

EpiPersona: Persona Projection and Episode Coupling for Pluralistic Preference Modeling

Yujie Zhang and Weikang Yuan and Zhuoren Jiang* and Pengwei Yan

Department of Information Resources Management,

Zhejiang University

{yj.zhang, yuanwk, jiangzhuoren, yanpw}@zju.edu.cn

* Corresponding author

Abstract

Pluralistic alignment is essential for adapting large language models (LLMs) to the diverse preferences of individuals and minority groups. However, existing personalized reward models often mix stable personal traits with episode-specific factors, limiting their ability to generalize across episodes. To address this challenge, we propose **EpiPersona**, a framework for explicit **persona–episode coupling**. EpiPersona first projects noisy preference feedback into a low-dimensional persona space, where similar personas are aggregated into shared discrete codewords. This process separates enduring personal characteristics from situational signals without relying on predefined preference dimensions. The inferred persona representation is then coupled with the current episode, enabling episode-aware preference prediction. Extensive experiments show that EpiPersona outperforms the baselines. It achieves notable performance gains in hard episodic-shift scenarios, while remaining effective with sparse preference data.

1 Introduction

Aligning artificial intelligence with human preferences has been a crucial point of both theoretical and applied research for AI systems (Kirk et al., 2024b; Zhang et al., 2024; Zhu et al., 2025; Guan et al., 2025; Xu et al., 2025). However, most existing approaches treat preferences as uniform, overlooking the heterogeneity found across individuals and groups (Li et al., 2026; Oh et al., 2025). Addressing the needs of a diverse population is therefore essential, making pluralistic alignment (Poddar et al., 2024; Chen et al., 2025; Ryan et al., 2025) a vital step toward building human-centered AI systems (Zheng et al., 2023; Zhang et al., 2025; Kirk et al., 2024a; Yuan et al., 2026).

Large language models can learn from human feedback (Christiano et al., 2017; Bai et al., 2022; Xie et al., 2024; Cross et al., 2025; Riemer et al.,

2025). Since keeping a human in the loop to provide continuous feedback is impractical (Kaufmann et al., 2025; Cai et al., 2026), there is a growing demand for personalized reward models (Ryan et al., 2025; Bose et al., 2025) which can capture individual preferences using a small amount of user-specific annotation (Cai et al., 2026; Zhao et al., 2024; Chen et al., 2025).

To collect such preference feedback, studies predominantly adopt the pairwise comparison approach (e.g., “Which response do you prefer?”) to identify personalized preferences due to its user-friendliness (Zheng et al., 2023; Kirk et al., 2024a; Ryan et al., 2025; Balepur et al., 2025; Li et al., 2026; Poddar et al., 2024). Existing approaches can be categorized into *non-parametric* modeling and *parametric* modeling. *Non-parametric* modeling leverages historical retrieval (Zollo et al., 2025) or text-based preference summarization (Balepur et al., 2025; Ryan et al., 2025) to infer the personalized preference. While intuitive, this approach is less effective at higher-level abstractions as preferences are particularly implicit and complex (Jang et al., 2023; Balepur et al., 2025). *Parametric* modeling compresses long-term interaction histories into a semantic space, such as predefined preference dimensions (Oh et al., 2025; Li et al., 2026) (e.g., harmlessness, helpfulness, and humor) or context compression (Zhao et al., 2024; Poddar et al., 2024) for downstream tasks such as pluralistic reward learning (Bradley and Terry, 1952). However, the unique challenges remain unsolved. Two interconnected challenges are highlighted:

First, existing methods mix stable personal traits with temporary episodic signals in the representation, failing to separate *stable* traits from *dynamic* episodic influences. As a result, static traits are not disentangled, limiting cross-episode generalization. *Second*, prevailing methods rely on predefined dimensions to model preferences. Human preference is better understood as a latent semantic space to be

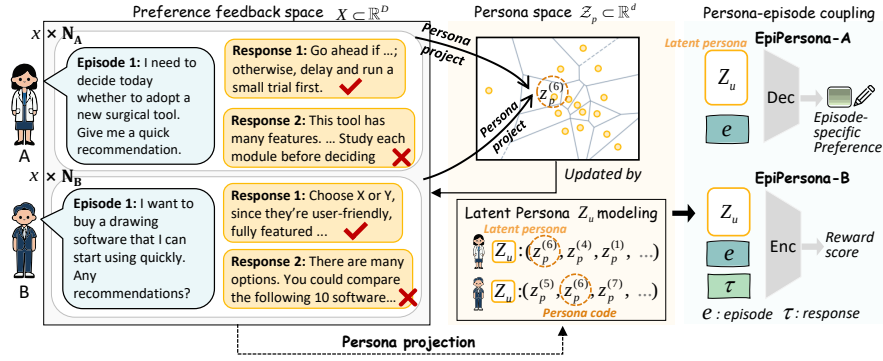


Figure 1: We propose **EpiPersona**. Based on an individual’s historical preference information, we model the individual’s latent persona Z_u . We further introduce two variants of EpiPersona that couple persona Z_u with episode context e for preference prediction.

discovered (Poddar et al., 2024), while predefined dimensions constrain its richness and introduce priori biases.

To address these challenges, we propose **persona projection**, which disentangles stable traits (i.e., the persona) from episode-specific preference feedback. It is motivated by the insight that individuals’ preferences are driven by their latent and stable personas. For example (see Figure 1), a user *A* in a *work decision* scenario and user *B* in an *online shopping* scenario may both prefer concise and direct responses; their feedback thus reflects the same *efficiency-driven* persona. Formally, our method projects individuals’ preference feedback from the high-dimensional token space $X \subset \mathbb{R}^D$ into a low-dimensional **persona space** $Z_p \subset \mathbb{R}^d$ ($d \ll D$), enabling feedback $x \in X$ with similar latent personas to be aggregated into shared persona codewords $z_p \in Z_p$. In other words, we learn a mapping $x \mapsto z_p$. This mapping function converts the originally redundant and noisy feedback into a combinatorial persona codeword, enabling robust generalization across individuals.

Building on persona projection, we propose EpiPersona, a method for **PERSONA-EPI**isode coupling. It constructs persona codewords automatically without relying on predefined dimensions by proposing *parameterized abductive reasoning* and *VQ-based mapping* (Van Den Oord et al., 2017). After disentangling personas, we further **couple** them with the current episode, capturing both stable persona and situational factors, enabling robust prediction of cross-episode preferences. Our main contributions are as follows:

- We propose EpiPersona, a framework that sep-

arates stable persona traits from episode-specific signals without relying on predefined preference dimensions.

- We propose parameterized abductive reasoning and VQ-based persona projection to ground preference feedback into a discrete persona codebook, coupled with episode context for contextualized preference prediction.

- Extensive experiments demonstrate that EpiPersona outperforms baselines, with particularly robust generalization in challenging settings such as the episodic-shift scenario.

2 Related work

Pluralistic alignment adapts the preferences of diverse groups, especially minority populations, thereby maximizing the collective benefits (Zollo et al., 2025) and promoting fairness across heterogeneous user communities (Kirk et al., 2024a,b). Studies show that even high-performing models struggle to satisfy preferences across different regions, cultures, and topics (Kirk et al., 2024a). These challenges underscore the diversity and complexity of human preferences. Due to the sparseness of individual preference data (Oh et al., 2025; Li et al., 2026), existing approaches use human feedback in the form of pairwise preference comparisons (Zheng et al., 2023; Poddar et al., 2024; Ryan et al., 2025; Balepur et al., 2025; Li et al., 2026; Kirk et al., 2024a). Prior personalized reward models employ the following methods to model pluralistic preferences:

Non-parametric modeling. *Preference summarization (text-based).* Preference dimensions are often defined as such attributes as helpfulness,

honesty, and harmlessness (Askill et al., 2021). However, simplified dimensions may not be able to capture the diversity of human preferences (Kirk et al., 2024b). Studies (Ryan et al., 2025; Balepur et al., 2025) utilize *preference attribution* methods to extract individual preferences. Among them, SynthesizeMe (Ryan et al., 2025) uses bootstrap reasoning on limited user interaction data and verifies inferred preferences on the validation set to synthesize the personas. PI (Balepur et al., 2025) leverages *text-based* abductive reasoning on preference data to infer users’ underlying needs and interests. They can be adaptive to in-context learning, such as *LLM-as-a-judge* optimizing (Gu et al., 2026) and personal LLM (Ryan et al., 2025; Li et al., 2026; Balepur et al., 2025; Du et al., 2025). *Retrieval-augmented generation (RAG)*. Personal-LLM (Zollo et al., 2025) combines context learning with historical information retrieval to produce the personalized responses. In contrast, EpiPersona employs the parameterized modeling approach to globally and abstractly represent personalized preferences.

