Published: 21 February 2026

WSDM '26:The Nineteenth ACM International Conference on Web Search and Data Mining
February 22 - 26, 2026
ID, Boise, USA

Conference Sponsors:
SIGKDD
SIGWEB
SIGIR
SIGMOD

RESEARCH-ARTICLE

Continuous-time Discrete-space Diffusion Model for Recommendation

CHENGYI LIU, The Hong Kong Polytechnic University, Hong Kong, Hong Kong, Hong Kong
XIAO CHEN, The Hong Kong Polytechnic University, Hong Kong, Hong Kong, Hong Kong
SHIJIE WANG, The Hong Kong Polytechnic University, Hong Kong, Hong Kong, Hong Kong
WENQI FAN, The Hong Kong Polytechnic University, Hong Kong, Hong Kong, Hong Kong
QING LI, The Hong Kong Polytechnic University, Hong Kong, Hong Kong, Hong Kong

Open Access Support provided by:
The Hong Kong Polytechnic University

Continuous-time Discrete-space Diffusion Model for Recommendation

Chengyi Liu
The Hong Kong Polytechnic University
Hong Kong SAR, China
chengyi.liu@connect.polyu.hk

Xiao Chen
The Hong Kong Polytechnic University
Hong Kong SAR, China
shawn.chen@connect.polyu.hk

Shijie Wang
The Hong Kong Polytechnic University
Hong Kong SAR, China
shijie.wang@connect.polyu.hk

Wenqi Fan
The Hong Kong Polytechnic University
Hong Kong SAR, China
wenqi.fan@polyu.edu.hk

Qing Li
The Hong Kong Polytechnic University
Hong Kong SAR, China
csqli@comp.polyu.edu.hk

Abstract

In the era of information explosion, Recommender Systems (RS) are essential for alleviating information overload and providing personalized user experiences. Recent advances in diffusion-based generative recommenders have shown promise in capturing the dynamic nature of user preferences. These approaches explore a broader range of user interests by progressively perturbing the distribution of user-item interactions and recovering potential preferences from noise, enabling nuanced behavioral understanding. However, existing diffusion-based approaches predominantly operate in continuous space through encoded graph-based historical interactions, which may compromise potential information loss and suffer from computational inefficiency. As such, we propose CDRec, a novel Continuous-time Discrete-space Diffusion Recommendation framework, which models user behavior patterns through discrete diffusion on historical interactions over continuous time. The discrete diffusion algorithm operates via discrete element operations (e.g., masking) while incorporating domain knowledge through transition matrices, producing more meaningful diffusion trajectories. Furthermore, the continuous-time formulation enables flexible adaptive sampling. To better adapt discrete diffusion models to recommendations, CDRec introduces: (1) a novel popularity-aware noise schedule that generates semantically meaningful diffusion trajectories, and (2) an efficient training framework combining consistency parameterization for fast sampling and a contrastive learning objective guided by multi-hop collaborative signals for personalized recommendation. Extensive experiments on real-world datasets demonstrate CDRec's superior performance in both recommendation accuracy and computational efficiency. Our codes are available at: https://github.com/ChengyiLIU-cs/CDRec.

CCS Concepts

• **Information systems → Collaborative filtering**; **Users and interactive retrieval**.

ACM ISBN 979-8-4007-2292-9/2026/02
https://doi.org/10.1145/3773966.3777987

Keywords

Recommender System; Diffusion Models, Continuous-time Discrete Diffusion Models.

ACM Reference Format:
Chengyi Liu, Xiao Chen, Shijie Wang, Wenqi Fan, and Qing Li. 2026. Continuous-time Discrete-space Diffusion Model for Recommendation. In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining (WSDM '26), February 22–26, 2026, Boise, ID, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3773966.3777987

1 Introduction

With the rapid growth of digital content, Recommender Systems (RS) have become essential to alleviate information overload and enhance user experience by providing personalised content across diverse scenarios, such as e-commerce and social media [6, 27]. Technically, as one of the most representative RS techniques, Collaborative Filtering (CF) methods aim to infer user preferences from user-item interaction behaviours in the history [36]. Graph Neural Network (GNN)-based approaches in CF have achieved notable success by leveraging their capability to model graph-structured data [3, 5, 8, 28]. These methods refine user and item representations on historical interaction graphs, capturing high-order collaborative signals within a discriminative recommendation paradigm [32]. Recent studies have focused on generative models that learn user interaction distributions to directly generate preferred items, effectively modeling the temporal dynamics of user preferences [20, 24, 25]. In particular, diffusion-based generative recommender models frame the recommendation task as simulating real-world interaction processes from noise, offering a promising solution [4, 14, 26].

Diffusion-based recommendation methods exhibit notable potential due to their ability to model complex data distributions and generate samples with broad coverage [9, 22]. Grounded in a solid theoretical foundation, the typical diffusion algorithm first perturbs the original data distribution into a known prior and consequently learns a parameterized reverse process to iteratively construct samples from noise. This diffusion mechanism aligns well with the objective of RS, which is to infer the distribution of potential user-item interactions from inherently incomplete and noisy historical data. To be specific, most existing methods directly apply continuous-space diffusion models on discrete graph-structured data, which

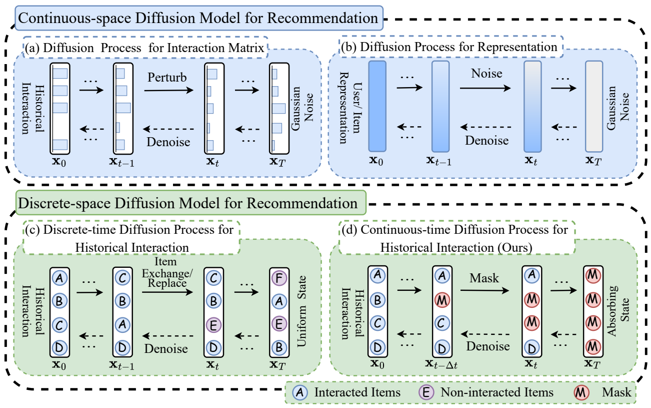


Figure 1: Comparison of diffusion-based RS across different state-spaces. Continuous-space diffusion algorithms apply isotropic Gaussian noise to either the collaborative graph's adjacency matrix or the representations, often compromising personalized information in the latent space. In contrast, discrete-space diffusion models operate through item-level operations toward a uniform distribution. Inspired by this formulation, our proposed CDRec perturbs historical interactions via masking operations in continuous time, enabling state transitions to occur at arbitrary time steps.

can be broadly categorized into two types: *applying diffusion algorithms either on the user-item interaction matrix or on user/item representations.* For instance, DiffRec [26] models user preferences by directly perturbing binary-encoded historical interactions and reconstructing the collaborative graph's adjacency matrix. In contrast, methods like DreamRec [32] initiate from Gaussian noise and progressively generate user and item representations in continuous space. By decomposing preference modeling into a multi-step denoising process, diffusion models enhance the capability to capture complex user-item interactions [35].

