CONDITIONAL GAUSSIAN MIXTURE DENOISING FOR CROSS-DOMAIN SEQUENTIAL RECOMMENDATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Cross-domain sequential recommendation is widely studied for alleviating the cold-start problem via knowledge transfer. Existing methods usually derive the user's noisy domain-shared multi-interest (NDI) from multi-domain interactions, with diffusion as an effective denoising strategy for adapting NDI to the target domain. However, this diffusion denoising process is initialized with a fixed, user-agnostic Gaussian prior, which may be far from the user's NDI, resulting in an excessively long denoising path. To address this, we propose a personalized prior initialization strategy, which enables efficient user adaptation by learning a user-specific NDI distribution as the denoising starting point. Specifically, a domain-shared preference modeling strategy employs a codebook-based Gaussian mixture distribution to capture the user's NDI, with a pairwise proximity regularizer preventing the mixture components from being excessively separated. Building on this, a target-domain adaptation mechanism reformulates the noising and denoising processes under the learned NDI prior, with a domain-specific log-density condition steering the denoising trajectory toward the target domain. Experiments on three real-world datasets show that our model converges in substantially fewer denoising steps than a variant using the standard Gaussian prior, without relying on step-skipping samplers, while also improving recommendation accuracy.

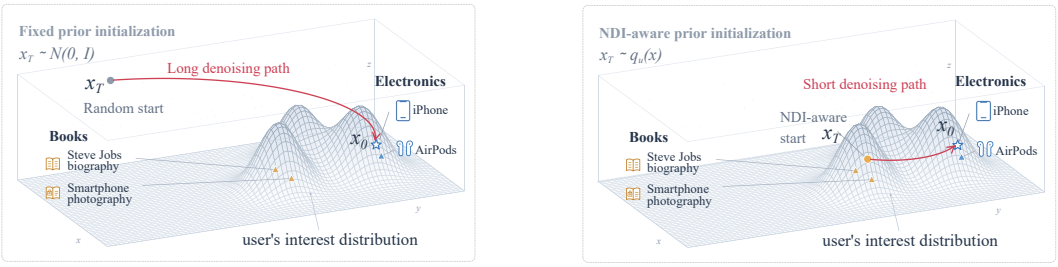
1 INTRODUCTION

Recommender systems are widely used in various online platforms, such as Taobao and Amazon. However, in real-world scenarios, many users interact with only a small fraction of available items, leading to the cold-start problem (Liu et al., 2021; Volkovs et al., 2017). Cross-Domain Recommendation (CDR (Zhang et al., 2023; Li et al., 2025a; Yang et al., 2024)) alleviates this issue by transferring knowledge from a source domain to improve recommendation performance in a target domain. Building upon CDR, Cross-Domain Sequential Recommendation (CDSR (Ning et al., 2025)) further incorporates sequential dependency modeling into the knowledge transfer process.

Most existing CDSR methods (Bian et al., 2025; Li et al., 2025a; Ning et al., 2025) exploit overlapping users as a bridge for knowledge transfer across domains, where the embedding-and-mapping paradigm and the unified distribution paradigm are two commonly adopted strategies. The embedding-and-mapping paradigm (Man et al., 2017; Sun et al., 2023) learns a unified cross-domain mapping function. However, as noted by Cao et al. (2022b), it assumes that all users share an identical mapping function, which often does not hold in real-world scenarios. In contrast, the unified distribution paradigm (Bian et al., 2025; Li et al., 2025a) learns a shared interest distribution by jointly modeling multi-domain interactions, and uses domain-adapters to extract transferable features for improving target-domain performance. Nevertheless, Zhao et al. (2023) point out that users may exhibit distinct or even conflicting preferences across domains, which introduces noise into the shared interest distribution and yields a noisy domain-shared multi-interest (NDI). To mitigate such noise, Li et al. (2025a) employ domain-conditioned diffusion as domain-adapters to extract domain-transferable features, achieving state-of-the-art performance.

However, conventional diffusion models (e.g., CD (Li et al., 2025a)) start denoising from a fixed, user-agnostic Gaussian prior $\mathcal{N}(0, I)$. Such a prior may differ greatly from the conditional data distribution, which makes denoising harder (Lee et al., 2022). As illustrated in Figure 1(a), a user-agnostic starting point can be far from the user's interest region, leading to a long denoising path.

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.



(a) User-agnostic Gaussian prior (b) User-specific NDI prior

Figure 1: Comparison of denoising initialization strategies.

JiT (Li & He, 2025) changes the prediction target (clean data instead of noise) but leaves the starting distribution unchanged, so the two design choices are complementary. In other words, a better prediction target alone cannot shorten a denoising path whose starting point is far from the user's interests. Moreover, user interactions usually concentrate around several latent interest topics. For example, as shown in Figure 1(b), interactions with a Steve Jobs biography, smartphone-photography books, and AirPods may reflect several related interests of the same user. Therefore, the denoising starting point should be drawn from a user-specific, multi-peaked interest distribution. To achieve this, we face two challenges.

Approximation of domain-shared multi-interest. Existing studies (Ning et al., 2025; Li et al., 2025b) typically model a user's NDI with a single Gaussian distribution, which is limited in capturing multiple interest patterns. For example, a user who reads technology biographies and buys audio devices exhibits distinct interest modes that a single Gaussian would blur into one. Gaussian mixture distributions (Dilokthanakul et al., 2016) can better represent such interests, but sparse user interactions make it difficult to learn distinguishable interest components. This motivates exploiting informative interaction patterns shared across users to support distribution estimation. Meanwhile, denoising may start from any component of a user's mixture distribution. If these components are too far apart, the sampled starting point may be distant from the target interest representation. The key challenge is therefore to exploit globally shared interaction patterns to learn personalized Gaussian mixture distributions whose components remain distinguishable without being excessively separated.

Rethinking denoising under a learnable Gaussian-mixture prior. Conventional diffusion denoising (Li et al., 2025a; Gong et al., 2023; Fang et al., 2024) samples from a fixed standard Gaussian prior and cannot accommodate a learnable Gaussian mixture prior. Since the learned prior is neither zero-mean nor isotropic, the closed-form marginal and posterior of standard diffusion no longer hold, so the noising and denoising processes must be redesigned to support mixture-based sampling, so that denoising starts from points encoding each user's mixture of domain-shared interests. For target-domain adaptation, the denoising model can follow JiT's clean-data prediction strategy (Li & He, 2025) and directly predict target-domain features, adapting domain-shared interests to target-domain preferences. However, a user's domain-shared interest components differ in relevance to the prediction target. Since the density of a noisy sample under the user's target-domain distribution, relative to that under the source-domain distribution, reflects how target-relevant the sample is, this density contrast can serve as a conditioning signal, which must also remain well-defined for users without target-domain history. The core challenge is thus to design a density-based domain-specific condition for mixture denoising that remains effective under sparse target-domain interactions.

Our method. To address these challenges, we propose a conditional **G**aussian **M**ixture **D**enoising (GMD) framework for CDSR, whose core idea is personalized prior initialization, i.e., learning a user-specific NDI distribution as the denoising starting point. GMD consists of two designs. **(1) Domain-shared preference modeling strategy.** To better approximate the user's noisy domain-shared multi-interest (NDI), we learn a codebook-based Gaussian mixture distribution from the user's multi-domain interactions, while a pairwise proximity regularizer prevents the mixture components from being excessively separated. **(2) Target-domain adaptation mechanism.** Instead of starting from pure Gaussian noise as in conventional diffusion models, we reformulate the controlled noise injection and reverse denoising processes under the learned NDI prior, so that denoising starts from informative NDI-aware samples. To guide the denoising trajectory toward the target domain, we further construct a domain-specific condition based on the log-density difference between the domain-specific features and the denoising samples.

2 Related Work

Cross-domain recommendation. Existing CDR methods (Liu et al., 2021; Yang et al., 2024) typically employ rich source-domain user interactions and overlapping users to alleviate target-domain data sparsity via knowledge transfer. Mapping-based methods such as MAP (Man et al., 2017) and MIT (Sun et al., 2023) learn cross-domain mappings of user representations, but assume a shared mapping for all users (Cao et al., 2022b). DM (Li et al., 2025b) instead learns a unified latent distribution via diffusion, while CD (Li et al., 2025a) employs conditional diffusion to recover shared representations from noise in a progressive manner.

Cross-domain sequential recommendation. CDSR extends CDR by modeling user interaction sequences across domains. Early work such as C2DSR (Cao et al., 2022a) leveraged graph neural networks to capture cross-domain transition patterns, while ASR (Ju et al., 2025) further enhanced sequence alignment via revisiting self-attention. ABXI (Bian et al., 2025) integrated domain-specific and LoRA modules to balance personalized and shared preferences, and I2VAE (Ning et al., 2025) modeled sequences as distributions with mutual information regularization to filter noise. Similar to the above work, we focus on ID-level cross-domain recommendation based solely on interaction IDs (Bian et al., 2025; Ju et al., 2025), making our approach orthogonal to LLM-based semantic recommendation methods. HzRec (Zha et al., 2026) fuses triple preferences via dual-oriented diffusion for CDSR but, like CD and DM, starts denoising from a user-agnostic prior.

A more detailed discussion of related diffusion priors is provided in Appendix D.

3 Problem Definition

In cross-domain sequential recommendation (CDSR), we consider a source domain A and a target domain B, defined as $D^A = \{\mathcal{U}^A, \mathcal{S}^A\}$ and $D^B = \{\mathcal{U}^B, \mathcal{S}^B\}$, where $\mathcal{U}$ denotes the user set and $\mathcal{S}$ represents the corresponding interaction sequences. Let $\mathcal{I}^A$ and $\mathcal{I}^B$ denote the item sets in the source and target domains, respectively. We denote the union of interaction sequences across domains as $\mathcal{S}^X = \mathcal{S}^A \cup \mathcal{S}^B$. In CDSR, two types of recommendation tasks are considered:

Few-interaction target-domain recommender (FR): For user u with limited historical interactions in the target domain ($\mathcal{S}_u^B \neq \varnothing$), the model recommends target-domain items $i_u^B \in \mathcal{I}^B$ based on $\mathcal{S}_u^A$, $\mathcal{S}_u^B$, and $\mathcal{S}_u^X$.
No-interaction target-domain recommender (NR): For user u with no historical interactions in the target domain ($\mathcal{S}_u^B = \varnothing$), the model predicts target-domain items $i_u^B \in \mathcal{I}^B$ based on $\mathcal{S}_u^A$ and $\mathcal{S}_u^X$.

4 Methodology

Now we introduce our GMD model for CDSR tasks, whose overall architecture is illustrated in Fig. 2. It consists of two modules: (1) domain-shared and domain-specific preference modeling (Section 4.1), which learns the user-specific Gaussian mixture prior and the domain-specific distributions used for conditioning; and (2) target-domain adaptation (Section 4.2), which performs Gaussian-mixture denoising under the domain-specific condition.

4.1 Domain-Shared and Domain-Specific Preference Modeling

To construct a global shared interest space (i.e., codebook), we employ two sequence masking prediction strategies. (M1) For a user u, mask only items in domain A (or B) of S_u^X to obtain $\widetilde{S}_u^{X_A}$ (or $\widetilde{S}_u^{X_B}$). (M2) For u, randomly mask items in S_u^X to obtain $\widetilde{S}_u^X$. The masked sequence set is $\widetilde{S}_u = \{\widetilde{S}_u^X, \widetilde{S}_u^{X_A}, \widetilde{S}_u^{X_B}\}$. The masked items serve as prediction targets for pre-training the codebook, defined as $\mathcal{C}^G = \{\mathcal{N}_k^G\}_{k=1}^K$. The codebook is similar to VQ-VAE (Van Den Oord et al., 2017), with the key difference that its entries are Gaussian distributions $\mathcal{N}_k^G$.

Specifically, the input sequence set $S = \{S_u, \widetilde{S}_u\}$, with $S_u = \{S_u^A, S_u^B, S_u^X\}$, where S_u^A and S_u^B are domain-specific and S_u^X is domain-shared, is fed into an encoder network.

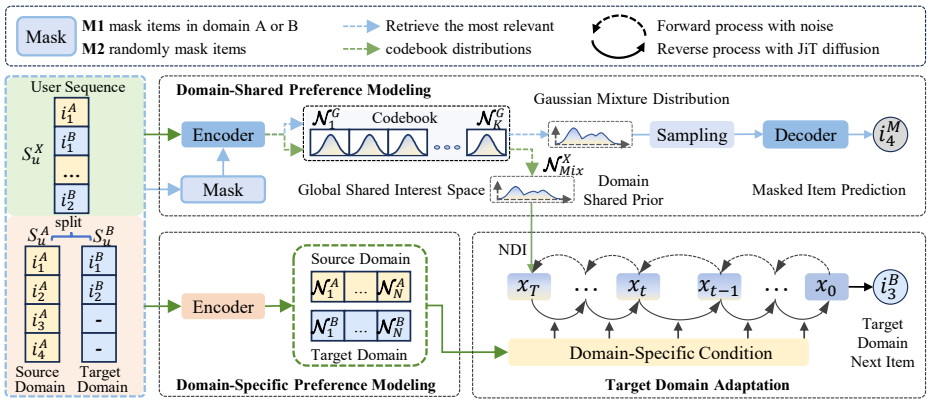


Figure 2: The architecture of the conditional Gaussian mixture denoising model.

$$\mathcal{N}_S = f_{\mathrm{e}}(S) = \{\mathcal{N}_S^A, \mathcal{N}_S^B, \mathcal{N}_S^X, \widetilde{\mathcal{N}}_S^X, \widetilde{\mathcal{N}}_S^{X_A}, \widetilde{\mathcal{N}}_S^{X_B}\} \tag{1}$$

where $\{\mathcal{N}_S^A, \mathcal{N}_S^B\}$ capture **user domain-specific preferences**. $\mathcal{N}_S^X$ consists of N Gaussian components $\{\mathcal{N}_n^X\}_{n=1}^N$, and the others are composed of Gaussian components similarly.

