual reward distribution, capturing poorly served users. Nash welfare computes the geomet-

ric mean of individual rewards, balancing efficiency and fairness by penalizing allocations in which any individual receives a very low reward. Together, these metrics characterize complementary aspects of policy assignment quality: average population performance, lower-tail fairness, and balanced collective welfare. Detailed definitions of the welfare evaluation metrics are provided in Appendix C.2.

4.2. Robustness to Noisy Human Labels

Figure 2 evaluates PREC under three labeling conditions: oracle labels and two progressively stronger B-Pref-based noise settings. We sample reward weights for 30 users, each of whom labels 50 independently sampled trajectory segments. The two endpoints define the baseline regimes. $K = 1$ corresponds to a single population-level policy trained from the pooled preference dataset, representing the conventional aggregation approach that collapses heterogeneous user preferences into one shared objective. In contrast, $K = 30$ corresponds to the conventional fully individualized PbRL setting, where each user receives a separate policy trained only on that user's preference dataset. PREC targets the intermediate regime, learning $K \in \{2, 3, 5, 10\}$ representative policies by sharing a population-level encoder and learning cluster-specific reward decoders.

Across the four MuJoCo environments, intermediate values of K generally achieve the best welfare. Higher utilitarian welfare shows that representative policies improve average alignment over a single global policy by preserving major preference modes. Higher egalitarian welfare indicates that these gains do not come only from well-served users, but also improve performance for lower-tail users. Higher Nash welfare further suggests a better efficiency–fairness trade-off, penalizing solutions that leave some users poorly served. These trends remain largely consistent under noisy labels, suggesting that clustering users and pooling their feedback stabilizes reward learning in sparse, noisy regimes where fully individualized policies can overfit. A controlled ablation in Appendix D.1 further separates the effect of SPR-based representation learning from the effect of reward clustering.

The bottom row of Figure 2 shows the corresponding training time. Because each learned reward model induces one downstream PPO policy, increasing K directly increases policy-optimization cost. Representative settings such as $K = 3$ or $K = 5$ achieve strong welfare gains while requiring far fewer policy-training runs than $K = 30$, making the resulting policy set easier to train. More importantly, they reduce the number of policy artifacts that must be validated, maintained, monitored, and deployed, which directly supports the deployability motivation of representative personalization.

4.3. Robustness to Preference Heterogeneity

The experiments in Figure 2 use approximately uniform preference distributions. To test whether PREC remains effective under stronger preference heterogeneity, we further consider increasingly polarized preference distributions. The first row of Figure 3 visualizes the user-specific weights over the three PCA-derived behavior features f_1, f_2, and f_3, ranging from broadly mixed preferences to populations concentrated around distinct preference modes.

Across the four MuJoCo environments, PREC settings generally achieve higher social welfare than the two endpoint baselines. Compared with the single population-level policy, intermediate values of K better preserve distinct preference modes instead of collapsing them into one averaged objective. Compared with fully individualized PbRL, they pool feedback among users with similar preferences, which stabilizes reward learning when each user provides only limited labels. These results indicate that PREC is robust not only to label noise, but also to changes in the underlying population preference structure. Additional visualizations of leaky-EM clustering under different preference distributions are provided in Appendix B.2.

4.4. Limitations

A limitation of our evaluation is that preference labels are generated by synthetic labelers and experiments are conducted in simulated locomotion environments. This design enables controlled comparisons across many users, policies, preference distributions, and noise conditions, but it does not fully capture the complexity of real human preferences or hardware deployment. Future work should validate representative reward learning with real user feedback and physical robotic systems.

5. Conclusion

We introduced Preference-based REward Clustering (PREC), a framework for deployable pluralistic alignment in robotics that learns representative reward models from offline trajectories and preference feedback. PREC learns a limited set of representative reward models, avoiding both the rigidity of a single global policy and the scalability challenges of fully individualized policies. By combining a population-level trajectory encoder, cluster-specific reward decoders, and leaky EM-based reward clustering, PREC preserves preference diversity while improving robustness under sparse and noisy feedback. Experiments across simulated robotic locomotion environments show that PREC improves aggregate user welfare across diverse preference distributions. These results highlight representative reward learning as a practical path toward scalable and certifiable robotic personalization.

Impact Statement

This paper advances deployable personalization in robotic systems, contributing to PbRL and human-robot interaction. The proposed approach may support more scalable personalization in user-facing and assistive robots, where safety, validation, and preference-data quality should be carefully considered. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted beyond these general considerations.

Acknowledgment

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No.RS-2024-00344732).

References

Bryant, D., Sadhwani, A., Fu, H., Smart, W. D., and Glas, D. F. Don't park there! learning socially-appropriate robot parking spots in the home. In *Proceedings of the 21st ACM/IEEE International Conference on Human-Robot Interaction*, pp. 1150–1159, 2026.

Cabi, S., Colmenarejo, S. G., Novikov, A., Konyushkova, K., Reed, S., Jeong, R., Zolna, K., Aytar, Y., Budden, D., Vecerik, M., et al. Scaling data-driven robotics with reward sketching and batch reinforcement learning. *arXiv preprint arXiv:1909.12200*, 2019.

Cheng, J., Xiong, G., Dai, X., Miao, Q., Lv, Y., and Wang, F.-Y. Rime: Robust preference-based reinforcement learning with noisy preferences. *arXiv preprint arXiv:2402.17257*, 2024.

Chhan, D., Novoseller, E., and Lawhern, V. J. Crowd-prefrl: Preference-based reward learning from crowds. *arXiv preprint arXiv:2401.10941*, 2024.

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

Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22, 1977.

Food and Drug Administration. 21 CFR 890.3480 – Powered Lower Extremity Exoskeleton. Electronic Code of Federal Regulations, 2015. URL https://www.ecfr.gov/current/title-21/chapter-I/subchapter-H/part-890/subpart-D/section-890.3480.

Forbes, M., Chung, M., Cakmak, M., and Rao, R. Robot programming by demonstration with crowdsourced action fixes. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 2, pp. 67–76, 2014.

Gabriel, I. Artificial intelligence, values, and alignment. *Minds and machines*, 30(3):411–437, 2020.

Gao, C.-X., Fang, S., Xiao, C., Yu, Y., and Zhang, Z. Hindsight preference learning for offline preference-based reinforcement learning. *arXiv preprint arXiv:2407.04451*, 2024.

Hejna, J. and Sadigh, D. Inverse preference learning: Preference-based rl without a reward function. *Advances in Neural Information Processing Systems*, 36:18806–18827, 2023.

Hejna, J., Rafailov, R., Sikchi, H., Finn, C., Niekum, S., Knox, W. B., and Sadigh, D. Contrastive preference learning: learning from human feedback without rl. *arXiv preprint arXiv:2310.13639*, 2023.

Hejna III, D. J. and Sadigh, D. Few-shot preference learning for human-in-the-loop rl. In *Conference on Robot Learning*, pp. 2014–2025. PMLR, 2023.

Hellou, M., Gasteiger, N., Lim, J. Y., Jang, M., and Ahn, H. S. Personalization and localization in human-robot interaction: A review of technical methods. *Robotics*, 10 (4):120, 2021.

Hwang, M., Weihs, L., Park, C., Lee, K., Kembhavi, A., and Ehsani, K. Promptable behaviors: Personalizing multi-objective rewards from human preferences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16216–16226, 2024.

