CONDITIONAL INFORMATION BOTTLENECK APPROACH FOR OUT-OF-DISTRIBUTION SEQUENTIAL RECOMMENDATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Sequential recommendation (SR) aims to suggest items users are most likely to engage with next based on their past interactions. However, in practice, SR systems often face the out-of-distribution (OOD) problem due to dynamic environmental factors (*e.g.*, seasonal changes), leading to significant performance degradation in the testing phase. Some methods incorporate distributionally robust optimization (DRO) into SR to alleviate OOD, but the sparsity of SR data challenges this. Other approaches use random data augmentations to explore the OOD, potentially distorting important information, as user behavior is personalized rather than random. Additionally, they often overlook users' varying sensitivity to distribution shifts during the exploration, which is crucial for capturing the evolution of user preferences in OOD contexts. In this work, inspired by information bottleneck theory (IB), we propose the Conditional Distribution Information Bottleneck (CDIB), a novel objective that creates diverse OOD distributions while preserving *minimal sufficient information* regarding the origin distribution conditioned on the user. Building on this, we introduce a framework with a learnable, personalized data augmentation method using a mask-then-generate paradigm to craft diverse and reliable OOD distributions optimized with CDIB. Experiments on four real-world datasets show our model consistently outperforms baselines. The code is available at `https://anonymous.4open.science/r/CDIB-51C8`.

1 INTRODUCTION

Nowadays, recommendation systems are important in addressing information overload across various applications, such as e-commerce, online retail platforms, and so on (Cen et al., 2020; Guy et al., 2010). SR is one of the crucial topics focusing on capturing users' dynamic interest to recommend content that aligns with it more accurately (Hidasi et al., 2016; Kang & McAuley, 2018).

Nevertheless, most methods assume that the popularity distribution during training and testing is independent and identically distributed, an unrealistic assumption in most cases (Zheng et al., 2021; Zhang et al., 2023). In SR, popularity distribution can shift due to time-sensitive environmental factors, leading to changes in user preferences (*e.g.*, the World Cup boosting soccer jersey sales or seasonal changes increasing T-shirt sales in summer and sweater sales in winter), which causes performance degradation of the model during the testing phases.

Furthermore, we observe that different users have varying sensitivity to distribution shifts, leading to different impacts from OOD scenarios. As shown in Figure 1, for blockbuster users who engage with trending content, the model can adjust and continue providing relevant recommendations as trends shift (①→③) due to its inherent bias toward popular items (Zhang et al., 2021). However, for niche users who follow mainstream items less, despite the model capturing their preferences during training, it often defaults to providing popular items when environmental factors change, likely due to unfamiliar behavior patterns, misaligning with niche users' true preferences (②→④).

To alleviate the OOD problem in SR, various models have been developed, employing techniques such as reweighting (Wang et al., 2022b), causal inference (Wang et al., 2023b; He et al., 2022), distributionally robust optimization (Yang et al., 2023b; Wen et al., 2022), and contrastive learning (CL) (Liu et al., 2021; Xie et al., 2022; Yang et al., 2023a; Qiu et al., 2022).

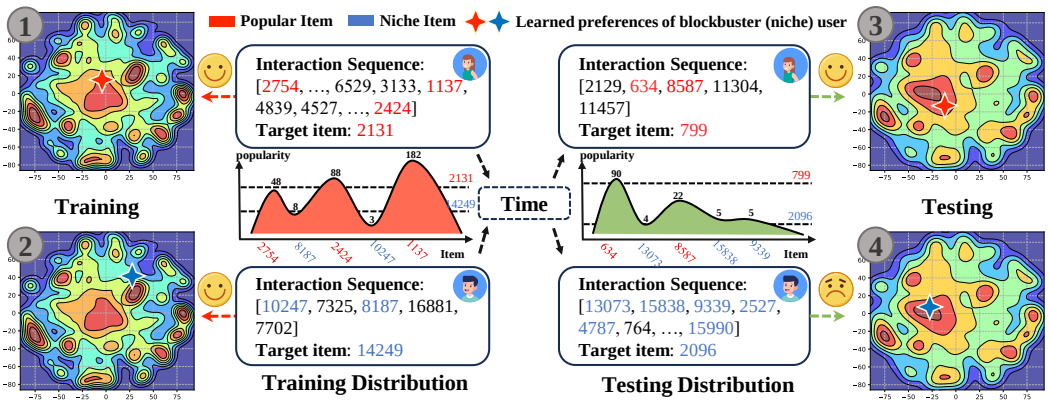


Figure 1: ① and ② show the overall user preference distribution learned by SASRec (Kang & McAuley, 2018) during training, along with learned blockbuster and niche user preferences, respectively, while ③ and ④ display the corresponding distributions during testing. (Blockbuster user: prefers mainstream items; Niche user: prefers mainstream items less.)[1]

However, existing methods have two main issues: **(i)** Methods like DRO in SR (Yang et al., 2023b) optimize the model for the worst-case distribution within a family of distributions around the training data, ensuring robustness to unknown distributions. However, the feasible distribution family for DRO is inherently limited by the sparse nature of recommendation data (Wang et al., 2024). **(ii)** Data augmentation models (*e.g.*, CL methods (Qiu et al., 2022; Xie et al., 2022)) can expand the training distribution but often rely on unguided or hand-crafted augmentations, risking the loss of important interaction data while retaining noisy or irrelevant information for augmentation, misleading user preference modeling. Additionally, they overlook users' varying sensitivity to distribution shifts mentioned above, which is essential for capturing the evolution of user preferences in OOD contexts.

To this end, inspired by IB theory (Gondek & Hofmann, 2003; Lee et al., 2023; Choi & Lee, 2023; Tishby & Zaslavsky, 2015; Alemi et al., 2016), we propose the Conditional Distribution Information Bottleneck (CDIB), a novel objective that generates diverse distributions while preserving *minimal sufficient information* from the original distribution, conditioned on the user. It aims to introduce more interaction patterns influenced by other environmental factors into model training, enhancing performance on unknown distributions. Specifically, it consists of a conditional generation term to diversify the generated OOD data considering the users' varying sensitivity to distribution shift and a conditional regularization term to preserve personalized critical information within the origin distribution. What's more, we provide theoretical analyses to justify the rationality of the generated distribution. On top of it, we propose a framework comprised of two processes:

- We propose a data augmentation strategy that employs a learnable mask to adaptively mask stable elements reflecting user interests (*e.g.*, a comic book during the World Cup) from distortion, while selecting elements sensitive to environmental factors (*e.g.*, a soccer jersey during the World Cup) for augmentation within the hidden space through a distribution generator based on a latent diffusion model (Wang et al., 2023a; Rombach et al., 2022; Ho et al., 2020).
- Optimizing with CDIB: Given an original distribution $\mathcal{D}$ and a generated distribution $\tilde{\mathcal{D}}$, CDIB diversify $\tilde{\mathcal{D}}$ by minimizing the mutual information between $\mathcal{D}$ and $\tilde{\mathcal{D}}$ conditioned on user feature. Simultaneously, the model preserves personalized information by maximizing the mutual information between $\tilde{\mathcal{D}}$ and the target, also conditioned on user feature.

To summarize, this work makes the following contributions: **(i)** We introduce CDIB, an objective that guides the generation of diverse and reliable distributions, and conduct theoretical analyses to prove its rationality. **(ii)** Based on the CDIB principle, we propose a framework which generates the distribution with a learnable and personalized augmentation method. **(iii)** Extensive experiments demonstrate the effectiveness and robustness of CDIB.

[1]The concepts and definition of Blockbuster and Niche are derived from (Wen et al., 2022)

2 PRELIMINARIES

This section begins by outlining the sequential recommendation scenario, including notations and the problem formulation (Section 2.1). Then, we introduce the IB theory (Section 2.2).

2.1 SEQUENTIAL RECOMMENDATION PARADIGM

Notations. Denote with $\mathcal{U}$ ($u \in \mathcal{U}$) the user set and with $\mathcal{I}$ ($i \in \mathcal{I}$) the item set, where $|\mathcal{U}|$ and $|\mathcal{I}|$ represent the number of users and items, respectively. Each user u is associated with a chronologically ordered interaction sequence $s_u = (i_u^1, \cdots, i_u^L)$, with L denoting the length of the sequence. The collection of all interaction sequences is denoted as $\mathcal{S}$ ($s_u \in \mathcal{S}$), which is further partitioned into the training set $\mathcal{S}_{tr}$ comprising historical interactions, and the test set $\mathcal{S}_{te}$ containing future interactions. Also, we denote the stable elements as $\mathbf{X}_s$ and the (environmental-)sensitive elements within the interaction as $\mathbf{X}_e$. Additionally, we define $\mathcal{D}_{tr}$ and $\mathcal{D}_{te}$ as the training and testing distributions.

Problem Formulation. Formally, the learning of sequential recommendation involves optimizing the model ξ through empirical risk minimization on the training distribution $\mathcal{D}_{tr}$ (Kang & McAuley, 2018; Sun et al., 2019), which only involves ID features:

$$\xi^* = \arg\min_{\xi} \hat{\mathbb{E}}_{\mathcal{D}_{tr}} \left[\ell \left(\xi \left(s_u \right), i_u^{L+1} \right) \right] = \arg\min_{\xi} \frac{1}{|\mathcal{S}_{tr}|} \sum_{s_u \in \mathcal{S}_{tr}} -\log p \left(i_u^{L+1} \mid \xi \left(s_u \right) \right), \quad (1)$$

where i_u^{L+1} represents the next item to interact with by user u, $\xi(s_u)$ represents the hypothesis generated by ξ, and $p\left(i_u^{L+1} | \xi\left(s_u\right)\right)$ denotes the probability of ξ recommending i_u^{L+1} to user u based on s_u. Subsequently, the derived ξ^* is applied to the future task.

2.2 INFORMATION BOTTLENECK PRINCIPLE

The information bottleneck principle (Tishby & Zaslavsky, 2015; Alemi et al., 2016) is an approach based on information theory designed to balance the trade-off between compressing a random variable and preserving its *minimum sufficient information* about the target variable. It aims to find a compact representation that retains as much information about the target as possible and discards the target-irrelevant information.

Definition 2.1 (IB) *Given input variable $\mathbf{X}$, target $\mathbf{Y}$, and bottleneck variable $\mathbf{Z}$, respectively, the IB aims to compress $\mathbf{X}$ to $\mathbf{Z}$, while keeping the information relevant for $\mathbf{Y}$:*

$$\min_{\mathbf{Z}} I(\mathbf{X}; \mathbf{Z}) - \beta I(\mathbf{Y}; \mathbf{Z}) \quad (2)$$

where $I(\mathbf{U}; \mathbf{V}) = \sum_{u,v} p(u,v) \log \frac{p(u,v)}{p(u)p(v)}$ is the mutual information between $\mathbf{U}$ and $\mathbf{V}$, and $\beta \in \mathbb{R}$ is a Lagrange multiplier balancing the two mutual information terms.

3 METHODOLOGY

In this section, we first formally introduce the distribution information bottleneck principle (DIB) in Section 3.1, and then we propose the conditional distribution information bottleneck principle (CDIB) in Section 3.2. Following the CDIB, we detail the overall architecture of the proposed framework and its optimization strategies in Section 3.3.

3.1 DISTRIBUTION INFORMATION BOTTLENECK PRINCIPLE

In this section, we introduce the DIB, which is anchored in the IB. This principle facilitates the generation of new distributions, represented as $\tilde{\mathcal{D}} = (\tilde{\mathbf{X}}_s, \tilde{\mathbf{X}}_e)$, derived from the training distribution $\mathcal{D}_{tr} = (\mathbf{X}_s, \mathbf{X}_e)$. With the DIB, while the $\tilde{\mathcal{D}}$ preserves the stable elements within the $\mathcal{D}_{tr}$, it introduces a spectrum of sensitive elements, thereby enhancing the diversity of the data.

Definition 3.1 (DIB) *Given original training distribution $\mathcal{D}_{tr}$, target $\mathbf{Y}$, and generated distribution $\tilde{\mathcal{D}}$, respectively, we define DIB as follows:*

$$\min_{\tilde{\mathcal{D}}} I(\mathcal{D}_{tr}; \tilde{\mathcal{D}}) - \beta I(\mathbf{Y}; \tilde{\mathcal{D}}) \quad (3)$$

where $\beta \in \mathbb{R}$ is a Lagrange multiplier balancing the diversity and reliability of the $\tilde{\mathcal{D}}$.

DIB seeks to foster a variety of distributions $\tilde{\mathcal{D}}$ that diverge from the original distribution by minimizing $I(\mathcal{D}_{tr}; \tilde{\mathcal{D}})$, while concurrently ensuring the preservation of critical information by maximizing $I(\mathbf{Y}; \tilde{\mathcal{D}})$. It can be demonstrated that this leads to $\tilde{\mathbf{X}}_s \simeq \mathbf{X}_s$ and $\tilde{\mathbf{X}}_e \not\sim \mathbf{X}_e$, achieving a balance between stability in stable elements and diversity in sensitive elements (*cf.* Appendix A.1).

3.2 Conditional Distribution Information Bottleneck Principle

Although DIB can generate a diverse and promising distribution $\tilde{\mathcal{D}}$, the generation process of the distributions is still not fully controllable due to the lack of constraints. Specifically, there is no certainty that the distributions obtained from the minimization of DIB will represent those encountered in the testing stage. This is because $\tilde{\mathbf{X}}_e$ can be generated in any direction that diverges from $\mathbf{X}_e$ as long as it can minimize the $I(\mathcal{D}_{tr}, \tilde{\mathcal{D}})$. Furthermore, the straightforward application of DIB fails to account for users' sensitivity to OOD. More concretely, some users are readily influenced by environmental factors like popular trends. In contrast, others display less susceptibility to such influences. Consequently, the direction of $\tilde{\mathbf{X}}_e$ generation should be personalized, suggesting that the generation of $\tilde{\mathcal{D}}$ should be more controlled.