Taking $\mathcal{N}_S^X$ as an example, it is used to index the codebook $\mathcal{C}^G = \{\mathcal{N}_k^G\}_{k=1}^K$ (i.e., **global shared interest space**). The effect of K is reported in Appendix C.2. The indexing operation is defined as:

$$\mathcal{N}_S^{G_X} = \{\mathcal{N}_k^G \mid k = \arg\min_j \mathcal{W}(\mathcal{N}_n^X, \mathcal{N}_j^G)\}_{n=1}^N \tag{2}$$

where $\mathcal{N}_S^{G_X} = \{\mathcal{N}_n^{G_X}\}_{n=1}^N$ denotes the set of codebook components selected from $\mathcal{C}^G$ by indexing with $\mathcal{N}_S^X$. Here, N is the number of interest modes per user, which controls the number of peaks of the prior, whereas model capacity is governed by the codebook size K; both effects are reported in Appendix C.2. $\mathcal{W}(\cdot,\cdot)$ denotes the Wasserstein distance (Ding & Niles-Weed, 2022; Feldman et al., 2024; Lv et al., 2024), which quantifies the similarity between sequence and codebook distributions.

Then, we derive mixture weights from the indexed codebook distributions and sequence distributions.

$$\pi_n^X = \frac{\sum_{m=1}^N \exp\Big(-P_{n,m} \cdot \mathcal{W}(\mathcal{N}_n^{G_X}, \mathcal{N}_m^X)\Big)}{\sum_{n'=1}^N \sum_{m=1}^N \exp\Big(-P_{n',m} \cdot \mathcal{W}(\mathcal{N}_{n'}^{G_X}, \mathcal{N}_m^X)\Big)} \tag{3}$$

where $P_{n,m}$ is learnable and the negative sign turns the distance into a similarity.

$\{\pi_n^X\}_{n=1}^N$ serves as mixture weights that assign greater influence to better-matched codebook distributions, thereby constructing a user-level Gaussian mixture (i.e., **domain-shared preferences**).

$$\mathcal{N}_{Mix}^X = \{\pi_n^X \cdot \mathcal{N}_n^{G_X}\}_{n=1}^N \tag{4}$$

Applying the same procedure to $\{\widetilde{\mathcal{N}}_S^X, \widetilde{\mathcal{N}}_S^{X_A}, \widetilde{\mathcal{N}}_S^{X_B}\}$ as in Eqs. (2-4), we obtain the corresponding indexed Gaussian mixture distributions $\widetilde{\mathcal{N}}_{\mathrm{Mix}} = \{\widetilde{\mathcal{N}}_{\mathrm{Mix}}^X, \widetilde{\mathcal{N}}_{\mathrm{Mix}}^{X_A}, \widetilde{\mathcal{N}}_{\mathrm{Mix}}^{X_B}\}$.

Then, we feed $\widetilde{\mathcal{N}}_{\mathrm{Mix}}$ into the decoder for mask prediction, which can be expressed as:

$$\hat{y}_m = f_{\mathrm{d}}\Big(\phi(\widetilde{\mathcal{N}}_{\mathrm{Mix}})\Big) \tag{5}$$

where $\phi(\cdot)$ denotes Gumbel-Softmax sampling, a standard differentiable relaxation (Potapczynski et al., 2020) for selecting mixture components.

We use S_u^L to denote the original items at the masked positions of $\widetilde{S}_u$, which serve as the labels corresponding to $\widetilde{S}_u = \{\widetilde{S}_u^X, \widetilde{S}_u^{X_A}, \widetilde{S}_u^{X_B}\}$. Following VQ-VAE (Van Den Oord et al., 2017), the loss

for pre-training the codebook is given by:

$$\mathcal{L}_{\text{pre}} = \|\hat{y}_m - S_u^L\|_2^2 + \|\mathcal{W}(\text{sg}[f_{\text{e}}(\widetilde{S}_u)],\, \mathcal{C}^G[\widetilde{S}_u])\|_2^2 + \|\mathcal{W}(f_{\text{e}}(\widetilde{S}_u),\, \text{sg}[\mathcal{C}^G[\widetilde{S}_u]])\|_2^2 + \frac{2}{N(N-1)} \sum_{i<j} \Big[\|\mu_i^{G_X} - \mu_j^{G_X}\|_2 - (\sigma_i^{G_X} + \sigma_j^{G_X})\Big]_+^2, \tag{6}$$

where $\text{sg}[\cdot]$ denotes the stop-gradient operator, $\mathcal{C}^G[\widetilde{S}_u]$ denotes the set of codebook distributions selected by $\widetilde{S}_u$, calculated following Eqs. (1)-(4), and $\mathcal{W}(\cdot,\cdot)$ denotes the Wasserstein distance function. The last term is the **pairwise proximity regularizer**, evaluated over the N components selected for the same user, with $[z]_+ = \max(z, 0)$. It penalizes a pair of components only when the distance between their means exceeds the sum of their standard deviations, i.e., when their high-density regions no longer overlap, thus preventing the mixture components from being excessively separated while still allowing them to encode distinct interests.

4.2 Target Domain Adaptation

4.2.1 Forward Process with Domain-Shared Prior

Unlike DDPM (Ho et al., 2020), whose forward process terminates at (and whose reverse process starts from) an isotropic Gaussian $p(x_T) = \mathcal{N}(x_T; 0, I)$, our approach adopts a user-level Gaussian mixture distribution as a domain-shared prior, which captures the multiple interests a user exhibits across domains (i.e., NDI).

Specifically, the reverse process starts from the user's mixture prior $\mathcal{N}_{Mix}^X$. To keep both processes tractable, we run N parallel component-wise trajectories: the n-th trajectory $x_{t,n}$ terminates at $\mathcal{N}\big(\pi_n^X \mu_n^{G_X}, (\pi_n^X \sigma_n^{G_X})^2 I\big)$, where the mixture weight π_n^X scales the component mean and standard deviation, and the N component-wise predictions are finally combined by Gumbel-Softmax according to $\{\pi_n^X\}$ (Eq. 12). For computational efficiency and to facilitate subsequent derivations, we organize the component-wise parameters in a concatenated form as follows:

$$\mu_{1:N}^{G_X} = \Big[\mu_1^{G_X}, \mu_2^{G_X}, \ldots, \mu_N^{G_X}\Big], \quad \sigma_{1:N}^{G_X} = \Big[\sigma_1^{G_X}, \sigma_2^{G_X}, \ldots, \sigma_N^{G_X}\Big] \tag{7}$$

The DDPM forward is defined as a fixed Markov chain that gradually adds Gaussian noise to the data according to a variance schedule $\beta_1, \ldots, \beta_T$. Using the notation $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^t \alpha_s$, the noise injection of DDPM at each step is given by $q(x_t \mid x_{t-1}) := \mathcal{N}\big(x_t; \sqrt{\alpha_t}\, x_{t-1}, \beta_t I\big)$. Following this variance schedule, the **controlled noise injection** based on $p(x_T)$ is defined as:

$$q(x_t \mid x_{t-1}) := \mathcal{N}\big(\sqrt{\alpha_t} x_{t-1} + (1 - \sqrt{\alpha_t})\pi_{1:N}^X \cdot \mu_{1:N}^{G_X},\ \beta_t (\pi_{1:N}^X \cdot \sigma_{1:N}^{G_X})^2 I\big) \tag{8}$$

where the choice of $1 - \sqrt{\alpha_t}$ is discussed in Appendix B.2. The effect of the number of diffusion steps is reported in Section 5.4. Then, we can derive the following formula (see Appendix B.1):

$$q(x_t \mid x_0) = \mathcal{N}\big(\sqrt{\bar{\alpha}_t}\, x_0 + (1 - \sqrt{\bar{\alpha}_t})\pi_{1:N}^X \cdot \mu_{1:N}^{G_X}, (1 - \bar{\alpha}_t)(\pi_{1:N}^X \cdot \sigma_{1:N}^{G_X})^2 I\big) \tag{9}$$

Thus, we can add sampling noise to x_t according to the formulation $x_t = \sqrt{\bar{\alpha}_t}\, x_0 + (1 - \sqrt{\bar{\alpha}_t})\, \pi_{1:N}^X \cdot \mu_{1:N}^{G_X} + \sqrt{1 - \bar{\alpha}_t}\, \pi_{1:N}^X \cdot \sigma_{1:N}^{G_X}\, \epsilon$, where this process perturbs the clean target $x_0 = [\text{emb}(i_u^B)]^{\oplus N}$, i.e., an N-fold replicated tensor of the ground-truth target item embedding.

4.2.2 Diffusion Model with Domain-Specific Condition

DDPM (Ho et al., 2020) requires training a network to predict noise, i.e., $\epsilon = \epsilon_\theta(x_t, t, c)$. Recently, JiT (Li & He, 2025) argues that directly predicting the noise demands a large model capacity and instead proposes predicting the clean data, i.e., $\hat{x}_0 = \text{x}_\theta(x_t, t, c)$. In this work, we adopt a **JiT-style diffusion** that directly predicts clean representations instead of noise. Besides, we construct a domain-specific condition by employing the probability density of x_t with respect to $\mathcal{N}_S^A = \big\{\mathcal{N}_n^A\big\}_{n=1}^N$ and $\mathcal{N}_S^B = \big\{\mathcal{N}_n^B\big\}_{n=1}^N$ from Eq. (1). $\mathcal{N}_n^{A'}$ and $\mathcal{N}_n^{B'}$ denote the N-fold replicated and concatenated forms of $\mathcal{N}_n^A$ and $\mathcal{N}_n^B$, respectively, ensuring dimensional alignment with x_t. A detailed derivation of the domain-specific condition and its corresponding closed-form expression is presented in Appendix B.3.

The model is then formulated as:

$$c_s = f_c\Big(\big\|_{n=1}^{N}\big[-\log p(x_t \mid \mathcal{N}_n^{A'}, t) \; + \log p(x_t \mid \mathcal{N}_n^{B'}, t)\big]\Big) \tag{10}$$

$$\hat{\mathrm{x}}_0 = \mathrm{x}_\theta(x_t, t, c_s) \tag{11}$$

where $\|$ denotes the concatenation operation, $f_c(\cdot)$ represents a fully connected layer, and $\mathrm{x}_\theta(\cdot)$ adopts the Transformer block architecture of the JiT model (Li & He, 2025). The condition c_s encodes the log-density difference between the domain-specific features (i.e., $\mathcal{N}_S^A$ and $\mathcal{N}_S^B$) and the denoising sample x_t, guiding denoising toward regions of high target-domain density. For NR users, $\mathcal{S}_u^B = \varnothing$, so we set the target-domain term $\log p(x_t \mid \mathcal{N}_n^{B'}, t)$ to zero and c_s reduces to the source-domain term, while the prior $\mathcal{N}_{Mix}^X$, indexed from a codebook learned on both domains, still carries target-domain information. $x_t = \sqrt{\bar{\alpha}_t}\, x_0 + (1-\sqrt{\bar{\alpha}_t})\pi_{1:N}^X \cdot \mu_{1:N}^{G_X} + \sqrt{1-\bar{\alpha}_t}\, \pi_{1:N}^X \cdot \sigma_{1:N}^{G_X}\, \epsilon$. The diffusion timestep t is uniformly sampled from $\{1, \ldots, T\}$. When $t = 1$, $\bar{\alpha}_t \approx 1$ and thus $x_t \approx x_0$. When $t = T$, $\bar{\alpha}_t \approx 0$ and thus $x_T \approx \pi_{1:N}^X \cdot \mu_{1:N}^{G_X} + \pi_{1:N}^X \cdot \sigma_{1:N}^{G_X} \epsilon$.

Each Gaussian component $x_{t,n} \in x_t$ produces a corresponding prediction $\hat{\mathrm{x}}_{0,n} \in \hat{\mathrm{x}}_0$. To enable differentiable stochastic selection over mixture components, we follow Potapczynski et al. (2020) and adopt Gumbel-Softmax (GS) sampling, with details provided in Appendix B.5, defined as:

$$\hat{\mathrm{x}}_{pre} = \frac{\sum_{n=1}^{N}[\exp\big(\log \pi_n^X + g_n\big) \cdot \hat{\mathrm{x}}_{0,n}]}{\sum_{m=1}^{N} \exp(\log \pi_m^X + g_m)}, \quad g_n, g_m \sim \mathrm{Gumbel}(0, 1) \tag{12}$$

Next, following JiT (Li & He, 2025), the loss function for diffusion training is formulated as:

$$\mathcal{L}_D = \frac{1}{1-\bar{\alpha}_t} \left\|\mathrm{emb}(i_u^B) - \hat{\mathrm{x}}_{pre}\right\|_2^2 \tag{13}$$

$$\mathcal{L} = \mathcal{L}_D + \mathcal{L}_I, \; \mathcal{L}_I = -\log \frac{\exp(\hat{\mathrm{x}}_{pre} \cdot e^+)}{\sum_{e \in \mathcal{E}} \exp(\hat{\mathrm{x}}_{pre} \cdot e)} \tag{14}$$

Eq. (14) defines the overall optimization objective of our model, which is constructed based on the widely adopted InfoNCE loss (Bian et al., 2025) for CDSR. Here, $\mathcal{E} = \{e^+, e^-\}$ denotes the set of positive and negative item embeddings, where e^+ corresponds to the embedding of the ground-truth target item, i.e., $\mathrm{emb}(i_u^B)$, and e^- represents negative item embeddings randomly sampled from interactions of other users (additional details are provided in Appendix B.5). The training procedure is summarized in Algorithm 1 (Appendix A).

