LEARNING LATENT PREFERENCE DISTRIBUTIONS FOR VAE RECOMMENDERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Variational autoencoder (VAE)-based collaborative filtering (CF) has emerged as a powerful framework for recommendation, efficiently inferring users' latent representations from user-item interactions without maintaining user-specific parameters. However, existing methods typically adopt the standard Gaussian prior to learn latent user representations, limiting their ability to model multifaceted user preferences. Moreover, existing VAE-based CF methods overlook potential collaborations among users, leaving sparse users with limited supervision signals and further limiting the quality of their latent representations. To address these limitations, we first theoretically analyze how Kullback-Leibler (KL) divergence regularization and reconstruction loss in vanilla VAE-based CFs constrain users' posterior distributions, limiting the overall performance of the methods. Building on these theoretical analyses, we propose LPD-VAE, a VAE-based recommendation framework designed to learn Latent Preference Distributions (LPD) and explore latent preference collaborations among users. First, we initialize a set of preference components based on the Gaussian Mixture Model prior and devise mixture-prior KL regularization to force user posterior distribution to approach these components, thereby learning each user's LPD to characterize rich user preferences. Then, based on the learned user LPDs, we reckon latent preference collaborations among users to discover potential positive samples from uninteracted items, alleviating the sparse interaction issue of VAE-based CFs. Finally, we design a preference-alignment loss supplementing to the original ELBO to encourage user latent representations to align with potential positive samples in the latent space, so as to refine user LPDs. Extensive experiments on three benchmark datasets demonstrate consistent gains over competitive SOTAs. Furthermore, experiments across users with varying interaction densities show that LPD-VAE maintains consistent performance gains under different levels of sparsity. The ablation experiment demonstrated that, compared to the interaction distance in the interaction space, the latent preference distance in the latent space is more effective in learning user preferences.

1 INTRODUCTION

Recommender systems play a vital role in modern web services by helping users discover personalized content. Collaborative filtering (CF) has become a fundamental approach in recommender systems, aiming to infer users' latent preferences from historical user-item interactions (He et al., 2020; Wang et al., 2019). Traditional latent factor models, such as matrix factorization (MF), assign an individual latent vector to each user and item, resulting in a parameter space that grows with the number of users and items. Such parameter growth increases computational and memory costs, complicating model optimization (Shenbin et al., 2020; Wang et al., 2022; Zhang et al., 2024). Moreover, sparse user-item interactions provide limited supervision for modeling users' latent preferences, making them challenging to capture the underlying collaborative signals among users (Guo et al., 2017; Yang et al., 2023; Liang et al., 2018). To alleviate these limitations, variational autoencoder (VAE)-based CF methods (Liang et al., 2018; Kim & Suh, 2019; Shenbin et al., 2020) adopt amortized variational inference (Kingma & Welling, 2014; Gershman & Goodman, 2014), which uses parameter-shared neural networks to learn users' latent representations from historical interac-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

tions. Consequently, users' preferences can be efficiently inferred without maintaining user-specific parameters.

Due to its simplicity and tractability, the standard Gaussian prior has been widely adopted in VAE-based CFs, constraining users' latent representations toward a single distribution. However, user-item interactions are inherently discrete and sparse, providing limited supervision for inferring users' preference distributions (Wang et al., 2023). Moreover, users may exhibit highly heterogeneous and multifaceted interests (Li et al., 2019; Cen et al., 2020; Yu et al., 2026), such that their preference distributions can differ substantially across users. Given this sparsity and heterogeneity, imposing the standard Gaussian prior on all users is likely overly restrictive, limiting the model's ability to capture diverse preference structures and potentially resulting in suboptimal latent representations. Beyond the standard Gaussian prior, more expressive priors have been explored in VAE-based representation learning methods mainly in computer vision, including variational mixture of posteriors prior (VampPrior) (Tomczak & Welling, 2018; Stirn & Knowles, 2025) and Gaussian Mixture Model (GMM) priors (Jiang et al., 2016; Dilokthanakul et al., 2016). However, existing expressive priors are mainly designed to increase the flexibility of latent representation distributions, without sufficiently considering the sparse and heterogeneous nature of user-item interactions in VAE-based CF.

Existing VAE-based CFs have primarily focused on improving user latent representation learning from observed user-item interactions. Multi-VAE (Liang et al., 2018) and RecVAE (Shenbin et al., 2020) improve the VAE objective and regularization to better exploit observed interaction signals, while Kim & Suh (2019) introduce VampPrior to enhance the flexibility of latent representations. More recently, PIA (Vuong et al., 2026) employs learnable item anchors to stabilize user representations under random masked input. While these efforts improve the quality of user representations, the reconstruction loss constrains posterior overlap through observed interactions, favoring local collaboration among users with similar interactions. However, since interaction similarity does not necessarily reflect users' underlying preferences, recent CF studies have explored potential collaborations derived from uninteracted items to provide complementary signals, particularly for users with sparse interaction histories (Yu et al., 2026; Wang et al., 2025; Ren et al., 2023). These findings inspire us to explore whether latent spaces of VAE-based CF can be used to discover potential user collaborations, and provide new opportunities for addressing interaction sparsity.

Therefore, two fundamental questions arise: *(1) How to learn latent user preferences to characterize divers user interests for VAE-based CFs? (2) How to discover potential user collaborations of VAE-based CFs for improving latent representations of sparse users?* Our work address these questions by introducing a novel VAE-based recommendation framework, named LPD-VAE, to learn Latent Preference Distributions (LPDs) and explore latent preference collaborations among users to augment sparse interactions. The contributions of this paper are summarized as follows:

- We provide a theoretical analysis of the collaboration mechanism in VAE-based CF, showing that Kullback-Leibler (KL) divergence regularization with a shared prior constrains posterior diversity, while reconstruction loss constrains posterior overlap according to observed interactions. Together, these two aspects constrain users' posterior distributions, limiting the overall performance.

- Motivated by this analysis, we initialize a set of preference components based on the Gaussian Mixture Model (GMM) prior and devise mixture-prior KL regularization to force user posterior distribution to approach these components, thereby learning each user's LPD to characterize rich user preferences. Then, based on the learned user LPDs, we reckon latent preference collaborations among users to discover potential positive samples from uninteracted items, alleviating the sparse interaction issue of VAE-based CFs. Finally, we design a preference-alignment loss supplementing to the original ELBO to encourage user latent representations to align with potential positive samples in the latent space, so as to refine user LPDs.

- Extensive experiments on three benchmark datasets demonstrate consistent improvements over competitive SOTAs. Experiments across users with varying interaction densities show that LPD-VAE maintains performance gains under different levels of sparsity. The ablation experiment demonstrated that, compared to the interaction distance in the interaction space, the latent preference distance in the latent space is more effective in learning user preferences.

2 COLLABORATION MECHANISM IN VAE-BASED CF

Notations. We represent the observed user-item interactions as a binary matrix $\mathbf{X} \in \{0,1\}^{\mathcal{U}\times\mathcal{I}}$, where $\mathcal{U}$ and $\mathcal{I}$ denote the numbers of users and items, respectively. For each user u, the interaction is denoted as $\mathbf{x}_u = [x_{u1}, x_{u2}, \ldots, x_{u\mathcal{I}}]$, which corresponds to the u-th row of $\mathbf{X}$. Each entry $x_{ui} = 1$ indicates an observed interaction between user u and item i, while $x_{ui} = 0$ corresponds to a missing interaction rather than explicit negative feedback. The general VAE-based CF formulation and the complete notation used throughout the paper are provided in Appendix A.

2.1 KL DIVERGENCE REGULARIZATION CONSTRAINS POSTERIOR DIVERSITY

We first examine how naive KL divergence regularization brings the posterior distributions of different users $\{q_\phi(\mathbf{z} \mid \mathbf{x}_u)\}_{u\in\mathcal{U}}$ toward a shared prior $p(\mathbf{z})$ and thereby constrains the distances between their posterior distributions.

Theorem 2.1. *Assume that the prior distribution $p(\mathbf{z})$ satisfies the Bobkov–Götze/Talagrand $T_1(C)$ inequality with constant $C > 0$. Then, for any two users u_1 and u_2, the 1-Wasserstein distance between their posterior distributions $W_1(q_\phi(\mathbf{z}|\mathbf{x}_1), q_\phi(\mathbf{z}|\mathbf{x}_2))$ satisfies the following inequality:*

$$\begin{aligned} &W_1(q_\phi(\mathbf{z}|\mathbf{x}_1), q_\phi(\mathbf{z}|\mathbf{x}_2)) \\ &\leq \sqrt{2C\mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x}_1)\|p(\mathbf{z}))} + \sqrt{2C\mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x}_2)\|p(\mathbf{z}))}. \end{aligned} \tag{1}$$

Theorem 2.1 shows that the 1-Wasserstein distance between the posterior distributions of any two users u_1 and u_2 is bounded by their respective KL divergence regularization to the shared prior $p(\mathbf{z})$. Thus, KL divergence regularization encourages each user posterior distribution $q_\phi(\mathbf{z}|\mathbf{x}_u)$ to remain close to the shared prior $p(\mathbf{z})$, thereby constraining the 1-Wasserstein distance between posterior distributions of different users. Consequently, imposing the same prior on different users constrains the differences between their posterior distributions, limiting the model's ability to capture heterogeneous user preference distributions.

Proof. See Appendix D.1. □

2.2 RECONSTRUCTION LOSS CONSTRAINS POSTERIOR OVERLAP THROUGH INTERACTIONS

We next analyze how the reconstruction loss constrains the overlap between posterior distributions $q_\phi(\mathbf{z} \mid \mathbf{x}_1)$ and $q_\phi(\mathbf{z} \mid \mathbf{x}_2)$ for user u_1 and u_2 under their interaction conditions $\mathbf{x}_1$ and $\mathbf{x}_2$. To characterize this constraint analytically, we consider a regular exponential family with a sufficient statistic $T(\mathbf{x})$ to model the reconstruction distribution $p_\theta(\mathbf{x} \mid \mathbf{z})$. Here, $T(\mathbf{x})$ captures the information in the interactions relevant to reconstruction. This regular exponential family reduces to the commonly used multinomial decoder if and only if $T(\mathbf{x}) = \mathbf{x}$. A detailed proof is provided in Appendix A.