Ingraham, K. A., Tucker, M., Ames, A. D., Rouse, E. J., and Shepherd, M. K. Leveraging user preference in the design and evaluation of lower-limb exoskeletons and prostheses. *Current Opinion in Biomedical Engineering*, 28:100487, 2023.

Ji, Z. and Chen, B. Pref-guide: Continual policy learning from real-time human feedback via preference-based learning. *arXiv preprint arXiv:2508.07126*, 2025.

Kim, C., Park, J., Shin, J., Lee, H., Abbeel, P., and Lee, K. Preference transformer: Modeling human preferences using transformers for rl. *arXiv preprint arXiv:2303.00957*, 2023.

Kirk, H. R., Vidgen, B., Röttger, P., and Hale, S. A. The benefits, risks and bounds of personalizing the alignment of large language models to individuals. *Nature Machine Intelligence*, 6(4):383–392, 2024.

Lee, K., Smith, L., and Abbeel, P. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. *arXiv preprint arXiv:2106.05091*, 2021a.

Lee, K., Smith, L., Dragan, A., and Abbeel, P. B-pref: Benchmarking preference-based reinforcement learning. *arXiv preprint arXiv:2111.03026*, 2021b.

Liang, X., Shu, K., Lee, K., and Abbeel, P. Reward uncertainty for exploration in preference-based reinforcement learning. *arXiv preprint arXiv:2205.12401*, 2022.

Liu, J., Qiu, Z., Li, Z., Dai, Q., Yu, W., Zhu, J., Hu, M., Yang, M., Chua, T.-S., and King, I. A survey of personalized large language models: Progress and future directions. *arXiv preprint arXiv:2502.11528*, 2025.

Madan, R., Lin, J., Goel, M., Xie, A., Liang, X., Lee, M., Guo, J., Thakkar, P. N., Banerjee, R., Barreiros, J., et al. Prioritouch: Adapting to user contact preferences for whole-arm physical human-robot interaction. *arXiv preprint arXiv:2509.18447*, 2025.

Metcalf, K., Sarabia, M., Fedzechkina, M., and Theobald, B.-J. Can you rely on synthetic labellers in preference-based reinforcement learning? it's complicated. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 10128–10136, 2024.

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

Park, H. W., Grover, I., Spaulding, S., Gomez, L., and Breazeal, C. A model-free affective reinforcement learning approach to personalization of an autonomous social robot companion for early literacy education. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 687–694, 2019.

Park, J., Seo, Y., Shin, J., Lee, H., Abbeel, P., and Lee, K. Surf: Semi-supervised reward learning with data augmentation for feedback-efficient preference-based reinforcement learning. *arXiv preprint arXiv:2203.10050*, 2022.

Santurkar, S., Durmus, E., Ladhak, F., Lee, C., Liang, P., and Hashimoto, T. Whose opinions do language models reflect? In *International conference on machine learning*, pp. 29971–30004. PMLR, 2023.

Schwarzer, M., Anand, A., Goel, R., Hjelm, R. D., Courville, A., and Bachman, P. Data-efficient reinforcement learning with self-predictive representations. *arXiv preprint arXiv:2007.05929*, 2020.

Shankar, K., Ding, D., and Gao, W. Low-burden llm-based preference learning: Personalizing assistive robots from natural language feedback for users with paralysis. *arXiv preprint arXiv:2604.01463*, 2026.

Shin, D., Dragan, A. D., and Brown, D. S. Benchmarks and algorithms for offline preference-based reward learning. *arXiv preprint arXiv:2301.01392*, 2023.

Slade, P., Kochenderfer, M. J., Delp, S. L., and Collins, S. H. Personalizing exoskeleton assistance while walking in the real world. *Nature*, 610(7931):277–282, 2022.

Soh, H. and Demiris, Y. Learning assistance by demonstration: Smart mobility with shared control and paired haptic controllers. *Journal of Human-Robot Interaction*, 4(3):76–100, 2015.

Tucker, M., Csomay-Shanklin, N., Ma, W.-L., and Ames, A. D. Preference-based learning for user-guided hzd gait generation on bipedal walking robots. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2804–2810. IEEE, 2021.

Wang, H., Chin, N., Gonzalez-Pumariega, G., Sun, X., Sunkara, N., Pace, M. A., Bohg, J., and Choudhury, S. Apricot: Active preference learning and constraint-aware task planning with llms. *arXiv preprint arXiv:2410.19656*, 2024.

Wang, R., Zhao, D., Suh, D., Yuan, Z., Chen, G., and Min, B.-C. Personalization in human-robot interaction through preference-based action representation learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7377–7384. IEEE, 2025.

Whitney, D., Rosen, E., and Tellex, S. Learning from crowdsourced virtual reality demonstrations. In *Proceedings of the 1st International Workshop on Virtual, Augmented, and Mixed Reality for HRI (VAM-HRI)*, 2018.

Wu, J., Antonova, R., Kan, A., Lepert, M., Zeng, A., Song, S., Bohg, J., Rusinkiewicz, S., and Funkhouser, T. Tidybot: Personalized robot assistance with large language models. *Autonomous Robots*, 47(8):1087–1102, 2023.

Xue, W., An, B., Yan, S., and Xu, Z. Reinforcement learning from diverse human preferences. *arXiv preprint arXiv:2301.11774*, 2023.

Zhang, Z., Sun, Y., Ye, J., Liu, T.-S., Zhang, J., and Yu, Y. Flow to better: Offline preference-based reinforcement learning via preferred trajectory generation. In *The Twelfth International Conference on Learning Representations*, 2023.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. Fine-tuning

language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A. Convergence of the Leaky EM Algorithm

This appendix establishes that, under the fixed-prior decoder-update formulation, the E-step and leaky M-step introduced in §3.3 jointly improve a single objective, and that the resulting sequence of leaky-ELBO values converges. The argument casts the leaky weight ω_{ik} as the unnormalized density of a restricted variational distribution over cluster assignments, and identifies the corresponding ELBO as the objective being improved.

A.1. Setup

Let $\Theta = (\rho_{1:K}, \psi^{\mathrm{dec}}_{1:K})$ collect the model parameters of the leaky mixture formulation, with $\rho_k > 0$ for all $k \in [K]$ and $\sum_{k=1}^{K} \rho_k = 1$. We assume finite decoder logits, so that $\mathcal{L}_{ik}(\psi^{\mathrm{dec}}_k) > 0$ and hence $F_{ik}(\Theta)$ and $\ell(\Theta)$ are finite. For user i and cluster k, define

$$F_{ik}(\Theta) \triangleq \log \rho_k + \log \mathcal{L}_{ik}(\psi^{\mathrm{dec}}_k),$$

so that $e^{F_{ik}(\Theta)} = \rho_k \mathcal{L}_{ik}$ is exactly the unnormalized posterior score used in the E-step. The marginal log-likelihood of the mixture model is

$$\ell(\Theta) \;=\; \sum_{i=1}^{N} \log \sum_{k=1}^{K} e^{F_{ik}(\Theta)} \;=\; \sum_{i=1}^{N} \log \sum_{k=1}^{K} \rho_k \, \mathcal{L}_{ik}. \tag{15}$$

Standard EM monotonically improves $\ell(\Theta)$ by iteratively maximizing an ELBO. We show below that our leaky EM procedure admits the same interpretation, with a specific restricted variational family in place of the full posterior.