4.2.3 Reverse Process with Domain-Shared Prior

During sampling, the neural reverse process $p_\theta(x_{t-1} \mid x_t)$ approximates the true posterior by first predicting the clean signal $\hat{x}_0 = \mathrm{x}_\theta(x_t, t, c_s)$, and then substituting it into the tractable conditional $q(x_{t-1} \mid x_t, x_0)$. Specifically, by Bayes' rule, the posterior factorizes as:

$$p_\theta(x_{t-1} \mid x_t) \;\approx\; q(x_{t-1} \mid x_t, x_0)\big|_{x_0=\hat{x}_0} \;=\; \left.\frac{q(x_t \mid x_{t-1}, x_0)\, q(x_{t-1} \mid x_0)}{q(x_t \mid x_0)}\right|_{x_0=\hat{x}_0} \tag{15}$$

$$q(x_{t-1} \mid x_t, \hat{x}_0) \sim \mathcal{N}\Big(a(x_t + \frac{\beta_t\sqrt{\bar{\alpha}_t}}{\hat{\alpha}_t}\hat{x}_0) + b_{1:N}, \;\; c_{1:N} \cdot I\Big) \tag{16}$$

By substituting Eq. (8), $q(x_t|\hat{x}_0) = \mathcal{N}\big(x_t; \sqrt{\bar{\alpha}_t}\,\hat{x}_0 + (1-\sqrt{\bar{\alpha}_t})\pi_{1:N}^X \cdot \mu_{1:N}^{G_X}, (1-\bar{\alpha}_t)\,(\pi_{1:N}^X \cdot \sigma_{1:N}^{G_X})^2 I\big)$ and $q(x_{t-1}|\hat{x}_0) = \mathcal{N}\big(x_{t-1}; \sqrt{\bar{\alpha}_{t-1}}\,\hat{x}_0 + (1-\sqrt{\bar{\alpha}_{t-1}})\,\pi_{1:N}^X \cdot \mu_{1:N}^{G_X}, (1-\bar{\alpha}_{t-1})\,(\pi_{1:N}^X \cdot \sigma_{1:N}^{G_X})^2 I\big)$ into Eq. (15), together with the notation $\hat{\alpha}_t := \alpha_t - \bar{\alpha}_t$, the posterior distribution can be rewritten into the closed-form formulation shown in Eq. (16). Specifically, the corresponding coefficients are defined as $a = \frac{\hat{\alpha}_t}{\sqrt{\alpha_t}(1-\bar{\alpha}_t)}$, $b_{1:N} = \frac{(1-\sqrt{\alpha_t})(1-\sqrt{\bar{\alpha}_{t-1}})}{1+\sqrt{\bar{\alpha}_t}}\,\pi_{1:N}^X \cdot \mu_{1:N}^{G_X}$, and $c_{1:N} = \frac{\hat{\alpha}_t\beta_t(\pi_{1:N}^X \cdot \sigma_{1:N}^{G_X})^2}{\alpha_t(1-\bar{\alpha}_t)}$. The detailed derivation procedure is provided in Appendix B.4. Based on the obtained posterior formulation, we can further infer x_{t-1} according to:

$$x_{t-1} = a\Big(x_t + \frac{\beta_t\sqrt{\bar{\alpha}_t}}{\hat{\alpha}_t}\mathrm{x}_\theta(x_t, t, c_s)\Big) + b_{1:N} + \sqrt{c_{1:N}} \cdot \epsilon \tag{17}$$

Based on the above formulations, we perform the reverse diffusion sampling process to predict the target item embedding. The reverse sampling is summarized in Algorithm 2 (Appendix A).

Table 1: Overall performance under the FR setting across all datasets. **Bold** values and underlined values denote the best and runner-up results. p-value indicates that, in the t-test, the best method significantly outperforms the runner-up with $p < 0.05$.

Dataset	Metric	VSAN	CVAE	MIT	DM	CD	ASR	ABXI	I2VAE	HzRec	**GMD**	*p-value*
KF-FR	HR@5	0.0955	0.1083	0.1659	0.1843	0.2114	0.1605	0.2133	0.2172	0.2196	**0.2430**	2.16e-5
	HR@10	0.1771	0.1781	0.2493	0.2852	0.3102	0.2629	0.3208	0.3389	0.3475	**0.3634**	4.57e-3
	NG@5	0.0594	0.0729	0.1083	0.1146	0.1263	0.1012	0.1320	0.1363	0.1432	**0.1662**	1.63e-4
	NG@10	0.0855	0.0954	0.1299	0.1451	0.1684	0.1340	0.1675	0.1708	0.1845	**0.2052**	6.84e-6
EB-FR	HR@5	0.0947	0.1110	0.1567	0.1625	0.1727	0.1422	0.1703	0.1875	0.1768	**0.2111**	3.08e-3
	HR@10	0.1426	0.1854	0.2439	0.2552	0.2708	0.2278	0.2571	0.2876	0.2785	**0.3060**	8.34e-4
	NG@5	0.0634	0.0709	0.1002	0.1036	0.1120	0.0960	0.1136	0.1185	0.1087	**0.1384**	6.10e-4
	NG@10	0.0789	0.0945	0.1282	0.1382	0.1473	0.1233	0.1417	0.1509	0.1448	**0.1692**	8.52e-5
BM-FR	HR@5	0.0857	0.0917	0.1420	0.1663	0.1755	0.1328	0.1486	0.1791	0.1785	**0.1994**	5.63e-3
	HR@10	0.1243	0.1673	0.1915	0.2455	0.2433	0.1963	0.2122	0.2525	0.2505	**0.2711**	7.99e-5
	NG@5	0.0438	0.0519	0.1022	0.1164	0.1226	0.0962	0.1018	0.1303	0.1284	**0.1467**	1.32e-4
	NG@10	0.0547	0.0745	0.1181	0.1388	0.1477	0.1167	0.1223	0.1539	0.1493	**0.1696**	1.57e-3

5 EXPERIMENTS

5.1 EXPERIMENTAL SETTINGS

Datasets. Following most existing methods (Sun et al., 2023; Bian et al., 2025; Ning et al., 2025), we adopt three widely used CDSR datasets collected from the Amazon website[1], which are organized into three source–target domain pairs, namely Kitchen → Food (KF), Electronics → Beauty (EB), and Book → Movie (BM).

Baseline methods. To evaluate our model, we compare it with several baseline methods. These include Single-Domain Sequential Recommender (**SDSR**): VSAN (Zhao et al., 2021) and CVAE (Wang et al., 2022); Cross-Domain Recommender (**CDR**): MIT (Sun et al., 2023), DM (Li et al., 2025b), and CD (Li et al., 2025a); Cross-Domain Sequential Recommender (**CDSR**): ASR (Ju et al., 2025), ABXI (Bian et al., 2025), I2VAE (Ning et al., 2025), and HzRec (Zha et al., 2026).

Implementation details. We train GMD with PyTorch, which is accelerated by an NVIDIA Tesla V100. The batch size is set to 1024 for all three datasets, and the learning rate is fixed at 0.001. The dropout rate is set to 0.5. The number of Gaussian components per sequence is set to $N = 3$ for KF and BM and $N = 4$ for EB. The item embedding dimension is 512. The model is optimized using Adam and trained for 200 epochs. We adopt NDCG@k and HR@k as evaluation metrics with $k \in \{5, 10\}$. Code is available at *https://anonymous.4open.science/r/GMD-B5F2*. Further details are given in Appendix C.1.

5.2 OVERALL PERFORMANCE COMPARISON

Table 1 and Table 2 show the results on Kitchen-Food, Electronics-Beauty, and Book-Movie in terms of HR@k and NDCG@k (abbreviated as NG@k). We have the following observations. Firstly, SDSR methods (i.e., VSAN, CVAE) underperform CDR methods (i.e., MIT, DM, CD) and CDSR methods (i.e., ASR, ABXI, I2VAE, HzRec, GMD), likely due to their indiscriminate modeling of all interactions as a single domain, which disregards cross-domain sequence discrepancies and results in suboptimal recommendation performance.

Secondly, among CDR methods designed for NR, unified-distribution paradigms (i.e., DM, CD) generally outperform the embedding-and-mapping strategy (i.e., MIT), likely because the latter assume all users share an identical mapping function, which rarely holds in practice. In contrast, unified-distribution methods avoid this limitation by learning a shared distribution over interactions from multiple domains. Besides, CDSR methods (i.e., ASR, ABXI, I2VAE, GMD) extend traditional CDR approaches by incorporating sequential features. I2VAE outperforms ASR and ABXI, potentially owing to its distributional modeling of user interests.

[1] https://cseweb.ucsd.edu/~jmcauley/datasets/amazon/links.html

Table 2: Overall performance under the NR setting across all datasets (notation as in Table 1).

Dataset	Metric	VSAN	CVAE	MIT	DM	CD	ASR	ABXI	I2VAE	HzRec	**GMD**	*p-value*
KF-NR	HR@5	0.0762	0.0931	0.1177	0.1254	0.1386	0.1126	0.1148	0.1239	0.1283	**0.1570**	3.41e-3
	HR@10	0.1354	0.1447	0.1734	0.1963	0.2115	0.1637	0.1662	0.1847	0.2035	**0.2312**	7.23e-4
	NG@5	0.0527	0.0641	0.0812	0.0816	0.0854	0.0731	0.0760	0.0804	0.0815	**0.1047**	2.85e-3
	NG@10	0.0716	0.0807	0.0992	0.1008	0.1047	0.0957	0.0963	0.0997	0.1033	**0.1287**	4.67e-5
EB-NR	HR@5	0.0203	0.0327	0.0823	0.0939	0.1084	0.0800	0.0927	0.1075	0.0935	**0.1282**	6.09e-4
	HR@10	0.0362	0.0573	0.1418	0.1684	0.1754	0.1458	0.1515	0.1748	0.1642	**0.2001**	1.74e-5
	NG@5	0.0125	0.0186	0.0499	0.0616	0.0694	0.0514	0.0642	0.0651	0.0646	**0.0871**	9.38e-5
	NG@10	0.0176	0.0264	0.0690	0.0855	0.0955	0.0723	0.0878	0.0922	0.0893	**0.1101**	5.52e-4
BM-NR	HR@5	0.0183	0.0228	0.0765	0.0896	0.1027	0.0868	0.0756	0.0993	0.0845	**0.1156**	8.16e-4
	HR@10	0.0262	0.0373	0.1097	0.1320	0.1416	0.1171	0.1083	0.1406	0.1245	**0.1659**	3.93e-4
	NG@5	0.0117	0.0133	0.0465	0.0568	0.0619	0.0512	0.0458	0.0602	0.0531	**0.0736**	2.47e-3
	NG@10	0.0137	0.0169	0.0575	0.0747	0.0806	0.0610	0.0564	0.0798	0.0673	**0.0897**	7.61e-4

Thirdly, for the NR task, CD and I2VAE achieve superior performance compared with ASR and ABXI, as the latter do not explicitly account for cold-user adaptation, which is critical in NR scenarios. HzRec, a diffusion-based CDSR method, achieves the runner-up results on KF-FR, but it is inferior to I2VAE on EB-FR and BM-FR and to CD in all NR settings, since it still starts denoising from a user-agnostic Gaussian prior. In contrast, GMD starts denoising from a user-specific Gaussian mixture prior and steers it with a domain-specific condition; the ablation **w/o P** (Table 3) confirms that this learned prior is a major source of the improvement. As a result, GMD consistently achieves the best performance across both FR and NR tasks, with absolute improvements over the strongest baseline of 1.59%-1.86% in HR@10 and 1.57%-2.07% in NDCG@10 for FR, and 1.97%-2.47% in HR@10 and 0.91%-2.40% in NDCG@10 for NR. In relative terms, the improvements are generally larger for NR (9.31%-25.50%) than for FR (4.58%-16.79%).

5.3 Ablation Study

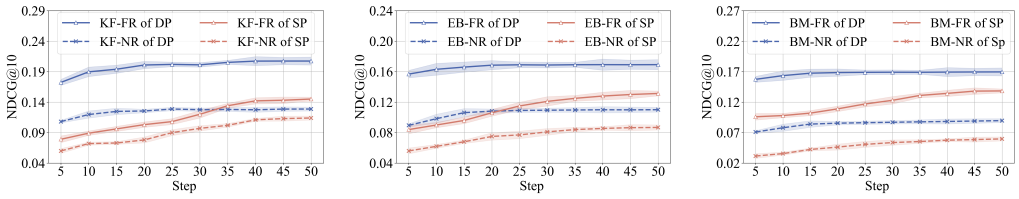
Next, we conduct an ablation study to evaluate the usefulness of key components, including: (1) **w/o M1**: removing mask strategy M1 (domain masking); (2) **w/o M2**: removing mask strategy M2 (random masking); (3) **w/o R**: removing pairwise proximity regularizer; (4) **w/o G**: removing the codebook, the model directly learns the mixed sequence distribution as the prior; (5) **w/o CA**: removing domain-A features from the conditioning; (6) **w/o CB**: removing domain-B features from the conditioning; (7) **w/o C**: removing conditioning mechanisms; (8) **w/o P**: removing the domain-shared prior and using a standard prior instead.

Table 3 reports the results of the ablation study. It is clear that GMD significantly outperforms all variants, indicating that all components are important. Specifically, we can observe that: (1) For domain-shared preference modeling, removing either masking strategy degrades performance, and removing M1 causes the larger drop, indicating that masking one domain and predicting it from the other (M1) provides the main cross-domain transfer signal, while random masking (M2) further strengthens within-domain co-occurrence modeling. Besides, the drop of **w/o R** shows that the pairwise proximity regularizer keeps the mixture components compact, facilitating adaptation from domain-shared to target-domain preferences. Moreover, **w/o G** shows that the shared codebook matches cold-start test users to cross-domain interests learned from overlapping users.