Theorem 2.2. *Assume that $p_\theta(\mathbf{x}|\mathbf{z})$, which represents the reconstruction process of user's interaction $\mathbf{x}$ from its latent representation $\mathbf{z}$, is a regular exponential family with sufficient statistics $T(\mathbf{x})$, natural parameter $\eta(\mathbf{z})$, and log-partition function A, with A^* denoting the convex conjugate of A. For two user interactions $\mathbf{x}_1, \mathbf{x}_2$, let $q_i(\mathbf{z}) = q_\phi(\mathbf{z}|\mathbf{x}_i)$, and define $Q(\mathbf{z}) = q_1(\mathbf{z}) + q_2(\mathbf{z})$ and $\alpha(\mathbf{z}) = \frac{q_1(\mathbf{z})}{q_1(\mathbf{z})+q_2(\mathbf{z})}$. Here, $Q(\mathbf{z})$ denotes the aggregated posterior density, while $\alpha(\mathbf{z})$ denotes the relative contribution of $q_1(\mathbf{z})$ at $\mathbf{z}$. Then, optimizing the mapping function $\eta(\cdot)$ aiming to minimize the reconstruction loss gives:*

$$\min_{\eta(\cdot)} \sum_{i=1}^{2} \mathbb{E}_{q_\phi(\cdot|\mathbf{x}_i)}[-\log p_\theta(\mathbf{x}_i|\mathbf{z})] = C + \int Q(\mathbf{z})\Delta_{A^*}(\mathbf{x}_1, \mathbf{x}_2; \alpha(\mathbf{z}))d\mathbf{z}, \tag{2}$$

where constant C is independent of $\eta(\cdot)$ and

$$\begin{aligned} \Delta_{A^*}(\mathbf{x}_1, \mathbf{x}_2; \alpha(\mathbf{z})) =& \alpha(\mathbf{z})A^*(T(\mathbf{x}_1)) + (1-\alpha(\mathbf{z}))A^*(T(\mathbf{x}_2)) \\ &- A^*(\alpha(\mathbf{z})T(\mathbf{x}_1) + (1-\alpha(\mathbf{z}))T(\mathbf{x}_2)) \geq 0. \end{aligned} \tag{3}$$

Furthermore, if the convex conjugate A^* of A is strictly convex, $\Delta_{A^*}(\mathbf{x}_1, \mathbf{x}_2; \alpha(\mathbf{z})) = 0$ if and only if either $T(\mathbf{x}_1) = T(\mathbf{x}_2)$ or $\alpha(\mathbf{z}) \in \{0, 1\}$ for all of $\mathbf{z}$.

Theorem 2.2 leads to the following conclusions about the relationship between the posterior distributions $q_\phi(\mathbf{z}|\mathbf{x}_1)$ and $q_\phi(\mathbf{z}|\mathbf{x}_2)$ under different interactions $\mathbf{x}_1$ and $\mathbf{x}_2$:

- When $T(\mathbf{x}_1) = T(\mathbf{x}_2)$, the two posterior distributions can overlap in the latent space without incurring an additional reconstruction loss, so $\Delta_{A^*}(\mathbf{x}_1, \mathbf{x}_2; \alpha(\mathbf{z})) = 0$ for any $\alpha(\mathbf{z})$.
- When $T(\mathbf{x}_1) \neq T(\mathbf{x}_2)$, the overlapping region where both posterior densities are nonzero corresponds to $\alpha(\mathbf{z}) \in (0, 1)$ and incurs a positive Jensen gap $\Delta_{A^*}(\mathbf{x}_1, \mathbf{x}_2; \alpha(\mathbf{z})) > 0$. As the overlap between the posterior distributions diminishing, the region contributing a positive Jensen gap becomes smaller, thereby reducing the additional reconstruction loss. When $\alpha(\mathbf{z})$ becomes 0 or 1, the Jensen gap vanishes at $\mathbf{z}$, resulting in non-overlapping posteriors.
- For sparse users, limited interactions may create larger interactive distances from other users, causing sparse users' posterior distributions to become separated from those of other users. Therefore, augmenting potential positive samples may assist reconstruction to alleviate the posterior separation pressure introduced by sparse users.

Overall, the reconstruction loss constrains posterior overlap according to the sufficient statistics of observed interactions. For sparse users, limited observed interactions provide less supervision information to learn their posterior distributions effectively. This motivate us to augment potential interactions and provide richer supervision to refine user posterior distributions.

Proof. See Appendix D.2. □

2.3 Learning Latent Preference Distributions beyond Interactions

Motivated by the above limitations, we aim to explore latent preference collaborations among users to identify potential positive samples from uninteracted items, thereby alleviating the interaction sparsity of VAE-based CFs.

We first define the interactive distance between users u and v in the interaction space as:

$$d_x(u, v) = \|\mathbf{x}_u - \mathbf{x}_v\|_1. \tag{4}$$

Definition 2.3 (Interaction collaboration)**.** *Given a predefined neighborhood radius $\delta > 0$, users u and v have an interaction collaboration if:*

$$d_x(u, v) \leq \delta.$$

Definition 2.4 (Latent Preference Distribution)**.** *We interpret the K mixture components of the GMM prior as global preference components. For user u, let $\gamma_u(c) = q_\phi(c \mid \mathbf{x}_u)$ denote the posterior probability of assignment to the c-th preference component. We collect these probabilities into:*

$$\boldsymbol{\gamma}_u = [\gamma_u(1), \ldots, \gamma_u(c), \ldots, \gamma_u(K)], \tag{5}$$

which is define as the LPD of user u over the K preference components.

Accordingly, We define the latent preference distance between users u and v in the latent space as:

$$d_\gamma(u, v) = \|\boldsymbol{\gamma}_u - \boldsymbol{\gamma}_v\|_1. \tag{6}$$

Definition 2.5 (Latent preference collaboration)**.** *For predefined preference radius $\epsilon > 0$, users u and v have a latent preference collaboration if:*

$$d_\gamma(u, v) \leq \epsilon.$$

Based on the above definitions, we can measure the distance between users in the latent space rather than the interaction space, allowing us to identify latent preference collaborations beyond interaction collaborations. These collaborations can then be exploited to discover potential positive samples from uninteracted items and alleviate interaction sparsity. Figure 1 illustrates the two limitations of existing VAE-based CFs and the goal of our work.

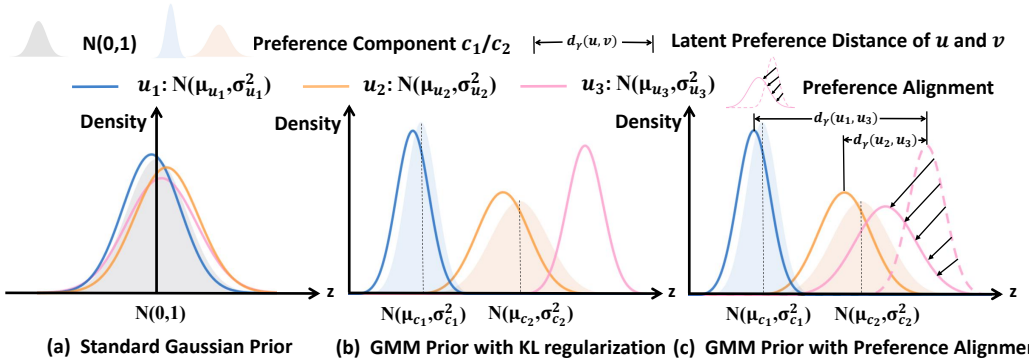

Figure 1: For visualization, the latent space is shown along a single dimension $\mathbf{z}$, with probability density on the vertical axis. Curves represent user posterior distributions $q_\phi(\mathbf{z} \mid \mathbf{x}_u) = \mathcal{N}(\boldsymbol{\mu}_u, \boldsymbol{\sigma}_u^2)$ produced by the encoder. (a) The standard Gaussian prior constrains user posterior distributions toward a single latent distribution, limiting the ability to capture diverse preferences across the three users. (b) Given two preference components c_1 and c_2 learned by GMM prior, three users are regularized by mixture prior KL divergence. However, due to the sparse interactions of u_3, its large interactive distances from u_1 and u_2 cause the reconstruction loss to impose separation on its posterior from those of other users, making u_3's preferences difficult to learn. (c) Our goal is to establish a latent user collaboration between u_2 and u_3 according to their latent preference distance $d_\gamma(u_2, u_3)$, which is short than that of u_1 and u_3 $d_\gamma(u_1, u_3)$. Based on this collaboration, we select the items from u_2 interacted item set, which have not been interacted by u_3, as the potential positive samples for u_3, and the preference alignment loss is then applied to further refine u_3's posterior distribution.