Restricted variational family. Fix a leak coefficient $\nu \in [0, 1)$ and let $z \in [K]^N$ denote a hard assignment vector. For each user i, define the probability distribution over $[K]$

$$q_{ik}^{(z_i,\nu)} \;\triangleq\; \frac{\mathbf{1}[z_i = k] \;+\; \nu\, \mathbf{1}[z_i \neq k]}{c_\nu}, \qquad c_\nu \;\triangleq\; 1 + \nu(K-1). \tag{16}$$

One checks $\sum_k q_{ik}^{(z_i,\nu)} = 1$, so $q_i^{(z_i,\nu)}$ is a valid distribution that places mass $1/c_\nu$ on the assigned cluster z_i and mass ν/c_ν on each of the remaining $K-1$ clusters. The family interpolates between two natural extremes: $\nu = 0$ recovers the Dirac delta at z_i (hard assignment), while $\nu \to 1$ approaches the uniform distribution on $[K]$.

The leaky weight $\omega_{ik} = \max(\mathbf{1}[\hat{z}_i = k], \nu)$ from the main text is precisely the unnormalized density of this distribution:

$$q_{ik}^{(\hat{z}_i,\nu)} \;=\; \frac{\omega_{ik}}{c_\nu}. \tag{17}$$

The normalizing constant c_ν does not affect any $\arg\max$ over Θ, which is why the main-text update in Eq. (13) uses the unnormalized ω_{ik} directly.

A.2. The leaky ELBO

For an assignment $z \in [K]^N$ and parameters Θ, define

$$\mathcal{F}_\nu(z, \Theta) \;\triangleq\; \sum_{i=1}^{N} \left[\sum_{k=1}^{K} q_{ik}^{(z_i,\nu)} \, F_{ik}(\Theta) \;+\; H\big(q_i^{(z_i,\nu)}\big) \right], \tag{18}$$

where $H(q_i) = -\sum_k q_{ik} \log q_{ik}$ denotes the entropy of $q_i^{(z_i,\nu)}$. Expression (18) is the evidence lower bound (ELBO) associated with the restricted family (16).

Lemma A.1 (ELBO property)**.** *For every $z \in [K]^N$, $\nu \in [0, 1)$, and Θ,*

$$\mathcal{F}_\nu(z, \Theta) \;\leq\; \ell(\Theta).$$

Proof. If $\nu = 0$, then $q_i^{(z_i,0)}$ is a Dirac delta at z_i, and

$$\mathcal{F}_0(z, \Theta) = \sum_{i=1}^{N} F_{iz_i}(\Theta) \leq \sum_{i=1}^{N} \log \sum_{k=1}^{K} e^{F_{ik}(\Theta)} = \ell(\Theta),$$

since $\log \sum_k e^{F_{ik}(\Theta)} \geq F_{iz_i}(\Theta)$ for every i.

Now consider $\nu \in (0,1)$. For each i, concavity of log and Jensen's inequality applied to the distribution $q_i^{(z_i,\nu)}$ give

$$\log \sum_{k=1}^{K} e^{F_{ik}(\Theta)} = \log \sum_{k=1}^{K} q_{ik}^{(z_i,\nu)} \frac{e^{F_{ik}(\Theta)}}{q_{ik}^{(z_i,\nu)}} \geq \sum_{k=1}^{K} q_{ik}^{(z_i,\nu)} \log \frac{e^{F_{ik}(\Theta)}}{q_{ik}^{(z_i,\nu)}} = \sum_{k=1}^{K} q_{ik}^{(z_i,\nu)} F_{ik}(\Theta) + H(q_i^{(z_i,\nu)}).$$

Summing over $i \in [N]$ yields $\mathcal{F}_\nu(z,\Theta) \leq \ell(\Theta)$. □

Remark A.2 (Slack interpretation). The slack in Lemma A.1 is $\ell(\Theta) - \mathcal{F}_\nu(z,\Theta) = \sum_i \mathrm{KL}\big(q_i^{(z_i,\nu)} \,\big\|\, \gamma_i(\Theta)\big)$, where $\gamma_i(\Theta) = (\gamma_{i1}, \ldots, \gamma_{iK})$ is the true posterior from the E-step. Soft EM closes this slack by choosing $q_i = \gamma_i$, whereas leaky EM restricts q_i to the two-level family (16). The leak coefficient ν thus controls a bias-variance trade-off: smaller ν sharpens cluster specialization but incurs a larger KL gap to the true posterior.

A.3. E-step and M-step as coordinate ascent on $\mathcal{F}_\nu$

We now show that the two updates defined in §3.3 are exactly coordinate-wise maximizers of $\mathcal{F}_\nu$.

Proposition A.3 (E-step maximizes $\mathcal{F}_\nu$ in z)**.** *For any fixed* Θ*,*

$$\hat{z}_i \;=\; \arg\max_{k\in[K]} \gamma_{ik}(\Theta) \;=\; \arg\max_{k\in[K]} F_{ik}(\Theta) \quad \textit{for all } i \in [N]$$

is a maximizer of $\mathcal{F}_\nu(\cdot,\Theta)$ *over* $z \in [K]^N$*.*

Proof. Decompose the inner expectation in (18) as

$$\sum_{k=1}^{K} q_{ik}^{(z_i,\nu)} F_{ik}(\Theta) \;=\; \underbrace{\frac{\nu}{c_\nu} \sum_{k=1}^{K} F_{ik}(\Theta)}_{\text{independent of } z_i} \;+\; \frac{1-\nu}{c_\nu} F_{iz_i}(\Theta). \tag{19}$$

The entropy $H(q_i^{(z_i,\nu)})$ depends only on the multiset of probabilities $\{1/c_\nu, \nu/c_\nu, \ldots, \nu/c_\nu\}$, which is invariant under the choice of z_i, so $H(q_i^{(z_i,\nu)})$ is also independent of z_i. Therefore

$$\arg\max_{z_i\in[K]} \mathcal{F}_\nu(z,\Theta) \;=\; \arg\max_{z_i\in[K]} F_{iz_i}(\Theta) \;=\; \arg\max_{k\in[K]} \big(\log \rho_k + \log \mathcal{L}_{ik}\big),$$

which coincides with $\arg\max_k \gamma_{ik}$ since $\gamma_{ik} \propto e^{F_{ik}}$. The maximization separates across i, so the coordinatewise optimum $\hat{z}$ is a global optimum over $[K]^N$. □

Proposition A.4 (Leaky M-step and $\mathcal{F}_\nu$)**.** *For any fixed* $z \in [K]^N$ *and fixed* $\rho_{1:K}$*, the leaky decoder update in Eq.* (13) *is the argmax of* $\mathcal{F}_\nu(z,\Theta)$ *over* $\psi_{1:K}^{\mathrm{dec}}$ *when solved exactly. In a generalized-EM implementation where the update only ensures an increase of the weighted log-likelihood objective (e.g., one or more gradient steps), the update likewise satisfies* $\mathcal{F}_\nu(z,\Theta^{(t+1)}) \geq \mathcal{F}_\nu(z,\Theta^{(t)})$*.*