(2) Removing the diffusion condition (**w/o C**) confirms its effectiveness. We further ablate the conditions from source domain A (**w/o CA**) and target domain B (**w/o CB**). On KF-FR, test users have on average fewer than three target-domain interactions versus about 16 source-domain interactions, yet removing the domain-B condition hurts as much as removing the domain-A condition, showing that even very limited target-domain interactions provide an informative condition. For NR, the domain-B term is set to zero, so **w/o CA** coincides with **w/o C**. Finally, **w/o P** verifies that the learnable domain-shared prior (i.e., NDI) is more effective than the standard Gaussian prior (see Appendix C.3 for more prior types). Overall, the absolute drops are generally larger on FR (1.79%-6.18%) than on NR (0.31%-3.00%), as NR users lack target-domain interactions.

Table 3: Ablation results on NDCG@10 across all datasets. FR and NR denote users with few and no interactions in the target domain, respectively.

Dataset	w/o M1	w/o M2	w/o R	w/o G	w/o CA	w/o CB	w/o C	w/o P	**GMD**
KF-FR	0.1724	0.1859	0.1873	0.1846	0.1479	0.1441	0.1434	0.1451	**0.2052**
KF-NR	0.1197	0.1249	0.1256	0.1204	0.1107	-	0.1107	0.1175	**0.1287**
EB-FR	0.1197	0.1324	0.1307	0.1458	0.1370	0.1301	0.1279	0.1314	**0.1692**
EB-NR	0.0861	0.0894	0.0858	0.0984	0.0923	-	0.0923	0.0868	**0.1101**
BM-FR	0.1286	0.1362	0.1478	0.1428	0.1385	0.1317	0.1309	0.1384	**0.1696**
BM-NR	0.0623	0.0642	0.0738	0.0694	0.0673	-	0.0673	0.0597	**0.0897**



(a) Diffusion step on KF (b) Diffusion step on EB (c) Diffusion step on BM

Figure 3: Impact of the number of diffusion steps.

5.4 Impact of the Number of Diffusion Steps

Fig. 3(a-c) compares the domain-shared prior (DP) with the standard prior (SP) on both FR and NR tasks across three benchmark datasets. All other hyperparameters are kept fixed, and each setting is repeated with five random seeds; the curves denote the mean performance, and the shaded bands indicate one standard deviation.

First, DP converges significantly faster in the diffusion process: on the KF dataset, it converges within 20 steps for FR and 15 steps for NR, whereas SP requires approximately 40 steps for both tasks, a pattern consistent across the remaining datasets. DP converges even faster on NR than on FR, as denoising for users without target-domain interactions relies more heavily on the prior. This indicates that DP provides a more informative starting point for denoising, reducing the number of inference steps without sacrificing recommendation performance. Second, DP consistently achieves higher accuracy than SP under identical diffusion configurations. This is because the codebook-based Gaussian mixture prior lets denoising start from a distribution that already carries the user's NDI rather than from a fixed, user-agnostic Gaussian, which improves the effectiveness of cross-domain knowledge transfer. Since DP and SP share all other components, these results confirm that the gain stems from the prior itself rather than from the condition or the x_0-prediction. Owing to the shorter denoising trajectory, GMD also requires less per-item inference time than the diffusion-based baselines DM, CD, and HzRec, as reported in Appendix C.5.

6 Conclusion

In summary, we propose GMD, a conditional Gaussian mixture denoising framework for CDSR that replaces the fixed, user-agnostic diffusion prior with a learned, user-specific Gaussian mixture prior and steers denoising with a domain-specific log-density condition. Experiments on three real-world datasets show that GMD improves recommendation accuracy and converges in substantially fewer denoising steps than a variant using the standard Gaussian prior. Moreover, the shared codebook allows cold-start users without target-domain interactions to benefit from cross-domain interests learned from overlapping users. These results suggest that, for cross-domain recommendation, learning where denoising starts is as important as learning how to denoise. Beyond accuracy, the learned prior shortens the denoising trajectory and thus reduces the inference cost of diffusion-based recommendation. In future work, we will extend GMD to heterogeneous cross-domain scenarios, such as video-music and news-shopping, and further reduce its reliance on overlapping training users.

AI USE STATEMENT

In this work, generative AI tools were used solely for language polishing and improving the readability of the manuscript. They were not used for idea generation, methodology design, mathematical derivation, experimental design, data analysis, or result interpretation. All content was reviewed and verified by the authors.

REFERENCES

Qingtian Bian, Marcus de Carvalho, Tieying Li, Jiaxing Xu, Hui Fang, and Yiping Ke. Abxi: invariant interest adaptation for task-guided cross-domain sequential recommendation. In *WWW*, pp. 3183–3192, 2025.

Denis Blessing, Xiaogang Jia, and Gerhard Neumann. End-to-end learning of gaussian mixture priors for diffusion sampler. In *ICLR*, volume 2025, pp. 58207–58233, 2025.

David Bolin and Jonas Wallin. Efficient methods for gaussian markov random fields under sparse linear constraints. *NeurIPS*, 34:9882–9894, 2021.

Jiangxia Cao, Xin Cong, Jiawei Sheng, Tingwen Liu, and Bin Wang. Contrastive cross-domain sequential recommendation. In *CIKM*, pp. 138–147, 2022a.

Jiangxia Cao, Jiawei Sheng, Xin Cong, Tingwen Liu, and Bin Wang. Cross-domain recommendation to cold-start users via variational information bottleneck. In *ICDE*, pp. 2209–2223. IEEE, 2022b.

Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016.

Yunzi Ding and Jonathan Niles-Weed. Asymptotics of smoothed wasserstein distances in the small noise regime. *NeurIPS*, 35:19203–19214, 2022.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. Remix-dit: Mixing diffusion transformers for multi-expert denoising. *NeurIPS*, 37:107494–107512, 2024.

Vivek Farias, Andrew Li, Tianyi Peng, and Andrew Zheng. Markovian interference in experiments. *NeurIPS*, 35:535–549, 2022.

Vitaly Feldman, Audra McMillan, Satchit Sivakumar, and Kunal Talwar. Instance-optimal private density estimation in the wasserstein distance. *NeurIPS*, 37:90061–90131, 2024.

Mingming Gong, Shaoan Xie, Wei Wei, Matthias Grundmann, Kayhan Batmanghelich, Tingbo Hou, et al. Semi-implicit denoising diffusion models (siddms). *NeurIPS*, 36:17383–17394, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020.

Kai Hu, Ye Xiao, Yuan Zhang, and Xieping Gao. Multi-view masked contrastive representation learning for endoscopic video analysis. *NeurIPS*, 37:47987–48014, 2024.

Nanshan Jia, Tingyu Zhu, Haoyu Liu, and Zeyu Zheng. Structured diffusion models with mixture of gaussians as prior distribution. *arXiv preprint arXiv:2410.19149*, 2024.

Clark Mingxuan Ju, Leonardo Neves, Bhuvesh Kumar, Liam Collins, Tong Zhao, Yuwei Qiu, Qing Dou, Sohail Nizam, Sen Yang, and Neil Shah. Revisiting self-attention for cross-domain sequential recommendation. In *SIGKDD*, pp. 1094–1105, 2025.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

Sang-gil Lee, Heeseung Kim, Chaehun Shin, Xu Tan, Chang Liu, Qi Meng, Tao Qin, Wei Chen, Sungroh Yoon, and Tie-Yan Liu. PriorGrad: Improving conditional denoising diffusion models with data-dependent adaptive prior. In *International Conference on Learning Representations (ICLR)*, 2022.

Hanyu Li, Jiayu Li, Weizhi Ma, Peijie Sun, Haiyang Wu, Jingwen Wang, Yuekui Yang, Min Zhang, and Shaoping Ma. Cd-cdr: Conditional diffusion-based item generation for cross-domain recommendation. In *SIGIR*, pp. 1789–1798, 2025a.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720*, 2025.

Tianxu Li, Kun Zhu, Juan Li, and Yang Zhang. Learning distinguishable trajectory representation with contrastive loss. *NeurIPS*, 37:64454–64478, 2024.

Xiaodong Li, Hengzhu Tang, Jiawei Sheng, Xinghua Zhang, Li Gao, Suqi Cheng, Dawei Yin, and Tingwen Liu. Exploring preference-guided diffusion model for cross-domain recommendation. In *KDD*, pp. 719–728, 2025b.

Weiming Liu, Jiajie Su, Chaochao Chen, and Xiaolin Zheng. Leveraging distribution alignment via stein path for cross-domain cold-start recommendation. *NeurIPS*, 34:19223–19234, 2021.

Jiaming Lv, Haoyuan Yang, and Peihua Li. Wasserstein distance rivals kullback-leibler divergence for knowledge distillation. *NeurIPS*, 37:65445–65475, 2024.

Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. Cross-domain recommendation: An embedding and mapping approach. In *IJCAI*, volume 17, pp. 2464–2470, 2017.

Stefan Matthes, Zhiwei Han, and Hao Shen. Towards a unified framework of contrastive learning for disentangled representations. *NeurIPS*, 36:67459–67470, 2023.

Xuying Ning, Wujiang Xu, Tianxin Wei, and Xiaolei Liu. i^{2}VAE: Interest information augmentation with variational regularizers for cross-domain sequential recommendation. *UAI*, 2025.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Andres Potapczynski, Gabriel Loaiza-Ganem, and John P Cunningham. Invertible gaussian reparameterization: Revisiting the gumbel-softmax. *NeurIPS*, 33:12311–12321, 2020.

Tim Salimans, Diederik Kingma, and Max Welling. Markov chain monte carlo and variational inference: Bridging the gap. In *ICML*, pp. 1218–1226. pmlr, 2015.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, pp. 2256–2265. pmlr, 2015.

Caiqi Sun, Jiewei Gu, BinBin Hu, Xin Dong, Hai Li, Lei Cheng, and Linjian Mo. Remit: reinforced multi-interest transfer for cross-domain recommendation. In *AAAI*, volume 37, pp. 9900–9908, 2023.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *NeurIPS*, 30, 2017.

Maksims Volkovs, Guangwei Yu, and Tomi Poutanen. Dropoutnet: Addressing cold start in recommender systems. *NeurIPS*, 30, 2017.

Yu Wang, Hengrui Zhang, Zhiwei Liu, Liangwei Yang, and Philip S. Yu. Contrastvae: Contrastive variational autoencoder for sequential recommendation. In *CIKM*, pp. 2056–2066, 2022.

Lianghao Xia, Yizhen Shao, Chao Huang, Yong Xu, Huance Xu, and Jian Pei. Disentangled graph social recommendation. In *ICDE*, pp. 2332–2344. IEEE, 2023.

Ziqi Yang, Zhaopeng Peng, Zihui Wang, Jianzhong Qi, Chaochao Chen, Weike Pan, Chenglu Wen, Cheng Wang, and Xiaoliang Fan. Federated graph learning for cross-domain recommendation. *NeurIPS*, 37:64865–64888, 2024.

Xi Yu, Xiang Gu, Haozhi Liu, and Jian Sun. Constructing non-isotropic gaussian diffusion model using isotropic gaussian diffusion model for image editing. *NeurIPS*, 36:76839–76851, 2023.

Zongsheng Yue, Jianyi Wang, and Chen Change Loy. Resshift: Efficient diffusion model for image super-resolution by residual shifting. *NeurIPS*, 36:13294–13307, 2023.

Yongfu Zha, Xinxin Dong, Haokai Ma, Yonghui Yang, and Xiaodong Wang. Align-for-fusion: Harmonizing triple preferences via dual-oriented diffusion for cross-domain sequential recommendation. In *SIGKDD*, pp. 1868–1879, 2026.

Tong Zhang, Chen Chen, Dan Wang, Jie Guo, and Bin Song. A vae-based user preference learning and transfer framework for cross-domain recommendation. *TKDE*, 35(10):10383–10396, 2023.

Yihan Zhang and Nir Weinberger. Mean estimation in high-dimensional binary markov gaussian mixture models. *NeurIPS*, 35:19673–19686, 2022.

Jing Zhao, Pengpeng Zhao, Lei Zhao, Yanchi Liu, Victor S Sheng, and Xiaofang Zhou. Variational self-attention network for sequential recommendation. In *ICDE*, pp. 1559–1570, 2021.

Yi Zhao, Chaozhuo Li, Jiquan Peng, Xiaohan Fang, Feiran Huang, Senzhang Wang, Xing Xie, and Jibing Gong. Beyond the overlapping users: Cross-domain recommendation via adaptive anchor link learning. In *SIGIR*, pp. 1488–1497, 2023.

Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising diffusion bridge models. In *International Conference on Learning Representations*, volume 2024, pp. 8160–8171, 2024.

Jiaming Zhuo, Yintong Lu, Hui Ning, Kun Fu, Bingxin Niu, Dongxiao He, Chuan Wang, Yuanfang Guo, Zhen Wang, Xiaochun Cao, et al. Unified graph augmentations for generalized contrastive learning on graphs. *NeurIPS*, 37:37473–37503, 2024.

Hanna Ziesche and Leonel Rozo. Wasserstein gradient flows for optimizing gaussian mixture policies. *NeurIPS*, 36:21058–21080, 2023.