3 Methodology

3.1 The Generative Process of Preference Components

Following the general VAE-based CF framework described in Appendix A, we replace the standard Gaussian prior with a GMM prior and define its K Gaussian components as global latent preference components. Each preference component c is parameterized by a mixture weight ω_c, a mean vector $\boldsymbol{\mu}_c$, and a variance vector $\boldsymbol{\sigma}_c^2$. $\{(\omega_c, \boldsymbol{\mu}_c, \boldsymbol{\sigma}_c^2)\}_{c=1}^K$ are jointly learned with the encoder and decoder parameters. Specifically, the c-th preference component is defined as:

$$p(\mathbf{z} \mid c) = \mathcal{N}\left(\mathbf{z} \mid \boldsymbol{\mu}_c, \mathrm{diag}(\boldsymbol{\sigma}_c^2)\right).$$

The mixture weights are represented by a global learnable vector $\boldsymbol{\omega} = [\omega_1, \ldots, \omega_K]$, which is shared across all users and represents the weight of each preference component:

$$p(c) = \omega_c, \omega_c \geq 0, \sum_{c=1}^{K} \omega_c = 1.$$

Accordingly, the prior distribution of the latent representation $\mathbf{z}$ is defined as:

$$p(\mathbf{z}) = \sum_{c=1}^{K} \omega_c \mathcal{N}\left(\mathbf{z} \mid \boldsymbol{\mu}_c, \mathrm{diag}(\boldsymbol{\sigma}_c^2)\right).$$

3.2 Interaction Augmentation via Latent Preference Collaboration

For each user u, we compute the posterior probability of assignment to the c-th preference component as:

$$\gamma_u(c) = \frac{\omega_c p(\mathbf{z}_u|c)}{\sum_{c'=1}^{K} \omega_{c'} p(\mathbf{z}_u|c')}, \tag{7}$$

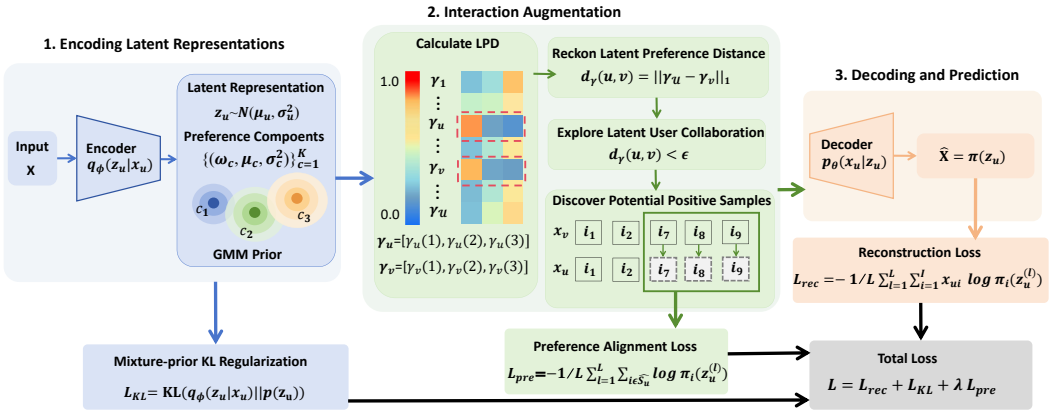

Figure 2: Overview of the proposed LPD-VAE framework. The framework consists of three steps: (1) Encoding latent representations, optimized with the mixture-prior KL regularization. (2) Interaction augmentation, guided by the preference alignment loss. (3) Decoding and prediction, optimized via the reconstruction loss. These losses are combined to form the total loss for model optimization.

The derivation is provided in Appendix D.3. Here, ω_c is the global mixture weight of the c-th preference component, and $p(\mathbf{z}_u|c)$ denotes the Gaussian density of $\mathbf{z}_u$ under the c-th preference component, as defined in Section 3.1. For each user u, the posterior assignment probabilities $\gamma_u(c)$ are collected into:

$$\boldsymbol{\gamma}_u = [\gamma_u(1), \ldots, \gamma_u(c), \ldots, \gamma_u(K)],$$

which represents the LPD of user u over the K preference components.

We then measure the latent preference distance between two users u and v by the L_1 distance:

$$d_\gamma(u, v) = \|\boldsymbol{\gamma}_u - \boldsymbol{\gamma}_v\|_1 .$$

For each user u, we select the M smallest distance users as the latent preference collaboration users:

$$\mathcal{N}_{\text{pref}}(u) = \text{TopM}_{v \neq u} \left(-d_\gamma(u, v)\right).$$

Let $S_u = \{i \mid x_{ui} = 1\}$ denote the set of items interacted by user u. Given the latent preference collaboration users in $\mathcal{N}_{\text{pref}}(u)$, we collect the items which they have interacted with but user u has not interacted as candidate potential positive items:

$$S_u^+ = \bigcup_{v \in \mathcal{N}_{\text{pref}}(u)} (S_v \setminus S_u).$$

The candidate items are then evaluated according to the current latent representation of user u. Specifically, the decoder output $\pi_i(\mathbf{z}_u)$ is used as the predicted probability of user u interacting with item i. We select the H highest predicted probabilities items as potential positive samples for user u:

$$\widetilde{S}_u = \text{TopH}_{i \in S_u^+} \pi_i(\mathbf{z}_u).$$

Finally, we define the preference-alignment loss w.r.t. potential positive samples as:

$$\mathcal{L}_{\text{pre}}(\mathbf{x}_u) = -\frac{1}{L} \sum_{l=1}^{L} \sum_{i \in \widetilde{S}_u} \log \pi_i(\mathbf{z}_u^{(l)}), \tag{8}$$

where L is the number of Monte Carlo samples. $\mathcal{L}_{\text{pre}}(\mathbf{x}_u)$ encourages the model to assign higher probabilities to potential positive items.

3.3 UPGRADING ELBO WITH PREFERENCE COMPONENTS

The ELBO can be further decomposed into a reconstruction loss and a mixture-prior KL regularization term as follows:

$$\begin{aligned}\mathcal{L}_{\mathrm{ELBO}}(\mathbf{x}) &= \mathbb{E}_{q_\phi(\mathbf{z},c|\mathbf{x})}\left[\log \frac{p(\mathbf{x},\mathbf{z},c)}{q_\phi(\mathbf{z},c|\mathbf{x})}\right] \\ &= \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right]}_{\text{Reconstruction Loss}} \\ &\quad \underbrace{- \sum_{c=1}^{K} \gamma(c) \mathrm{KL}\left(q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z}|c)\right) + \mathrm{KL}\left(q_\phi(c|\mathbf{x}) \| p(c)\right)}_{\text{Mixture-prior KL Regularization}}.\end{aligned} \tag{9}$$

The reconstruction loss encourages the decoder to recover observed interactions from the latent representations. The mixture-prior KL regularization consists of two terms. The first term encourages the user posterior $q_\phi(\mathbf{z}|\mathbf{x})$ to approach the preference components $p(\mathbf{z}|c)$ weighted by $\gamma(c)$. The second encourages the LPD $q_\phi(c|\mathbf{x})$ to align with the global mixture weight $p(c)$, incorporating global prior information into user-specific preference assignments. Together, the reconstruction loss and the Mixture-prior KL Regularization enable the model to preserve observed interactions while learning user representations structured by multiple preference components.

Building on the ELBO derived above, we further derive its explicit form under Equation 9 and use the SGVB estimator (Kingma & Welling, 2014) for optimization.

$$\begin{aligned}\mathcal{L}_{\mathrm{ELBO}}(\mathbf{x}_u) = &\frac{1}{L}\sum_{l=1}^{L}\sum_{i=1}^{\mathcal{I}} x_{ui} \log \pi_i(\mathbf{z}_u^{(l)}) \\ &- \frac{1}{2}\sum_{c=1}^{K}\gamma_u(c)\sum_{j=1}^{d}\left(\log \sigma_{cj}^2 + \frac{\sigma_{uj}^2}{\sigma_{cj}^2} + \frac{(\mu_{uj}-\mu_{cj})^2}{\sigma_{cj}^2}\right) \\ &+ \sum_{c=1}^{K}\gamma_u(c)\log\frac{\omega_c}{\gamma_u(c)} + \frac{1}{2}\sum_{j=1}^{d}\left(1+\log\sigma_{uj}^2\right).\end{aligned} \tag{10}$$

The detailed derivation from Equation 9 to Equation 10 is provided in Appendix D.4. Here, $\mathcal{I}$ denotes the number of items, K is the number of preference components, and d is the latent representation dimension. L is the number of Monte Carlo samples used in the SGVB estimator.

3.4 TOTAL LOSS

To incorporate the additional supervision derived from latent preference collaborations, we combine the ELBO with the preference-alignment loss defined in Section 3.2. The total training objective is:

$$\mathcal{L} = \sum_{u=1}^{\mathcal{U}} \left[-\mathcal{L}_{\mathrm{ELBO}}(\mathbf{x}_u) + \lambda \mathcal{L}_{\mathrm{pre}}(\mathbf{x}_u)\right], \tag{11}$$

where λ controls the contribution of the preference-alignment loss. We minimize $\mathcal{L}$ with respect to the encoder parameters ϕ, decoder parameters θ, and preference component parameters $\left\{\omega_c, \boldsymbol{\mu}_c, \boldsymbol{\sigma}_c^2\right\}_{c=1}^{K}$. The training procedure of LPD-VAE is summarized as Algorithm 1 presented in Appendix C.2.

4 EXPERIMENTS

4.1 OVERALL PERFORMANCE

We follow the preprocessing procedure from (Liang et al., 2018; Shenbin et al., 2020; Vuong et al., 2026). The detailed data preprocessing steps and train/validation/test dataset splitting method are presented in Section C.1. Our code is publicly available at `https://anonymous.4open.science/r/LPD-VAE-FEED`.

Table 1: Performance comparison of LPD-VAE and its variants against SOTAs. The best results are shown in bold. All results are averaged over five runs. Improv.% indicates the relative improvement over the corresponding base model or its PIA-enhanced variant.

Model	MovieLens-20M			Netflix Prize			Million Song		
	Recall @20	Recall @50	NDCG @100	Recall @20	Recall @50	NDCG @100	Recall @20	Recall @50	NDCG @100
Matrix Factorization & Linear Models									
Popularity	0.162	0.235	0.191	0.116	0.175	0.159	0.043	0.068	0.058
DirectAU	0.317	0.461	0.349	0.286	0.357	0.293	0.174	0.234	0.149
WMF	0.362	0.495	0.389	0.321	0.402	0.349	0.211	0.312	0.257
SLIM	0.370	0.495	0.401	0.347	0.428	0.379	0.235	0.341	0.283
LambdaNet	0.395	0.534	0.427	0.352	0.441	0.386	0.259	0.355	0.308
SaFeAU	0.392	0.528	0.424	0.351	0.438	0.372	0.251	0.346	0.304
Nonlinear Autoencoders									
CDAE	0.391	0.523	0.418	0.343	0.428	0.376	0.188	0.283	0.237
RaCT	0.403	0.543	0.434	0.357	0.450	0.392	0.268	0.364	0.319
Multi-VAE	0.395	0.537	0.426	0.351	0.444	0.386	0.266	0.364	0.316
Multi-VAE + PIA	0.408	0.546	0.437	0.360	0.448	0.392	0.275	0.372	0.326
Multi-VAE + LPD	0.410	0.550	0.439	0.365	0.451	0.396	0.278	0.374	0.328
Improv.% vs base (%)	3.79	2.42	3.05	3.99	1.58	2.59	4.51	2.75	3.80
Improv.% vs +PIA (%)	0.49	0.73	0.46	1.39	0.67	1.02	1.09	0.53	0.61
HVamp	0.405	0.540	0.437	0.365	0.451	0.392	0.263	0.358	0.311
HVamp + PIA	0.411	0.545	0.443	0.366	0.453	0.393	0.272	0.361	0.324
HVamp + LPD	0.418	0.551	0.447	0.367	0.455	0.397	0.274	0.366	0.328
Improv.% vs base (%)	3.21	2.04	2.29	0.55	0.89	1.28	4.18	2.23	5.47
Improv.% vs +PIA (%)	1.70	1.11	0.90	0.27	0.44	1.02	0.74	1.39	1.23
RecVAE	0.414	0.553	0.442	0.361	0.452	0.394	0.276	0.374	0.326
RecVAE + PIA	0.417	0.556	0.446	0.365	0.454	0.395	0.278	0.376	0.329
RecVAE + LPD	**0.423**	**0.562**	**0.449**	**0.367**	**0.456**	**0.401**	**0.279**	**0.378**	**0.332**
Improv.% vs base (%)	2.17	1.63	1.58	1.66	0.88	1.78	1.09	1.07	1.84
Improv.% vs +PIA (%)	1.44	1.08	0.67	0.55	0.46	1.52	0.36	0.53	0.91

Table 1 shows that incorporating LPD into VAE backbones (Multi-VAE, HVamp, and RecVAE) consistently improves performance over baselines and PIA. This indicates that LPDs capture latent preference collaborations beyond direct interactions and provide complementary supervision for recommendation.