To this end, we introduce the CDIB, aiming to guide the personalized generation of $\tilde{\mathcal{D}}$ and steer it, to a certain extent, towards aligning with the testing distribution:

Definition 3.2 (CDIB) *Given original training distribution $\mathcal{D}_{tr}$, target $\mathbf{Y}$, generated distribution $\tilde{\mathcal{D}}$, and user attributes $\mathbf{\Gamma}$ respectively. The formulation of CDIB is as follows:*

$$\min_{\tilde{\mathcal{D}}} I(\mathcal{D}_{tr}; \tilde{\mathcal{D}} \mid \mathbf{\Gamma}) - \beta I(\mathbf{Y}; \tilde{\mathcal{D}} \mid \mathbf{\Gamma}) \tag{4}$$

where $I(\mathbf{U}; \mathbf{V}|\mathbf{W}) = \sum_w p(w) \sum_{u,v} p(u, v|w) \log \frac{p(u,v|w)}{p(u|w)p(v|w)}$ is the mutual information between $\mathbf{U}$ and $\mathbf{V}$ conditioned on $\mathbf{W}$.

The first term, $I(\mathcal{D}_{tr}; \tilde{\mathcal{D}} \mid \mathbf{\Gamma})$, functioning as the conditional generation term, facilitates the personalized separation of $\tilde{\mathcal{D}}$ from $\mathcal{D}_{tr}$ by minimizing the mutual information between them that takes into account user features. The second term, $I(\mathbf{Y}; \tilde{\mathcal{D}} \mid \mathbf{\Gamma})$, serving as the conditional regularization term, prompts the $\tilde{\mathcal{D}}$ to preserve user-specific target-relevant information from the true labels. By optimizing these two terms, $\tilde{\mathcal{D}}$ includes the *minimum sufficient personalized information* about the target, along with elements that account for user sensitivity. We further demonstrate that introducing additional user features to generate a diverse distribution contributes to enhancing the model's generalizability.

Theorem 3.3 (Generalization Bound) *Let $\hat{\mathbb{E}}_{\mathcal{D}_{tr}}[\ell(f; \mathbf{Y})]$ be the empirical loss on the training set, $\mathcal{D}$ be the unknown distribution, and $\mathbb{E}_{\mathcal{D}}[\ell(f; \mathbf{Y})]$ be the expected loss on $\mathcal{D}$. Given any finite hypothesis space $\mathcal{F}$ of models, suppose $f \in [M_1, M_2]$, we have that with probability at least $1 - \delta$:*

$$\mathbb{E}_{\mathcal{D}}[\ell(f; \mathbf{Y})] \leq \hat{\mathbb{E}}_{\mathcal{D}_{tr}}[\ell(f; \mathbf{Y})] + 2\mathcal{R}_n(\mathcal{F}) + (M_2 - M_1)\sqrt{\frac{\log \frac{2}{\delta}}{m}} \tag{5}$$

where $\mathcal{R}_n(\mathcal{F})$ is the rademacher complexity of $\mathcal{F}$, reflecting its capacity to model random noise within a dataset, inherently linked to the dataset's properties, and m is the amount of the user features.

The proof is presented in the Appendix A.2. Theorem 3.3 shows that an increased value of m results in a tighter bound for $\mathbb{E}_{\mathcal{D}}[\ell(f; \mathbf{Y})]$, which is upper bounded by $\hat{\mathbb{E}}_{\mathcal{D}_{tr}}[\ell(f; \mathbf{Y})]$, thereby enhancing the model's generalizability in unknown distribution.

3.3 Model Architecture and Optimization

In this section, we formally introduce the generation process for the distribution $\tilde{\mathcal{D}}$, involving learnable mask, distribution generator, and Transformer Recommender (Section 3.3.1). Following this, we detail the optimization with the CDIB (Section 3.3.2).

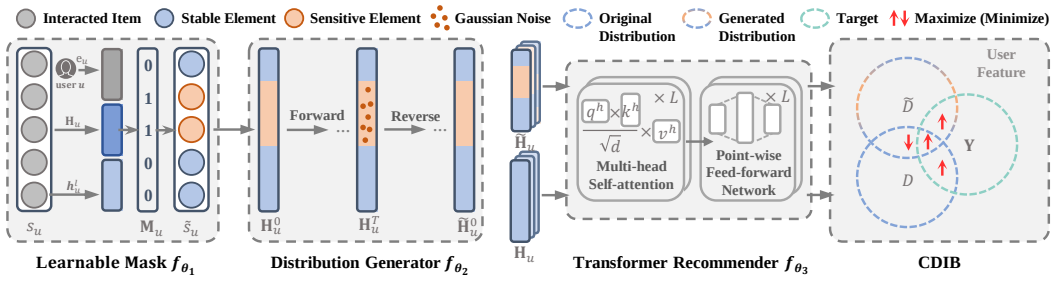


Figure 2: The overall framework of CDIB: The learnable mask first masks the stable elements, followed by the distribution generator augmenting the sensitive elements. Both original and augmented samples are then fed into the recommender to obtain $\mathcal{D}$ and $\tilde{\mathcal{D}}$, which are optimized using CDIB later.

3.3.1 Generation of Distribution $\tilde{\mathcal{D}}$

Embedding Layer. First, users and items are embedded into a d-dimensional latent space. Specifically, for user u and corresponding interaction sequence s_u, we obtain embedding vector $\mathbf{e}_u \in \mathbb{R}^d$ to capture the user features and embedding matrix $\mathbf{E}_u = \left(\mathbf{e}_u^1, ..., \mathbf{e}_u^L\right) \in \mathbb{R}^{L \times d}$ to model the item semantic, where $\mathbf{e}_u^l$ is the l-th item interacted by user u, and denote $\mathbf{\Gamma} = \{\mathbf{e}_u \mid \forall u \in \mathcal{U}\} \in \mathbb{R}^{|\mathcal{U}| \times d}$. Also, we initialize a learnable position embedding matrix $\mathbf{P} = (\mathbf{p}_1, ..., \mathbf{p}_L) \in \mathbb{R}^{L \times d}$ to model the temporal information, which is widely used in sequence modeling (Devlin et al., 2018; Sun et al., 2019). Thus, for a given sequence, we obtain the hidden representation $\mathbf{H}_u = \mathbf{E}_u + \mathbf{P}$ $(\mathbf{h}_u^l \in \mathbf{H}_u)$. To model the users' sensitivity to user features more effectively, we introduce a task based on the KL-divergence to align the user feature distribution with the interacted items' popularity distribution:

$$\mathcal{L}_{con} = \mathbb{E}_{\mathcal{D}_{tr}} \left[D_{KL} \left(\mathcal{N}\left(p\left(\mathbf{\Gamma}\right), \mathbf{I}\right) \parallel \mathcal{N}((O_{\mathbf{Y}} / \sum O_{\mathbf{Y}}), \mathbf{I}) \right) \right], \tag{6}$$

where $O_{\mathbf{Y}}$ represents the number of times each target has been observed, and $p(\cdot)$ is the sensitivity estimator, which is implemented using an MLP.

Learnable Mask Mechanism. We introduce a Learnable Mask that evaluates an item's significance within the interaction sequence considering user features as dictated by the CDIB principle. These measures work together to mask stable elements adaptively. An MLP network parameterized by θ_1 is employed to classify elements, summarized by the following formula:

$$\mathbf{M}_u = \left[\mathbf{M}_u^1, \cdots, \mathbf{M}_u^l, \cdots, \mathbf{M}_u^L\right], \text{where } \mathbf{M}_u^l = \sigma\left(\text{MLP}_{\theta_1}\left(\mathbf{h}_u^l \parallel \phi\left(\mathbf{H}_u\right) \parallel \mathbf{e}_u\right)\right), \tag{7}$$

where $\phi(\cdot)$ represents the interaction aggregation function, we have chosen to implement the mean aggregation. $\sigma(\cdot)$ denotes the sigmoid function, which scales the value of mask matrix $\mathbf{M}_u^l \in \mathbb{R}$ to (0, 1). To prevent $\mathbf{M}_u$ from converging on trivial solutions $\mathbf{0}$ during the diffusion process in the follow-up, we have formulated a self-supervised loss as a regularization: $\mathcal{L}_{mask} = -\sum_{u \in \mathcal{U}} \sum_{l=1}^{L} \mathbf{M}_u^l$.

Distribution Generator. Considering the impressive generative performance of the diffusion model (Rombach et al., 2022; Wang et al., 2023a) and computational efficiency, we utilize latent diffusion to generate more sensitive elements in the latent space. The core idea is to map all sensitive elements to a normal distribution by continuously adding noise, then sample from that distribution and denoise to generate a richer set of sensitive elements. Concretely, we first mask the stable elements by $\mathbf{H}_u^0 = \mathbf{H}_u \odot \mathbf{M}_u$, where $\odot$ is the broadcasted element-wise product, and then we incrementally introduced Gaussian noise into it, creating a sequence $\mathbf{H}_u^{1:T}$ through T steps in a Markov chain, which can be formulated as follows:

$$q\left(\mathbf{H}_u^t \mid \mathbf{H}_u^{t-1}\right) = \mathcal{N}\left(\mathbf{H}_u^t; \sqrt{1-\beta_t}\mathbf{H}_u^{t-1}, \beta_t \mathbf{I}\right), \tag{8}$$

where $\mathcal{N}$ indicates the Gaussian distribution and $\beta_t \in (0, 1)$ specifies the scale of noise introduced at each step t. Through the reparameterization trick and principle that the sum of two independent Gaussian noises is also Gaussian, $\mathbf{H}_u^t$ can be directly derived from $\mathbf{H}_u^0$ as $\mathbf{H}_u^t = \sqrt{\bar{\alpha}_t}\mathbf{H}_u^0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t$, with $\epsilon_t \sim \mathcal{N}\left(0, \mathbf{I}\right)$ as the added noise, and $\bar{\alpha}_t = \prod_{t'=1}^{T}\left(1-\beta_{t'}\right)$. After that, CDIB iteratively remove

the noise from $\mathbf{H}_u^t$ to reconstruct $\mathbf{H}_u^{t-1}$ and ultimately recover the original sample $\mathbf{H}_u^0$:

$$p_\theta\left(\mathbf{H}_u^{0:T}\right) = p\left(\mathbf{H}_u^T\right) \prod_{t=1}^{T} p_{\theta_2}\left(\mathbf{H}_u^{t-1} \mid \mathbf{H}_u^t\right), \tag{9}$$

where $p\left(\mathbf{H}_u^T\right) \sim \mathcal{N}(0, \mathbf{I})$ and $\prod_{t=1}^{T} p_{\theta_2}\left(\mathbf{H}_u^{t-1} \mid \mathbf{H}_u^t\right)$ denotes the process of sequentially deducing $\mathbf{H}_u^{t-1}$ by reversing the estimated Gaussian noise from $\mathbf{H}_u^t$ via a lightweight MLP network parameterized by θ_2. The learning objective is thereby distilled to:

$$\mathcal{L}_{diff} = \sum_{t=2}^{T} \mathbb{E}_{t,\epsilon}\left[\|\epsilon_t - \epsilon_{\theta_2}\left(\mathbf{H}_u^t, t\right)\|_2^2\right], \tag{10}$$

where ϵ_t represents the noise have been added to $\mathbf{H}_u^{t-1}$ in the forward process. Then, CDIB generate the external factors by firstly corrupting $\mathbf{H}_u^0$ via Equation 8, and executing reverse denoising on corrupted representation via Equation 9 to obtain the rich sensitive elements denoted as $\tilde{\mathbf{H}}_u^0$, then we obtain the generated sequence embedding as $\tilde{\mathbf{H}}_u = \tilde{\mathbf{H}}_u^0 + \mathbf{H}_u \odot (1 - \mathbf{M}_u)$.

Transformer Recommender. The effectiveness of the Transformer in learning sequential patterns inspires us to use it for the main recommendation task (Devlin et al., 2018; Vaswani et al., 2023). The core component of the Transformer architecture is the multi-head self-attentive mechanism, which can be formulated as follows:

$$\mathbf{H}_u^o = \varphi\left(\Big\|_{h=1}^{H} \mathbf{H}_u^h\right); \quad \mathbf{H}_u^h = \text{softmax}\left(\frac{\mathbf{H}_u\mathbf{W}_Q^h(\mathbf{H}_u\mathbf{W}_K^h)^T}{\sqrt{d/h}}\right)\mathbf{H}_u\mathbf{W}_V^h, \tag{11}$$

where $\varphi(\mathbf{x}) = \text{GELU}(\mathbf{W}\mathbf{x} + \mathbf{b})$, and $\mathbf{H}_u^o \in \mathbb{R}^{L\times d}$ represents the refined embedding of the items. This embedding is derived by concatenating $\mathbf{H}_u^h \in \mathbb{R}^{L\times d/H}$ then being calculated by $\varphi(\cdot)$. We also process the generated sequence embedding $\tilde{\mathbf{H}}_u$ through the Transformer to obtain the refined embedding $\mathbf{H}_u^g$. For both $\mathbf{H}_u^o$ and $\mathbf{H}_u^g$, the last position vector is considered as the representation of the entire interaction sequence (Kang & McAuley, 2018), denoted as $\mathbf{h}_u^o$ and $\mathbf{h}_u^g$, respectively. They share the same labels (*i.e.*, i_u^{L+1}). Notice that the distributions of all $\mathbf{h}_u^o$ and $\mathbf{h}_u^g$ are $\mathcal{D}_{tr}$ and $\tilde{\mathcal{D}}$, respectively. In the end, the total loss of generating distribution is $\mathcal{L}_{gd} = \mathcal{L}_{con} + \mathcal{L}_{mask} + \mathcal{L}_{diff}$.

3.3.2 Model Optimization with CDIB

Maximizing $I(\mathbf{Y}; \tilde{\mathcal{D}}|\mathbf{\Gamma})$. Directly maximizing the conditional regularization term proves challenging. Hence, according to (Choi & Lee, 2023), we instead derive and maximize the lower bound of $I(\mathbf{Y}; \tilde{\mathcal{D}}, \mathbf{\Gamma})$ via variational decomposition (*cf.* Appendix A.3), outlined as follows:

Proposition 3.4 (Lower bound of $I(\mathbf{Y}; \tilde{\mathcal{D}}, \mathbf{\Gamma})$) *Given label $\mathbf{Y}$, distribution $\tilde{\mathcal{D}}$, and user features $\mathbf{\Gamma}$, we have:*

$$I\left(\mathbf{Y}; \tilde{\mathcal{D}}, \mathbf{\Gamma}\right) \geq \mathbb{E}_{\mathbf{Y}, \tilde{\mathcal{D}}, \mathbf{\Gamma}}\left[\log p_{\theta_3}\left(\mathbf{Y} \mid \tilde{\mathcal{D}}, \mathbf{\Gamma}\right)\right] \tag{12}$$

where $\log p_{\theta_3}(\mathbf{Y} \mid \tilde{\mathcal{D}}, \mathbf{\Gamma})$ is the variational approximation of $\log p(\mathbf{Y} \mid \tilde{\mathcal{D}}, \mathbf{\Gamma})$.