Proof. The entropy term in (18) does not depend on Θ. Using (17), the Θ-dependent part of $\mathcal{F}_\nu$ is

$$\sum_{i,k} q_{ik}^{(z_i,\nu)} F_{ik}(\Theta) = \; \frac{1}{c_\nu} \sum_{i,k} \omega_{ik} \big(\log \rho_k + \log \mathcal{L}_{ik}(\psi_k^{\mathrm{dec}})\big). \tag{20}$$

With $\rho_{1:K}$ fixed, the term $\sum_{i,k} \omega_{ik} \log \rho_k$ is a constant in ψ^{dec}. The maximization over each ψ_k^{dec} therefore decouples across k and reduces to

$$\psi_k^{\mathrm{dec}\,(t+1)} \;=\; \arg\max_{\psi_k^{\mathrm{dec}}} \sum_{i=1}^{N} \omega_{ik} \log \mathcal{L}_{ik}(\psi_k^{\mathrm{dec}}) \;=\; \arg\min_{\psi_k^{\mathrm{dec}}} \sum_{i=1}^{N} \omega_{ik} \sum_{m=1}^{M_i} \mathrm{BCE}\Big(\sigma\big(\hat{R}_{\psi_k}(\tau_{i,m})\big),\, y_{i,m}\Big),$$

where the second equality uses $\log \mathcal{L}_{ik} = -\sum_m \mathrm{BCE}(\sigma(\hat{R}_{\psi_k}(\tau_{i,m})), y_{i,m})$. The overall constant $1/c_\nu$ does not affect the $\arg\max$. This is exactly the leaky M-step update in Eq. (13). □

A.4. Monotone convergence

Theorem A.5 (Monotone improvement of the leaky ELBO)**.** *Let $(z^{(t)}, \Theta^{(t)})$ denote the iterates produced by alternating (i) an E-step that maximizes $\mathcal{F}_\nu(z, \Theta^{(t)})$ over z, and (ii) a leaky decoder update with leak coefficient $\nu \in [0, 1)$ that increases $\mathcal{F}_\nu(z^{(t+1)}, \Theta)$ over the decoder parameters, with $\rho_{1:K}$ held fixed. Then the sequence $\{\mathcal{F}_\nu(z^{(t)}, \Theta^{(t)})\}_{t\geq 0}$ is monotonically non-decreasing and converges to a finite limit.*

Proof. By Proposition A.3, the E-step computes $z^{(t+1)} \in \arg\max_z \mathcal{F}_\nu(z, \Theta^{(t)})$, and therefore

$$\mathcal{F}_\nu\big(z^{(t+1)}, \Theta^{(t)}\big) \;\geq\; \mathcal{F}_\nu\big(z^{(t)}, \Theta^{(t)}\big).$$

By Proposition A.4, the leaky decoder update increases $\mathcal{F}_\nu(z^{(t+1)}, \Theta)$ over the decoder parameters (with $\rho_{1:K}$ held fixed), so

$$\mathcal{F}_\nu\big(z^{(t+1)}, \Theta^{(t+1)}\big) \;\geq\; \mathcal{F}_\nu\big(z^{(t+1)}, \Theta^{(t)}\big).$$

Combining the two inequalities shows that $\{\mathcal{F}_\nu(z^{(t)}, \Theta^{(t)})\}_{t\geq 0}$ is monotonically non-decreasing.

Finally, for each user i,

$$0 < \sum_{k=1}^{K} \rho_k \mathcal{L}_{ik} \leq 1,$$

which implies

$$\ell(\Theta) = \sum_{i=1}^{N} \log \sum_{k=1}^{K} \rho_k \mathcal{L}_{ik} \leq 0.$$

By Lemma A.1, we have

$$\mathcal{F}_\nu(z, \Theta) \leq \ell(\Theta) \leq 0,$$

so the monotone sequence $\{\mathcal{F}_\nu(z^{(t)}, \Theta^{(t)})\}_{t\geq 0}$ is bounded above and therefore converges to a finite limit. □

Corollary A.6 (Hard EM as the $\nu = 0$ case)**.** *At $\nu = 0$, $q_{ik}^{(z_i,0)} = \mathbf{1}[z_i = k]$, and*

$$\mathcal{F}_0(z, \Theta) \;=\; \sum_{i=1}^{N} \log\big(\rho_{z_i}\, \mathcal{L}_{iz_i}\big),$$

which is the standard hard-assignment ELBO. Theorem A.5 then specializes to the corresponding hard-assignment monotone-improvement result under the same fixed-prior setting.

Remark A.7 (Scope of the analysis). Theorem A.5 concerns the decoder parameters $\psi_{1:K}^{\mathrm{dec}}$ under the assumption that the cluster priors $\rho_{1:K}$ are held fixed at each iteration. Any additional stabilizer that modifies ρ_k—such as Dirichlet-style prior smoothing or balanced reassignment of users across clusters—falls outside this fixed-prior scope and is therefore omitted from the coordinate-ascent argument.

B. Visualization of User Clusters

In this section, we visualize the clustering results from Section 4.2 and Section 4.3.

B.1. Visualization of User Clusters in Section 4.2

Figure 4, Figure 5, Figure 6, and Figure 7 show example clustering results from a single seed for HalfCheetah, Ant, Hopper, and Walker2d, respectively, under the oracle setting and two noisy-label settings.

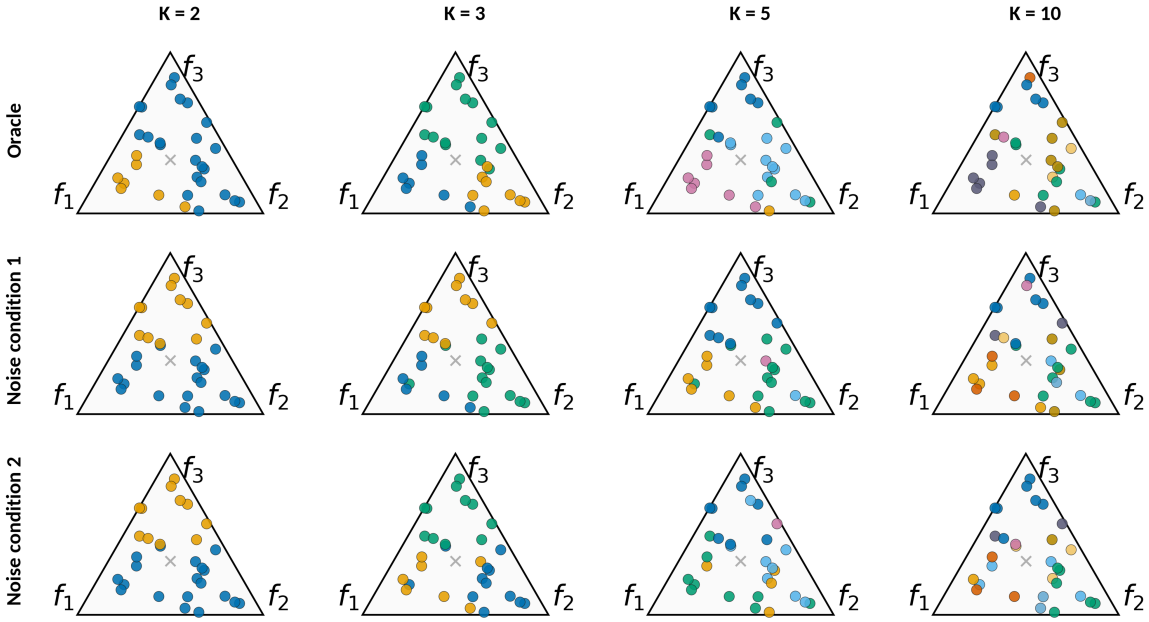

Figure 4. Each point denotes one user in the (f_1, f_2, f_3) preference simplex, and points with the same color belong to the same PREC cluster under oracle and noisy-label settings in HalfCheetah.