4.2 ABLATION EXPERIMENT

From the results in Table 2, we can draw the following observations:

- Compared with the standard Gaussian prior used in Multi-VAE, the Multi-VAE + GMM achieves better performance across datasets, demonstrating that a more flexible mixture prior can better capture heterogeneous user preferences. This supports our analysis in Section 2.1 that KL regularization with the a single distribution constrains the diversity of user posterior distributions.
- Notably, LPD-VAE consistently outperforms the baseline, supporting our analysis in Section 2.2 that augmented interactions provide sparse users with additional supervision to alleviate posterior separation induced by reconstruction loss.

Table 2: Ablation experiment of LPD-VAE. Multi-VAE adopts a standard Gaussian prior. Multi-VAE + GMM stands for replacing the standard Gaussian prior with the GMM prior. Multi-VAE + Rand-Align randomly selects items for aligning user preferences, while LPD-VAE(d_x) is a version of LPD-VAE that removes the GMM and instead interaction distance to explore user collaboration in the interaction space.

Dataset	Metric	Multi-VAE	Multi-VAE + GMM	Multi-VAE + Rand-Align	LPD-VAE(d_x)
MovieLens-20M	R@20	0.395	0.403	0.383	0.397
	R@50	0.537	0.540	0.528	0.538
	N@100	0.426	0.436	0.409	0.427
Netflix Prize	R@20	0.351	0.357	0.338	0.353
	R@50	0.444	0.446	0.426	0.444
	N@100	0.386	0.392	0.359	0.387
Million Song	R@20	0.266	0.271	0.253	0.268
	R@50	0.364	0.368	0.342	0.365
	N@100	0.316	0.323	0.306	0.319

- Multi-VAE + Rand-Align performs worse than Multi-VAE on all three datasets, suggesting that simply introducing additional samples in a random manner is not beneficial. Instead, it may introduce noisy or uninformative supervision, thereby damaging the learned representations.
- LPD-VAE generally achieves better results than LPD-VAE(d_x), indicating that latent preference distance captures user collaborations more effectively than interaction distance, and suggesting that LPDs can reveal potential user collaborations beyond observed interactions.

4.3 Performance across Interaction Densities

Table 3: Performance across user groups with varying interaction densities on MovieLens-20M.

	Multi-VAE			Multi-VAE+PIA			LPD-VAE		
User Group	R@20	R@50	N@100	R@20	R@50	N@100	R@20	R@50	N@100
[1-4]	0.142	0.167	0.122	0.158	0.213	0.136	0.245	0.337	0.192
[5-10]	0.463	0.625	0.318	0.473	0.629	0.323	0.489	0.644	0.342
[11-50]	0.422	0.596	0.429	0.424	0.598	0.432	0.452	0.613	0.448
[51-100]	0.313	0.479	0.497	0.314	0.479	0.501	0.316	0.481	0.504
[100+]	0.419	0.388	0.474	0.435	0.393	0.485	0.439	0.413	0.492

To further analyze sparse users, we evaluate users with fewer than 5 interactions as the very sparse group. Table 3 shows that LPD-VAE consistently improves across different interaction densities. For sparse users, LPD-VAE benefits from latent preference collaborations that provide additional supervision, while improvements on dense users demonstrate that LPDs capture collaborative signals beyond observed interactions.

5 Related Works

Due to space constraints, we defer the detailed discussion of related work to Appendix B.

6 Conclusion

To address the two challenges in existing VAE-based CFs, we propose LPD-VAE, which uses a mixture prior to learn user LPDs and discover latent preference collaborations. By exploiting these collaborations to identify potential positive samples, LPD-VAE provides additional supervision through a preference-alignment loss to alleviate sparsity. Extensive experiments show consistent improvements across datasets and interaction densities, and ablation studies demonstrate the superiority of latent preference distance over interaction distance for learning user preferences.

THE USE OF LARGE LANGUAGE MODELS

We utilized ChatGPT primarily as an editorial aid to enhance the clarity and flow of the manuscript. Its role was limited to language polishing, typo correction, and suggesting improvements to the presentation of mathematical proofs. All core research contributions, including the conceptualization, experimental design, and analysis, are entirely our own. We have carefully verified all AI-assisted edits and assume full responsibility for the accuracy and integrity of this work.

REFERENCES

Sergej G Bobkov and Friedrich Götze. Exponential integrability and transportation cost related to logarithmic sobolev inequalities. *Journal of Functional Analysis*, 163(1):1–28, 1999.

Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. Controllable multi-interest framework for recommendation. In Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash (eds.), *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pp. 2942–2951. ACM, 2020. doi: 10.1145/3394486.3403344. URL https://doi.org/10.1145/3394486.3403344.

Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31, 2018.

Nat Dilokthanakul, Pedro A. M. Mediano, Marta Garnelo, Matthew C. H. Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *CoRR*, abs/1611.02648, 2016. URL http://arxiv.org/abs/1611.02648.

Monroe D Donsker and SR Srinivasa Varadhan. Asymptotic evaluation of certain markov process expectations for large time, i. *Communications on pure and applied mathematics*, 28(1):1–47, 1975.

Werner Fenchel. On conjugate convex functions. In *Traces and Emergence of Nonlinear Programming*, pp. 125–129. Springer, 2013.

Santambrogio Filippo. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55:58–63, 2015.

Samuel Gershman and Noah D. Goodman. Amortized inference in probabilistic reasoning. In Paul Bello, Marcello Guarini, Marjorie McShane, and Brian Scassellati (eds.), *Proceedings of the 36th Annual Meeting of the Cognitive Science Society, CogSci 2014, Quebec City, Canada, July 23-26, 2014*. cognitivesciencesociety.org, 2014. URL https://escholarship.org/uc/item/34j1h7k5.

Guibing Guo, Huihuai Qiu, Zhenhua Tan, Yuan Liu, Jing Ma, and Xingwei Wang. Resolving data sparsity by multi-type auxiliary implicit feedback for recommender systems. *Knowledge-Based Systems*, 138:202–207, 2017.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pp. 639–648, 2020.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: An unsupervised and generative approach to clustering. *arXiv preprint arXiv:1611.05148*, 2016.

Daeryong Kim and Bongwon Suh. Enhancing vaes for collaborative filtering: flexible priors & gating mechanisms. In Toine Bogers, Alan Said, Peter Brusilovsky, and Domonkos Tikk (eds.), *Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019*, pp. 403–407. ACM, 2019. doi: 10.1145/3298689.3347015. URL https://doi.org/10.1145/3298689.3347015.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=H1kG7GZAW.

Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. Multi-interest network with dynamic routing for recommendation at tmall. In Wenwu Zhu, Dacheng Tao, Xueqi Cheng, Peng Cui, Elke A. Rundensteiner, David Carmel, Qi He, and Jeffrey Xu Yu (eds.), *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pp. 2615–2623. ACM, 2019. doi: 10.1145/3357384.3357814. URL https://doi.org/10.1145/3357384.3357814.

Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*, pp. 689–698, 2018.

Xubin Ren, Lianghao Xia, Jiashu Zhao, Dawei Yin, and Chao Huang. Disentangled contrastive collaborative filtering. In Hsin-Hsi Chen, Wei-Jou (Edward) Duh, Hen-Hsen Huang, Makoto P. Kato, Josiane Mothe, and Barbara Poblete (eds.), *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pp. 1137–1146. ACM, 2023. doi: 10.1145/3539618.3591665. URL https://doi.org/10.1145/3539618.3591665.

Ilya Shenbin, Anton Alekseev, Elena Tutubalina, Valentin Malykh, and Sergey I Nikolenko. Recvae: A new variational autoencoder for top-n recommendations with implicit feedback. In *Proceedings of the 13th international conference on web search and data mining*, pp. 528–536, 2020.

Andrew Stirn and David A. Knowles. The vampprior mixture model. In Yingzhen Li, Stephan Mandt, Shipra Agrawal, and Mohammad Emtiyaz Khan (eds.), *International Conference on Artificial Intelligence and Statistics, AISTATS 2025, Mai Khao, Thailand, 3-5 May 2025*, volume 258 of *Proceedings of Machine Learning Research*, pp. 1369–1377. PMLR, 2025. URL https://proceedings.mlr.press/v258/stirn25a.html.

Jakub M. Tomczak and Max Welling. VAE with a vampprior. In Amos J. Storkey and Fernando Pérez-Cruz (eds.), *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, volume 84 of *Proceedings of Machine Learning Research*, pp. 1214–1223. PMLR, 2018. URL http://proceedings.mlr.press/v84/tomczak18a.html.

Long Vuong, Julien Monteil, Hien Dang, Volodymyr Vaskovych, Trung Le, and Vu Nguyen. On the mechanisms of collaborative learning in vae recommenders. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 99115–99140, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/a102d6cb996be3482c059c1e18bbe523-Paper-Conference.pdf.