Given that the generation process of $\tilde{\mathcal{D}}$ incorporates the user features $\mathbf{\Gamma}$ as specified in Equation 7, we have $\log p_{\theta_3}(\mathbf{Y}|\tilde{\mathcal{D}}, \mathbf{\Gamma}) = \log p_{\theta_3}(\mathbf{Y}|\tilde{\mathcal{D}})$. Note that $\log p_{\theta_3}(\mathbf{Y}|\tilde{\mathcal{D}})$ essentially represents a recommendation task, where the input is the generated distribution ($h_u^g \in \tilde{\mathcal{D}}$) and the output is the next interacted item ($i_u^{L+1} \in \mathbf{Y}$). This suggests that the model is also trained to make recommendations beyond the training distribution. We employ the Transformer recommender f_{θ_3} to optimize this task using the objective defined in Equation 1, with the associated loss denoted as $\mathcal{L}_{reg}$.

Minimizing $I(\mathcal{D}_{tr}; \tilde{\mathcal{D}}|\mathbf{\Gamma})$. To minimize the conditional generation term, we first employ the chain rule for mutual information[2], applying it as follows: $I(\mathcal{D}_{tr}; \tilde{\mathcal{D}}|\mathbf{\Gamma}) = I(\mathcal{D}_{tr}; \tilde{\mathcal{D}}, \mathbf{\Gamma}) - I(\mathcal{D}_{tr}; \mathbf{\Gamma})$. Notice

[2]Given the random variables $\mathbf{X}$, $\mathbf{V}$, and $\mathbf{Z}$, then the chain rule gives $I(\mathbf{X}; \mathbf{V}|\mathbf{Z}) = I(\mathbf{X}, \mathbf{Z}; \mathbf{V}) - I(\mathbf{Z}; \mathbf{V})$.

that $I(\mathcal{D}_{tr}; \tilde{\mathcal{D}}, \mathbf{\Gamma}) = I(\mathcal{D}_{tr}; \tilde{\mathcal{D}})$ (*cf.* Appendix A.4). Intuitively, minimizing the first term personalized drives $\tilde{\mathcal{D}}$ away from $\mathcal{D}_{tr}$, thereby fostering a diverse distribution exploration. Maximizing the second term seeks to capture the personalized information from $\mathcal{D}_{tr}$ into $\mathbf{\Gamma}$. Inspired by (Wei et al., 2022), we adopt negative InfoNCE to estimate the mutual information (Gutmann & Hyvärinen, 2010) and contrastive learning to minimize the $I(\mathcal{D}_{tr}; \tilde{\mathcal{D}}|\mathbf{\Gamma})$. Specifically, for the $I(\mathcal{D}_{tr}; \tilde{\mathcal{D}})$, we treat the original sequence embedding $\mathbf{h}_u^o \in \mathcal{D}_{tr}$ and the corresponding augmented sequence embedding $\mathbf{h}_u^g \in \tilde{\mathcal{D}}$ as positive pairs, with in-batch instances serving as negative samples. For $I(\mathcal{D}_{tr}; \mathbf{\Gamma})$, the original sequence embedding $\mathbf{h}_u^o$ and the corresponding user embedding $\mathbf{e}_u$ are considered positive pairs, again with in-batch instances as negative samples. We define the contrastive loss as follows:

$$\mathcal{L}_{gen} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \left[\log \frac{e^{\phi(\mathbf{h}_u^o, \mathbf{h}_u^g)/\tau}}{\sum_{u' \in \mathcal{U}} e^{\phi(\mathbf{h}_u^o, \mathbf{h}_{u'}^g)/\tau)}} - \log \frac{e^{\phi(\mathbf{h}_u^o, \mathbf{e}_u)/\tau)}}{\sum_{u' \in \mathcal{U}} e^{\phi(\mathbf{h}_u^o, \mathbf{e}_{u'})/\tau)}} \right], \tag{13}$$

where $\phi(\cdot)$ denotes the similarity function and τ denotes the tunable temperature hyper-parameter to adjust the scale for softmax.

Overall Objective. Finally, we train the model using the specified final objective as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{pred} + \alpha_1 \mathcal{L}_{gd} + \alpha_2 \left(\beta \mathcal{L}_{reg} + \mathcal{L}_{gen} \right), \tag{14}$$

where $\mathcal{L}_{pred}$ is the primary recommendation loss, calculated by the f_{θ_3} (which is also employed to optimize the conditional regularization term), where the input is the training data ($h_u^o \in \mathcal{D}_{tr}$), and the output is the next item interacted with. The α_1, α_2 represent tunable hyperparameters that balance the significance of auxiliary losses. Ultimately, the trained f_{θ_3} serves as the ξ^* in the testing stage.

4 Experiment

Datasets. Our experiments are conducted on four public real-world datasets, *i.e.*, *MovieLens-100K, Retailrocket, Amazon-Beauty, and Amazon-Sports*. For each dataset, we chronologically select 80% of the historical interactions of each user as the training set, 10% of those as the validation set, and the remaining 10% as the test set. The detailed information is in Appendix E.1.

Baselines. We compare CDIB with nine methods from diverse research lines, covering (i) Naive Sequential Recommendation Methods: **GRU4Rec** Hidasi et al. (2016), **Caser** Tang & Wang (2018), and **SASRec** Kang & McAuley (2018). (ii) Reweighting Methods: **IPS** Schnabel et al. (2010). (iii) DRO Methods: **S-DRO** Wen et al. (2022) and **DROS** Yang et al. (2023b). (iv) Diffusion-based Augmentation Methods: **DiffuASR-CG** and **DiffuASR-CF** (Liu et al., 2023). (v) Contrastive Learning (CL) Methods: **CL4SRec** Xie et al. (2022), **DuoRec** Qiu et al. (2022), and **DCRec** Yang et al. (2023a). The details are in Appendix E.3.

4.1 Performance Comparison

Overall Performance Comparison. We assess the methods using the all-ranking protocol He et al. (2020), focusing on *HR@10* and *NDCG@10* metrics. The results are shown in Table 1, and we have several observations: **(i)** The DRO and CL methods outperform naive sequential recommendation models, demonstrating their effectiveness. Specifically, compared with SASRec, DROS shows improvements on the *ML-100K* and *Sports*, while DuoRec progresses on the *Retail* and *Beauty*. However, IPS and S-DRO only achieve marginal improvements or perform worse, suggesting their limitations when dealing with sparse data. **(ii)** Additionally, the efficacy of CL methods appeared to be hindered on the *Sports*, whose average interaction sequence length is the shortest. This indicates a sensitivity to hand-crafted data augmentation, which may limit the success of CL methods. **(iii)** Our model consistently outperforms the baseline models across all datasets, showing the effectiveness of the learnable data augmentation method and the optimization strategy with CDIB, which can create diverse and promising distributions and capture more robust information.

Robustness to Distribution Shift. To further evaluate the robustness of our model to distribution shifts, we conduct experiments and compare its performance to that of representative models across different time gaps. The results are shown in Figure 3, where T1 denotes the training stage, and T2 through T7 represents the testing stages, each with an increasing time gap. As the gap size increases, the overall accuracy of the baseline models generally shows a downward trend, highlighting the

Table 1: Overall performance. The best results and second-best are in **bold** and underline. All the numbers are percentage values with "%" omitted (mean±std). ♣ is the model's variants in the ablation study. The experiments are conducted 5 times.

Method	MovieLens-100K		Retailrocket		Amazon-Beauty		Amazon-Sports	
	HitRate ↑	NDCG ↑	HitRate ↑	NDCG ↑	HitRate ↑	NDCG ↑	HitRate ↑	NDCG ↑
GRU4Rec	10.26±0.22	4.90±0.09	12.03±0.29	5.90±0.10	6.49±0.27	3.44±0.16	3.47±0.15	1.80±0.10
Caser	6.22±0.39	2.90±0.21	7.28±0.26	3.16±0.18	3.74±0.13	1.83±0.07	2.02±0.11	1.00±0.07
SASRec	10.96±0.12	4.84±0.05	19.78±0.14	8.67±0.07	8.64±0.13	4.29±0.06	4.76±0.05	2.22±0.02
IPS	10.97±0.10	4.85±0.03	19.65±0.16	8.60±0.08	8.71±0.08	4.31±0.03	4.74±0.07	2.21±0.03
S-DRO	10.90±0.13	4.82±0.05	19.70±0.20	8.64±0.09	8.63±0.14	4.27±0.06	4.74±0.07	2.22±0.02
DROS	11.30±0.11	5.23±0.06	18.79±0.16	8.65±0.07	8.33±0.13	4.14±0.10	4.81±0.07	2.32±0.06
DiffuASR-CG	11.18±0.22	5.13±0.07	20.31±0.21	8.84±0.09	8.33±0.12	4.13±0.10	4.70±0.09	2.19±0.04
DiffuASR-CF	11.24±0.21	5.19±0.06	20.51±0.23	8.97±0.12	8.46±0.12	4.26±0.08	4.79±0.14	2.23±0.05
CL4SRec	11.07±0.35	5.16±0.07	19.72±0.26	8.67±0.10	8.80±0.06	4.39±0.05	4.77±0.14	2.26±0.04
DuoRec	11.21±0.17	5.17±0.06	20.63±0.11	9.10±0.06	8.74±0.41	4.41±0.19	4.49±0.10	2.21±0.04
DCRec	10.63±0.50	4.50±0.34	20.22±0.31	8.82±0.11	7.99±0.38	3.90±0.24	4.08±0.39	1.97±0.16
w/o LM♣	10.32±0.27	4.90±0.12	20.01±0.23	9.03±0.13	8.56±0.31	4.26±0.13	4.54±0.17	2.27±0.05
w/o LD♣	11.59±0.16	5.53±0.12	20.82±0.16	9.37±0.04	9.06±0.15	**4.60±0.06**	4.92±0.21	**2.40±0.08**
w/o IB♣	10.98±0.19	4.88±0.12	19.72±0.32	8.62±0.14	8.68±0.13	4.32±0.04	4.63±0.08	2.17±0.03
CDIB (Ours)	**11.89±0.16**	**5.67±0.09**	**21.12±0.14**	**9.41±0.10**	**9.17±0.04**	4.56±0.04	**4.95±0.11**	2.38±0.07

severe negative impact of temporal distribution shifts. For example, SASRec's performance drop 72.48% from T1 to T7 under the *Retailrocket* datasets, whereas CDIB remains more stable with the drop rate of 67.97%. We attribute it to the fact that generated distribution allows the model to recognize and adapt to these out-of-distribution situations to a certain extent at the training stage.

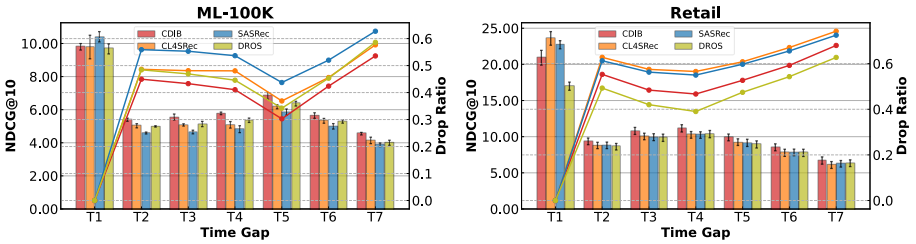


Figure 3: Models performance *w.r.t* time gap.

Performance on Different User Group. We investigated the model's effectiveness across different user groups as shown in Figure 4, with U1 representing niche users and U5 representing blockbuster users. User conformity increases progressively from U1 to U5. The experimental results reveal that the model's effectiveness declines as user conformity decreases, indicating the model's vulnerability to the influence of item popularity while neglecting individual user attributes. Throughout these tests, our model is basically superior to the baseline models. We attribute this superior performance to our optimization strategy, which utilizes user attributes to guide model optimization. This approach allows the model to better capture users' personalized interest and recommend more relevant content.

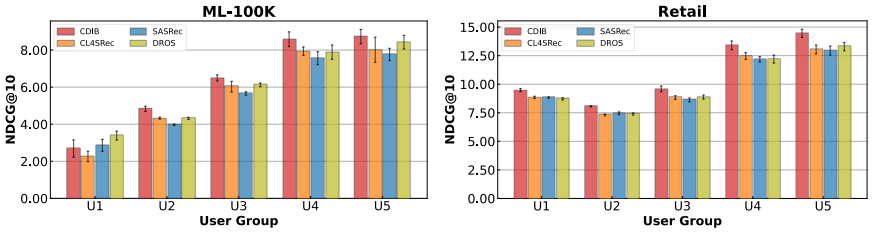


Figure 4: Models performance *w.r.t* user group.

4.2 SENSITIVITY ANALYSIS ON β

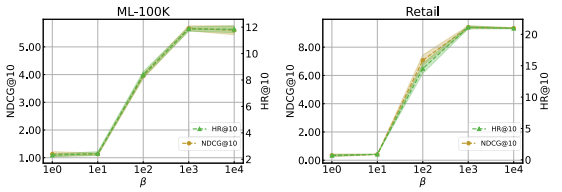


Figure 5: Sensitivity Analysis on β under *ML-100K* and *Retailrocket* datasets.

In this section, we analyze the model's sensitivity to β, which controls the trade-off between out-of-distribution exploration and prediction accuracy. The results are shown in Figure 5. Our observations are as follows: **(i)** The model fails to converge when $\beta \leq 1e1$. This issue arises because such low values of β encourage the model to aggressively generate distributions beyond the training distribution's scope without preserving stable factors, thereby introducing harmful noise. **(ii)** As β increases from $1e1$ to $1e3$, performance improves. However, when β reaches $1e4$, performance declines, possibly due to the model's excessive focus on prediction at the expense of sufficient out-of-distribution exploration. Thus, a tailored β is needed to balance the two.