Despite their demonstrated success, conventional diffusion models operating in continuous state-space face notable limitations when directly applied to discrete CF data. These algorithms typically introduce isotropic Gaussian noise to perturb encoded graph representations, resulting in undesirable information loss. As illustrated in Figure 1 (a), approaches that perturb the interaction matrix apply uniform noise across all items, ignoring item dependencies and disrupting the collaborative graph's topological structure [37]. In Figure 1 (b), methods that corrupt user/item representations with Gaussian noise risk degrading semantic consistency, leading to uninformative diffusion trajectories that hinder effective training and sampling [29].

To overcome these limitations, we investigate diffusion models in discrete state-spaces[1], which have demonstrated significant success in discrete text generation yet remain unexplored in the context of RS. These algorithms operate via state transitions, such as element swapping or replacement, progressively transforming discrete data toward a stationary distribution (Figure 1 (c)). More importantly, the discrete forward process facilitates direct integration of domain knowledge through transition matrices, yielding more informative diffusion trajectories for recommendation learning [1]. This formulation fits the recommendation setting, as it directly models interaction sequences, rather than treating them merely as conditional information [29]. Despite its great potential, directly applying discrete diffusion to recommendation tasks is highly challenging. First, simply perturbing user interactions toward a uniform distribution remains semantically meaningless for recommendation tasks, potentially degrading user preference information. Therefore, the discrete diffusion algorithm requires careful design that incorporates domain-specific knowledge (e.g., item popularity) to preserve personalized signals. Moreover, current theoretical foundations for parameterizing the reverse process remain developing, necessitating tailored adaptation for recommendation tasks to ensure efficient training and generation [16]. Additionally, establishing effective collaborative signal guidance is essential to learn interaction distributions for personalized recommendations.

In this work, we propose a novel Continuous-time Discrete-space Diffusion Recommendation framework (CDRec) to address the above challenges. The proposed CDRec employs an absorbing state to perturb historical interactions via a **masking operation**, and learns a parameterized reverse process for personalized recommendation, as shown in Figure 1 (d). Compared to the discrete-time algorithm, the continuous-time framework, formulated through a stochastic differential equation (SDE), enables flexible state transitions at arbitrary time intervals. This enhances modeling of user behavior patterns while generalizing beyond conventional ancestral sampling strategies [2]. Specifically, the proposed CDRec proposes a novel popularity-aware noise schedule, where items with higher interaction frequencies are assigned lower absorption probabilities during the forward diffusion process. This schedule encourages the diffusion model to simulate real-world interaction processes during reverse diffusion, which starts interaction from popular items, and thereby facilitates more accurate preference modeling. Instead of estimating the reverse transition rate, CDRec parameterizes the reverse diffusion process using a consistency function that models user behavior patterns over masked historical interactions. This formulation enables a balance between sampling quality and efficiency by supporting both single-step and multi-step generation.

Our main contributions are summarized as follows:

- We propose a novel recommendation framework (**CDRec**) that incorporates discrete diffusion processes to support efficient recommendation in continuous-time steps.
- To enhance training and sampling efficiency in the diffusion process, we introduce a novel popularity-aware noise schedule that generates informative diffusion trajectories by simulating real-world interaction dynamics.
- We design a consistency function to parameterize the reverse diffusion process, combined with a contrastive learning objective that integrates collaborative signals to guide the diffusion process for recommendation.
- Extensive experiments on three real-world datasets demonstrating our proposed CDRec's superior performance over state-of-the-art baselines, with ablation studies verifying each component's contribution.

[1]For brevity, "discrete-space diffusion" and "discrete diffusion" are interchangeable terms for diffusion models in discrete state-spaces.

2 The Proposed Method

In this section, we first introduce the key notations and definitions used throughout the paper. We then present an overview of the proposed framework, CDRec, followed by a detailed description of its components.

2.1 Notations and Definitions

In recommender systems, let $\mathcal{U}$ denote the set of n users and $\mathcal{V}$ indicate the set of m items. For each user $u \in \mathcal{U}$, the historical interactions are represented as $\mathbf{x}_u = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_l]$, where l is the predefined sequence length and each $\mathbf{v}_i \in \mathcal{V}$ is represented in one-hot form. To model collaborative signals, we represent users and items through learnable embeddings, where $\mathbf{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_n]^T \in \mathbb{R}^{n \times d}$ denotes the user embedding matrix and $\mathbf{Q} = [\mathbf{q}_1, \ldots, \mathbf{q}_m]^T \in \mathbb{R}^{m \times d}$ represents the item embedding matrix. These embeddings can be initialized using any arbitrary pre-trained GNN-based recommendation model.

2.2 An Overview of the Proposed Framework

We propose a continuous-time framework that directly models the distribution of historical interactions using the discrete-space diffusion algorithm. Specifically, CDRec initiates the forward diffusion process with a popularity-aware noise schedule that perturbs interactions via masking operations, progressively transitioning toward an absorbing state to simulate real-world interaction dynamics. In the reverse generation phase, the framework parameterizes the reverse process with a consistency function, enabling both efficient one-step generation and high-quality iterative sampling. To enhance the personalized recommendation, we incorporate contrastive learning with structural collaborative signals to guide the sampling process. Recommendations are generated by propagating synthesized items as user representations and selecting top-k items based on user–item embedding similarity. This approach fully exploits the generative capacity of diffusion models over interaction sequences, rather than treating them solely as conditional input. The overall framework is demonstrated in Figure 2.

2.3 Popularity-Aware Discrete Diffusion Algorithm

To mitigate information degradation in existing diffusion recommender models, we introduce a novel popularity-aware noise schedule to control the forward diffusion process, enabling the model to learn item distributions across popularity levels at different diffusion phases. This approach integrates prior knowledge to construct informative diffusion trajectories, thereby enhancing both training and sampling.

2.3.1 Forward Diffusion Process. The forward process constructs a Continuous-Time Markov Chain to transform the original data distribution p_0 into a stationary distribution p_{ref}. Formally, we model the user u's historical interactions $\mathbf{x}_u \sim p_0$ as the initial state, with an absorbing state serving as the prior distribution [1]. During forward diffusion, each item $\mathbf{v}_i$ is denoted as $\mathbf{x}_0 \in \mathbb{R}^{1 \times |\mathcal{V}|}$ under the SDE framework, and undergoes progressive replacement by a [MASK] token. The absorbing state distribution preserves the real user-interacted data throughout the diffusion trajectory, thereby providing more meaningful learning signals than corrupted embeddings [29]. The transition between time points t and $t + \Delta t$ can be described by the following SDE:

$$q(\mathbf{x}_{t+\Delta t} = y | \mathbf{x}_t = x) = \delta_{x,y} + \mathbf{R}_t(y, x)\Delta t + O(\Delta t), \quad (1)$$

where x and y represents the item state at time t and $t + \Delta t$ respectively, δ refers to the Kronecker delta, $\mathbf{R}_t \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ denotes the transition rate matrix, and $O(\Delta t)$ refers to the terms whose magnitude becomes negligible relative to Δt in the limit as $\Delta t \to 0$. This absorbing behavior is determined by the transition rate matrix $\mathbf{R}_t$ assigned to each item, where higher transition rates correspond to shorter absorption times.