Parametric modeling. Parameterized preferences have been used for personalized reward model training (Zhao et al., 2024; Poddar et al., 2024). For example, GPO (Zhao et al., 2024) adds a few-shot preference module on top of the base LLM and learns to predict group preference scores from historical user data. VPL (Poddar et al., 2024) compresses historical user preference data into a parameterized latent distribution using the VAE module. *Predefined preference dimensions.* AMPLe (Oh et al., 2025) models user preferences along multiple predefined dimensions via posterior updates, while Li et al. (2026) constructs a preference space grounded in psychological models. In contrast, EpiPersona requires no predefined dimensions, instead disentangling stable personas directly from episode-specific feedback.

3 Problem Setting

Let $\mathcal{U}$ denote a population of individuals and $\mathcal{E}$ the space of episodes. Each individual $u \in \mathcal{U}$ may participate in multiple episodes $e \in \mathcal{E}$, where each episode is associated with a query q_e . For a given episode e , the system presents a set of candidate responses $\{\tau_e^{(i)}\}_{i=1}^K$. We assume that preferences are generated by an underlying, unobserved reward function:

$$R_u(\tau_e) \rightarrow \mathbb{R}, \quad (1)$$

which reflects the utility assigned to response τ_e by individual u under episode e .

Observed feedback. The reward function R_u is not directly observable. Instead, for each individual u and episode e , we observe the pairwise preference feedback of the form:

$$x = (\tau_e^{(i)}, \tau_e^{(j)}, y_e), \quad (2)$$

where $y_e \in \{0, 1\}$. $y_e = 1$ indicates $\tau_e^{(i)}$ is preferred over $\tau_e^{(j)}$ in episode e . We denote the collection of observations for individual u as:

$$X_u = \{(\tau_e^{(i)}, \tau_e^{(j)}, y_e) \mid e \in \mathcal{E}_u\}. \quad (3)$$

Latent persona assumption. We define the episode-related response τ_e and a latent persona Z_u of individual u . We assume that the underlying reward function can be modeled as:

$$R_u(\tau_e) \hat{=} R_\theta(Z_u, \tau_e) \in \mathbb{R}, \quad (4)$$

where $R_\theta(\cdot, \cdot)$ is an estimated reward function. The latent persona Z_u is unobserved but can be inferred from observable features X_u :

$$Z_u \sim p(Z_u \mid X_u). \quad (5)$$

4 Method

We present an overview of EpiPersona in Figure 1. First, we disentangle individuals’ stable persona traits from their preference feedback via persona projection, mapping feedback with similar personas into a shared latent space (Section 4.1). Subsequently, the model couples the inferred latent persona with the specific episodes to enable contextualized preference prediction (Section 4.2). The EpiPersona achieves persona disentanglement through parameterized abductive reasoning (Section 4.3) and vector-quantized mapping (Section 4.4), automatically constructing a structured persona space without requiring predefined preference dimensions.

4.1 Persona projection

Formulation and assumptions. We assume that each observed preference feedback $x \in X_u$ from individual u is jointly determined by the episode-specific responses τ_e and a latent, stable persona Z_u . Formally, the preference formation process can be expressed as:

$$y_e \sim p(y_e \mid Z_u, \tau_e^{(i)}, \tau_e^{(j)}). \quad (6)$$

Latent persona modeling. The overall persona projection mapping function is illustrated in Figure 2. Specifically, we model an individual’s persona Z_u as a latent variable that potentially follows a *multimodal* distribution, reflecting that a single individual may exhibit multiple persona patterns (Chen et al., 2025). Concretely, we represent the latent persona of individual u as a set of persona codewords:

$$Z_u = \{z_p \mid z_p \in \mathcal{Z}_p\} \quad (7)$$

where each z_p corresponds to a stable persona codeword. Crucially, each persona codeword z_p lies in a *shared persona space* $\mathcal{Z}_p$, which is common to all individuals. To further represent the persona codeword z_p , we leverage the individual’s feedback $x \in X_u$. Specifically, we map feedback instance x into the persona codeword $z_p \in \mathcal{Z}_p$ via a persona projection function g :

$$g : x \mapsto z_p, \quad z_p \in \mathcal{Z}_p \subset \mathbb{R}^d, \quad (8)$$

where $d \ll D$ and D is the dimension of the original feedback $x \in \mathbb{R}^D$. It learns to automatically construct the persona space.

In this way, $Z_u = \{g(x) \mid x \in X_u\} \subseteq \mathcal{Z}_p$ captures the diverse persona traits of an individual, while grounding them in the shared persona space $\mathcal{Z}_p$. This shared space $\mathcal{Z}_p$ is also jointly updated from feedback across all individuals, enabling it to retain collective universality.

Objective of persona projection. In summary, the objective of persona projection is to map episode-specific feedback x into a low-dimensional persona codeword $z_p \in \mathcal{Z}_p$. This process aims to disentangle stable persona from episodic variability. We realize the mapping g through *parameterized abductive reasoning* (Section 4.3) and *VQ-based mapping* (Section 4.4).

4.2 Persona-episode coupling

Based on persona projection, we couple the user persona Z_u with the episode to model an individual’s preferences. This process captures how an individual’s persona may be activated in an episode.

Since downstream scenarios differ fundamentally in output space, we propose two variants: *generative* for preference generation (natural language), and *discriminative* for preference scoring.

EpiPersona-A (generative): This method generates a natural language description of the individual preference under the episode e . Note that the

input only includes the latent persona Z_u (based on historical feedback) and episode e . It allows the generated Pref_e to be used in a plug-and-play manner, such as judging the individuals’ potential choices. Formally, the process is defined as:

$$\text{Pref}_e = g_\psi(Z_u, e) \quad (9)$$

where g_ψ is a transformer-based language model trained with an NTP (Radford et al., 2018) objective. The output follows a JSON format containing keys such as *persona*, *value*, *identification*, *intent*, *style*, etc. Detailed comparison of EpiPersona-A with related methods is in Appendix A.4.

EpiPersona-B (discriminative): The reward is modeled via a Bradley-Terry preference comparison (Bradley and Terry, 1952) based on latent persona Z_u :

$$R_\theta(Z_u, \tau_e) = \mathbf{W}^\top \phi(h_{\theta_h}(Z_u, \tau_e)) \quad (10)$$

where h_{θ_h} is an encoder producing episode-persona coupled representation, ϕ is an optional nonlinear transformation, $\mathbf{W}$ maps the representation to a scalar reward. More detailed model architecture is shown as Figure A.2 in the Appendix. For the inference algorithm, see the Alg.1 and 2 in the Appendix.

4.3 Parameterized abductive reasoning

To realize the persona projection function g (see Eq.8), we first implement the instance-level encoding, mapping each preference-feedback instance to an estimated persona r .

We approach this using *abductive reasoning* following recent work (Balepur et al., 2025; Ryan et al., 2025), which infers a plausible persona from observed data. Building on this idea, we introduce *parameterized* abductive reasoning, which estimates latent preference by comparing choices. Specifically, it does not require the LLM to generate a human-readable reasoning trace. The *estimated persona* r is represented as a continuous learnable vector:

$$\lambda(\mathcal{P}, \tau_e^{(i)}, \tau_e^{(j)}, y_e) \mapsto r, \quad (11)$$

where $\mathcal{P}$ denotes the prompt that elicits the model to infer the persona underlying the user’s choice, λ is the LLM-based encoder. The estimated persona r encodes the rationale behind an individual’s preference feedback. At this stage, r is still instance-level and *not* shared across users.

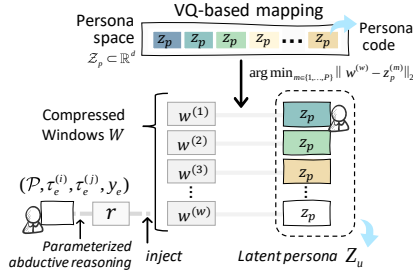


Figure 2: Overview of the persona projection mapping.

4.4 VQ-based mapping

Building on parameterized abductive reasoning (see Section 4.3), we then map the estimated persona r into a shared persona space using a VQ-based mapping, completing the persona projection function g . The procedure is as follows:

Variable persona compressed window. Let the estimated persona $r \in \mathbb{R}^d$ be injected into compressed windows as an embedding. The resulting window representation is then projected into a shared representation space via $f_{\theta_f}(\cdot)$, yielding $w^{(w)} = f_{\theta_f}(r)$. Each window $w^{(w)}$ thus encapsulates a latent estimated persona derived from observed individual feedback. The window size is flexibly configurable.

Persona codebook. We then define the persona codebook Z_p , which represents the shared persona space. P denotes the codebook size. Formally, Z_p is defined as:

$$Z_p = \{z_p^{(1)}, z_p^{(2)}, \dots, z_p^{(P)}\}, \quad z_p \in \mathbb{R}^d. \quad (12)$$

VQ-based mapping aims to map the compressed windows onto a set of learnable persona codewords z_p , without assuming the predefined dimensions.