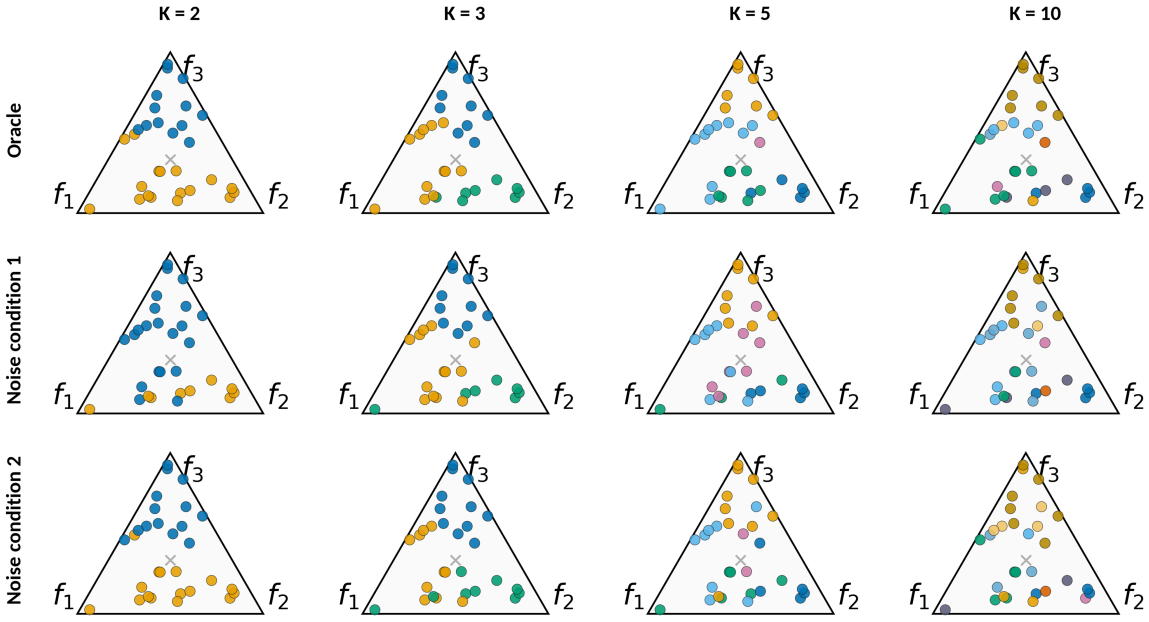

Figure 5. Each point denotes one user in the (f_1, f_2, f_3) preference simplex, and points with the same color belong to the same PREC cluster under oracle and noisy-label settings in Ant.

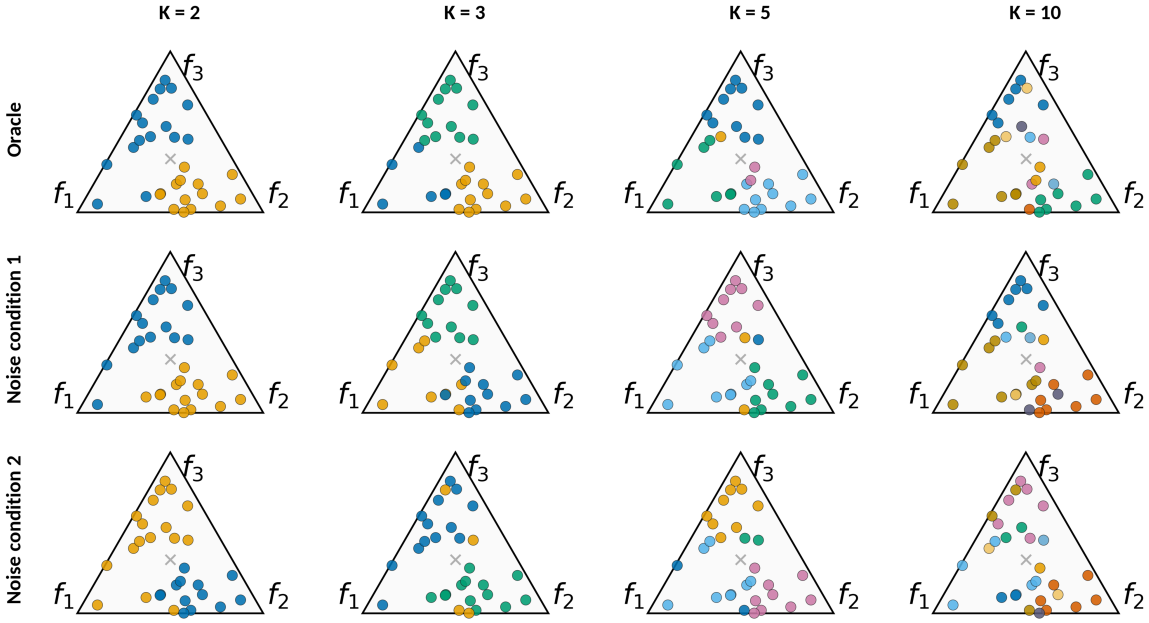

Figure 6. Each point denotes one user in the (f_1, f_2, f_3) preference simplex, and points with the same color belong to the same PREC cluster under oracle and noisy-label settings in Hopper.

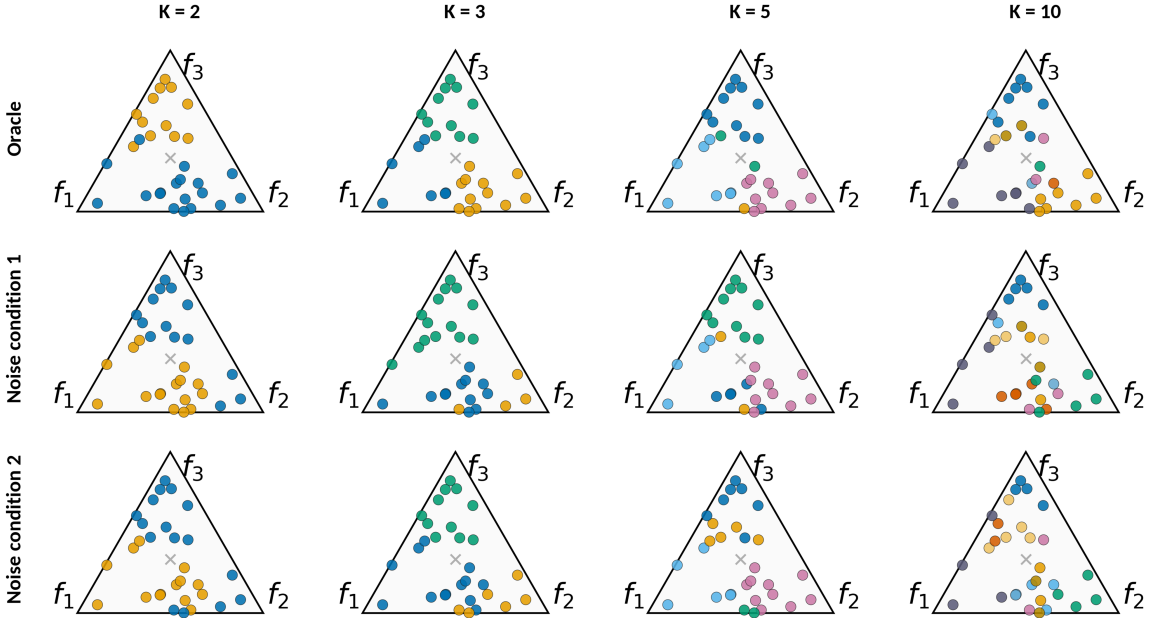

Figure 7. Each point denotes one user in the (f_1, f_2, f_3) preference simplex, and points with the same color belong to the same PREC cluster under oracle and noisy-label settings in Walker2d.