According to the law of total probability, the marginal distribution $p_t(\mathbf{x})$ at time t is given by [19]:

$$p_t(\mathbf{x}) = \int p_0(\mathbf{x}_0) q_{t|0}(\mathbf{x}_t | \mathbf{x}_0) d\mathbf{x}_0. \quad (2)$$

To derive an efficient transition kernel $q_{t|0}(\mathbf{x}_t | \mathbf{x}_0)$, CDRec adopts the standard formulation $\mathbf{R}_t = \beta(t)\mathbf{R}$, where $\beta(t)$ refers to the noise schedule, and $\mathbf{R}$ denotes the manually defined fixed base rate. Consequently, the analytical solution is derived as follows:

$$\begin{aligned} q_{t|0}(\mathbf{x}_t = j | \mathbf{x}_0 = i) &= \left(\mathrm{S}\exp(\Lambda \overline{\beta}(t)) \mathrm{S}^{-1} \right)_{ij}, \\ \overline{\beta}(t) &= \int_0^t \beta_t dt, \end{aligned} \quad (3)$$

where $\mathbf{R} = \mathrm{S}\Lambda\mathrm{S}^{-1}$ is the eigendecomposition of the base rate matrix $\mathbf{R}$, with S containing the eigenvectors of $\mathbf{R}$ and Λ being the diagonal matrix of eigenvalues. The $\exp(\cdot)$ denotes the element-wise exponential function, and i, j index distinct states.

2.3.2 Popularity-aware Noise Schedule. Existing approaches predominantly adopt linear or cosine schedules for $\beta(t)$, which are suboptimal for recommendation tasks. These conventional schedules typically assume uniform item contributions to preference modeling, thereby overlooking important inter-item correlations [23]. To address this limitation and construct more informative diffusion trajectories, we introduce a novel sequence-level noise schedule that simulates real-world interaction patterns by prioritizing popular items [10]. Since popular items typically reflect current trends and better characterize user behavior, they provide particularly meaningful information for interaction modeling [34]. Consequently, our method retains high-frequency items for extended durations during diffusion, thereby enabling more precise modeling of user preferences.

We formulate a popularity deviation metric $I(\cdot)$ that measures the relative popularity of an item $\mathbf{v}$ within the user's interaction sequence $\mathbf{x}_u$:

$$I(\mathbf{v}) = \mathrm{pop}(\mathbf{v}) - \frac{\sum_0^l \mathrm{pop}(\mathbf{v}_i)}{l}, \quad (4)$$

where $\mathbf{v}_i$ denotes i-th interacted item in the sequence $\mathbf{x}_u$, and $\mathrm{pop}(\mathbf{v}) \in [0, 1]$ denotes the frequency of item $\mathbf{v}$, normalized within the interaction sequence. Our design captures item correlations with respect to popularity by assigning positive deviation values to items with above-average interaction frequencies and negative values otherwise. Then the absorbing probability of each item $\mathbf{v}_i$ is

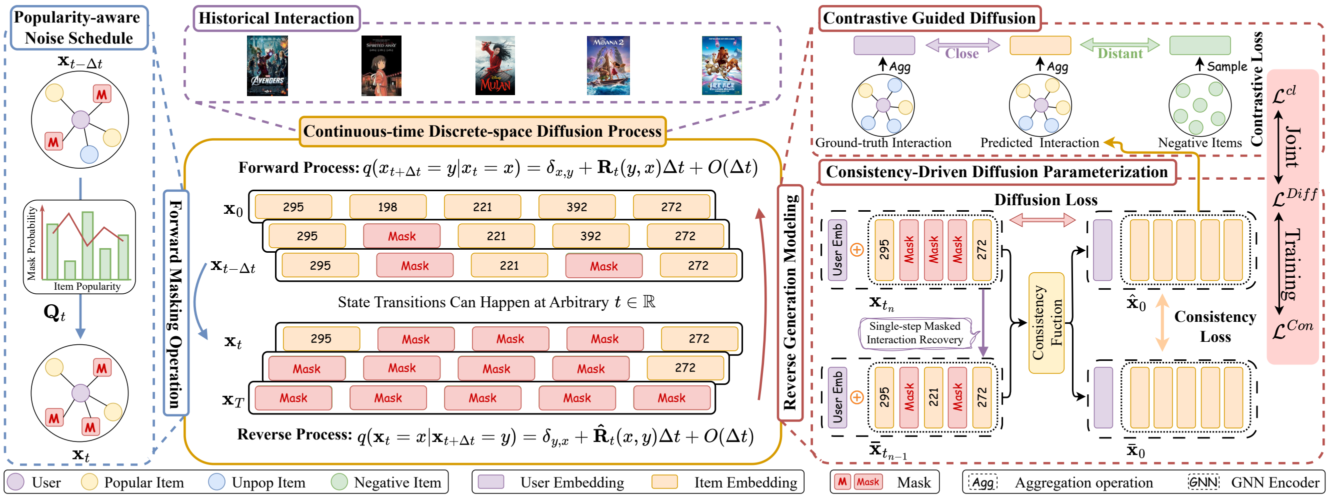


Figure 2: The proposed CDRec framework comprises three main modules: popularity-aware noise schedule, consistency-driven diffusion parameterization, and contrastive-guided diffusion. The left panel illustrates the forward diffusion process, where state transitions are controlled by item popularity to simulate interaction dynamics. The right panel presents the reverse generation modeling process, which leverages a consistency function for efficient parameterization and applies contrastive learning to guide personalized recommendations with the structural collaborative signal.

defined as :

$$\overline{\beta}(t)_{\mathbf{v}_i} = \frac{t}{T} - \omega \exp\left(-\frac{t-\frac{T}{2}}{2\sigma^2}\right) I(\mathbf{v}_i), \tag{5}$$

where ω and σ are hyperparameters to scale the noise schedule at time step t. The popularity deviation effect is adaptively scaled according to a normal distribution, as illustrated in Figure 3. Compared to linear schedules, this yields lower absorbing probabilities for relatively popular items and higher probabilities for unpopular ones. Our design guarantees that higher-frequency items ($I(\mathbf{v}_i) > I(\mathbf{v}_j)$) experience lower corruption rates ($\overline{\beta}(t)_{\mathbf{v}_i} < \overline{\beta}(t)_{\mathbf{v}_j}$) during forward diffusion. This creates an easy-to-hard reverse process, as popular items inherently exhibit more stable patterns [34]. This staged masking strategy adapts to popularity variations across diffusion stages, allowing the model to progressively learn items of varying popularity at different phases, thereby effectively capturing the interaction distribution.