4.3 ABLATION STUDY

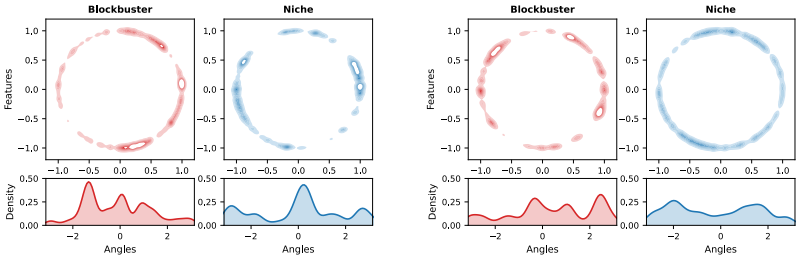
In this section, we explore the design rationale of sub-modules within our CDIB framework. We remove key modules to implement three variants of CDIB: **(i)** "w/o LM": CDIB without the learnable mask, setting $\mathbf{M}_u = \mathbf{1}$. **(ii)** "w/o IB": CDIB without optimization with ***CDIB***, using contrastive learning with InfoNCE loss instead. **(iii)** "w/o LD": CDIB without latent diffusion. Gaussian noise is added to $\mathbf{H}_u$ to get $\tilde{\mathbf{H}}_u$ (*i.e.*, $\tilde{\mathbf{H}}_u = \mathcal{N}(0, \mathbf{I}) \odot \mathbf{M}_u + \mathbf{H}_u \odot (1 - \mathbf{M}_u)$. From the results (Table 1), we observe that: **(i)** Removing the learnable mask significantly degrades performance, underscoring its essential role in identifying elements to be disturbed during data augmentation. Without this component, the model may fail to capture genuine user interest reflected in the interaction sequence, potentially leading to misguided model optimization. **(ii)** Removing latent diffusion for generating distribution shows a performance decline. However, on the *Beauty* and *Sports* datasets, where the average sequence length is the shortest, w/o LD performs better. **(iii)** The gap in performance between CDIB and w/o IB highlights its effectiveness in guiding the distribution generation process and boosting the model's generalization. The performance of w/o IB closely matches that of SASRec, which can be attributed to maximization of $I(\mathcal{D}_{tr}; \tilde{\mathcal{D}})$ in the standard contrastive learning with InfoNCE loss. Specifically, $\mathbf{M}_u$ may converge on trivial solutions $\mathbf{0}$ to fulfill the CL task, leading to $\mathcal{D}_{tr} = \tilde{\mathcal{D}}$. This indicates no OOD exploration, the same as SASRec.

4.4 VISUALIZATION OF BLOCKBUSTER AND NICHE USERS' PREFERENCE

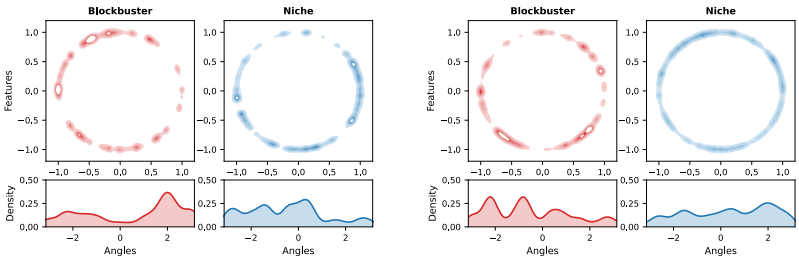
We visualized the interest distributions of blockbuster and niche users learned by CDIB on the *ML100K* and *Retailrocket* datasets during the testing stage. For both SASRec and CDIB, the preference distributions of blockbuster users exhibit significant clustering, likely around popular items (hotspots). However, for niche users, CDIB, compared to SASRec, shows a more uniform preference distribution and is less influenced by popular items, indicating that CDIB effectively models niche users without being affected by new trends, showing the rationality of our model design.

5 RELATED WORK

Sequential Recommendation is designed to predict the next item a user is likely to prefer based on their interaction history. Traditional methods have leveraged Markov chains to capture first-order item-to-item correlations through transition matrices (Rendle et al., 2010; He & McAuley, 2016). With the development of deep learning, which excels at modeling complex sequential patterns, various deep recommendation models have been developed. For instance, GRU4Rec (Hidasi et al., 2016) employs Gated Recurrent Unit (GRU) units to model the temporal dynamics of interaction sequences. SASRec (Kang & McAuley, 2018) and BERT4Rec (Sun et al., 2019) enhance computational efficiency in lengthy sequences by incorporating self-attention mechanisms. More recently, inspired by selective state space models (Gu & Dao, 2024), Mamba4Rec (Liu et al., 2024) has been introduced, utilizing the mamba framework to recommend items efficiently. Despite their capabilities, these models often suffer performance declines when OOD occurs. To address this, CDIB introduces a user



(a) Preferences learned by **SASRec** on ml100k (left two) and retail (right two).



(b) Preferences learned by **CDIB** on ml100k (left two) and retail (right two)

Figure 6: We visualize preference distributions using Gaussian kernel density estimation (KDE) in $\mathbb{R}^2$ and von Mises-Fisher (vMF) KDE for angular data (*i.e.*, $\arctan 2(y, x)$ for each point (x, y)).

feature-guided generation approach that proactively explores OOD scenarios during the training phase, enhancing the model's generalization capabilities.

Distributionally Robust Sequential Recommendation has recently attracted significant research interest, which aims to train a model that performs well not only at the training stage but also at the testing stage. Methods DRO (Schnabel et al., 2010; Bottou et al., 2013; Wang et al., 2022b; Yang et al., 2023b; Wen et al., 2022) optimize the model for the worst-case distribution to improve the robustness. For example, DROS (Yang et al., 2023b) unifies the DRO and sequential recommendation paradigms to enhance model robustness against distribution shifts. Causal inference methods capture real causal relationships but assume the causal graph is static (Wang et al., 2023b; He et al., 2022; Yang et al., 2020; Wang et al., 2022a), these methods face challenges with sparse data. While contrastive learning approaches seek to enrich the training data distribution through data augmentation (Liu et al., 2021; Xie et al., 2022; Yang et al., 2023a; Qiu et al., 2022; Zhao et al., 2023), but hardly rely on the hand-crafted data augmentation strategies. To fill the gap, we introduce the CDIB principle, using the user features to guide the exploration of the other distribution.

Information Bottleneck with Conditional Information is also widely utilized. The CIB (Gondek & Hofmann, 2003) theory has been applied in CGIB (Lee et al., 2023) to identify key structures in molecules that predict interaction behaviors between graph pairs, focusing on important subgraphs. Additionally, TimeCIB (Choi & Lee, 2023) extends the CIB to impute time series data, preserving vital temporal information. To the best of our knowledge, CDIB marks the first use of CIB to guide the distribution generation process. The detailed introduction of related works is in Appendix E.4.

6 Conclusion

In this work, to address the limitations of existing methods that struggle with sparse data or depend on hand-crafted augmentations, we introduce CDIB, an innovative principle that guides the generation of diverse and reliable distributions based on user features. Theoretical analyses demonstrate the rationality of this approach. Building on CDIB, we propose a framework that employs a learnable method to generate distributions for OOD exploration, guided by a conditional generation term and a conditional regularization term. Extensive experiments on four public datasets confirm the effectiveness and robustness of our model.

REFERENCES

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations (ICLR)*, 2016.

Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X. Charles, D. Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research (JMLR)*, 14(11), 2013.

Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. Controllable multi-interest framework for recommendation. In *International Conference on Knowledge Discovery & Data Mining (KDD)*, pp. 2942–2951, 2020.

MinGyu Choi and Changhee Lee. Conditional information bottleneck approach for time series imputation. In *International Conference on Learning Representations (ICLR)*, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

David Gondek and Thomas Hofmann. Conditional information bottleneck clustering. In *International Conference on Data Mining (ICDM)*, pp. 36–42. IEEE, 2003.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2024.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 297–304. JMLR Workshop and Conference Proceedings, 2010.

Ido Guy, Naama Zwerdling, Inbal Ronen, David Carmel, and Erel Uziel. Social media recommendation based on people and tags. In *International Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 194–201, 2010.

Ruining He and Julian McAuley. Fusing similarity models with markov chains for sparse sequential recommendation. In *International Conference on Data Mining (ICDM)*, pp. 191–200. IEEE, 2016.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *International Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 639–648, 2020.

Yue He, Zimu Wang, Peng Cui, Hao Zou, Yafeng Zhang, Qiang Cui, and Yong Jiang. Causpref: Causal preference learning for out-of-distribution recommendation. In *The Web Conference (WWW)*, pp. 410–421, 2022.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In *International Conference on Learning Representations (ICLR)*, 2016.

Jonathan Ho, Ajay Jain, and Abbeel Pieter. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:6840–6851, 2020.

Zhaolin Hu and L Jeff Hong. Kullback-leibler divergence constrained distributionally robust optimization. *Available at Optimization Online (AOO)*, 1(2):9, 2013.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *International Conference on Data Mining (ICDM)*, pp. 197–206. IEEE, 2018.

Namkyeong Lee, Dongmin Hyun, Gyoung S. Na, Sungwon Kim, Junseok Lee, and Chanyoung Park. Conditional graph information bottleneck for molecular relational learning. In *International Conference on Machine Learning (ICML)*, pp. 18852–18871, 2023.

Chengkai Liu, Jianghao Lin, Jianling Wang, Hanzhou Liu, and James Caverlee. Mamba4rec: Towards efficient sequential recommendation with selective state space models. *arXiv preprint arXiv:2403.03900*, 2024.

Qidong Liu, Fan Yan, Xiangyu Zhao, Zhaocheng Du, Huifeng Guo, Ruiming Tang, and Feng Tian. Diffusion augmentation for sequential recommendation. In *International Conference on Information & Knowledge Management (CIKM)*, pp. 1576–1586, 2023.

Zhiwei Liu, Yongjun Chen, Jia Li, Philip S Yu, Julian McAuley, and Caiming Xiong. Contrastive self-supervised sequential recommendation with robust augmentation. *arXiv preprint arXiv:2108.06479*, 2021.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. Contrastive learning for representation degeneration problem in sequential recommendation. In *International Conference on Web Search and Data Mining (WSDM)*, pp. 813–823, 2022.

Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *The Web Conference (WWW)*, pp. 811–820, 2010.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and BjÖrn Ommer. High-resolution image synthesis with latent diffusion models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2022.

Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, and Schmidt-Thieme Lars. Recommendations as treatments: Debiasing learning and evaluation. In *The Web Conference (WWW)*, pp. 811–820, 2010.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *International Conference on Information & Knowledge Management (CIKM)*, pp. 1441–1450, 2019.

Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *International Conference on Web Search and Data Mining (WSDM)*, pp. 565–573, 2018.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *IEEE Information Theory Workshop (ITW)*, pp. 1–5. IEEE, 2015.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research (JMLR)*, 9(11), 2008.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 30:18109–18131, 2023.

Bohao Wang, Jiawei Chen, Changdong Li, Sheng Zhou, Qihao Shi, Yang Gao, Yan Feng, Chun Chen, and Can Wang. Distributionally robust graph-based recommendation system. In *The Web Conference (WWW)*, pp. 3777–3788, 2024.

Wenjie Wang, Xinyu Lin, Fuli Feng, Xiangnan He, Min Lin, and Tat-Seng Chua. Causal representation learning for out-of-distribution recommendation. In *The Web Conference (WWW)*, pp. 3562–3571, 2022a.

Wenjie Wang, Yiyan Xu, Fuli Feng, Xinyu Lin, Xiangnan He, and Tat-Seng Chua. Diffusion recommender model. In *International Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 832–841, 2023a.

Zhenlei Wang, Shiqi Shen, Zhipeng Wang, Bo Chen, Xu Chen, and Ji-Rong Wen. Unbiased sequential recommendation with latent confounders. In *The Web Conference (WWW)*, pp. 2195–2204, 2022b.

Zhenlei Wang, Xu Chen, Rui Zhou, Quanyu Dai, Zhenhua Dong, and Ji-Rong Wen. Sequential recommendation with user causal behavior discovery. In *International Conference on Data Engineering (ICDE)*, pp. 28–40. IEEE, 2023b.

Chunyu Wei, Jian Liang, Di Liu, and Fei Wang. Contrastive graph structure learning via information bottleneck for recommendation. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:20407–20420, 2022.

Hongyi Wen, Xinyang Yi, Tiansheng Yao, Jiaxi Tang, Lichan Hong, and Ed H. Chi. Distributionally-robust recommendations for improving worst-case user experience. In *The Web Conference (WWW)*, pp. 3606–3610, 2022.

Zihao Wu, Xin Wang, Hong Chen, Kaidong Li, Yi Han, Lifeng Sun, and Wenwu Zhu. Diff4rec: Sequential recommendation with curriculum-scheduled diffusion augmentation. In *ACM International Conference on Multimedia (ACMMM)*, pp. 9329–9335, 2023.

Lianghao Xia, Yizhen Shao, Chao Huang, Yong Xu, Huance Xu, and Jian Pei. Disentangled graph social recommendation. In *International Conference on Data Engineering (ICDE)*, pp. 2332–2344. IEEE, 2023.

Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. Contrastive learning for sequential recommendation. In *International Conference on Data Engineering (ICDE)*, pp. 1259–1273. IEEE, 2022.

Chenxiao Yang, Qitian Wu, Qingsong Wen, Zhiqiang Zhou, Liang Sun, and Junchi Yan. Towards out-of-distribution sequential event prediction: A causal treatment. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:19314–19326, 2020.

Yuhao Yang, Chao Huang, Lianghao Xia, Chunzhen Huang, Da Luo, and Kangyi Lin. Debiased contrastive learning for sequential recommendation. In *The Web Conference (WWW)*, pp. 1063–1073, 2023a.

Zhengyi Yang, Xiangnan He, Jizhi Zhang, Jiancan Wu, Xin Xin, Jiawei Chen, and Xiang Wang. A generic learning framework for sequential recommendation with distribution shifts. In *International Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 331–340, 2023b.

Yaowen Ye, Lianghao Xia, and Chao Huang. Graph masked autoencoder for sequential recommendation. In *International Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 321–330, 2023.

Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *International Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 1294–1303, 2022.

An Zhang, Jingnan Zheng, Xiang Wang, Yancheng Yuan, and Chua Tat-Seng. Invariant collaborative filtering to popularity distribution shift. In *The Web Conference (WWW)*, pp. 1240–1251, 2023.

Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. Causal intervention for leveraging popularity bias in recommendation. In *International Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 11–20, 2021.

Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, et al. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *International Conference on Information & Knowledge Management (CIKM)*, pp. 4653–4664, 2021.