A Training and Sampling Algorithms

Algorithm 1: Training

Pre-train the codebook $\mathcal{C}^G = \{\mathcal{N}_k^G\}_{k=1}^K$ by $\widetilde{S}_u$, S^L
Input: S_u^A, S_u^B, S_u^X
Output: $\hat{x}_0 = \{\hat{x}_{0,n}\}_{n=1}^N$
repeat

$g_n, g_m \sim \text{Gumbel}(0,1)$; $\epsilon \sim \mathcal{N}(0, I)$; $t \sim \text{Uniform}(\{1, \dots, T\})$; $\mathcal{N}_{\text{Mix}}^{G_X} = \mathcal{C}^G\left[S_u^X\right]$
Take a gradient descent step on

$$\{\hat{\mathrm{x}}_{0,n}\}_{n=1}^N = \mathrm{x}_\theta(x_t, t, c_s) \tag{18}$$

$$\hat{\mathrm{x}}_{pre} = \frac{\sum_{n=1}^N [\exp\left(\log \pi_n^X + g_n\right) \cdot \hat{\mathrm{x}}_{0,n}]}{\sum_{m=1}^N \exp(\log \pi_m^X + g_m)} \tag{19}$$

$$\mathcal{L} = \frac{1}{1-\bar{\alpha}_t} \left\| \text{emb}(i_u^B) - \hat{\mathrm{x}}_{pre} \right\|_2^2 + \mathcal{L}_I \tag{20}$$

until converged;

Algorithm 2: Sampling

$x_T = \pi_{1:N}^X \cdot \mu_{1:N}^{G_X} + \pi_{1:N}^X \cdot \sigma_{1:N}^{G_X} \epsilon$
for $t = T, \dots, 1$ **do**

$\epsilon \sim \mathcal{N}(0, I)$ if $t > 1$; otherwise, $\epsilon = 0$

$$x_{t-1} = a\Big(x_t + \frac{\beta_t \sqrt{\bar{\alpha}_t}}{\hat{\alpha}_t} \mathrm{x}_\theta(x_t, t, c_s)\Big) + b_{1:N} + \sqrt{c_{1:N}} \cdot \epsilon \tag{21}$$

return x_0

B Mathematical Derivations

B.1 Derivation of the Closed-Form Marginal $q(x_t \mid x_0)$

Let $\bar{\alpha}_t \triangleq \prod_{s=1}^t \alpha_s$ and $\beta_t \triangleq 1 - \alpha_t$. The **controlled forward kernel** is

$$q(x_t \mid x_{t-1}) \;:=\; \mathcal{N}\big(\sqrt{\alpha_t}\, x_{t-1} + (1 - \sqrt{\alpha_t})\, \pi_{1:N}^X \cdot \mu_{1:N}^{G_X},\; \beta_t\, (\pi_{1:N}^X \cdot \sigma_{1:N}^{G_X})^2 I\big) \tag{22}$$

where $\pi_{1:N}^X \cdot \mu_{1:N}^{G_X} = \{\pi_n^X \cdot \mu_n^{G_X}\}_{n=1}^N$ and $\pi_{1:N}^X \cdot \sigma_{1:N}^{G_X} = \{\pi_n^X\, \sigma_n^{G_X}\}_{n=1}^N$ are the mixture-weighted mean and standard deviation of the Gaussian mixture prior $p(x_T) = \mathcal{N}_{\text{Mix}}^X = \{\pi_n^X \cdot \mathcal{N}_n^{G_X}\}_{n=1}^N$.

Since $\beta_t = 1 - \alpha_t$, the kernel (22) admits the reparameterisation

$$x_t = \sqrt{\alpha_t}\, x_{t-1} + (1 - \sqrt{\alpha_t})\, \pi_{1:N}^X \cdot \mu_{1:N}^{G_X} + \sqrt{1 - \alpha_t}\, \pi_{1:N}^X \cdot \sigma_{1:N}^{G_X}\, \epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0, I) \tag{23}$$

Lemma 1. *Let $X \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $Y \sim \mathcal{N}(\mu_2, \sigma_2^2)$ be independent. Then for any $a, b \in \mathbb{R}$,*

$$aX + bY \;\sim\; \mathcal{N}(a\mu_1 + b\mu_2,\; a^2\sigma_1^2 + b^2\sigma_2^2) \tag{24}$$

Proposition 1. *For all $t \geq 1$,*

$$q(x_t \mid x_0) = \mathcal{N}\Big(\sqrt{\bar{\alpha}_t}\, x_0 + (1 - \sqrt{\bar{\alpha}_t})\, \pi_{1:N}^X \cdot \mu_{1:N}^{G_X},\; (1 - \bar{\alpha}_t)\, (\pi_{1:N}^X \cdot \sigma_{1:N}^{G_X})^2 I\Big) \tag{25}$$

Proof. We proceed by induction on t.

Base case ($t = 1$). From (23),

$$x_1 = \sqrt{\alpha_1}\, x_0 + (1 - \sqrt{\alpha_1})\, \pi_{1:N}^X \cdot \mu_{1:N}^{G_X} + \sqrt{1 - \alpha_1}\, \pi_{1:N}^X \cdot \sigma_{1:N}^{G_X}\, \epsilon_1 \tag{26}$$

so

$$q(x_1 \mid x_0) = \mathcal{N}\Big(\sqrt{\alpha_1}\, x_0 + (1 - \sqrt{\alpha_1})\, \pi_{1:N}^X \cdot \mu_{1:N}^{G_X},\; (1 - \alpha_1)\, (\pi_{1:N}^X \cdot \sigma_{1:N}^{G_X})^2 I\Big) \tag{27}$$

which matches (25) with $\bar{\alpha}_1 = \alpha_1$.

Inductive step. Assume that for some $n \geq 1$,

$$x_n = \sqrt{\bar{\alpha}_n}\, x_0 + (1 - \sqrt{\bar{\alpha}_n})\, \pi^X_{1:N} \cdot \mu^{G_X}_{1:N} + \sqrt{1 - \bar{\alpha}_n}\, \pi^X_{1:N} \cdot \sigma^{G_X}_{1:N}\, \tilde{\epsilon}_n, \qquad \tilde{\epsilon}_n \sim \mathcal{N}(0, I) \tag{28}$$

Substituting (28) into (23) at step $n+1$:

$$\begin{aligned} x_{n+1} &= \sqrt{\alpha_{n+1}}\, x_n + (1 - \sqrt{\alpha_{n+1}})\, \pi^X_{1:N} \cdot \mu^{G_X}_{1:N} + \sqrt{1 - \alpha_{n+1}}\, \pi^X_{1:N} \cdot \sigma^{G_X}_{1:N}\, \epsilon_{n+1} \\ &= \sqrt{\alpha_{n+1}\bar{\alpha}_n}\, x_0 + \Big[\sqrt{\alpha_{n+1}}(1 - \sqrt{\bar{\alpha}_n}) + (1 - \sqrt{\alpha_{n+1}})\Big] \pi^X_{1:N} \cdot \mu^{G_X}_{1:N} \\ &\quad + \sqrt{\alpha_{n+1}(1 - \bar{\alpha}_n)}\, \pi^X_{1:N} \cdot \sigma^{G_X}_{1:N}\, \tilde{\epsilon}_n + \sqrt{1 - \alpha_{n+1}}\, \pi^X_{1:N} \cdot \sigma^{G_X}_{1:N}\, \epsilon_{n+1} \end{aligned} \tag{29}$$

Mean coefficient. Using $\bar{\alpha}_{n+1} = \alpha_{n+1}\bar{\alpha}_n$:

$$\begin{aligned} \sqrt{\alpha_{n+1}}(1 - \sqrt{\bar{\alpha}_n}) + (1 - \sqrt{\alpha_{n+1}}) &= \sqrt{\alpha_{n+1}} - \sqrt{\alpha_{n+1}\bar{\alpha}_n} + 1 - \sqrt{\alpha_{n+1}} \\ &= 1 - \sqrt{\bar{\alpha}_{n+1}}. \end{aligned} \tag{30}$$

Variance coefficient. Since $\tilde{\epsilon}_n \perp \epsilon_{n+1}$, Lemma 1 gives:

$$\begin{aligned} \alpha_{n+1}(1 - \bar{\alpha}_n) + (1 - \alpha_{n+1}) &= \alpha_{n+1} - \alpha_{n+1}\bar{\alpha}_n + 1 - \alpha_{n+1} \\ &= 1 - \bar{\alpha}_{n+1} \end{aligned} \tag{31}$$

Substituting (30) and (31) into (29) establishes (25) for $t = n+1$, completing the induction. □

During training, the noisy sample at step t is given by

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \big(1 - \sqrt{\bar{\alpha}_t}\big)\, \pi^X_{1:N} \cdot \mu^{G_X}_{1:N} + \sqrt{1 - \bar{\alpha}_t}\, \pi^X_{1:N} \cdot \sigma^{G_X}_{1:N}\, \epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0, I) \tag{32}$$

where $\pi^X_{1:N} \cdot \mu^{G_X}_{1:N}$ and $\pi^X_{1:N} \cdot \sigma^{G_X}_{1:N}$ are fixed constants determined by the mixture prior $p(x_T)$. Under the standard condition $\bar{\alpha}_T \to 0$, we have

$$q(x_{T,n} \mid x_{0,n}) \;\longrightarrow\; \mathcal{N}\big(\pi^X_n \mu^{G_X}_n,\, (\pi^X_n \sigma^{G_X}_n)^2 I\big), \quad n = 1, \ldots, N \tag{33}$$

confirming that each component trajectory of the controlled forward process converges to the corresponding terminal Gaussian of the learned prior, whose predictions are combined according to the mixture weights $\{\pi^X_n\}$ via Gumbel-Softmax.

B.2 Coefficient Design: Telescoping Structure and Boundary Conditions

The forward transition

$$q(x_t \mid x_{t-1}) = \mathcal{N}\Big(\underline{\sqrt{\alpha_t}}\, x_{t-1} + \underline{(1 - \sqrt{\alpha_t})}\, \pi^X_{1:N} \cdot \mu^{G_X}_{1:N},\; \underline{\beta_t}\, (\pi^X_{1:N} \cdot \sigma^{G_X}_{1:N})^2 I\Big) \tag{34}$$

is constructed to satisfy two principles (Sohl-Dickstein et al., 2015; Kingma & Welling, 2014; Ho et al., 2020): (i) a telescoping structure, and (ii) consistency with prescribed boundary distributions.

(i) Telescoping structure. Let $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. The coefficients are designed such that both the variance accumulation and the mean shift admit telescoping forms. For the variance term, we choose

$$\beta_t = 1 - \alpha_t \tag{35}$$

which corresponds to the discrete difference of the cumulative product $\bar{\alpha}_t$. For the mean term, the affine coefficients are defined as

$$\sqrt{\alpha_t} \quad \text{and} \quad (1 - \sqrt{\alpha_t}) \tag{36}$$

which together induce a telescoping structure on $\sqrt{\bar{\alpha}_t}$. Under these choices, the following difference identities hold:

$$\beta_s \prod_{j=s+1}^{t} \alpha_j = \prod_{j=s+1}^{t} \alpha_j - \prod_{j=s}^{t} \alpha_j \tag{37}$$

$$(1 - \sqrt{\alpha_s}) \prod_{j=s+1}^{t} \sqrt{\alpha_j} = \prod_{j=s+1}^{t} \sqrt{\alpha_j} - \prod_{j=s}^{t} \sqrt{\alpha_j} \tag{38}$$

Summing over s leads to cancellation of intermediate terms, yielding:

$$\sum_{s=1}^{t} \beta_s \prod_{j=s+1}^{t} \alpha_j = 1 - \bar{\alpha}_t \tag{39}$$

$$\sum_{s=1}^{t} (1 - \sqrt{\alpha_s}) \prod_{j=s+1}^{t} \sqrt{\alpha_j} = 1 - \sqrt{\bar{\alpha}_t} \tag{40}$$

(ii) Boundary conditions. The construction is consistent with the desired endpoints. At $t = 0$, since $\bar{\alpha}_0 = 1$,

$$q(x_0 \mid x_0) = \mathcal{N}(x_0, 0) \tag{41}$$

recovering the data distribution. Under a variance schedule chosen such that $\bar{\alpha}_T = \prod_{s=1}^{T} \alpha_s \approx 0$, we have

$$q(x_T \mid x_0) \approx \mathcal{N}\big(\pi^X_{1:N} \cdot \mu^{G_X}_{1:N},\ (\pi^X_{1:N} \cdot \sigma^{G_X}_{1:N})^2 I\big) \tag{42}$$

which matches the component-wise terminal Gaussians of the learned prior.

Resulting moments. The telescoping structure yields closed-form expressions for the conditional moments (see Appendix B.1):

$$\mathbb{E}[x_t \mid x_0] = \sqrt{\bar{\alpha}_t}\, x_0 + \big(1 - \sqrt{\bar{\alpha}_t}\big)\, \pi^X_{1:N} \cdot \mu^{G_X}_{1:N} \tag{43}$$

$$\mathrm{Var}(x_t \mid x_0) = (1 - \bar{\alpha}_t)\, (\pi^X_{1:N} \cdot \sigma^{G_X}_{1:N})^2 I \tag{44}$$

Summary. The coefficient design ensures that both drift and diffusion are governed by boundary terms through a shared telescoping mechanism, leading to a closed-form marginal that interpolates between the data distribution and the target Gaussian mixture prior $\mathcal{N}^X_{\mathrm{Mix}} = \{\pi^X_n \cdot \mathcal{N}^{G_X}_n\}_{n=1}^N$.

B.3 Domain-Specific Condition

Domain-conditional distribution as a diffusion-conditioned control signal. Consider a binary latent variable $z_n \in \{A, B\}$ indicating the domain assignment for the n-th component pair. Under the diffusion process, the posterior conditioned on the noisy state x_t at timestep t follows from Bayes' rule:

$$\log \frac{p(z_n = B \mid x_t, t)}{p(z_n = A \mid x_t, t)} = \log \frac{p(x_t \mid \mathcal{N}^{B'}_n, t)}{p(x_t \mid \mathcal{N}^{A'}_n, t)} + \log \frac{p(z_n = B)}{p(z_n = A)} \tag{45}$$

The second term is independent of x_t and can be treated as a constant offset for each component. We therefore define the *data-dependent log-density ratio* as:

$$r_n(x_t) = \log \frac{p(x_t \mid \mathcal{N}^{B'}_n, t)}{p(x_t \mid \mathcal{N}^{A'}_n, t)} = -\log p(x_t \mid \mathcal{N}^{A'}_n, t) + \log p(x_t \mid \mathcal{N}^{B'}_n, t) \tag{46}$$

which corresponds to the data-dependent term of the domain-conditional distribution. Here $\mathcal{N}^{A'}_n$ and $\mathcal{N}^{B'}_n$ denote the N-fold replicated and concatenated forms of $\mathcal{N}^A_n$ and $\mathcal{N}^B_n$, respectively, ensuring dimensional alignment with x_t. The component distributions $\mathcal{N}^A_n$ and $\mathcal{N}^B_n$ are obtained in Eq. (1) and are independent of the diffusion timestep t. Under this formulation, $r_n(x_t)$ evaluates how the current state x_t aligns with the domain-specific component structure, and thus serves as a unified conditioning signal across all diffusion steps.