Moreover, our popularity-aware noise schedule satisfies the standard transition rate requirements [2]: (1) guaranteed convergence of the data distribution to the terminal distribution, and (2) efficient computation of the transition kernel $q_{t|0}(\mathbf{x}_t|\mathbf{x}_0)$ through Equations 3 and 5, with the structured matrix $\mathbf{R}$ defined as:

$$\mathbf{R} = \begin{bmatrix} -1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & -1 & 0 \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix}. \tag{6}$$

2.3.3 Reverse Diffusion Process. The reverse SDE learns to generate personalized item interactions through iterative denoising from the absorbing state, effectively modeling the underlying dynamics of real user preference evolution. The reverse-time SDE can be formulated as:

$$q(\mathbf{x}_t = x | \mathbf{x}_{t+\Delta t} = y) = \delta_{y,x} + \hat{\mathbf{R}}_t(x,y)\Delta t + O(\Delta t), \tag{7}$$

where $\hat{\mathbf{R}}_t \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ represents the reverse transition rate matrix. Considering both the forward and reverse diffusion processes are Markovian, the $\mathbf{R}_t$ and $\hat{\mathbf{R}}_t$ satisfy the following correlation:

$$\hat{\mathbf{R}}_t(x,y) = \frac{q_t(y)}{q_t(x)}\mathbf{R}_t(y,x), \tag{8}$$

where $q_t(y)$ and $q_t(x)$ denote the marginal probabilities of state x and y at time t , respectively. The ratio $q_t(y)/q_t(x)$ serves as a concrete score, analogous to the score function $\nabla_{\mathbf{x}} \log p_t$ in continuous-state diffusion algorithms [23]. However, this concrete score is generally intractable and requires accurate parameterization to enable the reverse diffusion generation process.

2.4 Consistency-Driven Diffusion Parameterization

Current parameterization methods for discrete-space diffusion remain developing and can be broadly categorized into two classes: modeling the reverse density $p_{0|t}$ and modeling the score $q_t(y)/q_t(x)$, both of which face inherent limitations in the recommendation context [16]. Direct estimation of $p_{0|t}$ often yields suboptimal performance due to discrete distributions [2], while learning the score typically requires substantial computational resources and struggles to ensure positive transition rates during the training process, potentially introducing undesired bias [17, 23].

To address these limitations, we explore an alternative consistency function for parameterizing discrete diffusion processes, inspired by consistency models [21]. Our parameterization enables efficient single-step generation while supporting iterative refinement to balance sample quality and computational cost.

2.4.1 Consistency Function. The consistency function learns the diffusive trajectory of the masked historical interaction $\{\mathbf{x}_t\}_{t\in[\varepsilon,T]}$,

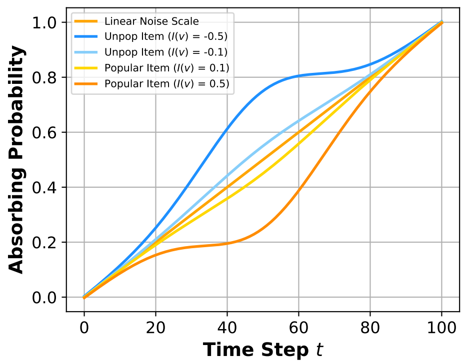


Figure 3: Absorbing probability versus diffusion time step for items with varying popularity deviation ($\omega = 0.5$).

bypassing explicit estimation of the reverse transition rate $\hat{\mathbf{R}}_t$. It directly outputs predicted user-item interactions by reconstructing the original preference distribution conditioned on the collaborative signal. Following the standard formulation [7], we define the consistency function as follows:

$$f : (\mathbf{x}_t, t) \mapsto \mathbf{x}_\varepsilon, \tag{9}$$

where ε denotes a fixed small positive time point, prior to which no state transitions occur.

This formulation naturally aligns with recommendation objectives by recovering latent user-item interactions. However, the well-established skip-connection architectures for consistency functions are fundamentally incompatible with discrete-space diffusion, as superposition operations on discrete states $\{\mathbf{x}_t\}_{t\in[\varepsilon,T]}$ lack semantic validity and implementation feasibility [21]. Thus, we parameterize the consistency model in a manner that aligns with the discrete nature of CDRec as

$$f_\theta(\mathbf{x}, t) \begin{cases} \mathbf{x}, & t = \varepsilon \\ F_\theta(\mathbf{x}, t), & t \in (\varepsilon, T] \end{cases} \tag{10}$$

where F_θ is a free-form deep neural network.

2.4.2 Consistency Training Strategy. The core of training the consistency function lies in minimizing the difference between outputs from data pairs at adjacent time points, $(\bar{\mathbf{x}}_{t-\Delta t}, \mathbf{x}_t)$, where $\bar{\mathbf{x}}_{t_{n-1}}$ is estimated from $\mathbf{x}_{t_n}$, enabling f_θ to estimate user preferences from samples at arbitrary time steps [7]. In the context of discrete-space diffusion, this strategy is inefficient, as state transitions may not occur within the infinitesimal interval Δt. To address this, we denote the current time step $t \sim \mathcal{U}(\varepsilon, T)$ as t_n, and we define $t_{n-1} = t_n - \nabla t$, where ∇t represents the fixed time interval. The training objective $\mathcal{L}^{Con}$, tailored for discrete-space diffusion, is formulated as:

$$\mathcal{L}^{Con} = \min_\theta \mathbb{E}_{\mathbf{x}_0, t_n}[\gamma(t_n) d(f_\theta(\mathbf{x}_{t_n}, t_n), f_{\theta^-}(\bar{\mathbf{x}}_{t_{n-1}}, t_{n-1})], \tag{11}$$

where $\gamma(\cdot)$ is a time-dependent weighing function, $d(\cdot,\cdot)$ denotes a distance metric between the outputs, and θ^- represents the exponential moving average of past values of θ, updated via $\theta^- \leftarrow stopgrad(\mu\theta^- + (1-\mu)\theta)$ to stabilize the training process. The primary difficulty in applying this training objective lies in the efficient estimation of $\bar{\mathbf{x}}_{t_{n-1}}$. In the context of recommendation, no pre-trained model exists to approximate the reverse transition kernel $p(\bar{\mathbf{x}}_{t_{n-1}}|\mathbf{x}_{t_n})$, and training a dedicated model would be computationally expensive. Moreover, unlike continuous-state SDEs, discrete-space diffusion algorithms lack an unbiased estimator for the score $q_t(y)/q_t(x)$ [16, 17].