Nearest-neighbor quantization. We represent the n windows of user u as $\mathcal{W} = [w_1, \dots, w_n]^T \in \mathbb{R}^{n \times d}$. $w^{(w)}$ is quantized by selecting the nearest persona codeword from Z_p :

$$m^{*(w)} = \arg \min_{m \in \{1, \dots, P\}} \|w^{(w)} - z_p^{(m)}\|_2. \quad (13)$$

The corresponding persona codeword is then defined as:

$$Z_u = (z_p^{(m^{*(w)})})_{w=1}^n. \quad (14)$$

Loss. To train the VQ module, we adopt two auxiliary objectives: a commitment loss $\mathcal{L}_{\text{commit}}$ and an entropy-based diversity loss $\mathcal{L}_{\text{div}}$.

Diversity regularisation. To avoid codebook collapse (Zhu et al., 2024), we propose an entropy-based diversity loss. For each window representation $w^{(w)}$, we compute a soft assignment over the codebook via cosine similarity with temperature τ :

$$a_c^{(w)} = \frac{\exp(\cos(w^{(w)}, z_p^{(c)}) / \tau)}{\sum_{c'=1}^P \exp(\cos(w^{(w)}, z_p^{(c')}) / \tau)}. \quad (15)$$

We average the soft assignments across all n windows to obtain the empirical codeword utilisation distribution $\bar{p}_c = \frac{1}{n} \sum_{w=1}^n a_c^{(w)}$, and define the diversity loss as its negative entropy:

$$\mathcal{L}_{\text{div}} = -H(\bar{\mathbf{p}}) = -\sum_{c=1}^P \bar{p}_c \log \bar{p}_c \leq 0. \quad (16)$$

Minimising $\mathcal{L}_{\text{div}}$ is equivalent to maximising $H(\bar{\mathbf{p}})$, thereby encouraging uniform utilisation across all P persona codewords. Crucially, the gradient flows through $w^{(w)}$ back to the encoder f_{θ_f} , while the codebook is detached, so only the encoder weights are steered toward producing more diverse representations.

Commitment loss. $\mathcal{L}_{\text{commit}}$ is a standard VQ commitment (Van Den Oord et al., 2017) loss that encourages the projected window representations to stay close to their assigned codebook entries. It prevents the encoder from producing representations that drift arbitrarily between codewords:

$$\mathcal{L}_{\text{commit}} = \sum_{w=1}^n \|w^{(w)} - \text{sg}[z_p^{(m^{*(w)})}]\|_2^2, \quad (17)$$

where $\text{sg}[\cdot]$ denotes the stop-gradient operator. Gradients are propagated back to the encoder via the straight-through estimator (Bengio et al., 2013), while the codebook entries are updated through an exponential moving average (EMA). The details of EMA are provided in Appendix A.6.

4.5 Overall training objective

The overall training objective combines the task loss $\mathcal{L}_{\text{task}}$ with the two VQ auxiliary terms:

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \alpha_{\text{commit}} \mathcal{L}_{\text{commit}} + \alpha_{\text{div}} \mathcal{L}_{\text{div}}, \quad (18)$$

where α_{commit} and α_{div} are the corresponding weights.

Corresponding to Section 4.2, the task loss of EpiPersona-A is the autoregressive next-token prediction loss:

$$\mathcal{L}_{\text{task}}^A = -\sum_{t=1}^T \log p_{\psi}(s_t | s_{<t}, Z_u, e), \quad (19)$$

where $s = (s_1, \dots, s_T)$, which is the episode-specific preference description.

The task loss of EpiPersona-B is the Bradley-Terry pairwise comparison loss:

$$\Delta R_e = R_\theta(Z_u, \tau_e^{(i)}) - R_\theta(Z_u, \tau_e^{(j)}), \quad (20)$$

then:

$$\mathcal{L}_{\text{task}}^B = -y_e \log \sigma(\Delta R_e) - (1 - y_e) \log(1 - \sigma(\Delta R_e)). \quad (21)$$

Together, these objectives jointly optimize the task performance and update VQ representations.

5 Experiment

5.1 Experiment setting

Baselines. We compare EpiPersona against two categories of methods: *a) Non-parametric preference modeling.* These methods model preferences using natural language and can be adapted to LLMs in a plug-and-play manner: **SynthesizeMe** (Ryan et al., 2025) synthesizes personas through LLM-based bootstrap reasoning and validation; **PersonaLLM** (Zollo et al., 2025) retrieves similar interactions as in-context examples to adapt to individual preferences; **PI** (Balepur et al., 2025) infers personas from preference feedback and incorporates them into context for alignment. *b) Parametric preference modeling.* **GPO** (Zhao et al., 2024) uses a transformer to predict preferences conditioned on context samples; **VPL** (Poddar et al., 2024) employs a VAE to model individual preferences as latent distributions; **Bradley-Terry (BT)** (Bradley and Terry, 1952) learns reward scores from pairwise comparisons without personalization. **LoRe** (Bose et al., 2025) represents individual reward functions as weighted combinations of shared basis functions in a low-rank subspace; **MRM** (Cai et al., 2026) reformulates personalized reward modeling as meta-learning via a MAML-style framework.

Tasks We evaluate on two tasks with different output spaces. This setting aims to assess the personalized preferences modeling ability comprehensively. Task selection is based on downstream task requirements. *a) Generative.* Following existing work (Ryan et al., 2025), this task evaluates the ability to generate natural language descriptions of individual preferences. Given an individual’s historical data and the current episode, the model generates the individual’s preference. An LLM

judge then uses it to predict which of two candidate responses the individual would prefer. *b) Discriminative.* This task evaluates the ability to predict personalized preference scores directly. Following standard practice (Zhao et al., 2023; Cai et al., 2026), we evaluate on pairwise comparisons and report accuracy. See Appendix A.1 for the complete task formulation.

Datasets *Prism* (Kirk et al., 2024a) is a values- and controversy-guided human feedback dataset linking sociodemographics and stated preferences of 1,500 participants from 75 countries. It provides pairwise comparisons of model responses within shared conversational contexts. *Arena* (Zheng et al., 2023) is a crowdsourced benchmark where users compare two anonymous LLM outputs and select their preferred response. The distribution and characteristics of the dataset are shown in Appendix A.2.

Dataset setting. The datasets contains no overlapping individuals between the train and test sets. Moreover, the historical and current samples do not share overlapping conversations. This setup ensures we evaluate on *unseen* individuals with *unseen* conversations, which aligns with real-world scenarios.

Implementation setup Fine-tuning is performed using the LoRA method applied to the main linear layers. The VQ codebook is initialized using K-means. λ is consistent with the finetuned backbone model used. Additional implementation details and hyperparameter experiments are provided in Appendix A.3 and A.9. **Additional setup for generative setting:** For downstream evaluation (LLM-as-a-Judge), LLMs including GPT-OSS-120B and Llama-3.3-70B-Instruct are employed. The backbone model used is Llama-3.1-8B. **Additional setup for discriminative setting:** The backbone model of all baselines used is Llama-3.1-8B and Llama-3.2-3B. The source code is available at <https://github.com/PiLab-ZJU/EpiPersona>.

5.2 Results

Experiments and analyses are presented in Sections 5.2.1 and 5.2.2, respectively.

5.2.1 Performance validation

We evaluate *EpiPersona* from two aspects: 1) Performance validation, 2) Contribution of each component to preference modeling.

Table 1: Comparison on generative task (LLM-as-a-judge). Values are percentages with standard deviations (95% CI).

	Prism		Arena	
	GPT-OSS-120B	Llama-3.3-70B	GPT-OSS-120B	Llama-3.3-70B
<i>Baseline</i>				
Random	50%	50%	50%	50%
LLM as a judge (Base model)	56.30 $\pm$ 1.26%	55.90 $\pm$ 1.22%	64.36 $\pm$ 3.91%	64.18 $\pm$ 3.91%
<i>Personalization Methods</i>				
SynthesizeMe (GPT-OSS-120B)	57.24 $\pm$ 1.27%	57.67 $\pm$ 1.27%	64.89 $\pm$ 4.11%	64.39 $\pm$ 3.93%
SynthesizeMe (Llama-3.1-8B)	57.27 $\pm$ 1.28%	56.86 $\pm$ 1.26%	63.83 $\pm$ 4.26%	64.17 $\pm$ 3.85%
PI (GPT-OSS-120B)	54.00 $\pm$ 1.37%	57.22 $\pm$ 1.28%	61.72 $\pm$ 4.78%	63.28 $\pm$ 3.98%
PI (Llama-3.1-8B)	57.48 $\pm$ 1.28%	56.58 $\pm$ 1.36%	63.06 $\pm$ 4.13%	63.97 $\pm$ 3.97%
PersonalLLM (demo num=1)	54.39 $\pm$ 1.26%	55.81 $\pm$ 1.24%	59.71 $\pm$ 4.10%	59.33 $\pm$ 4.09%
PersonalLLM (demo num=3)	56.29 $\pm$ 1.38%	56.42 $\pm$ 1.25%	62.01 $\pm$ 4.13%	63.13 $\pm$ 3.89%
PersonalLLM (demo num=5)	56.46 $\pm$ 1.34%	56.21 $\pm$ 1.24%	62.43 $\pm$ 4.97%	64.13 $\pm$ 3.92%
<i>General LLMs (Persona summarization)</i>				
Qwen3-235B	57.09 $\pm$ 1.21%	58.40 $\pm$ 1.12%	60.97 $\pm$ 4.13%	61.10 $\pm$ 4.03%
GPT-OSS-120B	56.98 $\pm$ 1.38%	57.79 $\pm$ 1.27%	63.04 $\pm$ 4.15%	64.32 $\pm$ 3.93%
GPT-4o-mini	57.48 $\pm$ 1.28%	58.21 $\pm$ 1.25%	61.39 $\pm$ 4.10%	60.37 $\pm$ 4.03%
LLAMA-3.1-8B	56.82 $\pm$ 1.64%	58.39 $\pm$ 1.24%	61.27 $\pm$ 4.15%	63.99 $\pm$ 4.10%
LLAMA-3.3-70B	56.78 $\pm$ 1.25%	58.18 $\pm$ 1.29%	63.89 $\pm$ 3.97%	62.68 $\pm$ 4.00%
<i>Ours</i>				
EpiPersona-A	59.38 $\pm$ 1.25%	59.15 $\pm$ 1.30%	66.07 $\pm$ 3.98%	65.01 $\pm$ 3.90%