B.2. Visualization of User Clusters in Section 4.3

Figure 8, Figure 9, Figure 10, and Figure 11 show example clustering results from a single seed for HalfCheetah, Ant, Hopper, and Walker2d, respectively, under different population preference distributions.

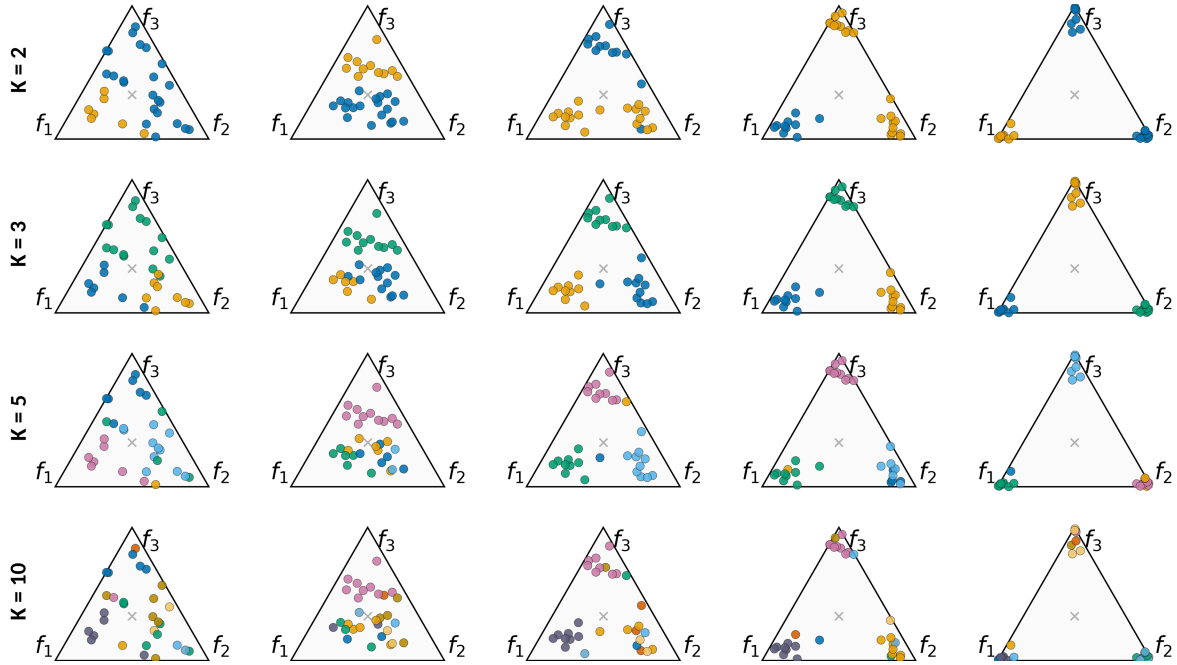

Figure 8. Each point denotes one user in the (f_1, f_2, f_3) preference simplex, and points with the same color belong to the same PREC cluster under different preference distributions in HalfCheetah.

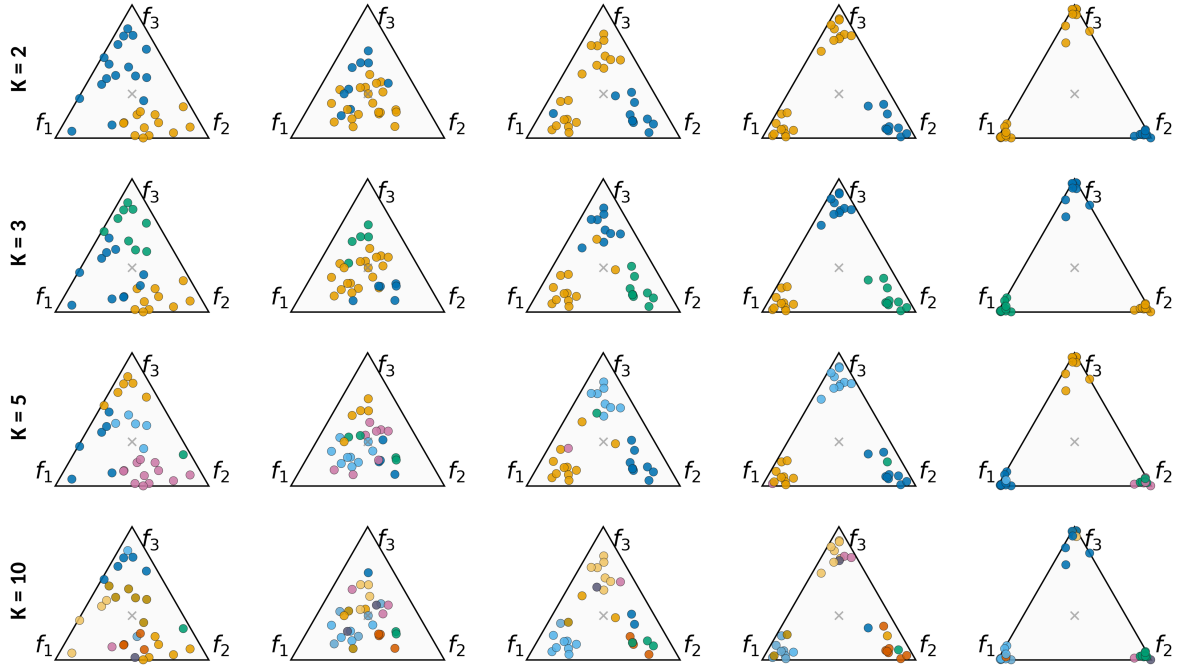

Figure 9. Each point denotes one user in the (f_1, f_2, f_3) preference simplex, and points with the same color belong to the same PREC cluster under different preference distributions in Ant.