Chenyang Wang, Yuanqing Yu, Weizhi Ma, Min Zhang, Chong Chen, Yiqun Liu, and Shaoping Ma. Towards representation alignment and uniformity in collaborative filtering. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 1816–1825, 2022.

Hao Wang, Zhichao Chen, Haotian Wang, Yanchao Tan, Pan Li, Tianqiao Liu, Xu Chen, Haoxuan Li, and Zhouchen Lin. Unbiased recommender learning from implicit feedback via weakly supervised learning. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/wang25p.html.

Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, pp. 165–174, 2019.

Xin Wang, Hong Chen, Yuwei Zhou, Jianxin Ma, and Wenwu Zhu. Disentangled representation learning for recommendation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(1):408–424, 2023. doi: 10.1109/TPAMI.2022.3153112. URL https://doi.org/10.1109/TPAMI.2022.3153112.

Xin Wang, Hong Chen, Si'ao Tang, Zihao Wu, and Wenwu Zhu. Disentangled representation learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46(12):9677–9696, 2024. doi: 10.1109/TPAMI.2024.3420937. URL https://doi.org/10.1109/TPAMI.2024.3420937.

Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the ninth ACM international conference on web search and data mining*, pp. 153–162, 2016.

Liangwei Yang, Zhiwei Liu, Chen Wang, Mingdai Yang, Xiaolong Liu, Jing Ma, and Philip S Yu. Graph-based alignment and uniformity for recommendation. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pp. 4395–4399, 2023.

Yajie Yu, Chenzhong Bin, Zhoubo Xu, Zhixin Zeng, Tongxin Xu, Cihan Xiao, and Jiafeng Wu. Beyond instance-level alignment and uniformity: Semantic factor learning for collaborative filtering. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '26, pp. 6326–6337, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400722592. doi: 10.1145/3770855.3817704. URL https://doi.org/10.1145/3770855.3817704.

Weizhi Zhang, Liangwei Yang, Zihe Song, Henry Peng Zou, Ke Xu, Liancheng Fang, and Philip S Yu. Do we really need graph convolution during training? light post-training graph-ode for efficient recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 3248–3258, 2024.

APPENDIX

This supplementary material provides detailed formulations, experimental settings, additional results, algorithmic descriptions, and theoretical analysis.

- We introduce the general formulation of VAE-based CF and summarize the notation used throughout the paper in Section A.
- We present the Related Work in Section B.
- Detailed experimental settings and implementation details are described in Section C.
- The pseudo-code of the algorithm is provided in Algorithm 1.
- We present all proofs relevant to the theory developed in our paper in Section D.

A VAE-BASED COLLABORATIVE FILTERING

We first summarize the general formulation of VAE-CF adopted throughout this paper. Given a user's historical interactions, VAE-CF infers a latent variable through an encoder and reconstructs

the observed interactions through a probabilistic decoder. This formulation defines the basic probabilistic components and notations used in our subsequent analysis. Table 4 summarizes the main notations used throughout the paper, including the VAE formulation, GMM prior, and theoretical analysis.

Generative Process. For each user u, let $\mathbf{z}_u \in \mathbb{R}^d$ denote the corresponding latent variable. Given $\mathbf{z}_u$, a neural decoder parameterized by θ transforms it into an unnormalized item score vector:

$$\mathbf{o}_u = f_\theta(\mathbf{z}_u), \tag{12}$$

where $f_\theta(\cdot) : \mathbb{R}^d \to \mathbb{R}^{\mathcal{I}}$ is a multilayer perceptron. The output scores are normalized through a softmax function to obtain a probability distribution over all items:

$$\pi_i(\mathbf{z}_u) = \frac{\exp(o_{ui})}{\sum_{j=1}^{I} \exp(o_{uj})}, \tag{13}$$

where the probability vector over all items is defined as $\pi(\mathbf{z}_u) = [\pi_1(\mathbf{z}_u), \pi_2(\mathbf{z}_u), \ldots, \pi_{\mathcal{I}}(\mathbf{z}_u)]^\top \in \mathbb{S}^{\mathcal{I}-1}$, where $\mathbb{S}^{\mathcal{I}-1}$ denotes the $(\mathcal{I}-1)$-dimensional probability simplex. Given the probability vector $\pi(\mathbf{z}_u)$, the user's observed interaction history is modeled as a multinomial distribution:

$$\mathbf{x}_u \sim \mathrm{Mult}(N_u, \pi(\mathbf{z}_u)), \tag{14}$$

where $N_u = \sum_i x_{ui}$ denotes the total number of observed interactions of user u. Accordingly, the decoder is optimized by maximizing the likelihood of the observed interaction history under the predicted distribution $\pi(\mathbf{z}_u)$. The corresponding log-likelihood is given by:

$$\log p_\theta(\mathbf{x}_u|\mathbf{z}_u) \stackrel{c}{=} \sum_i x_{ui} \log \pi_i(\mathbf{z}_u), \tag{15}$$

where the constant terms independent of θ are omitted.

Variational Inference. To approximate the intractable posterior, VAE employs an encoder $q_\phi(\mathbf{z}_u|\mathbf{x}_u)$ that infers the latent variable from the user's observed interactions. The posterior is parameterized as:

$$q_\phi(\mathbf{z}_u|\mathbf{x}_u) = \mathcal{N}(\mathbf{z}_u; \boldsymbol{\mu}_\phi(\mathbf{x}_u), \mathrm{diag}(\boldsymbol{\sigma}^2_\phi(\mathbf{x}_u))). \tag{16}$$

A latent representation is sampled via the reparameterization trick:

$$\mathbf{z}_u = \boldsymbol{\mu}_\phi(\mathbf{x}_u) + \boldsymbol{\sigma}_\phi(\mathbf{x}_u) \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}). \tag{17}$$

Training Objective. Let $p(\mathbf{z}_u)$ denote the prior distribution over the latent variable. The parameters ϕ and θ are optimized by minimizing the negative β-regularized evidence lower bound (ELBO):

$$\mathcal{L}_{\text{VAE}}(\mathbf{x}; \theta, \phi) = -\mathbb{E}_{q_\phi(\mathbf{z}_u|\mathbf{x}_u)}\big[\log p_\theta(\mathbf{x}_u \mid \mathbf{z}_u)\big] + \beta\, \mathrm{KL}\left(q_\phi(\mathbf{z}_u \mid \mathbf{x}_u) \,\|\, p(\mathbf{z}_u)\right). \tag{18}$$

B Related Works

Flexible Priors for VAE-based CF

VAE-based models have been widely studied in computer vision (Wang et al., 2024), where the standard Gaussian prior is commonly adopted as a simple and tractable choice for regularizing latent representations (Higgins et al., 2017; Chen et al., 2018; Kumar et al., 2018). Early VAE-based CF methods, such as Multi-VAE (Liang et al., 2018), adopt a standard Gaussian prior and adapt the VAE framework to implicit user-item interactions, particularly through the choice of likelihood and KL regularization. Kim & Suh (2019) incorporate VampPrior to provide a more flexible prior for latent representations. RecVAE (Shenbin et al., 2020) further adopts a composite prior together with a modified regularization scheme to improve the learning of latent representations.

Meanwhile, GMM-based priors have been extensively explored in broader VAE-based representation learning, particularly in computer vision, for modeling more complex latent distributions (Jiang

Table 4: Table of Notations

Symbol	Description
Input Data	
$\mathcal{U}, \mathcal{I}$	Number of users and number of items
$\mathbf{x}_u = [\mathbf{x}_{u1}, \mathbf{x}_{u2}, \ldots, \mathbf{x}_{u\mathcal{I}}]$	$\mathcal{I}$-dimensional binary vector (the u-th row of $\mathbf{X}$); $\mathbf{x}_{ui} = 1$ implies that user u has a positive interaction with item i; $\mathbf{x}_{ui} = 0$ indicates otherwise
$S_u = \{i \mid x_{ui} = 1\}$	Set of positive items interacted with by user u
$N_u \in \mathbb{N}$	Number of interactions of user u, $N_u = \sum_{i=1}^{I} x_{ui}$
VAE Models	
ϕ, θ	Parameters of the encoder q_ϕ and decoder p_θ
q_ϕ, p_θ	ϕ-parameterized and θ-parameterized neural networks
$\boldsymbol{\mu}_u,\ \boldsymbol{\sigma}_u^2$	Mean and variance of the posterior distribution for user u
$\mathbf{z}_u \in \mathbb{R}^d$	Latent variable of user u in the d-dimensional latent space
$\pi_i(\mathbf{z}_u) \in (0, 1)$	Predicted interaction probability based on $\mathbf{z}_u$ for item i
Latent Preference Components	
$K,\ c \in \{1, \ldots, K\}$	Number of preference components and index of the c-th component
$\boldsymbol{\omega} = [\omega_1, \ldots, \omega_K]$	Mixture-weight vector of the K preference components
$\boldsymbol{\mu}_c,\ \boldsymbol{\sigma}_c^2$	Mean and variance of the c-th preference component
$\boldsymbol{\gamma}_u = [\gamma_u(1), \ldots, \gamma_u(K)]$	LPD of user u, with $\gamma_u(c) = q(c \mid \mathbf{x}_u)$
Norms and Distance	
$\|\cdot\|_1$	L_1 norm (sum of absolute values)
$W_1(.,.)$	1-Wasserstein distance

et al., 2016; Dilokthanakul et al., 2016), but their application to VAE-based CF remains relatively limited.

These approaches enhance the expressiveness of latent priors, while how to design priors tailored to sparse and heterogeneous user-item interactions remains relatively underexplored in VAE-based CF.

Collaborative Signals in VAE-based CF

VAE-based CF methods learn user representations from historical interaction patterns, thereby implicitly incorporating collaborative information into the latent space. Multi-VAE (Liang et al., 2018) and RecVAE (Shenbin et al., 2020) jointly learn user representations using shared neural networks. PIA (Vuong et al., 2026) further introduces learnable item anchors to enhance the robustness of user representations against corrupted interaction inputs. Overall, these methods incorporate collaborative information through the representation learning process based on users' observed interaction patterns. Beyond VAE-based CF, recent MF-based methods (Yu et al., 2026; Wang et al., 2025; Ren et al., 2023) explore collaborative signals from unobserved interactions. Inspired by these studies, we explore latent preference collaborations in VAE latent space to identify potential positive samples.