Weiqi Zhao, Dian Tang, Xin Chen, Dawei Lv, Daoli Ou, Biao Li, Peng Jiang, and Kun Gai. Disentangled causal embedding with contrastive learning for recommender system. In *The Web Conference (WWW)*, pp. 406–410, 2023.

Jiawei Zheng, Qianli Ma, Hao Gu, and Zheng Zhenjing. Multi-view denoising graph auto-encoders on heterogeneous information networks for cold-start recommendation. In *International Conference on Knowledge Discovery & Data Mining (KDD)*, pp. 2338–2348, 2021.

A PROOFS

A.1 PROOF OF RATIONALITY OF DIB

Proof. Before the proof, we propose the following two assumptions:

(a) $\mathbf{X}_s$ and $\mathbf{Y}_s$ follow the same distribution.

(b) Since $\tilde{\mathcal{D}}$ is generated by latent diffusion model without introducing any information from $\mathbf{Y}$, $\tilde{\mathbf{X}}_e$ and $\mathbf{Y}_e$ are mutually independent.

Notice that $\mathbf{Y}_s$ and $\mathbf{Y}_e$ refer to the sensitive and stable attributes of the target item, respectively. For instance, considering a pair of shorts, $\mathbf{Y}_s$ would denote its stable features, such as the brand, while $\mathbf{Y}_e$ would indicate sensitive features like their status as a seasonal trend in summer.

We then obtain the fact that all stable factors and external factors are orthogonal. If this does not stand, $\exists\ x_s \in \mathbf{X}_s, x_e \in \mathbf{X}_e$ *s.t.* $corr(x_s, x_e) \neq 0$. When all external features other than x_e remain unchanged while x_e changes, x_s changes correspondingly. This contradicts with $\mathbf{X}_s$ is stable.

We are now ready to prove the proposition. With this fact, we can derive the following:

$$
\begin{aligned}
&I(\mathcal{D}_{tr};\tilde{\mathcal{D}}) - \beta I(\mathbf{Y};\tilde{\mathcal{D}}) \\
&\quad = I(\mathbf{X}_s,\mathbf{X}_e;\tilde{\mathbf{X}}_s,\tilde{\mathbf{X}}_e) - \beta I(\mathbf{Y}_s,\mathbf{Y}_e;\tilde{\mathbf{X}}_s,\tilde{\mathbf{X}}_e) \\
&\quad = I(\mathbf{X}_s;\tilde{\mathbf{X}}_s,\tilde{\mathbf{X}}_e) + I(\mathbf{X}_e;\tilde{\mathbf{X}}_s,\tilde{\mathbf{X}}_e|\mathbf{X}_s) - \beta(I(\mathbf{Y}_s;\tilde{\mathbf{X}}_s,\tilde{\mathbf{X}}_e) + I(\mathbf{Y}_e;\tilde{\mathbf{X}}_s,\tilde{\mathbf{X}}_e|\mathbf{Y}_s)) \\
&\quad = I(\mathbf{X}_s;\tilde{\mathbf{X}}_s,\tilde{\mathbf{X}}_e) + I(\mathbf{X}_e;\tilde{\mathbf{X}}_s,\tilde{\mathbf{X}}_e,\mathbf{X}_s) - I(\mathbf{X}_e;\mathbf{X}_s) - \\
&\qquad \beta(I(\mathbf{Y}_s;\tilde{\mathbf{X}}_s,\tilde{\mathbf{X}}_e) + I(\mathbf{Y}_e;\tilde{\mathbf{X}}_s,\tilde{\mathbf{X}}_e,\mathbf{Y}_s) - I(\mathbf{Y}_e;\mathbf{Y}_s)) \\
&\quad = I(\mathbf{X}_s;\tilde{\mathbf{X}}_s,\tilde{\mathbf{X}}_e) + I(\mathbf{X}_e;\tilde{\mathbf{X}}_s,\tilde{\mathbf{X}}_e) - \beta(I(\mathbf{Y}_s;\tilde{\mathbf{X}}_s,\tilde{\mathbf{X}}_e) + I(\mathbf{Y}_e;\tilde{\mathbf{X}}_s,\tilde{\mathbf{X}}_e)) \\
&\quad = I(\mathbf{X}_s;\tilde{\mathbf{X}}_s) + I(\mathbf{X}_s;\tilde{\mathbf{X}}_e|\tilde{\mathbf{X}}_s) + I(\mathbf{X}_e;\tilde{\mathbf{X}}_e) + I(\mathbf{X}_e;\tilde{\mathbf{X}}_s|\tilde{\mathbf{X}}_e) - \\
&\qquad \beta(I(\mathbf{Y}_s;\tilde{\mathbf{X}}_s) + I(\mathbf{Y}_s;\tilde{\mathbf{X}}_e|\tilde{\mathbf{X}}_s) + I(\mathbf{Y}_e;\tilde{\mathbf{X}}_e) + I(\mathbf{Y}_e;\tilde{\mathbf{X}}_s|\tilde{\mathbf{X}}_e)) \\
&\quad = I(\mathbf{X}_s;\tilde{\mathbf{X}}_s) + I(\mathbf{X}_s,\tilde{\mathbf{X}}_s;\tilde{\mathbf{X}}_e) - I(\tilde{\mathbf{X}}_s;\tilde{\mathbf{X}}_e) + I(\mathbf{X}_e;\tilde{\mathbf{X}}_e) + I(\mathbf{X}_e,\tilde{\mathbf{X}}_e;\tilde{\mathbf{X}}_s) - I(\tilde{\mathbf{X}}_s;\tilde{\mathbf{X}}_e) - \\
&\qquad \beta(I(\mathbf{Y}_s;\tilde{\mathbf{X}}_s) + I(\mathbf{Y}_s,\tilde{\mathbf{X}}_s;\tilde{\mathbf{X}}_e) - I(\tilde{\mathbf{X}}_s;\tilde{\mathbf{X}}_e) + I(\mathbf{Y}_e;\tilde{\mathbf{X}}_e) + I(\mathbf{Y}_e,\tilde{\mathbf{X}}_e;\tilde{\mathbf{X}}_s) - I(\tilde{\mathbf{X}}_s;\tilde{\mathbf{X}}_e)) \\
&\quad = I(\mathbf{X}_s;\tilde{\mathbf{X}}_s) + I(\mathbf{X}_e;\tilde{\mathbf{X}}_e) - \beta I(\mathbf{Y}_s;\tilde{\mathbf{X}}_s) - \beta I(\mathbf{Y}_e;\tilde{\mathbf{X}}_e) \\
&\quad = \gamma_1 I(\mathbf{X}_s;\tilde{\mathbf{X}}_s) + \gamma_2 I(\mathbf{X}_e;\tilde{\mathbf{X}}_e) - \beta\gamma_3 I(\tilde{\mathbf{X}}_s;\mathbf{Y}_s) - \beta\gamma_4 I(\tilde{\mathbf{X}}_e;\mathbf{Y}_e)
\end{aligned}
\tag{15}
$$

The second equation is derived from the $I(\mathbf{X},\mathbf{Y};\mathbf{Z},\mathbf{V}) = I(\mathbf{X};\mathbf{Z},\mathbf{V}) + I(\mathbf{Y};\mathbf{Z},\mathbf{V}|\mathbf{X})$. The third equation follows from $I(\mathbf{Y};\mathbf{Z},\mathbf{V}|\mathbf{X}) = I(\mathbf{Y};\mathbf{Z},\mathbf{V},\mathbf{X}) - I(\mathbf{Y};\mathbf{X})$. The fourth equation is based on the orthogonality of external and stable factors. The fifth equation is due to $I(\mathbf{X};\mathbf{Z},\mathbf{V}) = I(\mathbf{X};\mathbf{Z}) + I(\mathbf{X};\mathbf{V}|\mathbf{Z})$. The sixth equation is derived from $I(\mathbf{X};\mathbf{V}|\mathbf{Z}) = I(\mathbf{X},\mathbf{Z};\mathbf{V}) - I(\mathbf{Z};\mathbf{V})$, and the last equation also relies on the orthogonality of stable and external factors. Without losing generality, we use γ_1 through γ_4 to represent them here.

With assumption (a), the equation $I(\mathbf{X}_s;\tilde{\mathbf{X}}_s) = I(\tilde{\mathbf{X}}_s;\mathbf{Y}_s)$ holds, thus

$$
\gamma_1 I(\mathbf{X}_s;\tilde{\mathbf{X}}_s) - \beta\gamma_3 I(\tilde{\mathbf{X}}_s;\mathbf{Y}_s) = (\gamma_1 - \beta\gamma_3) I(\mathbf{X}_s;\tilde{\mathbf{X}}_s) \tag{16}
$$

With assumption (b), we have,

$$
I(\tilde{\mathbf{X}}_e;\mathbf{Y}_e) = 0 \tag{17}
$$

Plugging equation 16 and 17 into 3, the original minimization objective is equivalent to:

$$
\min_{\tilde{\mathcal{D}}} (\gamma_1 - \beta\gamma_3) I(\mathbf{X}_s;\tilde{\mathbf{X}}_s) + \gamma_2 I(\mathbf{X}_e;\tilde{\mathbf{X}}_e) \tag{18}
$$

When $\gamma_1 - \beta\gamma_3 < 0$, $I(\mathbf{X}_s;\tilde{\mathbf{X}}_s)$ is maximized, effectively rendering $\tilde{\mathbf{X}}_s \simeq \mathbf{X}_s$. Meanwhile, as $\gamma_2 > 0$, $I(\mathbf{X}_e;\tilde{\mathbf{X}}_e)$ is minimized, resulting in $\tilde{\mathbf{X}}_e \not\sim \mathbf{X}_e$. □

A.2 Proof of Generalization Bound

Proof. Before delving into the proof process, we first introduce the definition of Rademacher complexity and McDiarmid's Inequality:

Definition A.1 (Rademacher complexity) *Given a space Z and a fixed distribution $\mathcal{D}$ defined on Z, let $S = z_1, \ldots, z_n$ be a set of examples drawn from i.i.d. from $\mathcal{D}$. Furthermore, let $\mathcal{F}$ be a class of functions $f : Z \to \mathbb{R}$, the empirical Rademacher complexity of $\mathcal{F}$ is defined to be:*

$$\hat{\mathcal{R}}_n(\mathcal{F}) = \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{F}} \left(\frac{1}{n} \sum_{i=1}^{n} \sigma_i f(z_i) \right) \right] \tag{19}$$

where $\sigma_1, \ldots, \sigma_n$ are independent random variables uniform chosen from {-1,1}. The Rademacher complexity of $\mathcal{F}$ is defined as:

$$\mathcal{R}_n(\mathcal{F}) = \mathbb{E}_{\mathcal{D}} \left[\hat{\mathcal{R}}_n(\mathcal{F}) \right] \tag{20}$$

Theorem A.2 (McDiarmid's Inequality) *Let $X_1, \cdots, X_n$ be independent random variables, all taking values in the set $\mathcal{X}$. Let f: $\mathcal{X}_1 \times \cdots \times \mathcal{X}_n \to \mathbb{R}$ be any function with the $(c_1, \ldots, c_n)$-bounded difference property: $\forall i, \forall (x_1, \ldots, x_n), x_i' \in \mathcal{X}$, we have $|f(x_1, \ldots, x_i, \ldots, x_n) - f(x_1, \ldots, x_i', \ldots, x_n)| \leq c_i$. Then for any $\epsilon > 0$,*

$$\mathbb{P}\left(f(X_1, \cdots, X_n) - \mathbb{E}[f(X_1, \cdots, X_n)] \geq \epsilon \right) \leq \exp\left(-\frac{2\epsilon^2}{\sum_{i=1}^{n} c_i^2} \right) \tag{21}$$

Now, let's delve into the proof. In the CDIB framework, by introducing the i-th user feature Γ_i, we generate multiple distributions $\tilde{\mathcal{D}}_i(\Gamma)$ and combine them into a set $\tilde{\mathcal{D}}$. Next, we denote $\Phi(\hat{\mathcal{D}}(\Gamma)) = \sup_{f \in \mathcal{F}} \mathbb{E}_{\mathcal{D}}[\ell(f; \mathbf{Y})] - \hat{\mathbb{E}}_{\hat{\mathcal{D}}}[\ell(f; \mathbf{Y})]$, and we have:

$$\begin{aligned} \Phi(\tilde{\mathcal{D}}_i(\Gamma)) - \Phi(\mathcal{D}_{tr}(\Gamma)) &\leq \sup_{f \in \mathcal{F}} \hat{\mathbb{E}}_{\mathcal{D}_{tr}} \left[\ell(f; \mathbf{Y}) \right] - \hat{\mathbb{E}}_{\tilde{\mathcal{D}}_i(\Gamma)} \left[\ell(f; \mathbf{Y}) \right] \\ &= \sup_{f \in \mathcal{F}} \frac{f(\mathbf{h}^o, y) - f(\mathbf{h}^g, y)}{m} \leq \frac{M_2 - M_1}{m} \end{aligned} \tag{22}$$

The last inequality holds because CDIB is also trained on $\tilde{\mathcal{D}}$. Since Φ satisfies the bounded difference property, we can apply the McDiarmid's Inequality to find:

$$\mathbb{P}\left(\Phi(\mathcal{D}_{tr}) - \mathbb{E}_{\mathcal{D}_{tr}}[\Phi(\mathcal{D}_{tr})] \geq \epsilon \right) \leq \exp\left(-\frac{2\epsilon^2}{\sum_{i=1}^{m} \left(\frac{M_2 - M_1}{m} \right)^2} \right) = \exp\left(-\frac{2\epsilon^2 m}{(M_2 - M_1)^2} \right) \tag{23}$$

Setting the above probability to be less than δ (*i.e.*, $\exp\left(-\frac{2\epsilon^2 m}{(M_2 - M_1)^2} \right) = \delta$), we can solve that $\epsilon = (M_2 - M_1)\sqrt{\frac{\log \frac{2}{\delta}}{m}}$, and we have determined that with probability at least $1 - \delta$:

$$\Phi(\mathcal{D}_{tr}) \leq \mathbb{E}_{\mathcal{D}_{tr}}[\Phi(\mathcal{D}_{tr})] + (M_2 - M_1)\sqrt{\frac{\log \frac{2}{\delta}}{m}} \tag{24}$$