Generative setting. As shown in Table 1, we compare EpiPersona-A with non-parametric preference modeling approaches. Against the RAG-based *PersonalLLM*, our method improves by 2.92% on *Prism* and 3.64% on *Arena*. While RAG-based approaches capture localized or recent information, they overlook globally stable preferences. Conversely, methods like *SynthesizeMe* and *PI* model static global personas but struggle to adapt to episode-dependent preference variations. EpiPersona-A addresses both limitations by conditioning episode-specific preference predictions on a latent persona, with parametric history modeling enabling better generalization and robustness.

Discriminative setting. As shown in Table 2, we compare EpiPersona-B (ours) with parametric preference modeling methods on pluralistic reward learning. EpiPersona-B consistently achieves higher or comparable performance than other baselines, demonstrating the effectiveness of disentangling the persona from episode-related information via persona projection. Specifically, while *VPL* and *MRM* model latent individual information, their performance is sensitive to context and model size. In contrast, EpiPersona-B shows robust preference prediction, achieving the highest scores with both Llama-3.2-3B and Llama-3.1-8B.

Ablation of EpiPersona. To validate the contribution of each component in EpiPersona, we conduct ablation studies on two datasets, focusing on a more complex generative setting. Table 3 presents

Table 2: Comparison of the discriminative task. Values are percentages with standard deviations (95% CI).

	Prism		Arena	
	Llama-3.2-3B	Llama-3.1-8B	Llama-3.2-3B	Llama-3.1-8B
GPO	55.26 $\pm$ 1.25%	56.48 $\pm$ 1.24%	53.87 $\pm$ 4.14%	56.69 $\pm$ 4.05%
VPL	58.26 $\pm$ 1.75%	58.23 $\pm$ 1.75%	56.69 $\pm$ 5.81%	54.93 $\pm$ 5.71%
BT	56.58 $\pm$ 1.47%	57.11 $\pm$ 1.55%	51.94 $\pm$ 5.31%	54.06 $\pm$ 5.42%
LoRe	53.57 $\pm$ 1.81%	51.70 $\pm$ 1.78%	53.17 $\pm$ 5.99%	52.48 $\pm$ 5.63%
MRM	55.65 $\pm$ 1.75%	57.14 $\pm$ 1.71%	56.79 $\pm$ 5.63%	56.68 $\pm$ 5.63%
Ours	60.23 $\pm$ 1.76%	58.54 $\pm$ 1.76%	59.57 $\pm$ 5.67%	57.45 $\pm$ 5.12%

the results comparing two main ablation variants. Additionally, a detailed ablation study on the effect of the diversity loss on codebook utilization and training stability is provided in Appendix A.8.

Ablation variant 1: Text-based preference modeling (w/o modeling preference on latent persona). *Global persona prediction* summarizes the text-based persona from the historical information. *Episode-specific preference prediction* couples the text-based persona with the episode to predict the episode-specific preference (see prompt in Appendix A.11.4).

Ablation variant 2: Parametric preference modeling represents individual preferences with a latent persona Z_u . To assess the contribution of specific components, we conducted ablation studies by removing the VQ-based mapping (w/o *VQ-based mapping*) and parametric abductive reasoning (w/o *param. abductive reasoning*) modules.

Component-wise analysis. For **ablation variant 1 (Text-based preference modeling)**, using the sta-

ble global persona achieves moderate performance. Adding episode-specific preference prediction with different LLMs brings minor improvements, suggesting that *text-based* approaches alone have limited ability to capture episode-level preferences.

For **ablation variant 2 (Parametric preference modeling)**, representing individual preferences with a latent persona Z_u boosts performance. Ablation studies show that removing the VQ-based mapping or the parametric abductive reasoning module reduces accuracy, confirming that each component contributes to overall performance.

Table 3: Ablation study on EpiPersona-A.

	Prism	Arena
Ablation Variant 1: Text-based preference modeling		
<i>Global persona prediction</i>		
only stable persona gpt-oss-120b	57.54 $\pm$ 1.15%	64.05 $\pm$ 4.13%
<i>Episode-specific preference prediction</i>		
with Qwen3-235B	57.09 $\pm$ 1.21%	60.97 $\pm$ 4.13%
with distilled Llama-3.1-8B	56.98 $\pm$ 1.38%	62.52 $\pm$ 4.08%
Ablation Variant 2: Parameter-based preference modeling		
w/o VQ-based mapping	57.56 $\pm$ 1.13%	62.66 $\pm$ 4.01%
w/o param. abductive reasoning	57.99 $\pm$ 1.13%	64.36 $\pm$ 4.01%
EpiPersona (Ours)	59.38 $\pm$ 1.25%	66.07 $\pm$ 3.98%

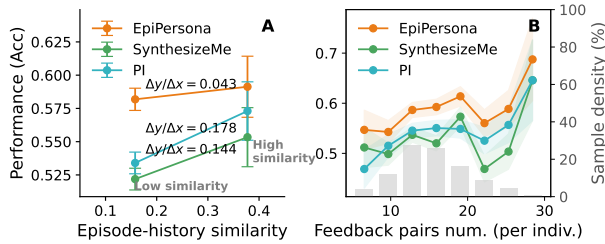


Figure 3: Evaluation of EpiPersona under different scenarios. (A) *Episode similarity*: performance under high vs. low episode similarity. (B) *Number of preference feedback instances*: model performance across users with varying amounts of historical feedback.

5.2.2 Analysis

To understand episode generalization, we analyze it from three perspectives: (1) *episode similarity*, (2) the *number of individual preference feedback instances*, i.e., reflecting the data distribution, and (3) insights from the codebook.

Analysis of episode similarity effects. *Episode similarity* refers to the similarity between an individual’s current episode and their historical preference episodes. Low similarity indicates a large episodic gap and shift (i.e., an *episode-shift scenario*), implying that historical episodes are less in-

formative to the current sample. Episode similarity is computed using *Qwen3-Embedding*. As shown in Figure 3(A), the results reveal two interesting findings. **Finding 1:** EpiPersona, *SynthesizeMe*, and *PI* all perform better under high episode similarity, suggesting that similar contexts help models more accurately infer individual preferences. This also indicates that samples in the *episode-shift scenario* are relatively hard. **Finding 2:** Under low episode similarity, our method shows a smaller performance drop, demonstrating its ability to *handle unseen episodes* and improve generalization during inference: our model drops only 0.043, compared to 0.178 and 0.144 for the other methods. In the low-episode similarity scenario, our method outperforms the SOTA by $\sim 5\%$ (see Figure 3(A)).

Analysis of instance number. As shown in Figure 3(B), across varying ranges of user preference counts, our method maintains consistently competitive performance compared to the baselines. The benefits of EpiPersona are especially apparent when user preferences are sparse.

These two analyses collectively validate EpiPersona’s strong generalization capability: it maintains robust performance under both episodic distribution shift and data scarcity, outperforming SOTA baselines in both challenging settings.

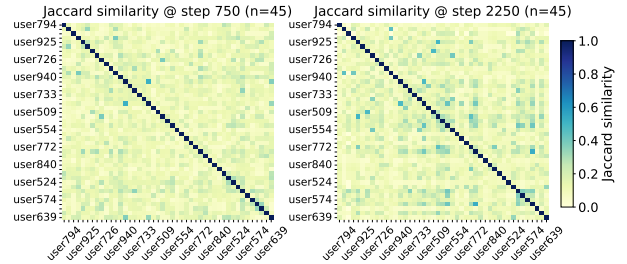


Figure 4: Pairwise Jaccard similarity of codebook index sets among 45 randomly sampled users.