C EXPERIMENTAL SETTINGS

C.1 DATASET

We validate our analysis using three real-world recommendation datasets: MovieLens-20M[1], Netflix[2], and Million Song (MSD)[3], where each record consists of a user-item pair along with a rating that the user has given to the item. We follow the preprocessing procedure from MultVAE (Liang et al., 2018). For MovieLens-20M and Netflix, we retain users who have rated at least five movies and treat ratings of four or higher as positive interactions. For MSD, we keep only users with at least 20 songs in their listening history and songs that have been listened to by at least 200 users.

Table 5: Dataset statistics.

	MovieLens-20M	Netflix	MSD
# of users	136,677	463,435	571,355
# of items	20,108	17,769	41,140
# of interactions	10.0M	56.9M	33.6M
% of interactions	0.36%	0.69%	0.14%
# of held-out users	10,000	40,000	50,000

The user data is split into training, validation, and test sets as presented in Table 5. For every user in the training set, we utilize all interaction history, whereas for users in the validation or test set, a fraction of the history (80%) is used to predict the remaining interaction.

C.2 IMPLEMENTATION DETAILS

Hyperparameters. The main hyperparameters of LPD-VAE include the number of latent preference components K, the number of preference-related users M, the number of augmented positive samples H, and the weight of the preference alignment loss λ.

For MovieLens-20M, the optimal hyperparameters are set to $K = 10$, $M = 5$, $H = 3$, and $\lambda = 0.5$. For Netflix Prize, we set $K = 10$, $M = 10$, $H = 2$, and $\lambda = 1$ based on the validation performance. For Million Song, the optimal settings are $K = 20$, $M = 5$, $H = 5$, and $\lambda = 0.1$.

Algorithm. The model is trained by optimizing the objective defined in Equation 11. The training procedure of LPD-VAE is summarized as Algorithm 1.

[1] https://grouplens.org/datasets/movielens/20m/

[2] https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data

[3] http://millionsongdataset.com

Algorithm 1 Training Procedure of LPD-VAE

Initialize: encoder q_ϕ, decoder p_θ, GMM prior parameters $\{\omega_c, \boldsymbol{\mu}_c, \boldsymbol{\sigma}_c^2\}_{c=1}^K$, and hyper-parameters M and λ.
for epoch=1 to n_{epochs} **do**
  Infer latent preference distributions
  for each user u **do**
    Compute latent preference distribution $\boldsymbol{\gamma}_u$ using Eq. 7
  end for
  Construct preference-based positive interactions
  for each user u **do**
    Identify preference neighbors: $\mathcal{N}_{\text{pref}}(u) \leftarrow \text{TopM}_{v \neq u}(-d_\gamma(u, v))$
    Augment interactions: $S_u^+ \leftarrow \bigcup_{v \in \mathcal{N}_{\text{pref}}(u)} (S_v \setminus S_u)$
  end for
  Optimize LPD-VAE
  for each mini-batch $\mathcal{B}$ **do**
    Sample $\mathbf{z}_u \sim q_\phi(\mathbf{z}|\mathbf{x}_u)$
    Compute ELBO loss $\mathcal{L}_{\text{ELBO}}$
    Compute augmented reconstruction loss $\mathcal{L}_{\text{rec}}^+$
    Update q_ϕ, p_θ, and $\{\omega_c, \boldsymbol{\mu}_c, \boldsymbol{\sigma}_c^2\}_{c=1}^K$
  end for
end for
Return: trained LPD-VAE model.

C.3 BASELINES

We have selected the following models as baselines:

- **Matrix factorization and linear models:** we consider Popularity, WMF, SLIM, DirectAU (Wang et al., 2022), LambdaNet, and SaFeAU (Yu et al., 2026). These methods learn user representations based on observed interaction patterns or collaborative regularization.
- **Nonlinear autoencoders:** we consider CDAE (Wu et al., 2016) and RaCT, which employ nonlinear neural architectures to reconstruct user-item interactions.
- **Variational autoencoders:** we consider representative VAE-based CFs, including Multi-VAE (Liang et al., 2018), HVamp (Kim & Suh, 2019), and RecVAE (Shenbin et al., 2020). PIA (Vuong et al., 2026) is further considered as an enhancement method that improves VAE-based CF by introducing learnable item anchors. We incorporate LPD into these models to investigate whether LPDs can provide additional collaborative signals beyond observed interactions.

D THEORETICAL DEVELOPMENT

In this Section, we present all proofs relevant to theory developed in our paper.

D.1 PROOF OF THEOREM 2.1

Lemma D.1. *Let $p(\mathbf{z})$ be a prior distribution satisfying the Bobkov–Götze/Talagrand $T_1(C)$ inequality with constant $C > 0$. For any probability distribution ν over the latent space with $\text{KL}(\nu\|p) < \infty$, we have*

$$W_1(\nu, p) \leq \sqrt{2C \,\text{KL}(\nu\|p)}. \tag{19}$$

Proof. According to the Kantorovich-Rubinstein dual form (Filippo, 2015):

$$W_1(\nu, p) = \sup_{\substack{f:\mathcal{X}\to\mathbb{R} \\ \text{Lip}(f)\leq 1}} \int_{\mathcal{X}} f \, d(\nu - p),$$

Here, f ranges over all 1-Lipschitz functions, and $W_1(\nu, p)$ is given by the supremum of the difference between the expected values of f under ν and p. For any measurable function g with $\int e^g \, dp < \infty$, the Donsker–Varadhan variational formula (Donsker & Varadhan, 1975) gives:

$$\mathrm{KL}(\nu\|p) \geq \int g \, d\nu - \log \int e^g dp.$$

Applying this inequality with $g = \lambda(f - \mathbb{E}_p f)$, where $\lambda > 0$, we obtain:

$$\mathrm{KL}(\nu\|p) \geq \lambda \int (f - \mathbb{E}_p f) d\nu - \log \int \exp(\lambda(f - \mathbb{E}_p f)) dp.$$

Since p satisfies the $T_1(C)$ inequality, the sub-Gaussian moment generating function bound (Bobkov & Götze, 1999) gives:

$$\log \int \exp(\lambda(f - \mathbb{E}_p f)) dp \leq \frac{C\lambda^2}{2}.$$

Thus,

$$\int f \, d(\nu - p) \leq \frac{1}{\lambda} \mathrm{KL}(\nu\|p) + \frac{C\lambda}{2}.$$

We choose $\lambda > 0$ to minimize the right-hand side, which gives:

$$\lambda^* = \sqrt{\frac{2\mathrm{KL}(\nu\|p)}{C}},$$

and hence

$$\int f \, d(\nu - p) \leq \sqrt{2C \, \mathrm{KL}(\nu\|p)}.$$

Taking the supremum over all 1-Lipschitz functions yields the upper bound for $W_1(\nu, p)$ and completes the proof.

□

Theorem D.2 (KL divergence regularization Constrains Posterior Diversity)**.** *Assume that the prior distribution $p(\mathbf{z})$ satisfies the Bobkov–Götze/Talagrand $T_1(C)$ inequality with constant $C > 0$. Then, for any two users u_1 and u_2, their posterior distributions $q_\phi(\mathbf{z}|\mathbf{x}_1)$ and $q_\phi(\mathbf{z}|\mathbf{x}_2)$,*

$$\begin{aligned} &W_1(q_\phi(\mathbf{z}|\mathbf{x}_1), q_\phi(\mathbf{z}|\mathbf{x}_2)) \\ &\leq \sqrt{2C\mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x}_1)\|p)} + \sqrt{2C\mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x}_2)\|p)}. \end{aligned} \tag{20}$$

Proof. By the triangle inequality of Wasserstein distance,

$$\begin{aligned} &W_1(q_\phi(\mathbf{z}|\mathbf{x}_1), q_\phi(\mathbf{z}|\mathbf{x}_2)) \\ &\leq W_1(q_\phi(\mathbf{z}|\mathbf{x}_1), p) + W_1(q_\phi(\mathbf{z}|\mathbf{x}_2), p). \end{aligned}$$

By the above lemma, the following bound holds for any distribution ν:

$$W_1(\nu, p) \leq \sqrt{2C\mathrm{KL}(\nu\|p)}.$$

Applying the above inequality to $q_\phi(\mathbf{z}|\mathbf{x}_1)$ and $q_\phi(\mathbf{z}|\mathbf{x}_2)$ proves the theorem. □

D.2 PROOF OF THEOREM 2.2

Theorem D.3 (Reconstruction-Induced Constraint on Posterior Overlap)**.** *Assume that $p_\theta(\mathbf{x}|\mathbf{z})$ is a regular exponential family with sufficient statistics $T(\mathbf{x})$, natural parameter $\eta(\mathbf{z})$, and log-partition function A. For two samples $\mathbf{x}_1, \mathbf{x}_2$, let $q_i(\mathbf{z}) = q_\phi(\mathbf{z}|\mathbf{x}_i)$ and define $Q(\mathbf{z}) = q_1(\mathbf{z}) + q_2(\mathbf{z})$, $\alpha(\mathbf{z}) = \frac{q_1(\mathbf{z})}{q_1(\mathbf{z})+q_2(\mathbf{z})}$, on the region where $Q(\mathbf{z}) > 0$. Then the optimal decoder parameter $\eta(\cdot)$ minimizes the reconstruction loss as*

$$\min_{\eta(\cdot)} \sum_{i=1}^{2} \mathbb{E}_{q_\phi(\cdot|\mathbf{x}_i)}[-\log p_\theta(\mathbf{x}_i|\mathbf{z})] = C + \int Q(\mathbf{z}) \Delta_{A^*}(\mathbf{x}_1, \mathbf{x}_2; \alpha(\mathbf{z})) d\mathbf{z}, \tag{21}$$

where C is independent of $\eta(\cdot)$ and

$$\begin{aligned} \Delta_{A^*}(\mathbf{x}_1, \mathbf{x}_2; \alpha) =& \alpha A^*(T(\mathbf{x}_1)) + (1-\alpha) A^*(T(\mathbf{x}_2)) \\ & - A^*(\alpha T(\mathbf{x}_1) + (1-\alpha) T(\mathbf{x}_2)) \geq 0. \end{aligned} \tag{22}$$

Furthermore, if A^* is strictly convex, $\Delta_{A^*}(\mathbf{x}_1, \mathbf{x}_2; \alpha) = 0$ if and only if either $T(\mathbf{x}_1) = T(\mathbf{x}_2)$ or $\alpha \in \{0, 1\}$ for almost all $\mathbf{z}$.