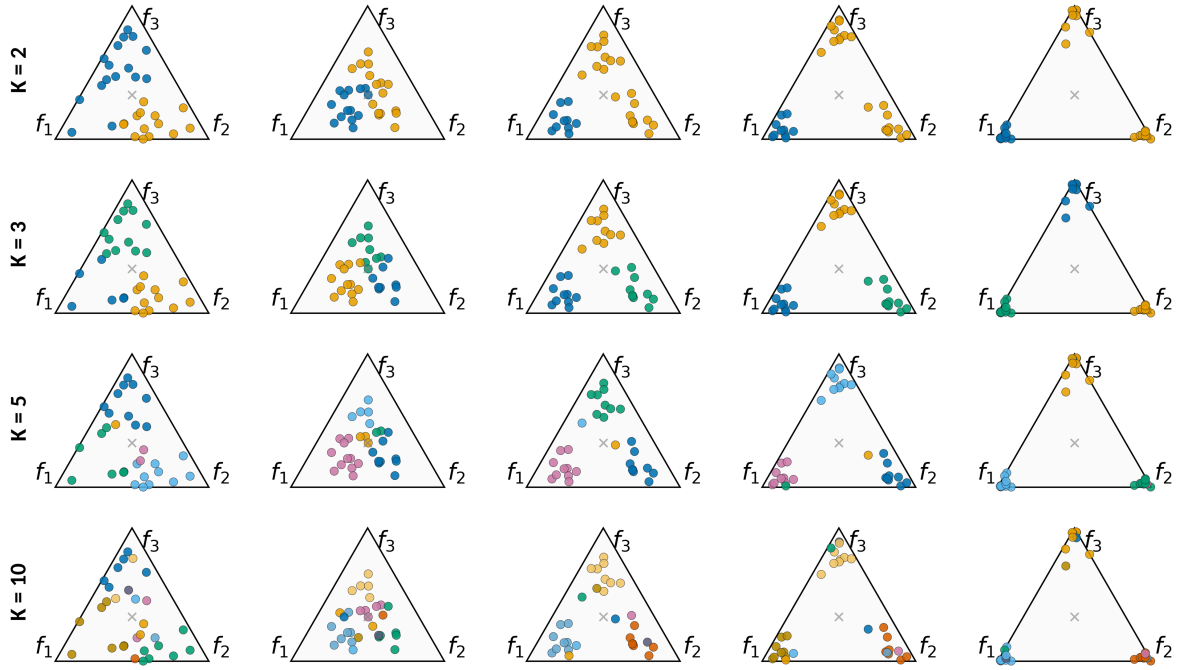

Figure 10. Each point denotes one user in the (f_1, f_2, f_3) preference simplex, and points with the same color belong to the same PREC cluster under different preference distributions in Hopper.

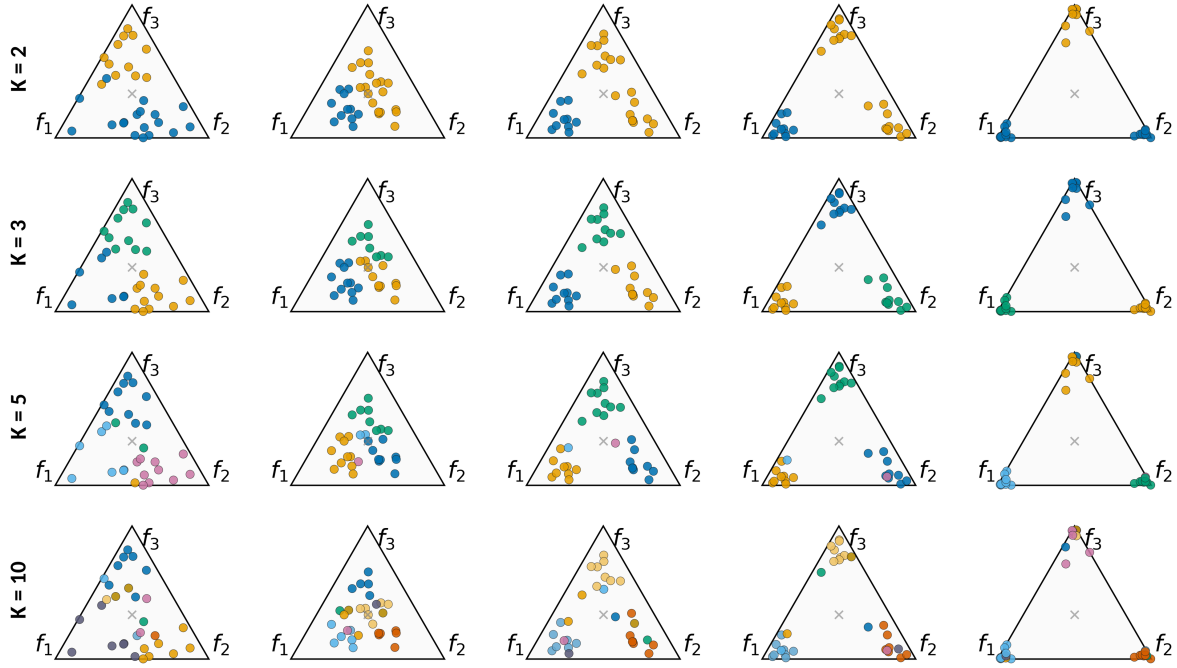

Figure 11. Each point denotes one user in the (f_1, f_2, f_3) preference simplex, and points with the same color belong to the same PREC cluster under different preference distributions in Walker2d.

C. Implementation Details

C.1. B-Pref Noise Adaptation

To evaluate robustness to imperfect human feedback, we adapt the simulated-teacher noise models from B-Pref (Lee et al., 2021b) to our offline binary-feedback setting. B-Pref originally defines irrational teachers for pairwise segment comparisons, including stochastic preferences, myopic preferences, accidental mistakes, skipped queries, and equal-preference responses. Since our data consist of single trajectory segments labeled independently by each user, we reformulate these five teacher irrationalities for a single-segment threshold-labeling problem.

For each user i and segment m, we compute a segment-level cumulative score

$$S_{i,m} = \sum_{t=1}^{T} r_i(s_{i,m,t}, a_{i,m,t}),$$

where r_i denotes the user-specific per-timestep reward and T is the segment length. We then obtain a binary label by thresholding this cumulative score,

$$y_{i,m} = \mathbb{I}\{S_{i,m} \geq \tau_i\},$$

where τ_i is the user-specific decision threshold.

Stochastic feedback. B-Pref applies a Bradley–Terry model to the cumulative-return difference between two trajectory segments. In our binary-feedback setting, we instead compare the cumulative score of a single segment against the user-specific threshold:

$$P(y_{i,m} = 1) = \sigma\left(\beta(S_{i,m} - \tau_i)\right).$$

Here, β is an inverse-temperature parameter controlling the sharpness of the stochastic threshold rule. The case $\beta = 1$ uses the unscaled score difference but still produces stochastic labels, whereas the deterministic threshold rule is recovered as $\beta \to \infty$.

Myopic feedback. B-Pref models a myopic teacher by discounting earlier rewards within each segment, so that the teacher places more emphasis on later timesteps. We use the same principle in our single-segment setting. Instead of using the original cumulative score $S_{i,m}$, we compute a myopic segment score

$$S_{i,m}^{\text{myopic}} = \sum_{t=1}^{T} w_t r_i(s_{i,m,t}, a_{i,m,t}), \quad w_t = \frac{\gamma^{T-t}}{\sum_{t'=1}^{T} \gamma^{T-t'}}.$$

The label is then obtained by thresholding $S_{i,m}^{\text{myopic}}$. When $\gamma < 1$, later timesteps receive larger weights, making the simulated user more sensitive to the end of the segment.