Analysis on codebook *Codebook evolution.* Figure 4 visualizes the pairwise Jaccard similarity of codebook index sets among 45 randomly sampled users across training steps. It shows that: • *Step 750:* Off-diagonal entries are predominantly near zero, indicating largely distinct index sets across users. • *Step 2250:* Partial clusters of similar users emerge, suggesting the codebook begins to organize users by shared preferences.

It indicates that the learned codebook is capable of partitioning the user population into preference

clusters, and the model is capturing individualized preference structure.

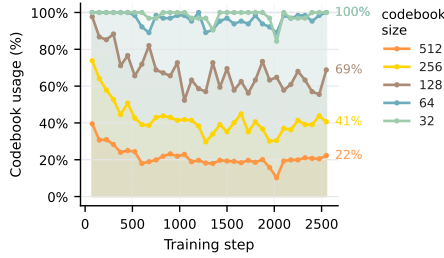


Figure 5: Codebook utilization rate for different codebook sizes (*window size=30*).

Codebook usage analysis. Figure 5 reports codebook utilization across training steps for sizes from 32 to 512. We observe two patterns: (1) Small codebooks ($K \leq 64$) maintain near 100% utilization throughout training, while larger codebooks stabilize at lower rates. This reflects the codebook’s adaptive use of capacity, matching the data’s intrinsic preference diversity. (2) To further illustrate this, the result shows that the absolute number of active entries converges to a similar range across all codebook sizes (see Appendix A.7).

6 Conclusion

We introduced EpiPersona, a novel method for pluralistic alignment that jointly models stable personas and dynamic episodes. By projecting individual preference feedback into a low-dimensional, shared persona space via parameterized abductive reasoning and vector quantization, our method effectively disentangles enduring personal traits from situational variations.

Limitations

While EpiPersona demonstrates promising results in open-domain dialogue personalization, several limitations warrant acknowledgment. *First*, our experiments are conducted on open-domain datasets. Future studies could further reflect the complexity of specialized domains. Extending EpiPersona to high-stakes and specialized settings such as medical consultation, legal advisory, and educational contexts would be an important step toward broader impact. *Second*, the datasets employed in this work are predominantly English-based, which restricts our ability to assess the framework’s robustness across diverse linguistic contexts. Future work

should explore multilingual benchmarks to address this gap.

Ethical Considerations

As with any personalization technology, our approach raises inherent **privacy considerations**. Learned persona representations may unintentionally encode sensitive attributes, such as *political beliefs, health conditions, or socioeconomic status*, or enable inference of private information from behavioral traces, even when no explicit personal data is provided. The episodic memory mechanism, while designed for preference tracking, could in principle reconstruct fine-grained behavioral profiles that users did not intend to disclose.

Although we mitigate this risk by (a) relying solely on publicly available, peer-reviewed benchmarks (Prism and Arena) that have undergone ethics review with established privacy protections, and (b) strictly adhering to dataset usage agreements, the potential for such issues in *real-world deployment* remains a concern. Future work should explore privacy-preserving personalization techniques, such as *differential privacy* over persona embeddings or *federated learning* schemes that keep user-specific parameters on-device.

Finally, while the proposed framework advances preference modeling and reward learning for pluralistic alignment, its application should be approached with caution to prevent **misuse**. Concretely, a deployed system could (a) exploit fine-grained persona representations to craft persuasive messages tailored to individual psychological vulnerabilities, or (b) repurpose preference signals as a covert profiling mechanism for *surveillance*. To guard against such scenarios, we will release our method under a *restrictive license* that explicitly prohibits commercial exploitation for targeted manipulation or user monitoring. Beyond our own release practices, we call on the community to develop complementary safeguards: (a) fairness evaluation protocols that audit personalization outcomes across demographic groups and episodic contexts, (b) red-teaming methodologies specific to persona-conditioned generation, and (c) institutional best practices for responsible deployment of pluralistic alignment systems.

References

Amanda Askill, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas

- Joseph, Ben Mann, Nova DasSarma, and 1 others. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, and 12 others. 2022. [Training a helpful and harmless assistant with reinforcement learning from human feedback](#). *Preprint*, arXiv:2204.05862.
- Nishant Balepur, Vishakh Padmakumar, Fumeng Yang, Shi Feng, Rachel Rudinger, and Jordan Lee Boyd-Graber. 2025. [Whose boat does it float? improving personalization in preference tuning via inferred user personas](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3371–3393, Vienna, Austria. Association for Computational Linguistics.
- Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.
- Avinandan Bose, Zhihan Xiong, Yuejie Chi, Simon Shaolei Du, Lin Xiao, and Maryam Fazel. 2025. Lore: Personalizing llms via low-rank reward modeling. *arXiv preprint arXiv:2504.14439*.
- Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.
- Hongru Cai, Yongqi Li, Tiezheng Yu, Fengbin Zhu, Wenjie Wang, Fuli Feng, and Wenjie Li. 2026. [One adapts to any: Meta reward modeling for personalized llm alignment](#). In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '26*, pages 110–121, New York, NY, USA. Association for Computing Machinery.
- Daiwei Chen, Yi Chen, Aniket Rege, Zhi Wang, and Ramya Korlakai Vinayak. 2025. [PAL: Sample-efficient personalized reward modeling for pluralistic alignment](#). In *The Thirteenth International Conference on Learning Representations*.
- Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.
- Logan Cross, Violet Xiang, Agam Bhatia, Daniel Yamins, and Nick Haber. 2025. Hypothetical minds: Scaffolding theory of mind for multi-agent tasks with large language models. In *International Conference on Learning Representations*, volume 2025, pages 6507–6546.
- Bangde Du, Ziyi Ye, Zhijing Wu, Jankowska Monika, Shuqi Zhu, Qingyao Ai, Yujia Zhou, and Yiqun Liu. 2025. Valuesim: Generating backstories to model individual value systems. *arXiv preprint arXiv:2505.23827*.
- Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, and 1 others. 2026. A survey on llm-as-a-judge. *The Innovation*, 7(6).
- Jian Guan, Junfei Wu, Jia-Nan Li, Chuanqi Cheng, and Wei Wu. 2025. A survey on personalized alignment—the missing piece for large language models in real-world applications. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5313–5333.
- Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. 2023. [Personalized soups: Personalized large language model alignment via post-hoc parameter merging](#). *Preprint*, arXiv:2310.11564.
- Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. 2025. [A survey of reinforcement learning from human feedback](#). *Trans. Mach. Learn. Res.*, 2025.
- Hannah R Kirk, Alexander Whitefield, Paul Röttger, Andrew Bean, Katerina Margatina, Juan Ciro, Rafael Mosquera, Max Bartolo, Adina Williams, He He, and 1 others. 2024a. The prism alignment dataset: What participatory, representative and individualised human feedback reveals about the subjective and multicultural alignment of large language models. *Advances in Neural Information Processing Systems*, 37:105236–105344.
- Hannah Rose Kirk, Bertie Vidgen, Paul Röttger, and Scott A Hale. 2024b. The benefits, risks and bounds of personalizing the alignment of large language models to individuals. *Nature Machine Intelligence*, 6(4):383–392.
- Jia-Nan Li, Jian Guan, Songhao Wu, Wei Wu, and Rui Yan. 2026. From 1,000,000 users to every user: Scaling up personalized preference for user-level alignment. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30144–30168.
- Minhyeon Oh, Seungjoon Lee, and Jungseul Ok. 2025. [Comparison-based active preference learning for multi-dimensional personalization](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33145–33166, Vienna, Austria. Association for Computational Linguistics.
- Sriyash Poddar, Yanming Wan, Hamish Ivison, Abhishek Gupta, and Natasha Jaques. 2024. Personalizing reinforcement learning from human feedback with variational preference learning. *Advances in Neural Information Processing Systems*, 37:52516–52544.

- Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, and 1 others. 2018. Improving language understanding by generative pre-training.
- Matthew Riemer, Zahra Ashktorab, Djallel Bouneffouf, Payel Das, Miao Liu, Justin D. Weisz, and Murray Campbell. 2025. [Position: Theory of mind benchmarks are broken for large language models](#). In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025 - Position Paper Track*. OpenReview.net.
- Michael J. Ryan, Omar Shaikh, Aditri Bhagirath, Daniel Frees, William Held, and Diyi Yang. 2025. [SynthesizeMe! inducing persona-guided prompts for personalized reward models in LLMs](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8045–8078, Vienna, Austria. Association for Computational Linguistics.
- Aaron Van Den Oord, Oriol Vinyals, and 1 others. 2017. Neural discrete representation learning. *Advances in neural information processing systems*, 30.
- Chengxing Xie, Canyu Chen, Feiran Jia, Ziyu Ye, Shiyang Lai, Kai Shu, Jindong Gu, Adel Bibi, Ziniu Hu, David Jurgens, James Evans, Philip Torr, Bernard Ghanem, and Guohao Li. 2024. [Can large language model agents simulate human trust behavior?](#) In *Advances in Neural Information Processing Systems*, volume 37, pages 15674–15729. Curran Associates, Inc.
- Yiyang Xu, Jinghao Zhang, Alireza Salemi, Xinting Hu, Wenjie Wang, Fuli Feng, Hamed Zamani, Xiangnan He, and Tat-Seng Chua. 2025. [Personalized generation in large model era: A survey](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24607–24649, Vienna, Austria. Association for Computational Linguistics.
- Weikang Yuan, Kaisong Song, Zhuoren Jiang, Junjie Cao, Yujie Zhang, Jun Lin, Kun Kuang, Ji Zhang, and Xiaozhong Liu. 2026. [LeCoDe: A benchmark dataset for interactive legal consultation dialogue evaluation](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36019–36048, San Diego, California, United States. Association for Computational Linguistics.
- Yujie Zhang, Weikang Yuan, and Zhuoren Jiang. 2025. [Bridging intuitive associations and deliberate recall: Empowering LLM personal assistant with graph-structured long-term memory](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 17533–17547, Vienna, Austria. Association for Computational Linguistics.
- Zhehao Zhang, Ryan A Rossi, Branislav Kveton, Yijia Shao, Diyi Yang, Hamed Zamani, Franck Dernoncourt, Joe Barrow, Tong Yu, Sungchul Kim, and 1 others. 2024. Personalization of large language models: A survey. *arXiv preprint arXiv:2411.00027*.
- Chuang Zhao, Hongke Zhao, Ming HE, Jian Zhang, and Jianping Fan. 2023. [Cross-domain recommendation via user interest alignment](#). In *Proceedings of the ACM Web Conference 2023, WWW '23*, page 887–896, New York, NY, USA. Association for Computing Machinery.
- Siyan Zhao, John Dang, and Aditya Grover. 2024. [Group preference optimization: Few-shot alignment of large language models](#). In *International Conference on Learning Representations*, volume 2024, pages 57965–57987.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. 2023. [Judging llm-as-a-judge with mt-bench and chatbot arena](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.
- Minjun Zhu, Yixuan Weng, Linyi Yang, and Yue Zhang. 2025. [Personality alignment of large language models](#). In *International Conference on Learning Representations*, volume 2025, pages 14206–14255.
- Yongxin Zhu, Bocheng Li, Yifei Xin, and Linli Xu. 2024. [Addressing representation collapse in vector quantized models with one linear layer](#).
- Thomas Zollo, Andrew Siah, Naimeng Ye, Li Li, and Hongseok Namkoong. 2025. [Personallm: Tailoring llms to individual preferences](#). In *International Conference on Learning Representations*, volume 2025, pages 66949–66971.

A Appendix

A.1 Task design

These two settings define the *general evaluation paradigm* rather than being tied to any specific model design.

Task 1: Generative setting. In this paradigm, an LLM is prompted to infer a natural-language description of an individual user’s preference based on historical interactions X_u , optionally conditioned on the current episode e . This formulation is applicable to a broad class of *non-parametric* baselines that rely on in-context prompting.

$$\text{Pref} = G(X_u, e) \text{ or } \text{Pref} = G(X_u) \quad (\text{A.1})$$

The generated preference Pref is expressed in natural language and summarizes the individual’s persona or preference. Subsequently, preference and a set of candidate responses $\{\tau_e^{(1)}, \tau_e^{(2)}\}$ are provided to a large language model acting as a

judge. $\hat{y}_e \in \{0, 1\}$ indicates whether the user selects the response. The judge predicts which response the user is more likely to select:

$$\hat{y}_e = LLM(\text{Pref}, \tau_e^{(1)}, \tau_e^{(2)}). \quad (\text{A.2})$$

This task is mainly designed for **EpiPersona-A** and non-parametric preference modeling, following the LLM-as-a-judge evaluation paradigm.

Task 2: Discriminative setting. Let e denote the current episode. The user’s historical preference information is X_u . Given a candidate response τ_e under episode e , the goal is to predict the probability that the user would select this response:

$$\hat{y}_e = \arg \max_{\tau_e \in \{\tau_e^{(1)}, \tau_e^{(2)}\}} \mathbb{P}(y_e = 1 \mid h(X_u), \tau_e). \quad (\text{A.3})$$

where h denotes the encoding of user context X_u . Model performance is evaluated using accuracy (Acc).

This task is primarily designed for **EpiPersona-B** and parametric preference modeling.

These two task settings correspond to different real-world application scenarios. The generative setting is better suited for plug-and-play settings, where the generated textual preference descriptions can be directly injected into the prompt of the LLM. The discriminative setting, on the other hand, is more appropriate for scenarios that require end-to-end training of personalized reward models. The two variants operate in fundamentally different output spaces, thereby providing a more comprehensive evaluation of the EpiPersona framework as a whole.

A.2 Dataset construction

A fundamental challenge in personalized preference research is that collecting real-world preference data with individual identity annotations is exceedingly difficult. Arena and Prism are among the few publicly available datasets that simultaneously satisfy the following criteria: (1) Each dataset contains multiple preference records per user across different episodes, enabling genuine longitudinal modeling of individual preferences. (2) Both datasets exhibit sufficient inter-individual heterogeneity to support meaningful pluralistic alignment research, capturing the diversity of real-world user populations.

We follow the existing study (Ryan et al., 2025) in adopting the dataset split (train, validation, and

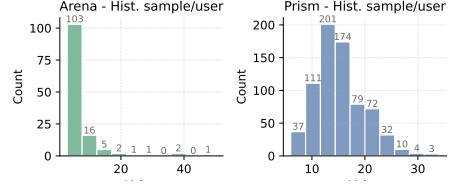


Figure A.1: The distribution of the history samples per user.

test sets) as well as the division of each individual’s historical and current data to ensure more consistent and continuous experimental validation. These two datasets compile the high-quality and challenging individual data to serve as a benchmark for personalized alignment. The distribution of historical sample numbers per individual is shown in Figure A.1. The statistics of historical/current pairs for both datasets are summarized in Table A.1.

Table A.1: Statistics of user numbers and historical/current pairs for Arena and Prism datasets.

	Arena	Prism
	Hist/cur pairs	Hist/cur pairs
Train	202/103	4192/2177
Validation	155/80	973/514
Test	514/ 284	5816/ 3033

A.3 Implementation details

The model is fine-tuned with LoRA (rank $r = 8$, $\alpha = 32$, dropout = 0.1). The vector quantization module uses a codebook size of 256 for Prism and 64 for Arena. The VQ commitment loss weight is set to 1.0, the diversity loss weight to 0.1, and the diversity softmax temperature to 0.5. User preference history is encoded with a sliding window of size 30. Preference descriptions s are generated by Qwen3-235B-A22B-Instruct based on users’ real feedback. We use a JSON dictionary format with fields such as *persona*, *value*, *preferred style*, and *intent* (see prompt in Appendix A.11.1).

Training is conducted on 2 NVIDIA GPUs (H20) with a per-device batch size of 2 and gradient accumulation steps of 2, yielding an effective global batch size of 8. We use the AdamW optimizer with learning rate 5×10^{-6} , weight decay 0.01, and a linear warmup over 100 steps followed by constant scheduling. Gradient clipping is applied at 1.0. Distributed training is managed via Deep-

Speed ZeRO Stage 2 with CPU optimizer offloading. Due to the limited size of the Arena dataset, we first pre-trained the transformer-based model on the Prism dataset (for EpiPersona-A) to stabilize training. We initialize the codebook via K-means rather than random initialization, with 20 iterations at initialization. We reset a codebook entry when its EMA usage frequency falls below 1.

A.4 Comparison of preference modeling methods across core capabilities

We formalize the core capabilities of preference modeling as: human-observable and interpretable (Human-O/I), machine-understandable (Machine-U), compression capability (Compact-C), and global preference abstraction capability (Global-A).

Text-based summarization and RAG offer interpretability but provide weaker compression, learnability, and abstraction, which may undesirably focus on users’ fragmented information. *Parameterized preference* modeling excels in learnability and abstraction but is less observable and plug-and-play. *Predefined dimensions* compress preferences into dense vectors, aiding learning and interpretability, yet rely heavily on manual design, limiting flexibility.

Our approach proposes a learnable personalized preference attribution (EpiPersona-A) that capitalizes on the strengths of these four capabilities. Specifically, *Machine-understandable*: EpiPersona-A parameterizes users’ static personas employing parameterized abductive reasoning and VQ-based mapping; *Compression capability*: Parameterized static personas serve as a highly compressed representation of user preferences, preventing loss of persona information via text summarization; *Global preference abstraction capability*: EpiPersona-A provides a macro-level, abstract characterization of user personalities, instead of isolated or fragmented local information; *Human-observable and interpretable*: Dynamic preferences are predicted in textual form, making them plug-and-play compatible with any LLM. See Table A.2 for the comparison of these models.

A.5 Figure of model details

Figure A.2 illustrates the detailed architecture of our model, including the parameterized abductive reasoning module and the VQ-based mapping mechanism.