Proof. Under the exponential-family assumption, the conditional likelihood $p_\theta(\mathbf{x}|\mathbf{z})$ admits the following representation:

$$p_\theta(\mathbf{x}|\mathbf{z}) = h(\mathbf{x}) \exp(\langle T(\mathbf{x}), \eta(\mathbf{z}) \rangle - A(\eta(\mathbf{z}))).$$

Therefore,

$$-\log p_\theta(\mathbf{x}_i|\mathbf{z}) = A(\eta(\mathbf{z})) - \langle T(\mathbf{x}_i), \eta(\mathbf{z}) \rangle - \log h(\mathbf{x}_i).$$

Let

$$\overline{T}_\alpha(\mathbf{z}) = \alpha(\mathbf{z}) T(\mathbf{x}_1) + (1 - \alpha(\mathbf{z})) T(\mathbf{x}_2).$$

The reconstruction objective can be expanded as

$$\begin{aligned} & \sum_{i=1}^{2} \mathbb{E}_{q_i}[-\log p_\theta(\mathbf{x}_i|\mathbf{z})] \\ =& \sum_{i=1}^{2} \int q_i(\mathbf{z}) \left(A(\eta(\mathbf{z})) - \langle T(\mathbf{x}_i), \eta(\mathbf{z}) \rangle - \log h(\mathbf{x}_i) \right) d\mathbf{z} \\ =& \int (q_1(\mathbf{z}) + q_2(\mathbf{z})) A(\eta(\mathbf{z})) d\mathbf{z} - \int \langle q_1(\mathbf{z}) T(\mathbf{x}_1) + q_2(\mathbf{z}) T(\mathbf{x}_2), \eta(\mathbf{z}) \rangle \, d\mathbf{z} - \sum_{i=1}^{2} \log h(\mathbf{x}_i) \int q_i(\mathbf{z}) d\mathbf{z}. \end{aligned}$$

Since $\int q_i(\mathbf{z}) d\mathbf{z} = 1$ and $Q(\mathbf{z}) = q_1(\mathbf{z}) + q_2(\mathbf{z})$, we have

$$\begin{aligned} & q_1(\mathbf{z}) T(\mathbf{x}_1) + q_2(\mathbf{z}) T(\mathbf{x}_2) \\ =& Q(\mathbf{z}) \left(\frac{q_1(\mathbf{z})}{Q(\mathbf{z})} T(\mathbf{x}_1) + \frac{q_2(\mathbf{z})}{Q(\mathbf{z})} T(\mathbf{x}_2) \right) \\ =& Q(\mathbf{z}) \left(\alpha(\mathbf{z}) T(\mathbf{x}_1) + (1 - \alpha(\mathbf{z})) T(\mathbf{x}_2) \right) \\ =& Q(\mathbf{z}) \overline{T}_\alpha(\mathbf{z}). \end{aligned}$$

Substituting the above equality into the reconstruction objective gives

$$\begin{aligned} & \sum_{i=1}^{2} \mathbb{E}_{q_i}[-\log p_\theta(\mathbf{x}_i|\mathbf{z})] \\ =& \int Q(\mathbf{z}) \left(A(\eta(\mathbf{z})) - \left\langle \overline{T}_\alpha(\mathbf{z}), \eta(\mathbf{z}) \right\rangle \right) d\mathbf{z} - \sum_{i=1}^{2} \log h(\mathbf{x}_i). \end{aligned}$$

Recall the convex conjugate A^* and Fenchel–Young inequality (Fenchel, 2013):

$$A^*(y) := \sup_{\eta}\{\langle y, \eta\rangle - A(\eta)\}, \qquad A(\eta) - \langle y, \eta\rangle \geq -A^*(y),$$

with equality when $y \in \partial A(\eta)$; in the regular (Legendre) case, A is essentially smooth and strictly convex, so $y = \nabla A(\eta)$ is the unique equality condition and A^* is strictly convex on its (convex) effective domain.

Because the integrand is separable in $\mathbf{z}$ and $Q(\mathbf{z}) \geq 0$, minimizing the integral over all measurable $\eta(\cdot)$ reduces to pointwise minimization:

$$\begin{aligned}\min_{\eta(\cdot)} \int Q(\mathbf{z})\Big(A(\eta(\mathbf{z})) - \langle \overline{T}_\alpha(\mathbf{z}), \eta(\mathbf{z})\rangle\Big)\, d\mathbf{z} &= \int Q(\mathbf{z}) \min_{\eta}\big\{A(\eta) - \langle \overline{T}_\alpha(\mathbf{z}), \eta\rangle\big\}\, d\mathbf{z} \\ &= -\int Q(\mathbf{z})\, A^*\big(\overline{T}_\alpha(\mathbf{z})\big)\, d\mathbf{z}.\end{aligned}$$

Hence

$$\min_{\eta(\cdot)} \sum_{i=1}^{2} \mathbb{E}_{q_i}\big[-\log p_\theta(\mathbf{x}_i \mid \mathbf{z})\big] = -\int Q(\mathbf{z})\, A^*\big(\overline{T}_\alpha(\mathbf{z})\big)\, d\mathbf{z} - \sum_{i=1}^{2} \log h(\mathbf{x}_i).$$

(When $\overline{T}_\alpha(\mathbf{z})$ lies in the interior of $\mathrm{dom}(A^*)$, the minimizer is $\eta^*(\mathbf{z}) = \nabla A^*(\overline{T}_\alpha(\mathbf{z}))$; equivalently, $\nabla A(\eta^*(\mathbf{z})) = \overline{T}_\alpha(\mathbf{z})$.)

Add and subtract the quantity

$$\int Q(\mathbf{z})\Big(\alpha(\mathbf{z})\, A^*(T(\mathbf{x}_1)) + (1-\alpha(\mathbf{z}))\, A^*(T(\mathbf{x}_2))\Big)\, d\mathbf{z} = \sum_{i=1}^{2} A^*(T(\mathbf{x}_i)) \underbrace{\int q_i(\mathbf{z})\, d\mathbf{z}}_{=1},$$

and collect the terms independent of $\eta(\cdot)$ into the constant

$$C := -\sum_{i=1}^{2}\big(\log h(\mathbf{x}_i) + A^*(T(\mathbf{x}_i))\big).$$

Using Eq. (20), we obtain

$$\begin{aligned}\min_{\eta(\cdot)} \sum_{i=1}^{2} \mathbb{E}_{q_i}\big[-\log p_\theta(\mathbf{x}_i \mid \mathbf{z})\big] &= C + \int Q(\mathbf{z})\Big(\alpha(\mathbf{z})\, A^*(T(\mathbf{x}_1)) + (1-\alpha(\mathbf{z}))\, A^*(T(\mathbf{x}_2)) - A^*\big(\overline{T}_\alpha(\mathbf{z})\big)\Big)\, d\mathbf{z} \\ &= C + \int Q(\mathbf{z})\, \Delta_{A^*}(\mathbf{x}_1, \mathbf{x}_2; \alpha(\mathbf{z}))\, d\mathbf{z},\end{aligned}$$

where

$$\Delta_{A^*}(\mathbf{x}_1, \mathbf{x}_2; \alpha) := \alpha\, A^*(T(\mathbf{x}_1)) + (1-\alpha)\, A^*(T(\mathbf{x}_2)) - A^*(\overline{T}_\alpha).$$

By convexity of A^*, $\Delta_{A^*}(\mathbf{x}_1, \mathbf{x}_2; \alpha) \geq 0$ for all $\alpha \in [0, 1]$ (Jensen gap). In the regular (Legendre) case, A^* is strictly convex on its effective domain, so $\Delta_{A^*}(\mathbf{x}_1, \mathbf{x}_2; \alpha) = 0$ iff either

- $T(\mathbf{x}_1) = T(\mathbf{x}_2)$, in which case the three arguments of A^* coincide, or
- $\alpha \in \{0, 1\}$, i.e. $Q(\mathbf{z})\alpha(\mathbf{z})\big(1-\alpha(\mathbf{z})\big) = 0$ for Q-a.e. $\mathbf{z}$. Equivalently, the posteriors have disjoint supports w.r.t. the measure $Q(\mathbf{z})\, d\mathbf{z}$ (on each point with $Q > 0$ exactly one of q_1, q_2 is zero).

(If A^* were affine on the segment $[T(\mathbf{x}_1), T(\mathbf{x}_2)]$, equality could also occur with $0 < \alpha < 1$, but strict convexity rules this out unless $T(\mathbf{x}_1) = T(\mathbf{x}_2)$.) □

D.3 PROOF OF EQUATION 7

Theorem D.4. *To maximize $\mathcal{L}_{ELBO}(\mathbf{x})$, $\mathrm{KL}(q(c|\mathbf{x})||p(c|\mathbf{z})) \equiv 0$ should be satisfied. As a result, we use the following equation to compute $q(c|\mathbf{x})$:*

$$q(c|\mathbf{x}) = p(c|\mathbf{z}) \equiv \frac{p(c)p(\mathbf{z}|c)}{\sum_{c'=1}^{K} p(c')p(\mathbf{z}|c')} \tag{23}$$

Proof. Specifically, $\mathcal{L}_{\mathrm{ELBO}}(\mathbf{x})$ can be rewritten as:

$$\begin{aligned}
\mathcal{L}_{\mathrm{ELBO}}(\mathbf{x}) &= E_{q(\mathbf{z},c|\mathbf{x})}\left[\log \frac{p(\mathbf{x},\mathbf{z},c)}{q(\mathbf{z},c|\mathbf{x})}\right] \\
&= \int_{\mathbf{z}} \sum_{c} q(c|\mathbf{x})q(\mathbf{z}|\mathbf{x}) \left[\log \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{q(\mathbf{z}|\mathbf{x})} + \log \frac{p(c|\mathbf{z})}{q(c|\mathbf{x})}\right] d\mathbf{z} \\
&= \int_{\mathbf{z}} q(\mathbf{z}|\mathbf{x}) \log \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{q(\mathbf{z}|\mathbf{x})} d\mathbf{z} - \int_{\mathbf{z}} q(\mathbf{z}|\mathbf{x})\, \mathrm{KL}(q(c|\mathbf{x})||p(c|\mathbf{z}))d\mathbf{z}
\end{aligned} \tag{24}$$