The domain-specific condition is then constructed by aggregating the N component-wise ratios through an encoder f_c:

$$c_s = f_c\Big(\big\|_{n=1}^{N} \big[-\log p(x_t \mid \mathcal{N}^{A'}_n, t) + \log p(x_t \mid \mathcal{N}^{B'}_n, t)\big]\Big) \tag{47}$$

which is consistent with Eq. (10).

Closed-form expression of the domain-specific condition. Assume diagonal Gaussian components

$$p(x_t \mid \mathcal{N}_n^{D'}, t) = \mathcal{N}(x_t \mid \mu_n^{D'},\, \mathrm{diag}((\sigma_n^{D'})^2)) \propto \exp\Big(-\tfrac{1}{2}(x_t - \mu_n^{D'})^\top \Lambda_n^{D'} (x_t - \mu_n^{D'})\Big) \tag{48}$$

where $\Lambda_n^{D'} = \mathrm{diag}((\sigma_n^{D'})^{-2})$ is the precision matrix of component $\mathcal{N}_n^{D'}$.

The log-density ratio

$$r_n(x_t) = \log \frac{p(x_t \mid \mathcal{N}_n^{B'}, t)}{p(x_t \mid \mathcal{N}_n^{A'}, t)} \tag{49}$$

is obtained by expanding each log-density and collecting terms by power of x_t, using $(x_t-\mu)^\top \Lambda (x_t-\mu) = x_t^\top \Lambda x_t - 2\mu^\top \Lambda x_t + \mu^\top \Lambda \mu$. This gives the closed-form quadratic expression:

$$r_n(x_t) = \frac{1}{2}\, x_t^\top (\Lambda_n^{A'} - \Lambda_n^{B'})\, x_t + \big(\Lambda_n^{B'} \mu_n^{B'} - \Lambda_n^{A'} \mu_n^{A'}\big)^\top x_t + C_n \tag{50}$$

where

$$C_n = \frac{1}{2}\Big[(\mu_n^{A'})^\top \Lambda_n^{A'} \mu_n^{A'} - (\mu_n^{B'})^\top \Lambda_n^{B'} \mu_n^{B'}\Big] + \frac{1}{2} \log \frac{|\Lambda_n^{B'}|}{|\Lambda_n^{A'}|} \tag{51}$$

is a constant fully determined by the component parameters $\{\mu_n^{A'}, \sigma_n^{A'}, \mu_n^{B'}, \sigma_n^{B'}\}$.

B.4 Derivation of $p_\theta(x_{t-1} \mid x_t)$

Posterior of controlled diffusion. Let the forward process be defined as

$$q(x_t \mid x_{t-1}) = \mathcal{N}\Big(\sqrt{\alpha_t}\, x_{t-1} + (1 - \sqrt{\alpha_t})\, \pi_{1:N}^X \cdot \mu_{1:N}^{G_X},\ \beta_t (\pi_{1:N}^X \cdot \sigma_{1:N}^{G_X})^2 I\Big) \tag{52}$$

and define $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$, $\hat{\alpha}_t := \alpha_t - \bar{\alpha}_t$, $\beta_t := 1 - \alpha_t$. Then the posterior admits a closed-form Gaussian:

$$q(x_{t-1} \mid x_t, \hat{x}_0) = \mathcal{N}\Big(\mu_t(x_t, \hat{x}_0),\ c_{1:N} I\Big) \tag{53}$$

where

$$c_{1:N} = \frac{\hat{\alpha}_t \beta_t (\pi_{1:N}^X \cdot \sigma_{1:N}^{G_X})^2}{\alpha_t (1 - \bar{\alpha}_t)} \tag{54}$$

and

$$\mu_t = a\Big(x_t + \frac{\beta_t \sqrt{\bar{\alpha}_t}}{\hat{\alpha}_t} \hat{x}_0\Big) + b_{1:N} \tag{55}$$

with

$$a = \frac{\hat{\alpha}_t}{\sqrt{\alpha_t}(1 - \bar{\alpha}_t)}, \quad b_{1:N} = \frac{(1 - \sqrt{\alpha_t})(1 - \sqrt{\bar{\alpha}_{t-1}})}{1 + \sqrt{\bar{\alpha}_t}}\, \pi_{1:N}^X \cdot \mu_{1:N}^{G_X} \tag{56}$$

Proof. Recently, JiT (Li & He, 2025) points out that directly predicting the noise typically requires substantial model capacity, and instead advocates predicting the clean data, i.e., $\hat{x}_0 = \mathrm{x}_\theta(x_t, t, c_s)$. Adopting this parameterization, we replace x_0 with its estimator $\hat{x}_0$, and the reverse process is approximated via Bayes' rule as:

$$p_\theta(x_{t-1} \mid x_t) \;\approx\; q(x_{t-1} \mid x_t, x_0)\big|_{x_0 = \hat{x}_0} \;=\; \left.\frac{q(x_t \mid x_{t-1}, x_0)\, q(x_{t-1} \mid x_0)}{q(x_t \mid x_0)}\right|_{x_0 = \hat{x}_0} \tag{57}$$

By the first-order Markov property (Salimans et al., 2015; Farias et al., 2022; Bolin & Wallin, 2021) of the forward diffusion process, x_t is conditionally independent of $\hat{x}_0$ given x_{t-1}, which yields:

$$q(x_t \mid x_{t-1}, \hat{x}_0) = q(x_t \mid x_{t-1}) \tag{58}$$

Next, the forward transition distribution is given by:

$$q(x_t \mid x_{t-1}) = \mathcal{N}\Big(\sqrt{\alpha_t}\, x_{t-1} + (1 - \sqrt{\alpha_t})\, \pi_{1:N}^X \cdot \mu_{1:N}^{G_X},\ \beta_t\, (\pi_{1:N}^X \cdot \sigma_{1:N}^{G_X})^2\Big) \tag{59}$$

Owing to the linear Gaussian structure, the marginal distributions conditioned on the estimator $\hat{x}_0$ admit closed-form solutions:

$$q(x_t \mid \hat{x}_0) = \mathcal{N}\Big(\sqrt{\bar{\alpha}_t}\,\hat{x}_0 + \big(1-\sqrt{\bar{\alpha}_t}\big)\,\pi^X_{1:N}\cdot\mu^{G_X}_{1:N},\ (1-\bar{\alpha}_t)\,(\pi^X_{1:N}\cdot\sigma^{G_X}_{1:N})^2 I\Big) \tag{60}$$

$$q(x_{t-1} \mid \hat{x}_0) = \mathcal{N}\Big(\sqrt{\bar{\alpha}_{t-1}}\,\hat{x}_0 + \big(1-\sqrt{\bar{\alpha}_{t-1}}\big)\,\pi^X_{1:N}\cdot\mu^{G_X}_{1:N},\ (1-\bar{\alpha}_{t-1})\,(\pi^X_{1:N}\cdot\sigma^{G_X}_{1:N})^2 I\Big) \tag{61}$$

To derive the posterior in closed form, we make use of the fact that a Gaussian density satisfies $\mathcal{N}(x;\mu,\sigma^2) \propto \exp\Big(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\Big)$. Substituting Eqs. (59)-(61) into Eq. (57), and expanding each Gaussian in exponential form, we obtain:

$$\begin{aligned} q(x_{t-1} \mid x_t, \hat{x}_0) \propto \exp\Bigg\{ -\frac{1}{2}\Bigg(& \frac{\|x_t - \sqrt{\alpha_t}x_{t-1} - (1-\sqrt{\alpha_t})\,\pi^X_{1:N}\cdot\mu^{G_X}_{1:N}\|^2}{(1-\alpha_t)(\pi^X_{1:N}\cdot\sigma^{G_X}_{1:N})^2} \\ & + \frac{\|x_{t-1} - \sqrt{\bar{\alpha}_{t-1}}\hat{x}_0 - (1-\sqrt{\bar{\alpha}_{t-1}})\,\pi^X_{1:N}\cdot\mu^{G_X}_{1:N}\|^2}{(1-\bar{\alpha}_{t-1})(\pi^X_{1:N}\cdot\sigma^{G_X}_{1:N})^2} \\ & - \frac{\|x_t - \sqrt{\bar{\alpha}_t}\hat{x}_0 - (1-\sqrt{\bar{\alpha}_t})\,\pi^X_{1:N}\cdot\mu^{G_X}_{1:N}\|^2}{(1-\bar{\alpha}_t)(\pi^X_{1:N}\cdot\sigma^{G_X}_{1:N})^2} \Bigg)\Bigg\} \end{aligned} \tag{62}$$

Collecting the terms in x_{t-1} from Eq. (62), the posterior can be written as

$$q(x_{t-1} \mid x_t, \hat{x}_0) \propto \exp\Bigg\{ -\frac{1}{2(\pi^X_{1:N}\cdot\sigma^{G_X}_{1:N})^2}\Big(A\|x_{t-1}\|^2 - 2Bx_{t-1} + C\Big)\Bigg\} \tag{63}$$

where C is independent of x_{t-1}, and

$$A = \frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}} \tag{64}$$

$$B = \frac{\sqrt{\alpha_t}}{\beta_t}\Big(x_t - (1-\sqrt{\alpha_t})\,\pi^X_{1:N}\cdot\mu^{G_X}_{1:N}\Big) + \frac{1}{1-\bar{\alpha}_{t-1}}\Big(\sqrt{\bar{\alpha}_{t-1}}\hat{x}_0 + (1-\sqrt{\bar{\alpha}_{t-1}})\,\pi^X_{1:N}\cdot\mu^{G_X}_{1:N}\Big) \tag{65}$$

Completing the square w.r.t. x_{t-1} yields

$$q(x_{t-1} \mid x_t, \hat{x}_0) \propto \exp\Bigg\{ -\frac{A}{2(\pi^X_{1:N}\cdot\sigma^{G_X}_{1:N})^2}\left\|x_{t-1} - \frac{B}{A}\right\|^2 + C\Bigg\} \tag{66}$$

Closed-form coefficients. Using $\bar{\alpha}_t = \prod_{s=1}^{t}\alpha_s$, $\hat{\alpha}_t := \alpha_t - \bar{\alpha}_t$, $\beta_t := 1-\alpha_t$, we obtain

$$q(x_{t-1} \mid x_t, \hat{x}_0) = \mathcal{N}\Big(\mu_t,\ c_{1:N} I\Big), \quad \mu_t = \frac{B}{A}, \quad c_{1:N} = (\pi^X_{1:N}\cdot\sigma^{G_X}_{1:N})^2\frac{1}{A} \tag{67}$$

where $c_{1:N} = \frac{(\pi^X_{1:N}\cdot\sigma^{G_X}_{1:N})^2}{A} = \frac{\hat{\alpha}_t\beta_t\,(\pi^X_{1:N}\cdot\sigma^{G_X}_{1:N})^2}{\alpha_t(1-\bar{\alpha}_t)}$, and the mean can be written as $\mu_t = \frac{B}{A} = a(x_t + \frac{\beta_t\sqrt{\bar{\alpha}_t}}{\hat{\alpha}_t}\hat{x}_0) + b_{1:N}$, with $a = \frac{\hat{\alpha}_t}{\sqrt{\alpha_t}(1-\bar{\alpha}_t)}, b_{1:N} = \frac{(1-\sqrt{\alpha_t})(1-\sqrt{\bar{\alpha}_{t-1}})}{1+\sqrt{\bar{\alpha}_t}}\ \pi^X_{1:N}\cdot\mu^{G_X}_{1:N}$. □

B.5 Gumbel-Softmax Sampling and Contrastive Loss Details

Gumbel-Softmax sampling. Given the N component-wise predictions $\hat{\mathrm{x}}_{0,n}$ produced by the Gaussian mixture diffusion model, we require a differentiable mechanism to obtain a single prediction $\hat{\mathrm{x}}_{pre}$. This requirement follows from the generative structure of the Gaussian mixture prior $\mathcal{N}^X_{\text{Mix}} = \{\pi^X_n \cdot \mathcal{N}^{G_X}_n\}_{n=1}^N$. In a Gaussian mixture model (Zhang & Weinberger, 2022; Ziesche & Rozo, 2023), each sample is generated from a *single* component selected by a latent categorical variable:

$$n^* \sim \text{Categorical}(\pi^X_1, \ldots, \pi^X_N), \quad x \sim \mathcal{N}^{G_X}_{n^*} \tag{68}$$

Thus, training should approximately follow the same single-component selection principle. However, directly applying a hard $\arg\max$ over mixture components is non-differentiable and prevents gradient-based optimization.