A.6 The detail of VQ-based mapping

Exponential moving average update. We adopt an exponential moving average scheme to stabilize codebook learning. For each persona codeword $z_p^{(m)}$, we maintain an EMA count N_m and an EMA embedding E_m . At each update step, we first accumulate statistics over the current batch:

$$N_m \leftarrow \gamma N_m + (1 - \gamma) \sum_{w=1}^n \mathbb{I}[m^{*(w)} = m], \quad (\text{A.4})$$

$$E_m \leftarrow \gamma E_m + (1 - \gamma) \sum_{w=1}^n \mathbb{I}[m^{*(w)} = m] \times \mathbf{w}^{(w)} \quad (\text{A.5})$$

where $\gamma \in (0, 1)$ is the EMA decay rate and $\mathbb{I}[\cdot]$ denotes the indicator function. The persona codeword is then updated as:

$$z_p^{(m)} \leftarrow \frac{E_m}{N_m}. \quad (\text{A.6})$$

A.7 Codebook usage analysis

As shown in Figure A.3, the codebook usage dynamics reveal several consistent patterns across all codebook sizes:

1. **Codebook utilization dynamics.** At the very beginning of training, codebook usage starts high due to the K -means initialization. After the initialization phase, the actual number of utilized codewords gradually stabilizes as training progresses.
2. **Size-dependent utilization ceiling.** Larger codebooks (size $\in \{256, 512\}$) converge to high absolute utilization levels, whereas smaller codebooks (size $\in \{32, 64\}$) saturate earlier with nearly all entries occupied. This indicates that an excessively small codebook reaches its utilization ceiling prematurely, thereby limiting the model’s expressive power.
3. **Dataset-driven convergence.** As training progresses, codebooks of varying sizes converge to their respective stable usage levels. This implies that the effective number of required codebook entries is closely tied to the intrinsic preference diversity of the dataset, rather than being an artifact of model capacity alone.

Table A.2: Comparing various preference modeling methods in terms of key attributes. Predef.: Predefined dimensions; Text-based sum.: text-based preference summarization; Parameterized.: Parameterized modeling; EpiPersona-A: Our proposed method.

Method	Human-O/I	Machine-U	Compact-C	Global-A
Text-based sum.	✓	-	-	-
Parameterized.	-	✓	✓	✓
Predef.	✓	✓	✓	-
RAG	✓	-	-	-
EpiPersona-A	✓	✓	✓	✓

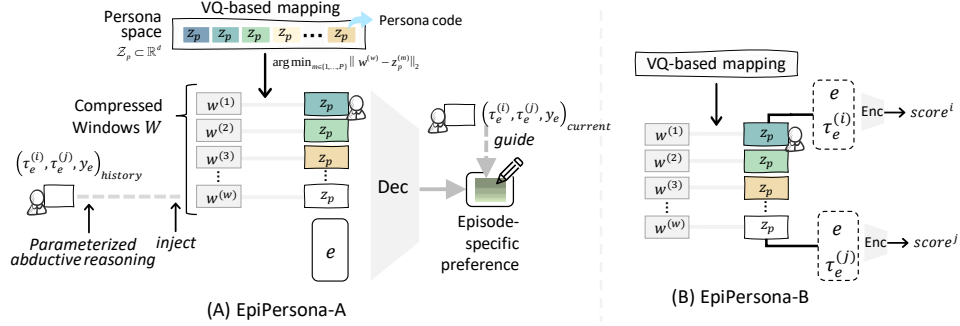


Figure A.2: Model architecture details (parameterized abductive reasoning and VQ-based mapping).

A.8 Effect of diversity loss on codebook utilization

To validate the effectiveness of the diversity loss, we conduct an ablation study comparing codebook usage with and without it under two codebook sizes (128 and 256), as shown in Figure A.4. To quantify training stability, we measure oscillation as the rolling standard deviation of codebook utilization over a window of 4 consecutive training steps.

Reduced Oscillation. Without the diversity loss, codebook usage exhibits severe oscillation throughout training. The rolling standard deviation reaches up to 30% for codebook size 128 and approximately 20% for codebook size 256. With the diversity loss, oscillation is suppressed to below 10% across both settings, indicating substantially more stable training dynamics.

Alleviated Codebook Collapse. Without the diversity loss, codebook utilization frequently drops below 25% for size 128 and below 10% for size 256, reflecting recurring collapse. With the diversity loss, utilization is maintained consistently above 50% and 30% for the two sizes, respectively. These results confirm that the diversity loss is essential for stable and efficient codebook utilization.

A.9 Hyperparameter experiment

To investigate the effect of key hyperparameters, we conduct ablation studies on codebook size and window size, as shown in Figure A.5.

For codebook size, accuracy remains stable across 32, 64, 128, 256, 512. For window size, performance follows an inverted U-shaped trend over 10, 15, 20, 25, 30, 35, 40. A window that is too small fails to capture sufficient context, while an overly large window introduces irrelevant noise. In both cases, accuracy variance remains within 0.5%, indicating that our model is relatively robust to these hyperparameters.

A.10 Inference algorithm

We summarize the inference procedure for EpiPersona. Given user preference feedback and an episode, the algorithms first encode feedback using parameterized abductive reasoning and convert it to latent persona representations via VQ-based mapping. We then generate either episode-specific preferences (EpiPersona-A) or reward scores for candidate responses (EpiPersona-B) (see algorithm 1 and 2).

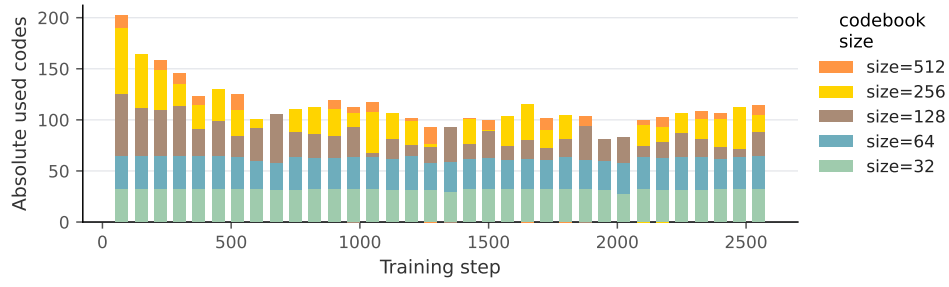


Figure A.3: Codebook utilization across different codebook sizes during training (*window size=30*).

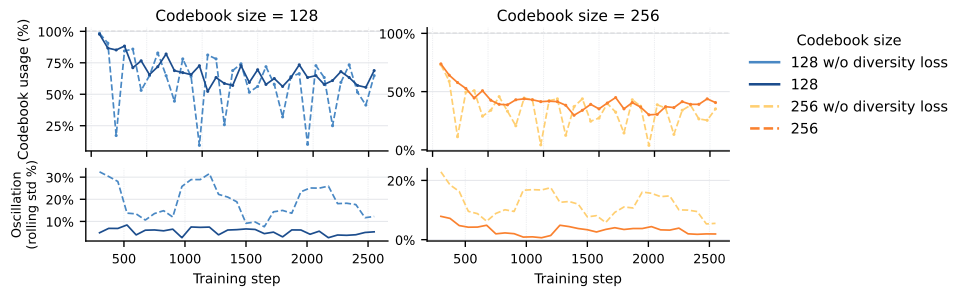


Figure A.4: Ablation study on the diversity loss. Top: codebook utilization rate over training steps. Bottom: oscillation measured by rolling standard deviation. The diversity loss reduces both codebook collapse and utilization instability across codebook sizes 128 and 256.

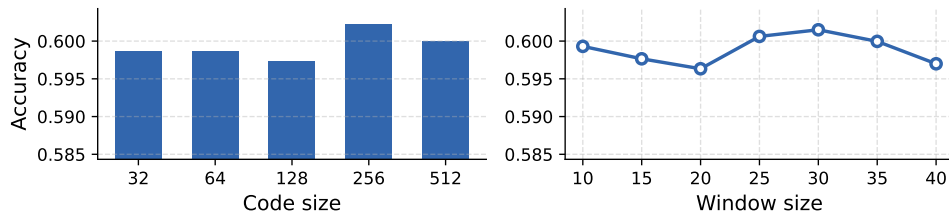


Figure A.5: Hyperparameter experiment on *Prism*.