Since $\mathbf{E}_{\mathcal{D}'}[\hat{\mathbf{E}}_{\mathcal{D}'}[\ell(f; \mathbf{Y})] | \mathcal{D}_{tr}] = \mathbf{E}_{\mathcal{D}}[\ell(f; \mathbf{Y})]$ and $\mathbf{E}_{\mathcal{D}'}[\hat{\mathbf{E}}_{\mathcal{D}_{tr}}[\ell(f; \mathbf{Y})] | \mathcal{D}_{tr}] = \hat{\mathbf{E}}_{\mathcal{D}_{tr}}[\ell(f; \mathbf{Y})]$, where $\mathcal{D}'$ is a "ghost sample" independently drawn identically to $\mathcal{D}_{tr}$, we can rewrite the expectation:

$$\begin{aligned} \mathbb{E}_{\mathcal{D}_{tr}}[\Phi(\mathcal{D}_{tr})] &= \mathbb{E}_{\mathcal{D}_{tr}} \left[\sup_{f \in \mathcal{F}} \mathbb{E}_{\mathcal{D}}[\ell(f; \mathbf{Y})] - \hat{\mathbb{E}}_{\mathcal{D}_{tr}}[\ell(f; \mathbf{Y})] \right] \\ &= \mathbb{E}_{\mathcal{D}_{tr}} \left[\sup_{f \in \mathcal{F}} \mathbb{E}_{\mathcal{D}'} \left(\hat{\mathbb{E}}_{\mathcal{D}'}[\ell(f; \mathbf{Y})] - \hat{\mathbb{E}}_{\mathcal{D}_{tr}}[\ell(f; \mathbf{Y})] \right) \right] \end{aligned} \tag{25}$$

Since sup is a convex function, we can apply Jensen's Inequality to move the sup inside the expectation:

$$\mathbb{E}_{\mathcal{D}_{tr}} \left[\sup_{f \in \mathcal{F}} \mathbb{E}_{\mathcal{D}'} \left(\hat{\mathbb{E}}_{\mathcal{D}'}[\ell(f; \mathbf{Y})] - \hat{\mathbb{E}}_{\mathcal{D}_{tr}}[\ell(f; \mathbf{Y})] \right) \right] \leq \mathbb{E}_{\mathcal{D}_{tr}, \mathcal{D}'} \left[\sup_{f \in \mathcal{F}} \hat{\mathbb{E}}_{\mathcal{D}'}[\ell(f; \mathbf{Y})] - \hat{\mathbb{E}}_{\mathcal{D}_{tr}}[\ell(f; \mathbf{Y})] \right] \tag{26}$$

Multiplying each term in the summation by a Rademacher variable σ_i will not change the expectation since $\mathbb{E}[\sigma_i] = 0$. Furthermore, negating a Rademacher variable does not change its distribution. Combining these two facts,

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}_{tr},\mathcal{D}'}\left[\sup_{f\in\mathcal{F}} \hat{\mathbb{E}}_{\mathcal{D}'}[\ell(f;\mathbf{Y})] - \hat{\mathbb{E}}_{\mathcal{D}_{tr}}[\ell(f;\mathbf{Y})]\right] \\
&\quad= \mathbb{E}_{\mathcal{D}_{\text{tr}},\mathcal{D}'}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}\left(f(\mathbf{h}',y) - f(\mathbf{h}^o,y)\right)\right] \\
&\quad= \mathbb{E}_{\sigma,\mathcal{D}_{\text{tr}},\mathcal{D}'}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}\sigma_i\left(f(\mathbf{h}',y) - f(\mathbf{h}^o,y)\right)\right] \\
&\quad\le \mathbb{E}_{\sigma,\mathcal{D}'}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}\sigma_i f(\mathbf{h}',y)\right] + \mathbb{E}_{\sigma,\mathcal{D}_{tr}}\left[\sup_{f\in\mathcal{F}} -\frac{1}{n}\sum_{i=1}^{n}\sigma_i f(\mathbf{h}^o,y)\right] \\
&\quad= 2\mathbb{E}_{\sigma,\mathcal{D}_{tr}}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}\sigma_i f(\mathbf{h}^o,y)\right] = 2\mathcal{R}_n(\mathcal{F})
\end{aligned} \tag{27}$$

The inequality is due to $\sup(A+B) \le \sup A + \sup B$. Substituting this bound into inequality 24 gives us exactly the Theorem 3.3. □

A.3 Proof of Lower bound of $I(\mathbf{Y};\tilde{\mathcal{D}},\mathbf{\Gamma})$

Proof. According to the chain rule of mutual information, the conditional prediction term $I(\mathbf{Y};\tilde{\mathcal{D}}|\mathbf{\Gamma})$ can be decomposed as: $I(\mathbf{Y};\tilde{\mathcal{D}}|\mathbf{\Gamma}) = I(\mathbf{Y};\tilde{\mathcal{D}},\mathbf{\Gamma}) - I(\mathbf{Y};\mathbf{\Gamma})$. Intuitively, minimizing the $I(\mathbf{Y};\mathbf{\Gamma})$ aims to reduce the model's capture of personalized interests, which is harmful to satisfying recommendations. Therefore, we only maximize the $I(\mathbf{Y};\tilde{\mathcal{D}},\mathbf{\Gamma})$. Similar to the derivation process in (Choi & Lee, 2023), we have:

$$\begin{aligned}
I(\mathbf{Y};\tilde{\mathcal{D}},\mathbf{\Gamma}) &= \mathbb{E}_{\mathbf{Y},\tilde{\mathcal{D}},\mathbf{\Gamma}}\left[\log\frac{p(\mathbf{Y}\mid\tilde{\mathcal{D}},\mathbf{\Gamma})}{p(\mathbf{Y})}\right] \\
&= \mathbb{E}_{\mathbf{Y},\tilde{\mathcal{D}},\mathbf{\Gamma}}\left[\log\frac{p_{\theta_3}(\mathbf{Y}\mid\tilde{\mathcal{D}},\mathbf{\Gamma})}{p(\mathbf{Y})}\right] + \mathbb{E}_{\tilde{\mathcal{D}},\mathbf{\Gamma}}\left[D_{KL}(p(\mathbf{Y}\mid\tilde{\mathcal{D}},\mathbf{\Gamma})\parallel p_{\theta_3}(\mathbf{Y}\mid\tilde{\mathcal{D}},\mathbf{\Gamma})\right] \\
&\ge \mathbb{E}_{\mathbf{Y},\tilde{\mathcal{D}},\mathbf{\Gamma}}\left[\log\frac{p_{\theta_3}(\mathbf{Y}\mid\tilde{\mathcal{D}},\mathbf{\Gamma})}{p(\mathbf{Y})}\right] \\
&= \mathbb{E}_{\mathbf{Y},\tilde{\mathcal{D}},\mathbf{\Gamma}}\left[\log p_{\theta_3}(\mathbf{Y}\mid\tilde{\mathcal{D}},\mathbf{\Gamma})\right] + \mathcal{H}(\mathbf{Y}) \\
&\ge \mathbb{E}_{\mathbf{Y},\tilde{\mathcal{D}},\mathbf{\Gamma}}\left[\log p_{\theta_3}(\mathbf{Y}\mid\tilde{\mathcal{D}},\mathbf{\Gamma})\right]
\end{aligned} \tag{28}$$

where $\mathcal{H}(\mathbf{Y})$ represents the entropy of $\mathbf{Y}$ and $\log p_{\theta_3}(\mathbf{Y}|\tilde{\mathcal{D}},\mathbf{\Gamma})$ denotes the variational approximation of $\log p(\mathbf{Y}|\tilde{\mathcal{D}},\mathbf{\Gamma})$. The first and second inequalities hold because of the non-negative inherent in KL-divergence and entropy. So, the lower bound of $I(\mathbf{Y};\tilde{\mathcal{D}},\mathbf{\Gamma})$ is $\mathbb{E}_{\mathbf{Y},\tilde{\mathcal{D}},\mathbf{\Gamma}}[\log p_{\theta_3}(\mathbf{Y}|\tilde{\mathcal{D}},\mathbf{\Gamma})]$. □

A.4 Proof of $I(\mathcal{D}_{tr};\tilde{\mathcal{D}},\mathbf{\Gamma}) = I(\mathcal{D}_{tr};\tilde{\mathcal{D}})$

Proof. $\mathcal{D}_{tr}$ is training distribution, $\tilde{\mathcal{D}}$ is the generated distribution, and $\mathbf{\Gamma}$ is user features. Remind that $\tilde{\mathcal{D}}$ contains all the information of $\mathbf{\Gamma}$ due to its generation process involving $\mathbf{\Gamma}$ (see, Equation 7). This implies that $\tilde{\mathcal{D}}$ is a deterministic function of $\mathbf{\Gamma}$, *i.e.*, $\mathbf{\Gamma} = f(\tilde{\mathcal{D}})$ for some function f.

Since $\tilde{\mathcal{D}}$ is a function of $\mathbf{\Gamma}$, the joint distribution $p(\mathbf{h}_u^o,\mathbf{h}_u^g,\mathbf{e}_u)$ can be expressed in terms of $p(\mathbf{h}_u^o,\mathbf{h}_u^g)$ and the deterministic relationship $\mathbf{\Gamma} = f(\tilde{\mathcal{D}})$. Thus, we can write:

$$p(\mathbf{h}_u^o,\mathbf{h}_u^g,\mathbf{e}_u) = p(\mathbf{h}_u^o,\mathbf{h}_u^g)\cdot\delta_{\mathbf{e}_u,f(\mathbf{h}_u^g)} \tag{29}$$

where δ is the Kronecker delta function, which is 1 if its arguments are equal and 0 otherwise.

The mutual information $I(\mathcal{D}_{tr}; \tilde{\mathcal{D}}, \mathbf{\Gamma})$ is given by:

$$I(\mathcal{D}_{tr}; \tilde{\mathcal{D}}, \mathbf{\Gamma}) = \sum_{\mathbf{h}_u^o, \mathbf{h}_u^g, \mathbf{e}_u} p(\mathbf{h}_u^o, \mathbf{h}_u^g, \mathbf{e}_u) \log \frac{p(\mathbf{h}_u^o, \mathbf{h}_u^g, \mathbf{e}_u)}{p(\mathbf{h}_u^o) p(\mathbf{h}_u^g, \mathbf{e}_u)} \tag{30}$$

Given $p(\mathbf{h}_u^o, \mathbf{h}_u^g, \mathbf{e}_u) = p(\mathbf{h}_u^o, \mathbf{h}_u^g) \cdot \delta_{\mathbf{e}_u, f(\mathbf{h}_u^g)}$, we can rewrite the mutual information as:

$$I(\mathcal{D}_{tr}; \tilde{\mathcal{D}}, \mathbf{\Gamma}) = \sum_{\mathbf{h}_u^o, \mathbf{h}_u^g} p(\mathbf{h}_u^o, \mathbf{h}_u^g) \cdot \delta_{\mathbf{e}_u, f(\mathbf{h}_u^g)} \log \frac{p(\mathbf{h}_u^o, \mathbf{h}_u^g) \cdot \delta_{\mathbf{e}_u, f(\mathbf{h}_u^g)}}{p(\mathbf{h}_u^o) p(\mathbf{h}_u^g, \mathbf{e}_u)} \tag{31}$$

Since $\delta_{\mathbf{e}_u, f(\mathbf{h}_u^g)}$ is 1 when $\mathbf{e}_u = f(\mathbf{h}_u^g)$ and 0 otherwise, the term $\delta_{\mathbf{e}_u, f(\mathbf{h}_u^g)}$ effectively restricts the summation to the cases where $\mathbf{e}_u = f(\mathbf{h}_u^g)$. Thus, $p(\mathbf{h}_u^g, \mathbf{e}_u)$ in the denominator simplifies to $p(\mathbf{h}_u^g, f(\mathbf{h}_u^g))$, which is equal to $p(\mathbf{h}_u^g)$. The mutual information simplifies to:

$$I(\mathcal{D}_{tr}; \tilde{\mathcal{D}}, \mathbf{\Gamma}) = \sum_{\mathbf{h}_u^o} \sum_{\mathbf{h}_u^g} p(\mathbf{h}_u^o, \mathbf{h}_u^g) \log \frac{p(\mathbf{h}_u^o, \mathbf{h}_u^g)}{p(\mathbf{h}_u^o) p(\mathbf{h}_u^g)} \tag{32}$$

This proves that when $\tilde{\mathcal{D}}$ contains all the information of $\mathbf{\Gamma}$, the mutual information between $\mathcal{D}_{tr}$ and the joint variables $\tilde{\mathcal{D}}$ and $\mathbf{\Gamma}$ is equal to the mutual information between $\mathcal{D}_{tr}$ and $\tilde{\mathcal{D}}$ alone. Intuitively, knowing $\tilde{\mathcal{D}}$ uniquely determines $\mathbf{\Gamma}$, $\mathbf{\Gamma}$ provide no additional information about $I(\mathcal{D}_{tr}; \tilde{\mathcal{D}}, \mathbf{\Gamma})$ beyond what is already known about $\tilde{\mathcal{D}}$. □

B Additional Experiments

B.1 Case Study

To verify the CDIB's effectiveness in handling distribution shifts, similar to the case mentioned in the introduction section (see Figure 1), we visualize the interest space learned by CDIB for different users in both the training and testing stages. The result is shown in Figure 7. The result reveals that compared with SASRec, CDIB is less influenced by new trending items (*i.e.*, 634, 8587) or unseen collaboration patterns when modelling the interest of niche users at the testing phase. This indicates the capability of CDIB to capture the users' true interest when faced with distribution shifts.

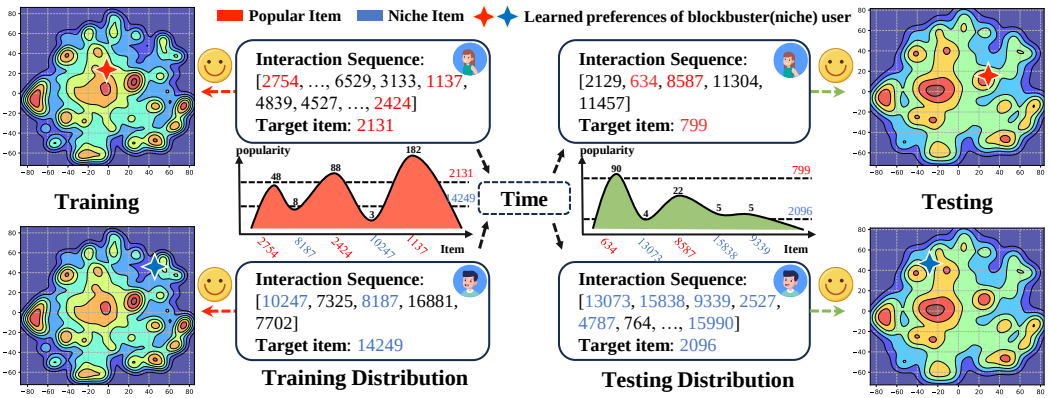


Figure 7: Visualization of the interest space learned by CDIB on Retailrocket dataset.