To address this, we adopt the Gumbel-Softmax (GS) trick (Potapczynski et al., 2020), which provides a differentiable approximation to discrete component selection. Concretely, let π_n^X denote the mixing weight of the n-th Gaussian component. For each component n, we draw independent Gumbel noise:

$$g_n = -\log(-\log u_n), \quad u_n \sim \mathrm{Uniform}(0,1) \tag{69}$$

which satisfies $g_n \sim \mathrm{Gumbel}(0,1)$. The perturbed log-weight is:

$$\tilde{w}_n = \log \pi_n^X + g_n \tag{70}$$

The normalized weight is computed as:

$$\hat{\pi}_n = \frac{\exp(\tilde{w}_n)}{\sum_{m=1}^N \exp(\tilde{w}_m)} \tag{71}$$

with $\hat{\pi}_n > 0$ and $\sum_{n=1}^N \hat{\pi}_n = 1$. The final prediction is:

$$\hat{\mathrm{x}}_{pre} = \frac{\sum_{n=1}^N \exp\big(\log \pi_n^X + g_n\big) \cdot \hat{\mathrm{x}}_{0,n}}{\sum_{m=1}^N \exp(\log \pi_m^X + g_m)} = \sum_{n=1}^N \hat{\pi}_n \cdot \hat{\mathrm{x}}_{0,n} \tag{72}$$

Preservation of the mixture structure via parallel component diffusion. We define the forward process independently on each component: $x_t = [x_{t,1}; \ldots; x_{t,N}]$ is the concatenation of N component-wise trajectories, and the marginal factorizes as $q(x_t \mid x_0) = \prod_{n=1}^N q(x_{t,n} \mid x_{0,n})$ with $q(x_{t,n} \mid x_{0,n}) = \mathcal{N}\big(\sqrt{\bar{\alpha}_t} x_{0,n} + (1-\sqrt{\bar{\alpha}_t})\pi_n^X \mu_n^{G_X},\ (1-\bar{\alpha}_t)(\pi_n^X \sigma_n^{G_X})^2 I\big)$ (Proposition 1). As $\bar{\alpha}_T \to 0$, each $x_{T,n}$ converges to $\mathcal{N}(\pi_n^X \mu_n^{G_X}, (\pi_n^X \sigma_n^{G_X})^2 I)$. During reverse diffusion, every trajectory independently produces a clean prediction $\hat{\mathrm{x}}_{0,n}$. To aggregate these component predictions while retaining the ability to sample from the discrete mixture, we employ Gumbel-Softmax reparameterization (Eq. 72): independent Gumbel noise g_n is added to the log-weights $\log \pi_n^X$, and a softmax with unit temperature yields a differentiable relaxation of one-hot selection, which explores all mixture components during training and prevents collapse to a single mode. During inference, we set the Gumbel noise to zero, which yields a deterministic, mixture-weighted aggregation $\hat{\mathrm{x}}_{pre} = \sum_n \pi_n^X \hat{\mathrm{x}}_{0,n}$ and ensures reproducible predictions.

Positive and negative sample construction. The InfoNCE loss $\mathcal{L}_I$ requires a set of contrastive pairs $\mathcal{E} = \{e^+, e^-\}$ for each training instance (Hu et al., 2024; Li et al., 2024; Zhuo et al., 2024). We now describe how these are constructed.

Positive sample. For a given user u, the positive sample e^+ is defined as the embedding of the target item in domain B:

$$e^+ = \mathrm{emb}(i_u^B) \tag{73}$$

which is the ground-truth next item that user u interacted with in domain B. This is the same target used in the diffusion loss $\mathcal{L}_D$, ensuring consistency between the two objectives. The model is trained to produce a prediction $\hat{\mathrm{x}}_{pre}$ that is close to e^+ in the embedding space.

Negative samples. The negative samples e^- are item embeddings randomly sampled from the interactions of *other* users in domain B. Formally, for user u, the negative set is:

$$\mathcal{E}^- = \big\{\mathrm{emb}(i_{u'}^B) \;\big|\; u' \neq u,\ i_{u'}^B \in \mathcal{I}^B\big\} \tag{74}$$

where $\mathcal{I}^B$ denotes the item set of domain B. These negatives are drawn uniformly at random from items not associated with user u, providing negative samples that encourage the model to produce user-specific predictions rather than generic item embeddings.

InfoNCE objective. Given the aggregated prediction $\hat{\mathrm{x}}_{pre}$, the positive embedding e^+, and the negative embeddings $e^- \in \mathcal{E}^-$, the InfoNCE loss (Matthes et al., 2023; Oord et al., 2018) is:

$$\mathcal{L}_I = -\log \frac{\exp(\hat{\mathrm{x}}_{pre} \cdot e^+)}{\sum_{e \in \mathcal{E}} \exp(\hat{\mathrm{x}}_{pre} \cdot e)} \tag{75}$$

Table 4: Statistics of the CDSR datasets. #Overlap denotes the number of overlapping users in the training set (with the user overlap ratio in parentheses); Avg. Len denotes the average sequence length.

Dataset	# Users	# Items	# Interactions	# Overlap	# Avg. Len
Kitchen	7,087	16,258	85,964	1,837 (15.17%)	12.1
Food	6,859	11,837	81,957	1,837 (15.17%)	11.9
Electronics	3,296	14,188	61,385	1,043 (19.64%)	18.6
Beauty	3,058	10,379	48,582	1,043 (19.64%)	15.9
Book	23,596	90,958	386,275	6,432 (16.27%)	16.4
Movie	22,378	35,712	327,850	6,432 (16.27%)	14.7

where $\mathcal{E} = \{e^+\} \cup \mathcal{E}^-$. This loss maximizes the similarity between $\hat{\mathrm{x}}_{pre}$ and the ground-truth target e^+, while simultaneously pushing $\hat{\mathrm{x}}_{pre}$ away from the negative embeddings e^-. Combined with the diffusion loss $\mathcal{L}_D$, the total training objective is:

$$\mathcal{L} = \mathcal{L}_D + \mathcal{L}_I \tag{76}$$

which jointly optimizes for accurate denoising reconstruction and discriminative contrastive alignment in the item embedding space.

C Experiments

C.1 Experimental Settings

Datasets. Following most existing methods (Sun et al., 2023; Bian et al., 2025; Ning et al., 2025), we adopt three widely used CDSR datasets collected from the Amazon website[2], which are organized into three source–target domain pairs, namely Kitchen → Food (KF), Electronics → Beauty (EB), and Book → Movie (BM). Each dataset consists of user interaction sequences from two related domains. For overlapping users, sequences contain interactions from both domains and are strictly ordered by timestamp, while non-overlapping users have sequences from a single domain only. Dataset statistics are summarized in Table 4; the user overlap ratios are 15.17%, 19.64%, and 16.27% for KF, EB, and BM, respectively.

Baselines. We compare the performance of our model against representative baselines from three categories: Single-Domain Sequential Recommender (SDSR), including VSAN (Zhao et al., 2021) and CVAE (Wang et al., 2022); Cross-Domain Recommender (CDR), including MIT (Sun et al., 2023), DM (Li et al., 2025b), and CD (Li et al., 2025a); and Cross-Domain Sequential Recommender (CDSR), including ASR (Ju et al., 2025), ABXI (Bian et al., 2025), I2VAE (Ning et al., 2025), and HzRec (Zha et al., 2026), to demonstrate the effectiveness of our method. Since some method names are too long to fit in the result tables, we use shortened names throughout the paper: CVAE for ContrastVAE, MIT for REMIT, DM for DMCDR, CD for CD-CDR, ASR for AutoCDSR, I2VAE for i^{2}VAE, and HzRec for HorizonRec.

Specifically, the baselines are organized as follows:

- **VSAN (Zhao et al., 2021):** models sequential user behavior with variational state evolution, enabling robust preference modeling in single-domain settings.
- **CVAE (Wang et al., 2022):** integrates variational autoencoders with contrastive learning to enhance representation learning of user-item interactions, improving the structure and separability of latent embeddings.
- **MIT (Sun et al., 2023):** improves cross-domain transfer by maximizing mutual information between domain-specific representations, which helps preserve informative shared signals across domains.
- **DM (Li et al., 2025b):** introduces a diffusion-based framework that refines latent representations through an iterative denoising process, enabling effective cross-domain preference transfer.

[2] https://cseweb.ucsd.edu/~jmcauley/datasets/amazon/links.html

- **CD (Li et al., 2025a):** captures cross-domain knowledge transfer by learning shared and domain-specific latent factors, allowing flexible decomposition of user preferences across domains.
- **ASR (Ju et al., 2025):** models cross-domain sequential dependencies by aligning behavior sequences with attention-based architectures, capturing transferable sequential patterns across domains.
- **ABXI (Bian et al., 2025):** incorporates attention mechanisms to aggregate and reweight historical behaviors across domains, enabling more adaptive cross-domain sequential preference modeling.
- **I2VAE (Ning et al., 2025):** models item-level transitions via variational inference and treats interaction sequences as stochastic processes, further improving robustness with mutual information regularization.
- **HzRec (Zha et al., 2026):** a diffusion-based CDSR method that aligns and fuses triple preferences via dual-oriented diffusion, starting denoising from a user-agnostic Gaussian prior.

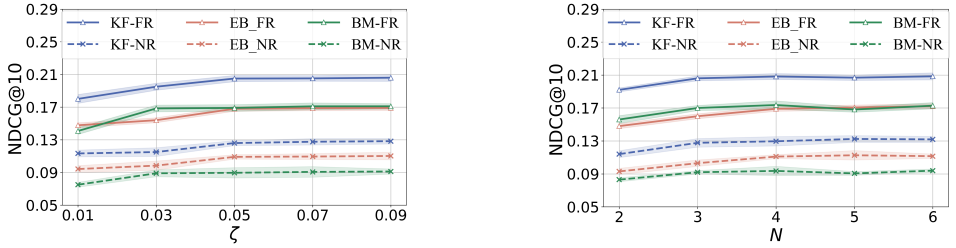
Implementation details. We implement GMD using the PyTorch framework and conduct all experiments on an NVIDIA Tesla V100 GPU. The batch size is set to 1024 across all datasets, and the learning rate is fixed at 0.001. The dropout rate is 0.5. The item embedding dimension is set to 512. Optimization is performed using Adam, and the model is trained for 200 epochs. By default, the diffusion process consists of 20 steps. We adopt a two-stage training strategy. In the pretraining stage, we employ a masked item prediction objective to learn the shared codebook. In the main training stage, we optimize the target domain next item prediction task, while keeping the codebook parameters frozen to preserve the learned cross-domain structure. During training, Gumbel-Softmax sampling is employed to enable differentiable approximation of discrete variables, where $g_n, g_m \sim \mathrm{Gumbel}(0, 1)$ introduce stochastic perturbations. During inference, we set $g_n = g_m = 0$, which removes randomness and yields a deterministic, mixture-weighted aggregation of the component predictions. For the codebook, we set the number of Gaussian components per sequence to 3 for KF-FR, KF-NR, BM-FR, and BM-NR, while setting it to 4 for EB-FR and EB-NR. We further define the codebook size as $K = \zeta \cdot N_{\text{cross}}$, where N_{cross} denotes the number of cross-domain interactions of overlapping users. Specifically, we set $\zeta = 0.05$ for KF-FR, KF-NR, EB-FR, and EB-NR, and $\zeta = 0.03$ for BM-FR and BM-NR. We adopt NDCG@k and HR@k as evaluation metrics with $k \in \{5, 10\}$.

C.2 Impact of the Codebook

Number of intra-codebook distributions. The codebook is introduced to capture shared preferences across domains. We study the effect of intra-codebook distributions by setting $K = \zeta \cdot N_{\text{cross}}$, where N_{cross} denotes the number of cross-domain interactions of overlapping users. The scaling factor $\zeta \in \{1, 3, 5, 7, 9\} \times 10^{-2}$ controls the relative codebook capacity. Larger ζ yields finer-grained representations but may also introduce redundancy. We further conduct a controlled sensitivity analysis on ζ. For each setting, the model is independently trained and evaluated five times with different random seeds to account for stochasticity in optimization and sampling. For each ζ, we report the mean NDCG@10 across five runs as the final performance. The variability is quantified by the standard deviation, visualized as a shaded band around the mean curve, where the solid line or dashed line denotes the average performance and the translucent region indicates one standard deviation. This protocol provides a robust evaluation of sensitivity to ζ, reducing the influence of outlier runs and clearly characterizing performance trends under varying codebook capacity.

As shown in Fig. 4(a), the effect of ζ varies across datasets. For KF and EB, performance improves up to approximately $\zeta = 0.05$, after which the gains become marginal, whereas BM reaches a plateau earlier, at approximately $\zeta = 0.03$. A larger ζ yields a finer-grained partition of the shared interest space, but once the main shared patterns are covered, further increasing ζ mainly over-partitions already well-represented structures and brings limited additional improvement.

Number of Gaussian components. We further study the effect of the number of Gaussian components N, where $N \in \{2, 3, 4, 5, 6\}$, to control the granularity of preference modeling. For each setting of N, we independently train and evaluate the model five times using different random seeds to account for stochasticity in optimization and sampling. We report the mean NDCG@10 over the five runs as the final performance for each configuration. To quantify performance variability, we compute the standard deviation across runs and visualize it as a shaded region around the mean curve, where the line denotes the average performance and the shaded band represents one standard deviation.



(a) Number of intra-codebook distributions (b) Number of Gaussian components

Figure 4: Impact of the codebook.

As shown in Fig. 4(b), the effect of N varies across datasets. For KF-FR, KF-NR, BM-FR, and BM-NR, performance improves up to approximately $N = 3$, after which the gains become marginal. In contrast, EB keeps improving up to $N = 4$ and then gradually plateaus. A larger N enables a more fine-grained decomposition of latent preference patterns, while potentially introducing redundancy among components, as reflected in both the mean performance and its variance.

This difference may be attributed to variations in sequence complexity across datasets. EB has the largest average sequence length (around 18.6 and 15.9), indicating that user behavior sequences contain richer and more diverse preference patterns. In this case, a larger number of Gaussian components is beneficial to better capture the multi-pattern structure within long sequences. As a result, performance continues to benefit from increasing N until a higher threshold.