Algorithm 1 EpiPersona-A: Persona-episode coupled preference generation

Input: Feedback set X_u for individual u ; target episode $e = (\tau_e^{(i)}, \tau_e^{(j)})$

Output: Personalized preference description Pref_e

// Step 1: Parameterized abductive reasoning (Section 4.3)

for each feedback $x = (\tau_e^{(i)}, \tau_e^{(j)}, y_e) \in X_u$ **do**
 Infer estimated persona: $r \leftarrow \lambda(\mathcal{P}, \tau_e^{(i)}, \tau_e^{(j)}, y_e)$

end for

Collect all estimated personas: $R_u = \{r \mid x \in X_u\}$

// Step 2: VQ-based mapping (Section 4.4)

Project into shared space: $\mathbf{w}^{(w)} = f_\theta(r)$ for each $r \in R_u$

Quantize each window via nearest-neighbor lookup:

$$m^{*(w)} = \arg \min_{m \in \{1, \dots, P\}} \|\mathbf{w}^{(w)} - z_p^{(m)}\|_2$$

Obtain latent persona set: $Z_u = (z_p^{(m^{*(w)})})_{w=1}^n$
// Step 3: Persona-episode coupling (Section 4.2)

Generate preference: $\text{Pref}_e = g_\psi(Z_u, e)$

Format Pref_e as JSON with keys: persona, value, identification, intent, style

return Pref_e

Algorithm 2 EpiPersona-B: Episode-aware persona-based reward modeling

Input: Feedback set X_u for individual u ; episode e ; candidate responses $\tau_e^{(i)}, \tau_e^{(j)}$

Output: Rewards $R_\theta(Z_u, \tau_e^{(i)})$ and $R_\theta(Z_u, \tau_e^{(j)})$

// Step 1: Parameterized abductive reasoning (Section 4.3)

for each feedback $x = (\tau_e^{(i)}, \tau_e^{(j)}, y_e) \in X_u$ **do**
 Infer estimated persona: $r \leftarrow \lambda(\mathcal{P}, \tau_e^{(i)}, \tau_e^{(j)}, y_e)$

end for

Collect all estimated personas: $R_u = \{r \mid x \in X_u\}$

// Step 2: VQ-based mapping (Section 4.4)

Project into shared space: $\mathbf{w}^{(w)} = f_\theta(r)$ for each $r \in R_u$

Quantize each window via nearest-neighbor lookup:

$$m^{*(w)} = \arg \min_{m \in \{1, \dots, P\}} \|\mathbf{w}^{(w)} - z_p^{(m)}\|_2$$

Obtain latent persona set: $Z_u = (z_p^{(m^{*(w)})})_{w=1}^n$

// Step 3: Persona-episode coupling & reward computation (Section 4.2)

for each candidate response $\tau_e^{(k)} \in \{\tau_e^{(i)}, \tau_e^{(j)}\}$ **do**

 Compute episode-persona coupled representation: $h_\theta(Z_u, \tau_e^{(k)})$

 Compute reward: $R_\theta(Z_u, \tau_e^{(k)}) = \mathbf{W}^\top \phi(h_\theta(Z_u, \tau_e^{(k)}))$

end for

Predict preference via Bradley-Terry comparison:

$$\hat{y}_e = \mathbf{1} \left[R_\theta(Z_u, \tau_e^{(i)}) > R_\theta(Z_u, \tau_e^{(j)}) \right]$$

return $R_\theta(Z_u, \tau_e^{(i)}), R_\theta(Z_u, \tau_e^{(j)}), \hat{y}_e$

A.11 Prompt

This section lists the prompts used for modeling user beliefs and preferences, including:

1. Prompt for fine-grained preference description extraction;
2. LLM-as-a-Judge Prompt for EpiPersona-A;
3. LLM-as-a-Judge Prompt with Retrieval Augmentation;
4. Text-based Episode-specific Preference Prediction Prompt (*used in ablation test*);
5. Parameterized Abductive Reasoning Prompt ($\mathcal{P}$);
6. EpiPersona-A Prompt for episode-specific context coupled with user representation Z_u .

3. The output must be strictly formatted as JSON and must not contain comments.

---Output Format---

```
{
  "stable_preferences": {
    "persona": "<persona>",
    "value": "<value>",
    "identification": "<identification>"
  },
  "dynamic_preferences": {
    "style": "<style>",
    "intent": "<intent>"
  }
}
```

1. Prompt for fine-grained preference description extraction (guidance for EpiPersona-A)

Given the dialogue context below and two possible responses—one selected by the user and one not:

Dialogue context: {context}

Chosen answer: {chosen}

Rejected answer: {rejected}

---Task---

Analyze the user's selection and extract explicitly defined preference elements.

---Definitions---

The extracted preferences should be divided into the following categories:

1. Stable preferences (long-term, consistent traits):
persona: the user's personality or character traits.
value: the user's values, opinions, or attitudes.
identification: the user's social identity, role, or profession if inferable.

2. Dynamic preferences (context-dependent and situational):
style: the user's preferred expression style, tone, or format.
intent: the user's goal or intention in choosing the response.

---Rules---

1. If a preference element cannot be inferred, return None.
2. Do not guess or invent information beyond what can be reasonably inferred from the context.

2. LLM-as-a-Judge Prompt (EpiPersona-A)

Your input fields are:

1. conversation (str): The conversation context leading up to the completions.
2. first_completion (str): The first of the two candidate responses.
3. second_completion (str): The second of the two candidate responses.

Your output fields are:

1. reasoning (str): A brief explanation of the judgment.
2. preference (str): The response more likely to be preferred by the user.
The value must be one of: "First" or "Second".

All interactions follow the structure below.

```
[[ ## conversation ## ]]  
{conversation}
```

```
[[ ## first_completion ## ]]  
{first_completion}
```

```
[[ ## second_completion ## ]]  
{second_completion}
```

The output must be a valid JSON object with the following structure:

```
{  
  "reasoning": "<reasoning>",  
  "preference": "<First | Second>"  
}
```

Your objective:

Given a conversation and two responses generated by different models, determine which response the human user is more likely to prefer.

The user you are judging for has the following preference profile:

```
===  
{user_preference}  
===
```

Consider the user's persona and preferences carefully, and make a rational, preference-consistent judgment.

3. LLM-as-a-Judge Prompt with Retrieval Augmentation

User History (Retrieved Demonstrations):

```
---  
User's historical selection i (can be used  
as a reference if relevant):
```

```
[[ ## conversation ## ]]  
{demo_conversation_i}
```

```
[[ ## first_completion ## ]]  
{demo_first_completion_i}
```

```
[[ ## second_completion ## ]]  
{demo_second_completion_i}
```

User's choice: "First" or "Second"

(The above block may repeat for multiple retrieved demonstrations.)

Current Dialogue Context:
{conversation}

Candidate Responses:

```
[[ ## first_completion ## ]]  
{first_completion}
```

```
[[ ## second_completion ## ]]  
{second_completion}
```

Task:

Given the current dialogue context and the user's historical selections, infer which candidate response the user is more likely to choose.

You should support your judgment by reasoning about how the current responses relate to patterns observed in the user's history.

Output Format:

The output must be a valid JSON object and contain no additional text.

```
{  
  "preference": "<First | Second>"  
}
```

4. Prompt for Text-based Preference Modeling (Episode-specific Preference Prediction)

Given the user's historical stable preferences and the current context, infer the Stable preferences and Dynamic preferences that are relevant to the specific episode.

Historical Stable Preferences:
{stable_preferences}

Current Context:
{context}

Definitions:

1. Stable preferences (long-term, consistent traits):
 - persona: the user's personality or character traits.
 - value: the user's values, opinions, or attitudes.
 - identification: the user's social identity, role, or profession (if inferable).
2. Dynamic preferences (context-dependent and situational):
 - style: the user's preferred style, tone, or format of expression.
 - intent: the user's immediate goal or purpose in this context.

Rules:

- Infer only preferences that are reasonably supported by the historical data and the current context.
- If a preference element cannot be inferred, set its value to null.
- Do not introduce unsupported assumptions.
- The output must be valid JSON and contain no extra text.

Output Format:

```
{
  "related_stable_preferences": {
    "persona": "<persona>",
    "value": "<value>",
    "identification": "<identification>"
  },
  "dynamic_preferences": {
    "style": "<style>",
    "intent": "<intent>"
  }
}
```

5. Parameterized Abductive Reasoning ($\mathcal{P}$)

Task:

Given a dialogue context and two alternative responses—one chosen by the user and one rejected—identify the stable preference implied by the user's selection.

Definition of Stable Preference:

A stable preference reflects enduring aspects of the user's persona, values, or identity. Such preferences are transferable across dialogue contexts and are unlikely to change due to short-term or situational factors.

Inputs:

Dialogue Context:
{context}

Chosen Answer:
{chosen}

Rejected Answer:
{rejected}

Inference Objective:

Infer the stable preference that best explains why the chosen answer was preferred over the rejected one. If no stable preference can be reasonably inferred, return an empty string.

Output Format:

The output must be a valid JSON object and contain no additional text.

```
{
  "stable_preferences": "<preference or empty string>"
}
```

6. EpiPersona-A Prompt for Episode-specific Context (Coupled with Z_u)

Given the current context, infer Stable preferences and Dynamic preferences relevant to the specific context.

Context:
{context}

Respond ONLY in the following JSON structure:

```
{
  "related_stable_preferences": {
    "persona": "<persona>",
    "value": "<value>",
    "identification":
      "<identification>"
  },
  "dynamic_preferences": {
    "style": "<style>",
    "intent": "<intent>"
  }
}
```