Since the discrete process operates through scheduled item masking, the transition kernel $p(\bar{\mathbf{x}}_{t_{n-1}}|\mathbf{x}_{t_n})$ can be simplified via masked element recovery algorithms, bypassing complex mathematical derivations. To enable efficient training of the consistency function for the discrete-space diffusion, we propose two methods to construct the data pair $(\bar{\mathbf{x}}_{t_{n-1}}, \mathbf{x}_{t_n})$ by transit masked items to original state from sampled $\mathbf{x}_{t_n}$. The first method, termed one-step recovery, aims to sample $\bar{\mathbf{x}}_{t_{n-1}}$ by recovering only one masked item for simplicity. Given a sample $\mathbf{x}_{t_n}$ drawn from the transition density $q_{t|0}(\mathbf{x}_{t_n}|\mathbf{x}_0)$, we obtain the masking probabilities for each position as $\left[\overline{\beta}(t_n)_{\mathrm{v}_1}, \overline{\beta}(t_n)_{\mathrm{v}_2}, \ldots, \overline{\beta}(t_n)_{\mathrm{v}_I}\right]$. To construct $\bar{\mathbf{x}}_{t_{n-1}}$, we simply recover the masked item with the lowest masking probability. This method is simple and efficient, but requires careful design of the time interval ∇t to mitigate exposure bias [18], which refers to the behavioral gap between training and sampling. Consequently, recovering only one item may not accurately reflect the data distribution at time step t_{n-1}. The second method adopts a pseudo-Euler approach to adaptively estimate the unmasking probabilities of items over the time interval ∇t [23]. The reverse transition probability for each item is defined as:

$$p_{t_{n-1}|t_n}(\mathbf{x}^{\mathrm{v}_i}_{t_{n-1}} = x|\mathbf{x}^{\mathrm{v}_i}_{t_n} = y) \begin{cases} 1 - (B_{t_n} - \frac{\nabla t}{T} B_{t_n}), & \text{if } x \neq [M], y = [M] \\ B_{t_n} - \frac{\nabla t}{T} B_{t_n}, & \text{if } x = y = [M] \\ 1 & \text{if } x = y \neq [M], \end{cases} \tag{12}$$

where $B_{t_n} = \overline{\beta}(t_n)_{\mathrm{v}_i}$ for brevity, and $[M]$ represents the absorbing state. This method recovers masked items with probabilities scaled by the time interval and original masking rate, aligning well with the forward diffusion process. Given the data pair $(\bar{\mathbf{x}}_{t-\Delta t}, \mathbf{x}_t)$, we can efficiently train the consistency function f_θ. This enables single-step generation while maintaining iterative sampling capability, as detailed in Algorithm 1.

The $\mathcal{L}^{Con}$ enforces identical predictions under data perturbations to model the interaction distribution in a step-wise manner, but it may over-emphasize popular items, as they tend to persist longer in the diffusive trajectory. To mitigate this bias, we empirically introduce a diffusion loss $\mathcal{L}^{Diff}$ as an auxiliary denoising objective, directly modeling historical interactions through the following formulation:

$$\mathcal{L}^{Diff} = -\sum_{i=1}^{I} \left(\mathbf{x}^i_u \log p^i_\theta(\mathbf{x}_0|\mathbf{x}_t)\right), \tag{13}$$

where p^i_θ represents the predicted categorical probability for i-th element generated by the consistency model F_θ, $\mathbf{x}^i_u$ denotes the ground-truth value of i-th element. The experiments demonstrate that this loss function stabilizes the training process while enhancing the model's capacity, consistent with previous studies [1, 9].

2.4.3 Practical Implementation. In practice, the consistency model F_θ is instantiated as a Transformer encoder. For a user u with perturbed interactions $\mathbf{x}_t$ at time step t, the encoded interaction sequence is prefixed with a collaborative user embedding to preserve personalized information: $\langle \mathbf{p}_u, \mathbf{q}_{v_1}, \ldots, \mathbf{q}_{v_l} \rangle$. We adopt a layer-wise time embedding strategy, where the embedding of time step t is added to the user representation $\mathbf{p}_u$. The output $\hat{\mathbf{x}}_0$, formulated as $\langle \hat{\mathbf{p}}_u, \hat{\mathbf{q}}_{v_1}, \ldots, \hat{\mathbf{q}}_{v_l} \rangle$, is projected back to the discrete item space using cosine similarity with the item embedding matrix $\mathbf{Q}$. The i-th generated item is represented as a one-hot vector $\hat{\mathbf{x}}^{\mathbf{v}_i}$. Additionally, the distance metric $d(\cdot)$ is defined using Kullback–Leibler divergence to encourage F_θ to produce consistent outputs across different partially masked historical interactions of users.

Algorithm 1 Multistep Sampling Process

Require: F_θ: Consistency function, $\mathbf{x}_T$: Absorbing state initialization, T: Total diffusion step, $[\delta_1, \delta_2 \ldots \delta_N = T]$: Discretized diffusion steps, N: Sampling steps.
  $\hat{\mathbf{x}} \leftarrow F_\theta(\mathbf{x}_T, T)$
  for $n = N - 1$ to 0 **do**
    $\mathbf{x}_{\delta_n} \sim q_{t|0}(\cdot|\hat{\mathbf{x}}^{\mathbf{v}_i}) = \exp(\overline{\beta}(\delta_n)\mathbf{R})_{\hat{\mathbf{x}}^{\mathbf{v}_i}}$.
    $\hat{\mathbf{x}} \leftarrow F_\theta(\mathbf{x}_{\delta_n}, \delta_n)$
  end for
Output: $\hat{\mathbf{x}}$: Prediction of user-item interaction

2.5 Contrastive Guided Diffusion

Although discrete-space diffusion algorithms demonstrate strong capability in modeling data distributions, these methods primarily focus on learning direct historical interactions (i.e., one-hop neighbors in collaborative graphs) while neglecting valuable multi-hop relational information. To bridge this gap, CDRec incorporates contrastive learning to guide the diffusion process with structural information encoded in the user representations $\mathbf{P}$.