B.2 Visualization of $\mathbf{M}_u$

To explore the underlying mechanism of the factor discriminator, we utilize heatmaps to visualize the popularity of each item in the interaction sequences of six users from the *RetailRocket* dataset, along with the corresponding $1 - \mathbf{M}_u$ learned by the factor discriminator. It is important to note that

the higher the $1 - \mathbf{M}_u$ value for an item, the less the model intends to interfere with it, suggesting these items may reflect the user's true interest. The visualization results are displayed in Figure 8. The results show that for niche users (User 509, User 444, User 117), their interaction sequences predominantly feature niche items, which often represent their unique interest. The factor discriminator not only protects popular items from being altered but also tends to shield these less common items from interference, as highlighted by the blue dashed box. Conversely, for blockbuster users (User 168, User 107, User 161), who typically interact with trending topics, the factor discriminator often identifies these popular items as the users' interest and refrains from modifying them, as indicated by the red dashed box. Compared to traditional hand-crafted data augmentation methods, the factor discriminator's ability to adaptively select elements for augmentation is crucial. It can intelligently choose which elements to enhance based on user attributes without distorting the original user interest. This approach helps generate more promising augmented samples and, to some extent, avoids introducing extraneous noise.

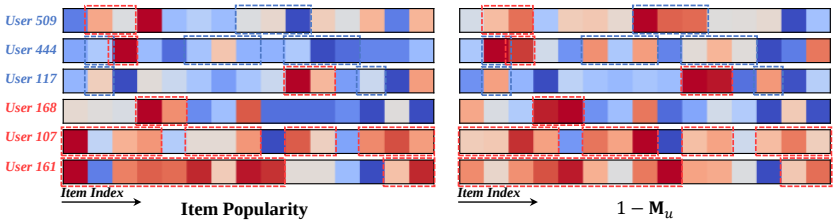


Figure 8: The visualization of interacted items' popularity and the corresponding $1 - \mathbf{M}_u$.

B.3 Visualization of $\tilde{\mathcal{D}}$

We visualize the representation space of some origin and augmented items to explore its inner mechanisms. Specifically, we reduce the dimensionality of items within the users' interaction sequences and those enhanced by latent diffusion using t-SNE (Van der Maaten & Hinton, 2008), then visualize their two-dimensional outcomes. Additionally, considering that β is a critical hyperparameter for balancing the diversity and reliability of generated distribution, we also visualized the generation results for different β values, as depicted in Figure 9. From the visualization results, it can be concluded that at lower β values (*i.e.*, 1e0, 1e1), feature collapse occurs in the generated items. This collapse may happen because a small β shifts the model's focus towards generating low similarity distributions (minimizing $I(\mathcal{D}_{tr}; \tilde{\mathcal{D}}|\mathbf{\Gamma})$, thereby neglecting the intrinsic features of the original dataset. Such generated distributions can introduce unnecessary noise to the model, complicating its learning process. Hence, the model performs poorly when β is small (see Section 4.2). As β increases, the distribution generated by latent diffusion gradually aligns more closely with the original distribution and retains a clustering effect of popular items to some extent, while the overall distribution becomes more uniform, aiding the model's learning process (Yu et al., 2022). When β reaches 1e4, there is no distinguishable difference between the generated and original distributions. At this level, the model prioritizes preserving the original data characteristics as much as possible (maximizing $I(\mathbf{Y}; \tilde{\mathcal{D}}|\mathbf{\Gamma})$, which limits the exploration of out-of-distribution scenarios and decreases overall effectiveness.

B.4 Results on Diverse Augmentation Methods

Within the scope of our ablation study, we added Gaussian noise to the original data for augmentation rather than using diffusion-based augmentation techniques. We also conducted experiments with random data augmentation strategies. Specifically, we performed random masking, cropping, and reordering on the original interaction sequences to augment the data. The overall results are presented in Table 2. It can be seen that our diffusion-based augmentation method performs better than the other two techniques.

B.5 Sensitivity Analysis on α_1 and α_2

We have further added sensitivity analysis of hyperparameters for α_1 and α_2 on ml-100k and retailrocket, respectively. Specifically, we evaluate our model by varying the α_1 and α_2 in {0.01,

Table 2: Performance over diverse data augmentation methods.

Method	MovieLens-100K		Retailrocket		Amazon-Beauty		Amazon-Sports	
	HitRate ↑	NDCG ↑	HitRate ↑	NDCG ↑	HitRate ↑	NDCG ↑	HitRate ↑	NDCG ↑
random	11.09±0.22	5.08±0.13	19.84±0.17	8.76±0.11	8.98±0.17	4.42±0.05	4.90±0.18	2.30±0.10
Gaussian Noise	11.59±0.16	5.53±0.12	20.82±0.16	9.37±0.04	9.06±0.15	**4.60±0.06**	4.92±0.21	**2.40±0.08**
CDIB	**11.89±0.16**	**5.67±0.09**	**21.12±0.14**	**9.41±0.10**	**9.17±0.04**	4.56±0.04	**4.95±0.11**	2.38±0.07

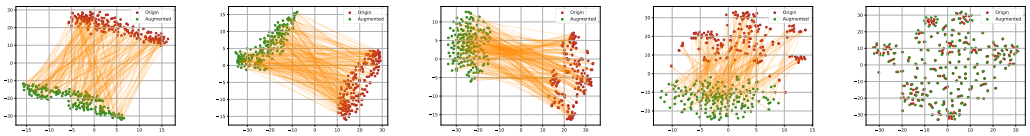
0.1, 1.0, 5.0, 10.0}, respectively. The results are present in Table 3 and Table 4. We conclude our observations as follows: **(i)** Optimal performance across both datasets is attained at $\alpha_1 = 1.0$, marking a peak in performance that rises to it and then begins to decline. If α_1 is set too low, the diffusion model's generative capabilities are diminished, potentially leading to the creation of noise samples that can hinder model training. Conversely, if α_1 is too high, the auxiliary task takes precedence in the model's optimization process, which can adversely impact the model's recommendation capabilities. **(ii)** Optimal model performance is consistently achieved at $\alpha_2 = 1.0$ across all datasets, after which there is a notable decline in performance as α_2 increases to 10.0, with pronounced effects on the Retail dataset. This decrease may be due to the model's overly focus on model performance on the generated data distribution at higher α_2 values, potentially obscuring the model's capacity to extract essential information from the original datasets' distribution.

Table 3: Sensitivity analysis on α_1

α_1	MovieLens-100K		Retailrocket	
	HitRate ↑	NDCG ↑	HitRate ↑	NDCG ↑
0.01	**11.91±0.19**	5.58±0.13	20.92±0.15	9.28±0.12
0.1	11.76±0.17	5.59±0.14	20.96±0.15	9.38±0.11
1.0	11.89±0.16	**5.67±0.09**	**21.12±0.14**	9.41±0.10
5.0	11.16±0.14	5.30±0.08	21.10±0.15	**9.44±0.13**
10.0	10.99±0.17	5.12±0.12	20.98±0.12	9.40±0.11

Table 4: Sensitivity analysis on α_2

α_2	MovieLens-100K		Retailrocket	
	HitRate ↑	NDCG ↑	HitRate ↑	NDCG ↑
0.01	11.11±0.18	4.92±0.14	19.69±0.15	8.58±0.12
0.1	11.44±0.17	5.19±0.11	20.28±0.15	8.81±0.13
1.0	**11.89±0.16**	**5.67±0.09**	**21.12±0.14**	**9.41±0.10**
5.0	8.42±0.14	3.81±0.07	16.00±0.13	7.83±0.12
10.0	5.73±0.13	2.63±0.6	5.13±0.15	2.50±0.13



(a) $\beta = 1e0$ (b) $\beta = 1e1$ (c) $\beta = 1e2$ (d) $\beta = 1e3$ (e) $\beta = 1e4$

Figure 9: The visualization of generated distribution *w.r.t* β.

B.6 APPROXIMATE PERCENTAGE OF VALUES OF 1 IN $\mathbf{M}_u$

we record the percentage of values of 1 in $\mathbf{M}_u$ before and after adding the $\mathcal{L}_{mask}$, the results are shown in Table 5. As we can see, after the introduction of the $\mathcal{L}_{mask}$, the percentage of values of 1 in $\mathbf{M}_u$ increased from 0.0% to an average of 17.4%.

Table 5: approximate percentage of values of 1 in $\mathbf{M}_u$

	ML-100K	Retailrocket	Amazon-Beauty	Amazon-Sports
w/o $\mathcal{L}_{mask}$	0.0%	0.0%	0.0%	0.0%
w $\mathcal{L}_{mask}$	13.5%	22.1%	15.5%	18.4%

C IMPLEMENTATION DETAILS

The CDIB model is implemented using Pytorch 1.13.0 (Paszke et al., 2019) and Python 3.8.13. Experiments are conducted using two NVIDIA GeForce RTX 3090 GPUs. To ensure a fair comparison,

we adopt the widely used experimental environment RecBole (Zhao et al., 2021). The parameters are initialized using a Gaussian distribution $\mathcal{N}(0, 0.02)$ and optimized with the Adam optimizer at a learning rate of 0.001. Moreover, we adopt the early-stop strategy to train the models and set the maximum sequence length to 50. We run the codes for GRU4Rec, Caser, SASRec, and CL4SRec, which are reproduced by the RecBole team[3]. We also reproduce the implementations of IPS, S-DRO, DROS, DuoRec, and DCRec to the RecBole environment.

C.1 ALGORITHM

The pseudo-algorithms for the training and inference stages of CDIB are presented in Algorithm 1 and Algorithm 2, respectively.

Algorithm 1: The Training Stage of the Proposed CDIB Algorithm

Input: user set $\mathcal{U} = \{u\}$, item set $\mathcal{I} = \{i\}$, interaction sequences $\mathcal{S}_{tr}$, learning rate η, and batch size B.
Output: trained recommender ξ^*.

Initialize all parameters;
while *not converge* **do**
  Sample a batch $\{u\}_1^B$ and $\{s_u\}_1^B$ from $\mathcal{S}_{tr}$
  Embed users $\{u\}_1^B$ and items $\{s_u\}_1^B$ to get the corresponding embedding $\mathbf{\Gamma}$ and $\mathbf{H}_u$
  # Generating Distribution $\tilde{\mathcal{D}}$
  Calculate the mask $\mathbf{M}_u$ using the Factor Discriminator
  Mask the stable factors by $\mathbf{H}_u^0 = \mathbf{H}_u \odot \mathbf{M}_u$
  Forward diffusion process: $\mathcal{N}(\mathbf{H}_u^t; \sqrt{1-\beta_t}\mathbf{H}_u^{t-1}, \beta_t\mathbf{I})$
  Reverse diffusion process: $p(\mathbf{H}_u^T)\prod_{t=1}^{T} p_{\theta_2}(\mathbf{H}_u^{t-1}|\mathbf{H}_u^t)$
  Sample external factors to get diverse data $\tilde{\mathbf{H}}_u = \tilde{\mathbf{H}}_u^0 + \mathbf{H}_u \odot (1 - \mathbf{M}_u)$
  Calculate the $\mathcal{L}_{gd} = \mathcal{L}_{con} + \mathcal{L}_{mask}$
  # Optimizing with CDIB
  Encode origin and augmented interaction sequence using Transformer Recommender
  Obtain the overall loss: $\mathcal{L}_{total} = \mathcal{L}_{pred} + \alpha_1\mathcal{L}_{gd} + \alpha_2(\beta\mathcal{L}_{reg} + \mathcal{L}_{gen})$
  Update ξ to minimize $\mathcal{L}_{total}$
end
return trained recommender ξ^*

Algorithm 2: The testing Stage of the Proposed CDIB Algorithm

Input: interaction sequences $\mathcal{S}_{te}$, trained recommender ξ^*, and item set $\mathcal{I}$.
Output: recommended items.

for $s_u \in \mathcal{S}_{te}$ **do**
  return $\arg\max_{i \in \mathcal{I}} p(i|\xi^*(s_u))$
end

C.2 HYPERPARAMETER

We tune the hyperparameters as follows: Batch Size $\in \{64, 128, 256, 512, 1024\}$; Dropout Rate $\in \{0.1, 0.3, 0.5, 0.7\}$; $\beta \in \{10, 100, 1000, 10000\}$. For a fair comparison, we standardized all common hyperparameters across models and configured the unique hyperparameters according to the settings provided by the corresponding authors. The hyperparameters settings of CDIB are present in Table 6.

[3]https://github.com/RUCAIBox/RecBole/tree/master/recbole/model/sequential_recommender

Table 6: Hyperparameter specifications

Dataset	ML-100K	Retailrocket	Amazon-Beauty	Amazon-Sports
Optimizer	Adam	Adam	Adam	Adam
Batch Size	512	512	512	512
Learning Rate	0.001	0.001	0.001	0.001
Embedding Size	64	64	64	64
Hidden Size	256	256	256	256
Dropout Rate	0.5	0.5	0.5	0.5
Temperature τ	1.0	1.0	1.0	1.0
Lagrange multiplier β	1000	1000	1000	10000

C.3 Details about $\mathcal{L}_{pred}$ and $\mathcal{L}_{reg}$

$\mathcal{L}_{pred}$ is a negative log-likelihood function of the expected next item i_{L+1} of an origin interaction sequence s_u, where we adopt cross-entropy loss under the full set of items:

$$\mathcal{L}_{pred} = \frac{1}{|\mathcal{U}|}\sum_{u=1}^{|\mathcal{U}|} -\log\left(\frac{\exp(\mathbf{h}_u^o \cdot \mathbf{e}_u^{L+1})}{\sum_{i\in\mathcal{I}}\exp(\mathbf{h}_u^o \cdot \mathbf{e}_i)}\right) \tag{33}$$

where $\mathbf{h}_u^o$ is the representation of the origin interaction sequence, $\mathbf{e}_u^{L+1}$ is the embedding of the next interacted item, and $\mathbf{e}_i$ is the embedding of item i.