Mistake feedback. B-Pref models accidental mistakes by flipping the preference label with probability ϵ. We use the same binary-flip mechanism in our threshold-labeling setting:

$$y_{i,m} \leftarrow 1 - y_{i,m} \quad \text{with probability } \epsilon.$$

Thus, positive labels become negative and negative labels become positive with probability ϵ.

Skip feedback. In B-Pref, skip noise is reward-conditional: a teacher may skip a query when both compared segments are considered insufficiently informative or low-quality. Since our data collection setting is offline and each data point contains only a single labeled segment, we model skipped feedback as random missing supervision. Specifically, each segment label is removed from reward-model training with probability p_{skip} by assigning it zero sample weight.

Equal-preference feedback. B-Pref assigns an equal-preference response when the two compared segments have similar cumulative returns. In our single-segment setting, the analogous case occurs when a segment lies close to the user-specific decision boundary. We therefore assign an uncertain soft label,

$$y_{i,m} = 0.5 \quad \text{if} \quad |S_{i,m} - \tau_i| < \delta,$$

where δ controls the width of the ambiguity region around the threshold.

In addition to adapting the five B-Pref noise types from pairwise comparison to binary threshold labeling, we consider a label-level mixed-noise setting. For each noisy user and each segment label, one of the five noise types is sampled uniformly at random and applied to that label. Therefore, different labels from the same user may be corrupted by different noise mechanisms. When noise is applied, it is applied to all users in the dataset. We use two composite noise levels, summarized in Table 1.

Table 1. B-Pref-style label-noise levels used in our binary-feedback setting.

Noise condition	Stochastic β	Myopic γ	Mistake ϵ	Skip p_{skip}	Equal δ
Noise condition 1	1.00	0.90	0.10	0.10	0.10
Noise condition 2	0.50	0.85	0.20	0.25	0.20

C.2. Welfare Evaluation Metrics

For each individual i, we first compute an individual reward r_i that measures how well the assigned policy aligns with the individual's true preference over the PCA features. Let $w_i \in \Delta^{C-1}$ denote the preference weight of individual i, a_i denote the cluster assignment, and $\bar{f}_{a_i}^{(e)} \in [0,1]^C$ denote the episode-level PCA feature vector produced by the assigned policy in evaluation episode e. The individual reward is defined as

$$r_i = \frac{1}{E} \sum_{e=1}^{E} \left\langle w_i, \bar{f}_{a_i}^{(e)} \right\rangle,$$

which gives the reward vector $r = (r_1, \ldots, r_N) \in [0,1]^N$. We evaluate the resulting population outcome using three welfare metrics:

$$W_{\text{util}} = \frac{1}{N} \sum_{i=1}^{N} r_i, \qquad W_{\text{egal}} = Q_{0.10}(r), \qquad W_{\text{nash}} = \left(\prod_{i=1}^{N} r_i \right)^{1/N}.$$

Here, utilitarian welfare measures the average population reward, egalitarian welfare measures the lower tail of the reward distribution using the 10th percentile, and Nash welfare balances efficiency and fairness by penalizing outcomes in which any individual receives a very low reward.

C.3. Policy Learning Details

After EM clustering, we train one policy for each learned representative reward model using PPO from Stable-Baselines3. Each policy is optimized independently with the learned cluster-specific reward. The PPO hyperparameters are reported in Table 2.

Table 2. PPO hyperparameters used for policy optimization with learned reward models.

Hyperparameter	Value
Total environment steps	1.5M
Rollout length	2048 steps per environment
Mini-batch size	64
PPO update epochs	10
Optimizer	Adam
Learning rate	3×10^{-4}, constant schedule
Discount factor γ	0.99
GAE parameter λ	0.95

D. Ablation study

D.1. Ablation on the population-level encoder

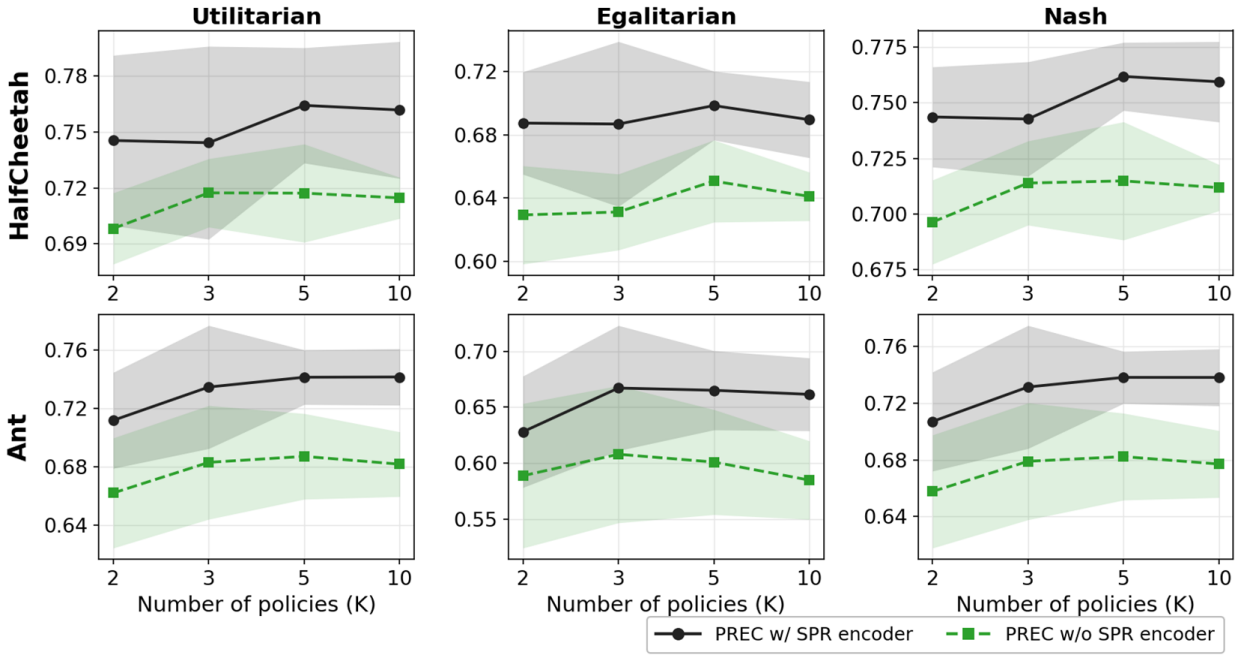

Figure 12. **Ablation of the SPR-based population-level encoder in the HalfCheetah and Ant environments.** We compare PREC with the SPR encoder and PREC without the SPR encoder across different numbers of representative policies. The top row reports results on HalfCheetah, and the bottom row reports results on Ant. Using the SPR encoder consistently improves utilitarian, egalitarian, and Nash welfare in both environments, indicating that the population-level representation learning provides a more stable and informative feature space for reward clustering and decoder learning.

Figure 12 isolates the effect of the population-level SPR encoder. Across all values of K, PREC with the SPR encoder achieves higher welfare than the variant without the encoder under all three metrics. This consistent improvement suggests that the shared encoder helps extract behaviorally meaningful state-action representations before preference-supervised reward learning. As a result, cluster-specific decoders can be trained on a more stable representation space, improving both average alignment and fairness-oriented welfare. The gap is especially important because the clustering and policy-training pipeline is otherwise unchanged, indicating that the SPR-based encoder contributes independently of the reward-clustering mechanism.