2.5.1 Contrastive Learning. Given the generated interaction sequence, $\langle \hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \ldots, \hat{\mathbf{v}}_l \rangle$, we aggregate the corresponding collaborative embeddings to construct the user representation $\mathbf{e}_u$, simulating the propagation behavior of a single GNN layer [11, 33]. To effectively transfer domain knowledge from the collaborative graph, we utilize the pre-trained user embedding $\mathbf{p}_u$ as a positive sample while employing a randomly sampled non-interacted item set $\mathcal{V}_u^-$ as negative samples for contrastive learning. The CDRec employs the InfoNCE loss as our self-supervised learning objective to guide the diffusion generation, which can be defined as:

$$\mathcal{L}^{cl} = \sum_{u \in \mathcal{U}} -\log \frac{\varsigma(\mathbf{e}_u, \mathbf{p}_u)}{\varsigma(\mathbf{e}_u, \mathbf{p}_u) + \sum_{v \in \mathcal{V}_u^-} \varsigma(\mathbf{e}_u, \mathbf{q}_v))}, \tag{14}$$

where the similarity function is defined as $\varsigma(\hat{\mathbf{e}}_u, \mathbf{p}_*) = \exp\left(\cos(\hat{\mathbf{e}}_u, \mathbf{p}_*)/\tau\right)$ to compute the cosine similarity between embeddings, and τ is the temperature parameter.

2.5.2 Joint Training Mechanism. The joint optimization objective for CDRec combines multiple learning objectives through the following weighted formulation:

$$\mathcal{L} = \lambda_1 \mathcal{L}^{Con} + (1 - \lambda_1)\mathcal{L}^{Diff} + \lambda_2 \mathcal{L}^{cl}, \tag{15}$$

where λ_1 balances two complementary objectives: the consistency loss, which emphasizes pattern consistency by producing identical predictions for interactions with varying masking levels, and the diffusion loss, which performs behavior reconstruction by directly recovering the original user–item interactions; while λ_2 balances the guidance from the structural information of the collaborative graph with the modeling of direct one-hop relations. The training procedure for the denoising model is presented in Algorithm 2.

Algorithm 2 Training Algorithm of CDRec

Require: $f_\theta(\mathbf{x}, t)$: Consistency function, ∇t represents the fixed time interval, $\mathbf{RecEnc}(\cdot)$: Pre-trained GNN-based recommendation model.
  repeat
    Sample the historical interaction $\mathbf{x}_u = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_l]$ of user u as $\mathbf{x}_0$.
    $t_n \leftarrow t \sim \mathcal{U}(\varepsilon, T)$
    $\mathbf{x}_{t_n} \sim q_{t_n|0}(\cdot|\mathbf{x}_0^{\mathbf{v}_i}) = \exp(\overline{\beta}(t_n)\mathbf{R})_{\mathbf{x}_0^{\mathbf{v}_i}}$.
    $t_{n-1} \leftarrow t_n - \nabla t$
    if Using One-step Recovery **then**
      Sample $\bar{\mathbf{x}}_{t_{n-1}}$ by recovering only one masked item with the lowest absorbing probability.
    else if Using Pseudo-Euler **then**
      Sample $\bar{\mathbf{x}}_{t_{n-1}}$ via Equation 12.
    end if
    Encode the $\mathbf{x}_{t_n}$ and $\bar{\mathbf{x}}_{t_{n-1}}$ with $\mathbf{RecEnc}(\cdot)$.
    Compute the $\mathcal{L}^{Con}$ via Equation 11.
    Compute the $\mathcal{L}^{Diff}$ via Equation 13.
    Sample non-interacted item set $\mathcal{V}_u^-$.
    Compute the $\mathcal{L}^{cl}$ via Equation 14.
    Update the parameters θ via gradient descent.
  until converged
Output: f_θ: Optimized consistency function

3 Experiment

In this section, we conduct comprehensive experiments to assess the performance of our proposed CDRec framework.

3.1 Experiment Settings

3.1.1 Datasets. Experiments are conducted on three real-world datasets: Ciao [2], MovieLens-1M [3], and Dianping [4], where user-item ratings follow a $[1-5]$ scale. Following a widely adopted setting, we treat ratings of 3 and above as implicit interactions. Each dataset is partitioned into training, validation, and testing sets with an $8:1:1$ ratio. Dataset statistics are summarized in Table 2.

3.1.2 Evaluation Metrics. We evaluate top-K recommendation performance using two standard ranking metrics: Recall@K (R@K) and Normalized Discounted Cumulative Gain (N@K), with $K \in \{5, 10\}$ under the full-ranking setting. All results are averaged over five independent runs to ensure reliability.

3.1.3 Baselines. We evaluate the performance of CDRec against state-of-the-art baselines from three categories: GNN-based recommenders (NGCF [27], LightGCN [8], SGL [28]), VAE-based models (Mult-VAE [12], VGCL [31]), and diffusion-based approaches

[2]Ciao: https://www.cse.msu.edu/~tangjili/trust.html
[3]MovieLens-1M: https://grouplens.org/datasets/movielens/1m/
[4]Dianping: https://grouplens.org/datasets/movielens/1m/

Table 1: Comparison of overall performance on three recommendation datasets. The best performance is highlighted in bold, and the second best is underlined. %Improve denotes the relative improvement of CDRec over the strongest baseline.

Dataset	Ciao				MovieLens-1M				Dianping			
Method	R@10	R@5	N@10	N@5	R@10	R@5	N@10	N@5	R@10	R@5	N@10	N@5
NGCF	0.0483	0.0266	0.0396	0.0321	0.1201	0.0724	0.1212	0.1136	0.0531	0.0316	0.0381	0.0317
LightGCN	0.0525	0.0312	0.0441	0.0370	0.1297	0.0783	0.1231	0.1193	0.0543	0.0333	0.0421	0.0339
SGL	0.0553	0.0364	0.0493	0.0438	0.1319	0.0852	0.1277	0.1207	0.0546	0.0339	0.0433	0.0365
Muti-VAE	0.0517	0.0361	0.0409	0.0341	0.1337	0.0863	0.1193	0.1152	0.0547	0.0341	0.0415	0.0345
VGCL	0.0568	0.0346	0.0477	0.0422	0.1339	0.0846	0.1315	0.1273	0.0551	0.0336	0.0431	0.0353
DiffRec	0.0436	0.0289	0.0397	0.0361	0.1351	0.0839	0.1379	0.1309	0.0556	0.0345	0.0439	0.0367
L-DiffRec	0.0458	0.0293	0.0417	0.0371	0.1326	0.0799	0.1351	0.1169	0.0539	0.0327	0.0435	0.0371
BSPM	0.0590	0.0363	0.0518	0.0447	0.1399	0.0871	0.01347	0.1301	0.0557	0.0329	0.0441	0.0359
GiffCF	0.0584	0.0372	0.0473	0.0396	0.1426	0.0857	0.1395	0.1293	0.0505	0.0339	0.0412	0.0361
CDRec	**0.0617**	**0.0390**	**0.0539**	**0.0473**	**0.1486**	**0.0931**	**0.1469**	**0.1384**	**0.0586**	**0.0355**	**0.0471**	**0.0394**
%Improve	4.58%	4.84%	4.06%	5.81%	4.21%	6.88%	5.30%	5.72%	5.21%	2.90%	6.80%	6.19%