$\mathcal{L}_{reg}$ is the regularization loss, which encourages that essential information to the target item i_{L+1} is preserved, where we utilized the cross entropy loss under the full set of items:

$$\mathcal{L}_{reg} = \frac{1}{|\mathcal{U}|}\sum_{u=1}^{|\mathcal{U}|} -\log\left(\frac{\exp(\mathbf{h}_u^g \cdot \mathbf{e}_u^{L+1})}{\sum_{i\in\mathcal{I}}\exp(\mathbf{h}_u^g \cdot \mathbf{e}_i)}\right) \tag{34}$$

where $\mathbf{h}_u^g$ is the representation of the generated interaction sequence, $\mathbf{e}_u^{L+1}$ is the embedding of the next interacted item, and $\mathbf{e}_i$ is the embedding of item i.

D Complexity Analyses

Time Complexity Analysis. For one batch of training data, the computational cost of the factor discriminator is $O(4Ld)$, and the distribution generator is $O(Ld+TLd^2)$, which means the time cost of generating new distribution by CDIB is $O(5Ld + TLd^2)$ in the training stage. With the attention calculations, the time complexity of the Transformer Recommender is $O(2L^2hd + Lhd^2)$, which is also the total time cost in the testing stage. Moreover, the time complexity to compute $\mathcal{L}_{gd}$ is $O(L(2d + 1))$, $\mathcal{L}_{reg}$ is $O(|\mathcal{I}|d)$, $\mathcal{L}_{gen}$ is $O(2|\mathcal{U}|d)$, and $\mathcal{L}_{pred}$ is $O(|\mathcal{I}|d)$, therefore, the total time cost to compute the loss is $O(L(2d+1) + 2|\mathcal{I}|d + 2|\mathcal{U}|d)$. Consider that $|\mathcal{I}|$ and $|\mathcal{U}|$ are much larger than L empirically, the overall time complexity of CDIB is $O((|\mathcal{I}| + |\mathcal{U}|)d + TLd^2)$.

Space Complexity Analysis. Compared to the naive sequential recommendation model SASRec (Kang & McAuley, 2018), our CDIB uses extra parameters costs $O(|\mathcal{B}|d)$ to represent the users' attributes, which only occur in the training stage, and $O(d^2)$ to develop the factor discriminator and distribution generator, which is affordable. The running time and model size in the training stage are shown in Table 7, and that of the testing stage is present in Table 8, where also present the performance improvement compared with SASRec.

E Datasets, Baselines, Metrics, and Related Works

E.1 Detailed Datasets Description

ML-100K. ML-100K is sourced from MovieLens[4], a recommendation system and virtual community website established by the GroupLens Research Project at the University of Minnesota's School of Computer Science and Engineering.

[4]https://movielens.org/

Table 7: Running time and Model size at the training stage

Dataset	SASRec		DROS		DCRec		CDIB	
	Running Time	Model Size	Running Time	Model Size	Running Time	Model Size	Running Time	Model Size
ML-100K	00h 15m 02s	0.21 M	01h 48m 01s	0.21 M	01h 26m 55s	0.22 M	00h 37m 21s	0.42 M
Retail	01h 02m 32s	1.24 M	02h 25m 38s	1.24 M	06h 36m 12s	1.25 M	01h 19m 24s	2.80 M
Beauty	01h 03m 39s	0.87 M	01h 39m 34s	0.87 M	02h 11m 21s	0.89 M	01h 08m 40s	2.45 M
Sports	01h 34m 48s	1.27 M	03h 35m 04s	1.27 M	02h 42m 11s	1.29 M	00h 50m 38s	3.70 M

Table 8: Running time, Model size and Performance Improvement at the testing stage

Dataset	SASRec		DROS			DCRec			CDIB		
	Running Time	Model Size	Running Time	Model Size	Performance Improvement	Running Time	Model Size	Performance Improvement	Running Time	Model Size	Performance Improvement
ML-100K	0.13s	0.21 M	0.12s	0.21 M	$\uparrow$ 8.10%	0.14s	0.22 M	$\downarrow$ 6.94%	0.12s	0.21 M	$\uparrow$ 17.23%
Retailrocket	0.37s	1.24 M	0.45s	1.24 M	$\downarrow$ 0.32%	0.43s	1.25 M	$\uparrow$ 1.72%	0.38s	1.24 M	$\uparrow$ 8.53%
Beauty	0.34s	0.87 M	0.32s	0.87 M	$\downarrow$ 3.54%	0.38s	0.89 M	$\downarrow$ 9.13%	0.31s	0.87 M	$\uparrow$ 6.34%
Sports	0.47s	1.27 M	0.48s	1.27 M	$\uparrow$ 4.41%	0.58s	1.29 M	$\downarrow$ 11.34%	0.48s	1.27 M	$\uparrow$ 7.29%

Retailrocket. The Retailrocket dataset was collected from a real-world e-commerce website[5]. We utilize viewing sequences to train and test our model. Following the approach in (Yang et al., 2023b), we exclude items interacted with fewer than five times to mitigate the cold-start issue. Additionally, sequences shorter than three interactions are removed.

Amazon. The Amazon-Beauty and Amazon-Sports datasets compile user-item interactions from Amazon[6] in the Beauty and Sports product categories, respectively. Employing preprocessing similar to that used for Retailrocket, we filter out items with fewer than five interactions and sequences shorter than three. The statistics of our evaluation datasets are detailed in Table 9.

Table 9: Dataset statics

Dataset	ML-100K	Retailrocket	Amazon-Beauty	Amazon-Sports
#Users	944	22179	22364	35599
#Items	1683	17804	12102	18358
#Interactions	100000	240938	198502	296337
Avg. actions of users	106.04	10.86	8.87	8.32
Avg. actions of items	59.45	13.53	16.40	16.14
Sparsity	93.71%	99.94%	99.93%	99.95%

E.2 Detailed Baselines Description

We compare CDIB with nine methods from diverse research lines, covering:

1. **Naive Sequential Recommendation Methods**: These methods have been effective techniques to capture the evolving pattern of users' interest.
   - **GRU4Rec** (Hidasi et al., 2016): GRU4Rec utilizes the Gated Recurrent Unit (GRU) for session-based recommendations, providing strong sequence modelling capabilities.
   - **Caser** (Tang & Wang, 2018): Caser is a CNN-based approach that employs horizontal and vertical convolutional filters to capture sequential patterns.
   - **SASRec** (Kang & McAuley, 2018): SASRec applies a multi-head self-attention mechanism to encode item-wise sequential correlations, suitable for long sequence data.
2. **Reweighting Methods**: These methods aim to develop a more unbiased and robust model by adjusting the weight of each training instance.
   - **IPS** (Schnabel et al., 2010): IPS re-weights each training instance with inverse popularity score to eliminate popularity bias.

[5]https://www.kaggle.com/datasets/retailrocket/ecommerce-dataset/

[6]https://www.amazon.com/

3. **DRO Methods**: DRO Methods integrate the distributionally robust optimization (Hu & Hong, 2013) to the sequential recommendation to obtain a recommender with better generalization ability.
   - **S-DRO** (Wen et al., 2022): This model adds streaming optimization improvement to the Distributionally Robust Optimization (DRO) framework to mitigate the amplification of Empirical Risk Minimization (ERM) on popularity bias.
   - **DROS** (Yang et al., 2023b): It introduces a carefully designed distribution adaption paradigm, which considers the dynamics of data distribution and explores possible distribution shifts between training and testing.
4. **Diffusion-based Augmentation Methods**: Diffusion-based Augmentation Methods utilize diffusion technical to enrich the sparse training data to improve the model performance.
   - **DiffuASR** (Liu et al., 2023): This model designs a Sequential U-Net to capture sequence information while predicting the added noise. Additionally, two guiding strategies (DiffuASR-CG and DiffuASR-CF) are implemented to steer DiffuASR, ensuring it generates items that align more closely with the preferences in the original sequence.
5. **Contrastive Learning Methods**: CL methods adopt data augmentation to enhance the robustness of recommenders.
   - **CL4SRec** (Xie et al., 2022): CL4SRec employs random corruption techniques like cropping, masking, and reordering to generate contrastive views.
   - **DuoRec** (Qiu et al., 2022): DuoRec introduces supervised positive sampling to obtain high-quality positive pairs.
   - **DCRec** (Yang et al., 2023a): DCRec unifies sequential pattern encoding with global collaborative relation modelling through adaptive conformity-aware augmentation.

E.3 Detailed Metrics Description

We focus on top-N item recommendations and utilize two widely used metrics for evaluation: Hit Rate (HR)@N and Normalized Discounted Cumulative Gain (NDCG)@N. These metrics are crucial for assessing the recommendation accuracy at the top-N ranked positions (Kang & McAuley, 2018; Yang et al., 2023a; Xia et al., 2023). The models are evaluated using an all-ranking protocol (He et al., 2020), which provides a robust and comprehensive performance assessment. The metrics are formally calculated as follows:

$$HR@N = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} r_{i,j}}{M}; \quad NDCG@N = \sum_{i=1}^{M} \frac{\sum_{j=1}^{N} r_{i,j}/\log_2(j+1)}{M \cdot IDCG_i} \tag{35}$$

where M denotes the number of tested users, $r_{i,j} = 1$ if the j-th item in the ranked list for the i-th user is positive, and $r_{i,j} = 0$ otherwise. The numerator of $NDCG@N$ is the discounted cumulative gain (DCG) at N, and $IDCG_i$ is the ideal maximum $DCG@N$ value for the i-th tested user.

E.4 More Related Works

Sequential Recommendation is designed to predict the next item a user is likely to prefer based on their interaction history. Traditional methods have leveraged Markov chains to capture first-order item-to-item correlations through transition matrices (Rendle et al., 2010; He & McAuley, 2016). With the development of deep learning, which excels at modeling complex sequential patterns, various deep recommendation models have been developed. For instance, GRU4Rec (Hidasi et al., 2016) employs Gated Recurrent Unit (GRU) units to model the temporal dynamics of interaction sequences. Caser (Tang & Wang, 2018) uses a time convolutional neural network (TCN) to account for both long-term and short-term user interests in personalized recommendations. SASRec (Kang & McAuley, 2018) and BERT4Rec (Sun et al., 2019) enhance computational efficiency in lengthy sequences by incorporating self-attention mechanisms. More recently, inspired by selective state space models (Gu & Dao, 2024), Mamba4Rec (Liu et al., 2024) has been introduced, utilizing the mamba framework to recommend items efficiently. Despite their capabilities, these models often suffer performance declines when OOD occurs. To address this, CDIB introduces a user feature-guided generation approach that proactively explores OOD scenarios during the training phase, enhancing the model's generalization capabilities.

Distributionally Robust Sequential Recommendation has recently attracted significant research interest, which aims to train a model that performs well not only at the training stage but also at the testing stage. Methods like reweighting and DRO (Schnabel et al., 2010; Bottou et al., 2013; Wang et al., 2022b; Yang et al., 2023b; Wen et al., 2022) presume that the test dataset's distribution can be inferred from prior knowledge. For example, IPS (Schnabel et al., 2010) re-weight each instance with the inverse propensity score, which implicitly assumes the testing distribution is uniform (Zhang et al., 2023). DROS (Yang et al., 2023b) unifies the DRO and sequential recommendation paradigms to enhance model robustness against distribution shifts but faces challenges with sparse data. Causal inference methods capture real causal relationships but assume the causal graph is static (Wang et al., 2023b; He et al., 2022; Yang et al., 2020; Wang et al., 2022a), while contrastive learning approaches seek to enrich the training data distribution through data augmentation (Liu et al., 2021; Xie et al., 2022; Yang et al., 2023a; Qiu et al., 2022; Zhao et al., 2023), but hardly rely on the data augmentation strategies. What's more, most of the existing models ignore the user's sensitivity during the process of distribution shift. To fill the gap, we introduce the CDIB principle, using the user features to guide the exploration of the other distribution.

Information Bottleneck with Conditional Information has been increasingly utilized in recent research. Various studies have adopted the information bottleneck (IB) principle by incorporating conditional, aiming to extract information that aligns with specific objectives. The conditional information bottleneck (CIB) theory (Gondek & Hofmann, 2003) has been applied in methods such as CGIB (Lee et al., 2023) to identify crucial molecular structures that predict interactions between graph pairs, with a focus on significant subgraphs. TimeCIB (Choi & Lee, 2023) extends the CIB to time series data imputation, ensuring the preservation of essential temporal information. Drawing inspiration from these precedents, CDIB employs CIB to steer the generation of distributions, enhancing the model's robustness. To the best of our knowledge, CDIB is the first application of CIB to guide the distribution generation process.

Diffusion-based Augmentation Models Earlier approaches like Diff4Rec (Wu et al., 2023) and DiffuASR (Liu et al., 2023) followed a three-step process: training the diffusion model, generating new data with the diffusion model, and then training the recommendation model on new data, which can lead to a disconnect between the generation and downstream tasks due to the discrete nature of these stages, preventing the flow of gradient information. Our model, however, employs an end-to-end training approach, which maintains the alignment between the generation and downstream tasks. What's more, in the SR scenario, interaction data is very sensitive (Ye et al., 2023), and there is a risk of losing significant information during the data augmentation phase, which may compromise the quality of the generated data, a concern overlooked in previous methods. Our model addresses this by utilizing a learnable mask mechanism to safeguard critical interactions adaptively and is guided by IB theory in the generation process. Visualization (*cf.* Appendix B.2) demonstrates that our model can mitigate the data quality issues.

F Limitation and Future Work

Although CDIB outperforms the baseline models, it currently relies solely on ID features to model user attributes, and its ability to guide generating distributions is constrained by cold-start problems. In future work, we plan to investigate using side information or multi-modal data to model user attributes, which may help mitigate the cold start issues. Additionally, to maintain high computational efficiency, we employ a lightweight MLP model as the backbone for the denoising process. While the suitability of MLP for recommendation scenarios is not the focus of this work, it remains an important question. Therefore, we will explore which architectures are both lightweight and effective for recommendation scenarios, such as Mamba (Gu & Dao, 2024), in our future studies.