In contrast, KF and the other datasets have shorter average sequence lengths (around 12.1 and 11.9 for KF and 16.4 and 14.7 for BM). These shorter sequences contain relatively simpler preference transitions. Therefore, a smaller number of Gaussian components is sufficient to model the main patterns, and increasing N beyond 3 brings limited additional benefit. Overall, datasets with longer and more complex user behavior sequences benefit from a larger number of Gaussian components, while datasets with shorter sequences tend to saturate earlier with smaller N.

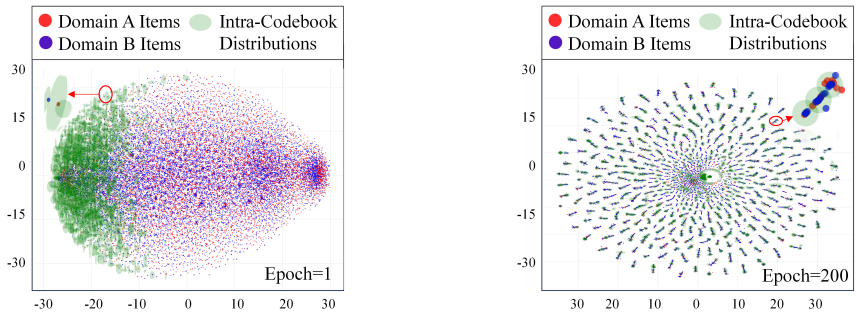
C.3 Comparison of Prior Types

To isolate the effect of the prior, we keep all other components of GMD unchanged (codebook pre-training, domain-specific condition, and x_0-prediction) and only replace the prior from which denoising starts. We compare the following variants: (1) **Standard**: the isotropic Gaussian $\mathcal{N}(0, I)$, as in DDPM (Ho et al., 2020) (identical to **w/o P** in Table 3); (2) **Data-dependent**: a single Gaussian whose mean and variance are computed from the user's own sequence distribution $\mathcal{N}_S^X$, in the spirit of PriorGrad (Lee et al., 2022); (3) **Single-codebook**: a single Gaussian indexed from the codebook ($N = 1$); (4) **Mixture w/o codebook**: a Gaussian mixture learned directly from the user's sequence without the shared codebook (identical to **w/o G** in Table 3); (5) **GMD**: the codebook-based Gaussian mixture prior. Table 5 reports NDCG@10.

Table 5: Comparison of prior types (NDCG@10). Only the prior is changed; all other components are kept identical.

Prior	KF-FR	KF-NR	EB-FR	EB-NR	BM-FR	BM-NR
Standard $\mathcal{N}(0, I)$	0.1451	0.1175	0.1314	0.0868	0.1384	0.0597
Data-dependent (single Gaussian)	0.1631	0.1187	0.1402	0.0915	0.1404	0.0626
Single-codebook ($N = 1$)	0.1748	0.1201	0.1435	0.0948	0.1416	0.0664
Mixture w/o codebook	0.1846	0.1204	0.1458	0.0984	0.1428	0.0694
GMD	**0.2052**	**0.1287**	**0.1692**	**0.1101**	**0.1696**	**0.0897**

The codebook-based mixture prior of GMD performs best on all datasets, and all learned priors outperform the standard prior. On all six settings, performance increases steadily from the standard prior to the data-dependent single Gaussian, the single-codebook Gaussian, and the mixture prior,



(a) Visualization at Epoch 1

(b) Visualization at Epoch 200

Figure 5: t-SNE visualization of A/B domain items and the codebook. Ellipses represent intra-codebook distributions, where centers denote means and radii denote standard deviations.

indicating that both a user-specific starting point and multi-peaked modeling contribute to the gain. Comparing the single-codebook variant ($N = 1$) with GMD isolates the effect of multi-peaked modeling, which yields absolute NDCG@10 gains of 0.86%-3.04%. Moreover, GMD outperforms the codebook-free mixture prior on all datasets by 0.83%-2.68%, confirming the benefit of the globally shared codebook.

C.4 Case Study

Fig. 5 visualizes item embeddings and intra-codebook distribution embeddings (including means and standard deviations) on the KF dataset using t-SNE. At the early stage of pre-training, item embeddings are randomly scattered throughout the space, and the codebook captures only a small portion of the data, often very poorly. At the later stage of training (including pre-training) as shown in Fig. 5(b), the codebook distributions better capture items from multiple domains and exhibit clearer separation, reflecting a more organized embedding space. The zoomed-in view in the top-right corner shows that a mixture of distributions can better approximate multi-domain user interactions, which is essential for effective cross-domain learning. Furthermore, samples drawn from these mixture distributions serve as NDI-aware starting points for denoising.

C.5 Efficiency Study

Table 6: Efficiency Study. T-T denotes the per-epoch training time (minutes), whereas T-I represents the per-item inference time (seconds). The best results are underlined (GMD-D is excluded as a pretraining-only stage without inference). Results of our model are **highlighted in bold**. † indicates the worst-performing model.

	KF-FR		EB-FR		BM-FR		KF-NR		EB-NR		BM-NR	
Model	*T-T*	*T-I*	*T-T*	*T-I*	*T-T*	*T-I*	*T-T*	*T-I*	*T-T*	*T-I*	*T-T*	*T-I*
MIT	2.10m	0.80e-3	1.90m	0.76e-3	5.10m	0.78e-3	2.02m	0.79e-3	1.80m	0.75e-3	5.00m	0.77e-3
ASR	2.85m	1.05e-3	2.55m	1.01e-3	7.10m	1.03e-3	2.75m	1.04e-3	2.45m	1.00e-3	6.95m	1.02e-3
ABXI	2.73m	0.89e-3	2.01m	0.87e-3	6.07m	0.88e-3	2.70m	0.88e-3	1.98m	0.86e-3	6.00m	0.87e-3
I2VAE	2.95m	0.96e-3	2.18m	0.93e-3	6.55m	0.95e-3	2.90m	0.95e-3	2.12m	0.92e-3	6.48m	0.94e-3
DM	7.60m†	2.75e-3†	5.90m†	2.65e-3†	20.53m†	2.70e-3†	7.40m†	2.70e-3†	5.75m†	2.60e-3†	20.2m†	2.68e-3†
CD	6.65m	2.30e-3	5.05m	2.20e-3	17.8m	2.25e-3	6.50m	2.28e-3	4.95m	2.18e-3	17.5m	2.24e-3
HzRec	6.35m	2.24e-3	5.01m	2.12e-3	16.8m	2.22e-3	6.36m	2.16e-3	4.45m	2.05e-3	16.8m	2.15e-3
GMD-D	**0.97m**	-	**0.73m**	-	**2.81m**	-	**0.95m**	-	**0.71m**	-	**2.78m**	-
GMD	**4.05m**	**1.45e-3**	**3.10m**	**1.42e-3**	**11.84m**	**1.46e-3**	**3.98m**	**1.44e-3**	**3.02m**	**1.43e-3**	**11.65m**	**1.45e-3**

We follow Xia et al. (2023) to assess the efficiency of GMD by comparing its training and inference costs with CDR and CDSR baselines. Table 6 reports the per-epoch training time (T-T, in minutes) and per-item inference time (T-I, in seconds) across three datasets. GMD-D denotes the training cost of the domain-shared preference modeling stage, which is used to pretrain the codebook for capturing cross-domain shared preferences. GMD represents the total cost, including both the pretraining and main training phases.

Several observations can be drawn. First, the pretraining stage is computationally lightweight. Specifically, GMD-D requires 0.71–2.81 min/epoch, which is substantially lower than the overall training cost of GMD (i.e., 3.02–11.84 min/epoch). This efficiency stems from the lightweight design of the pretraining stage, where the domain-shared codebook is learned without full diffusion modeling. In contrast, the main training phase involves a complete iterative denoising process with a larger step budget.

Second, MIT achieves the lowest computational cost among all methods, requiring only 1.80–5.10 min/epoch for training and 0.75–0.80×10^{-3} s per item for inference. This efficiency is attributed to its embedding-and-mapping paradigm without iterative refinement, which makes both training and inference essentially one-pass operations. However, this design also limits its modeling capacity, leading to inferior recommendation performance. Besides, non-diffusion-based methods (i.e., MIT, ASR, ABXI, and I2VAE) generally exhibit lower computational costs than diffusion-based approaches, which introduce additional iterative denoising overhead.

Third, among diffusion-based approaches, GMD achieves a more favorable efficiency–effectiveness trade-off. Compared with DM (i.e., 5.75–20.53 min/epoch and 2.60–2.75×10^{-3} s per item), CD (i.e., 4.95–17.80 min/epoch and 2.18–2.30×10^{-3} s per item), and HzRec (i.e., 4.45–16.80 min/epoch and 2.05–2.24×10^{-3} s per item), GMD significantly reduces training cost to 3.02–11.84 min/epoch and inference cost to 1.42–1.46×10^{-3} s per item. This efficiency gain is mainly attributed to the learned prior: since denoising starts from a user-specific distribution close to the target, GMD requires only 20 denoising steps (Section 5.4), while the pretraining stage itself is lightweight. As a result, the iterative denoising trajectory becomes shorter while maintaining effective reconstruction quality. Consequently, GMD achieves superior overall efficiency compared with other diffusion-based models, while retaining much of the expressive power of multi-step diffusion modeling.

D Supplementary Related Work

Diffusion models for recommendation. Diffusion models (Ho et al., 2020) have recently been adopted for cross-domain recommendation. DM (Li et al., 2025b) learns a preference-guided diffusion model to transfer user preferences across domains, CD (Li et al., 2025a) generates target-domain items via conditional diffusion, and HzRec (Zha et al., 2026) aligns and fuses triple preferences via dual-oriented diffusion for CDSR. All of these methods start denoising from the standard prior $\mathcal{N}(0, I)$, whereas GMD starts from a learned, user-specific Gaussian mixture prior.

Diffusion models with non-standard priors. Several works in speech and image generation replace the standard prior. PriorGrad (Lee et al., 2022) builds a data-dependent Gaussian prior from conditional statistics for speech synthesis; ResShift (Yue et al., 2023) shifts the residual between low- and high-resolution images to shorten the diffusion chain for super-resolution; denoising diffusion bridge models (Zhou et al., 2024) connect two paired distributions; and non-isotropic Gaussian diffusion (Yu et al., 2023) relaxes the isotropic noise assumption for image editing. Mixture-of-Gaussian priors have also been studied for image generation under limited training resources (Jia et al., 2024) and for diffusion-based samplers (Blessing et al., 2025). These methods are designed for continuous signals such as audio and images, where the prior is derived from a paired condition or fixed per dataset. In contrast, GMD targets sequential recommendation, where the prior must be estimated per user from sparse interactions; it therefore learns the mixture components from a globally shared codebook, derives the corresponding noising and denoising processes, and conditions denoising on domain-specific densities for cross-domain adaptation.

Building blocks. GMD adopts several existing techniques as building blocks: x_0-prediction from JiT (Li & He, 2025), Gaussian mixture modeling from GM-VAE (Dilokthanakul et al., 2016), the codebook design from VQ-VAE (Van Den Oord et al., 2017), and conditional diffusion from CD (Li et al., 2025a). Our contribution lies in the learnable, user-specific prior and the correspondingly redesigned diffusion process.

E Limitations and Future Work

In this paper, we propose a conditional Gaussian Mixture Denoising (GMD) framework for cross-domain sequential recommendation. The proposed method models domain-shared interests via a

codebook-based Gaussian mixture distribution and performs diffusion-based denoising for effective target-domain adaptation. Moreover, domain-specific sequential distributions are incorporated as conditions to guide the diffusion process, enabling more accurate knowledge transfer across domains. Extensive experiments on multiple real-world datasets demonstrate the superiority and effectiveness of the proposed approach. Although the model achieves promising results, it still has certain limitations, as discussed below.

E.1 Limitations

Computational cost of iterative denoising. Although the proposed framework achieves more efficient cross-domain adaptation compared to existing diffusion-based recommendation methods such as CD (Li et al., 2025a) and DM (Li et al., 2025b), it still relies on an iterative denoising process during inference. Compared with one-shot generative paradigms based on variational autoencoders (VAEs), such as I2VAE (Ning et al., 2025), the multi-step diffusion procedure introduces additional computational overhead. Nevertheless, because denoising starts from a learned user-specific prior rather than from $\mathcal{N}(0, I)$, GMD converges in substantially fewer steps (Section 5.4), which partially alleviates this overhead. Moreover, we employ a lightweight diffusion setting with a relatively small number of denoising steps, making the inference process feasible under the experimental settings considered in this work. Further improving sampling efficiency for latency-sensitive recommendation scenarios remains an important direction for future research.

Dependence on overlapping users. Like existing CDSR methods, GMD relies on overlapping training users to learn cross-domain associations, since the two domains share no items. Exploiting side information (e.g., item text) to bridge domains without overlapping users is left for future work.

Evaluation scope. Following existing baselines, we evaluate on Amazon cross-domain datasets, whose domains are relatively homogeneous. Evaluating GMD on heterogeneous or large-scale industrial cross-domain scenarios (e.g., video-music, news-shopping) remains future work, as suitable public benchmarks are currently scarce.

E.2 Future Work

Efficient diffusion sampling. Future work can further improve the efficiency of the proposed framework through accelerated diffusion sampling strategies. Since the proposed formulation is compatible with recent efficient diffusion paradigms, one promising direction is to distill the iterative denoising process into a few-step or even single-step generator via progressive distillation, thereby reducing inference overhead while preserving recommendation quality. In addition, deterministic samplers, adaptive timestep scheduling, and truncated diffusion trajectories may further shorten the denoising process. Furthermore, extending the current framework with more scalable latent diffusion architectures and efficient codebook retrieval mechanisms may further improve its applicability in large-scale recommendation systems.

F Broader Impact Statement

This work advances cross-domain sequential recommendation by introducing a diffusion-based framework with conditional Gaussian mixture denoising for effective knowledge transfer under sparse interactions. It focuses on methodological improvements and does not involve sensitive data or privacy-related concerns, thus posing no negative societal impact.