Table 2: Statistics of the datasets

Dataset	Ciao	MovieLens-1M	Dianping
#Users	7,373	6,038	131,629
#Items	91,091	3,533	10,932
#Interactions	227,392	575,281	1,412,544
Interaction Density	0.0338%	2.6967%	0.0981%

(DiffRec [26], DiffRec-L [26], BSPM [4], GiffCF [37]). Descriptions of these baselines are provided below:

- NGCF [27]: NGCF employs graph convolutional networks to model collaborative signals through embedding propagation.
- LightGCN [8]: LightGCN simplifies NGCF by eliminating feature transformation and nonlinear activation layers, focusing specifically on collaborative filtering tasks.
- SGL [28]: SGL introduces multiple auxiliary tasks to capture structural relatedness across different views, leveraging this information to enhance the supervision signals.
- Mult-VAE [12]: Mult-VAE produces interaction vectors via variational inference based on a multinomial distribution.
- VGCL [31]: VGCL integrates VAE with contrastive learning by employing variational graph reconstruction to generate multiple views for collaborative filtering.
- DiffRec [26]: DiffRec introduces the diffusion algorithm by perturbing the user's historical interactions across the item vector.
- L-DiffRec [26]: L-DiffRec encodes collaborative data using a VAE framework and performs the diffusion process in the latent space.
- BSPM [4]: BSPM leverages score-based generative models for collaborative filtering by incorporating a blurring and sharpening process on the interaction matrix.
- GiffCF [37]: GiffCF applies a diffusion algorithm, grounded in the heat equation, to the item–item similarity graph.

3.1.4 Parameter Settings. The proposed CDRec framework is implemented in PyTorch. We set the batch size to 1024 for all three datasets, with the sequence length l set to 20. CDRec is optimized using the Adam optimizer, and the learning rate is tuned from $\{0.001, 0.0001, 0.0001\}$. The number of sampling steps N is selected from $\{1, 3, 5, 10, 30, 50, 100\}$, with T is evaluated in the range $\{N, 2N\}$ and the corresponding step size adjusted within ∇t within $\{1, 5, 10, 20\}$. The hyperparameter λ_1 is searched in the range $[0.1, 1]$ while the λ_2 is selected from $\{1.0, .01, 0.001, 0.0001\}$. We empirically set the $\omega = 0.5$, and $\sigma = T/10$. In this work, user and item representations are initialized using a pre-trained 3-layer LightGCN model.

3.2 Overall Performance

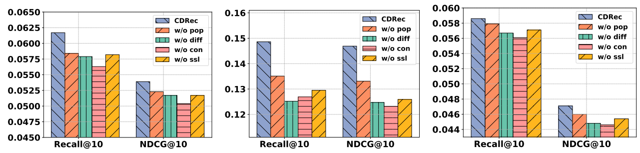
We conduct a comprehensive evaluation of recommendation performance by comparing the proposed CDRec framework against existing baselines. As shown in Table 1, we report Recall and NDCG metrics across three benchmark datasets: Ciao, MovieLens-1M, and Dianping. Our main findings include:

- Generative recommender systems outperform GNN-based approaches, particularly on large-scale sparse datasets, by directly modeling user interaction distributions rather than learning low-rank embeddings from high-dimensional interaction matrices.
- While diffusion models outperform baseline approaches by capturing complex data distributions more effectively, their performance improvements remain modest, likely due to information loss during discrete-to-continuous data transformation.
- CDRec demonstrates consistent superiority over all baseline methods across all evaluated datasets. The performance gains can be attributed to three key components. Our popularity-aware noise scheduler effectively simulates real-world interaction patterns through an easy-to-hard sampling process. The proposed consistency function accurately captures interaction distributions conditioned on the collaborative signals, leading to substantially improved recommendation accuracy. Moreover, the contrastive component bridges one-hop and multi-hop neighborhood representations, enhancing personalized sampling.

3.3 Ablation Study

To systematically evaluate the core components of CDRec, we design four architectural variants:

- *w/o pop*: Ablates the popularity-aware component, implementing standard uniform masking with linear noise schedule.
- *w/o con*: Removes the consistency loss $\mathcal{L}^{con}$, forcing the model to estimate reverse densities $p_{0|t}$ in the discrete state space.



(a) Ciao (b) MovieLens-1M (c) Dianping

Figure 4: Ablation study of CDRec and its variants on three datasets with evaluation metrics Recall@10 and NDCG@10.

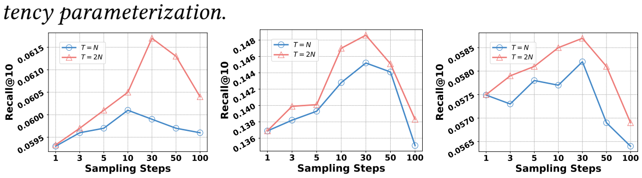
- *w/o diff*: Eliminate the diffusion loss $\mathcal{L}^{diff}$, relying solely on the consistency loss for interaction distribution learning.
- *w/o ssl*: Ablates the contrastive learning objective $\mathcal{L}^{cl}$, resulting in a denoising process that is solely conditioned on the user representation prefix.

Figure 4 presents our ablation study results. The popularity-aware noise scheduler proves significantly more effective than linear scheduling, demonstrating its ability to capture real-world interaction patterns while improving training stability and sampling quality. Performance suffers most severely when removing either the consistency or diffusion losses, confirming their fundamental role in modeling interaction distributions. Additionally, the contrastive learning component contributes measurably to performance, verifying its effectiveness in exploiting multi-hop relational patterns for personalized recommendation generation.

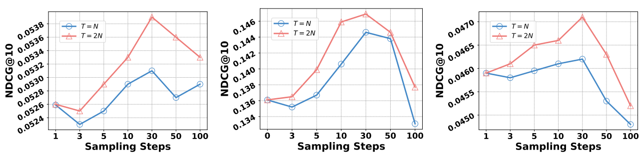
3.4 Parameter Sensitivity Study

In this section, we examine the impact of CDRec's key parameters.

3.4.1 Diffusion Steps. We evaluate the sampling steps N, with corresponding diffusion steps $T \in \{T = N, T = 2N\}$. As shown in Figure 5, optimal performance occurs at 30 sampling steps with $T = 60$ across all datasets. After this point, performance degrades with additional steps, likely due to noise accumulation from excessive sampling iterations. Moreover, experimental results indicate that larger T values generally improve performance with the same sampling step, suggesting that an extended step T better simulates the continuous-time diffusion trajectory and encourages the model to learn the interaction distribution. Importantly, *the one-step recommendation of CDRec achieves performance comparable to diffusion-based baselines, demonstrating the efficiency of the consistency parameterization.*

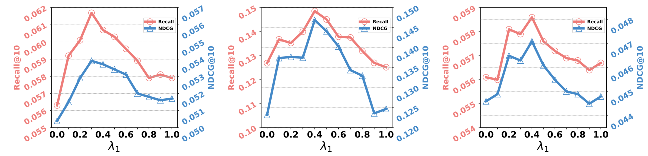


(a) Ciao (b) MovieLens-1M (c) Dianping



(d) Ciao (e) MovieLens-1M (f) Dianping

Figure 5: Effect of sampling steps N on Recall@10 and NDCG@10.