In Equation 24, the first term has no relationship with c and the second term is non-negative. Hence, to maximize $\mathcal{L}_{\mathrm{ELBO}}(\mathbf{x})$, $\mathrm{KL}(q(c|\mathbf{x})||p(c|\mathbf{z})) \equiv 0$ should be satisfied. As a result, we use the following equation to compute $q(c|\mathbf{x})$ in VaDE:

$$q(c|\mathbf{x}) = p(c|\mathbf{z}) \equiv \frac{p(c)p(\mathbf{z}|c)}{\sum_{c'=1}^{K} p(c')p(\mathbf{z}|c')} \tag{25}$$

By using Equation 25, the information loss induced by the mean-field approximation can be mitigated, since $p(c|\mathbf{z})$ captures the relationship between c and $\mathbf{z}$. It is worth noting that $p(c|\mathbf{z})$ is only an approximation to $q(c|\mathbf{x})$, and we find it works well in practice. □

D.4 PROOF OF EQUATION 10

Theorem D.5. *By substituting the terms in Equation 9 with Equations 4, 5, 6 and 11, and using the SGVB estimator and the reparameterization trick, the $\mathcal{L}_{ELBO}(\mathbf{x})$ can be rewritten as:*

$$\begin{aligned}
\mathcal{L}_{ELBO}(\mathbf{x}) = &\frac{1}{L}\sum_{l=1}^{L}\sum_{i=1}^{D} x_i \log \pi_i(\mathbf{z}^{(l)}) \\
&- \frac{1}{2}\sum_{c=1}^{K}\gamma_c \sum_{j=1}^{J}(\log \boldsymbol{\sigma}_c^2|_j + \frac{\tilde{\boldsymbol{\sigma}}^2|_j}{\boldsymbol{\sigma}_c^2|_j} + \frac{(\tilde{\boldsymbol{\mu}}|_j - \boldsymbol{\mu}_c|_j)^2}{\boldsymbol{\sigma}_c^2|_j}) \\
&+ \sum_{c=1}^{K}\gamma_c \log \frac{\pi_c}{\gamma_c} + \frac{1}{2}\sum_{j=1}^{J}(1 + \log \tilde{\boldsymbol{\sigma}}^2|_j)
\end{aligned} \tag{26}$$

where L is the number of Monte Carlo samples in the SGVB estimator, D is the dimensionality of $\mathbf{x}$ and $\boldsymbol{\mu}_x^{(l)}$, x_i is the ith element of $\mathbf{x}$, J is the dimensionality of $\boldsymbol{\mu}_c$, $\boldsymbol{\sigma}_c^2$, $\tilde{\boldsymbol{\mu}}$ and $\tilde{\boldsymbol{\sigma}}^2$, and $*|_j$ denotes the jth element of $*$, K is the number of clusters, π_c is the prior probability of cluster c, and γ_c denotes $q(c|\mathbf{x})$ for simplicity.

Proof. To simplify the computation, we adopt a mean-field variational posterior $q_\phi(\mathbf{z}, c|\mathbf{x})$ and factorize it as:

$$q_\phi(\mathbf{z}, c|\mathbf{x}) = q_\phi(\mathbf{z}|\mathbf{x})q_\phi(c|\mathbf{x}). \tag{27}$$

Under the model formulation in Section 3.1, the observed interaction $\mathbf{x}$ is generated from the latent representation $\mathbf{z}$ through the decoder $p_\theta(\mathbf{x}|\mathbf{z})$. Given $\mathbf{z}$, the preference component c provides no

additional information about $\mathbf{x}$. Thus, $\mathbf{x}$ is conditionally independent of c given $\mathbf{z}$, i.e., $p(\mathbf{x}|\mathbf{z}, c) = p_\theta(\mathbf{x}|\mathbf{z})$. By the chain rule of probability, the joint distribution factorizes as:

$$p(\mathbf{x}, \mathbf{z}, c) = p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z}|c)p(c). \tag{28}$$

The $\mathcal{L}_{\text{ELBO}}(\mathbf{x})$ can be rewritten as:

$$\begin{aligned}\mathcal{L}_{\text{ELBO}}(\mathbf{x}) &= E_{q_\phi(\mathbf{z},c|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z}, c)}{q_\phi(\mathbf{z}, c|\mathbf{x})}\right] \\ &= E_{q_\phi(\mathbf{z},c|\mathbf{x})}\left[\log p_\theta(\mathbf{x}, \mathbf{z}, c) - \log q_\phi(\mathbf{z}, c|\mathbf{x})\right] \\ &= E_{q_\phi(\mathbf{z},c|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z}) + \log p_\theta(\mathbf{z}|c) \\ &\quad + \log p_\theta(c) - \log q_\phi(\mathbf{z}|\mathbf{x}) - \log q_\phi(c|\mathbf{x})]\end{aligned}$$

- $\mathbb{E}_{q(\mathbf{z},c|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})]$

$$\begin{aligned}\mathbb{E}_{q(\mathbf{z},c|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] \\ &\approx \frac{1}{L}\sum_{l=1}^{L} \log p(\mathbf{x}|\mathbf{z}^{(l)}) \\ &\overset{c}{=} \frac{1}{L}\sum_{l=1}^{L}\sum_{i=1}^{D} x_i \log \pi_i(\mathbf{z}^{(l)}).\end{aligned}$$

L is the number of Monte Carlo samples in the SGVB estimator and can be set to 1. D is the dimensionality of $\mathbf{x}$, x_i is the ith element of $\mathbf{x}$. However, directly sampling latent variables from $q_\phi(\mathbf{z}|\mathbf{x})$ breaks gradient propagation to the inference network. Therefore, we employ the reparameterization trick and rewrite the sampling process as:

$$\mathbf{z}^{(l)} = \tilde{\boldsymbol{\mu}} + \tilde{\boldsymbol{\sigma}} \circ \boldsymbol{\epsilon}^{(l)} \quad \text{and} \quad \boldsymbol{\epsilon}^{(l)} \sim \mathcal{N}(0, \mathbf{I}) \tag{29}$$

where $\circ$ denotes the element-wise product.

- $\mathbb{E}_{q(z,c|x)}[\log p(z|c) - \log q(z|x)]$

$$\begin{aligned}&\mathbb{E}_{q(z,c|x)}[\log p(z|c) - \log q(z|x)] \\ &= \mathbb{E}_{q(c|x)}\left[\mathbb{E}_{q(z|x)}[\log p(z|c) - \log q(z|x)]\right] \\ &= \mathbb{E}_{q(c|x)}\left[-\text{KL}(q(z|x)\|p(z|c))\right]\end{aligned}$$

The variational posterior and the cluster-dependent prior are modeled as diagonal Gaussian distributions:

$$q_\phi(z|x) = \mathcal{N}(\mu_x, \text{diag}(\sigma_x^2)),$$

$$p(z|c) = \mathcal{N}(\mu_c, \text{diag}(\sigma_c^2)),$$

where λ_c denotes the variance of the Gaussian distribution associated with cluster c. Since both distributions are diagonal Gaussian distributions, the KL divergence regularization has the following closed-form expression:

$$\text{KL}(q(z|x)\|p(z|c)) = \frac{1}{2}\sum_{j=1}^{J}\left(\log \frac{\sigma_{c,j}^2}{\sigma_{x,j}^2} + \frac{\sigma_{x,j}^2 + (\mu_{x,j} - \mu_{c,j})^2}{\sigma_{c,j}^2} - 1\right)$$

Furthermore, $q(c \mid x) = \gamma_c$, where γ_c denotes the posterior responsibility of sample x belonging to the c-th cluster; therefore:

$$
\begin{aligned}
& - \mathbb{E}_{q(c|x)}[\mathrm{KL}(q(z|x)||p(z|c))] \\
= & - \sum_{c=1}^{K} \gamma_c \, \mathrm{KL}(q(z|x)||p(z|c)) \\
= & - \frac{1}{2} \sum_{c=1}^{K} \gamma_c \sum_{j=1}^{J} \left(\log \frac{\sigma_{c,j}^2}{\sigma_{x,j}^2} + \frac{\sigma_{x,j}^2 + (\mu_{x,j} - \mu_{c,j})^2}{\sigma_{c,j}^2} - 1 \right)
\end{aligned}
$$

J is the dimensionality of $\boldsymbol{\mu}_c$, $\boldsymbol{\sigma}_c^2$, $\tilde{\boldsymbol{\mu}}$ and $\tilde{\boldsymbol{\sigma}}^2$, and $*|_j$ denotes the jth element of $*$, K is the number of clusters, π_c is the prior probability of cluster c, and γ_c denotes $q(c|\mathbf{x})$ for simplicity.

- $\mathbb{E}_{q(z,c|x)}[\log p_\theta(c) - \log q_\phi(c|\mathbf{x})]$

$$
\begin{aligned}
& \mathbb{E}_{q(z,c|x)}[\log p_\theta(c) - \log q_\phi(c|x)] \\
= & \sum_{c=1}^{K} \int q(z, c|x)[\log p_\theta(c) - \log q_\phi(c|x)]dz \\
= & \sum_{c=1}^{K} \int q(z|c, x)q(c|x)[\log p_\theta(c) - \log q_\phi(c|x)]dz \\
= & \sum_{c=1}^{K} q(c|x)[\log p_\theta(c) - \log q_\phi(c|x)] \underbrace{\int q(z|c, x)dz}_{=1} \\
= & \sum_{c=1}^{K} q(c|x) \left[\log p_\theta(c) - \log q_\phi(c|x)\right] \\
= & \sum_{c=1}^{K} q(c|x) \log \frac{p_\theta(c)}{q_\phi(c|x)} \\
= & \sum_{c=1}^{K} \gamma_c \log \frac{\pi_c}{\gamma_c}
\end{aligned}
$$

□