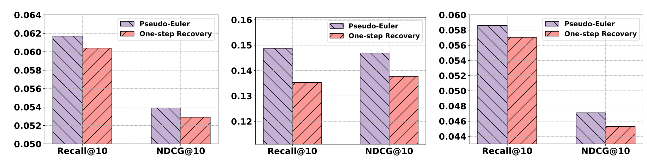


(a) Ciao (b) MovieLens-1M (c) Dianping

Figure 6: Effect of λ_1 on Recall@10 and NDCG@10.

3.4.2 Loss Weights. We conduct the sensitivity study on λ_1 to examine CDRec's training dynamics, with results shown in Figure 6. The best performance is achieved at $\lambda_1 = 0.4$, with a decline observed as λ_1 deviates in either direction. These results suggest that incorporating $\mathcal{L}^{Diff}$ effectively enhances the modeling of interaction distributions in the recommendation context while mitigating undesired bias.

3.4.3 Construction of $(\bar{\mathbf{x}}_{t_{n-1}}, \mathbf{x}_{t_n})$. We evaluate two methods, pseudo-Euler and one-step recovery, for constructing the data pair $(\bar{\mathbf{x}}_{t_{n-1}}, \mathbf{x}_{t_n})$ to train the consistency model F_θ. The results, shown in Figure 7, indicate that pseudo-Euler consistently outperforms one-step recovery across all three datasets. This advantage likely arises because Equation 12 adaptively demasks items based on time intervals ∇t, thereby more accurately simulating the diffusion process.



(a) Ciao (b) MovieLens-1M (c) Dianping

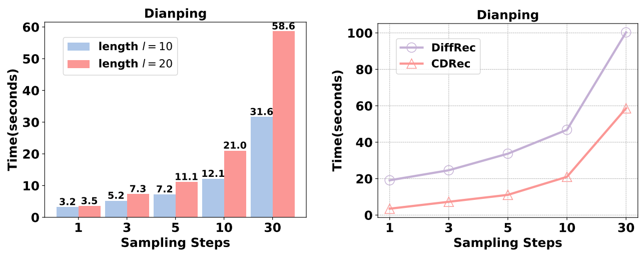
Figure 7: Effect of methods to construct $(\bar{\mathbf{x}}_{t_{n-1}}, \mathbf{x}_{t_n})$ on Recall@10 and NDCG@10.

3.5 Model Investigation

This section presents a comprehensive evaluation of CDRec, including its time efficiency and case study analyses.

3.5.1 Time Analysis. We systematically evaluate the sampling efficiency of CDRec under varying parameter settings and conduct comparative analysis with the baseline DiffRec model. Figure 8(a) illustrates the time cost with respect to two key parameters: sequence length and sampling steps. The results show that time cost increases sharply with the number of sampling steps, underscoring the importance of achieving the single-step recommendation while maintaining comparable performance. Figure 8(b) demonstrates the comparison of sampling time of CDRec and DiffRec on the Dianping dataset under the same settings (test batch size is 1024). CDRec outperforms DiffRec as its discrete diffusion algorithm perturbs data through masking operations rather than applying Gaussian noise to the entire item vector.

3.5.2 Case Study. We illustrate the forward masking diffusion process for a specific user in the MovieLens-1M dataset across the diffusion steps. The observations are as follows: (1) The user's preferences span action films, comedies, and dramas, with relatively popular films in these categories typically watched first, aligning with our assumption of user behavior. (2) The proposed popularity deviation metric guides the forward diffusion process, allowing



(a) Sequence Length **(b) CDRec vs. DiffRec**

Figure 8: Time Analysis

more popular movies to be retained longer. However, the diffusion mechanism introduces randomness, so the masking order does not strictly follow interaction frequency. This randomness gives the model a greater opportunity to model less popular items, helping to mitigate potential bias.

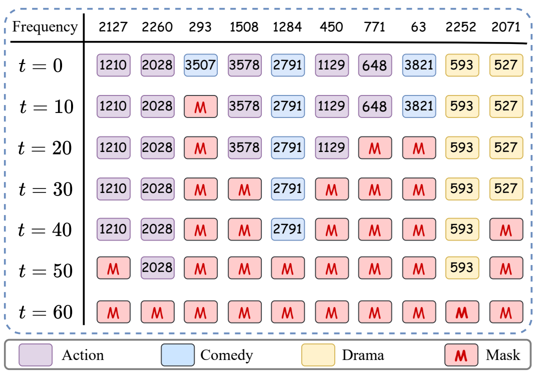


Figure 9: Case study of the forward diffusion process for user ID 98 in MovieLens-1M with $\omega = 0.5$. Movies are initially ordered by interaction timestamps for illustration only.

4 Related Work

This section reviews the related work in diffusion models for recommender systems and discrete-space diffusion algorithms.

4.1 Diffusion Recommender Model

Diffusion models have emerged as a powerful generative approach for RS, achieving remarkable performance by modeling complex data distributions. In collaborative filtering, DiffRec [26] pioneered diffusion-based recommendation by applying DDPM [9] to binary interaction vectors to model user–item interactions in latent space. Building on this, GiffCF [37] integrates graph-level diffusion to capture higher-order relations, while BSPM [4] employs a continuous-time diffusion framework on the adjacency matrix. On the other hand, DreamRec [32] adopts the learn-to-generate paradigm, applying diffusion processes to target representations conditioned on historical interactions. PreferDiff [15] introduces negative samples during training for sequential recommendation, potentially aligning with the Direct Preference Optimization (DPO) framework. Despite their advancements, these methods require encoding discrete collaborative graph data into continuous space, which poses a risk of collaborative information loss.

DCDR [13] introduced discrete-state diffusion into recommendation for reranking, marking an early exploration in this direction. However, discrete diffusion in DCDR [13] only perturbs item positions within interactions, without fully exploiting the modeling capacity of discrete diffusion frameworks. This highlights the necessity of developing a discrete diffusion algorithm that directly models the distribution of user preferences while enabling an efficient sampling process.

4.2 Discrete Diffusion Model

Discrete diffusion models have attracted growing attention for their capacity to model complex distributions in discrete data spaces [30]. D3PM [1] proposed the first discrete-state diffusion framework for categorical random variables in discrete time. Instead of Gaussian noise, D3PM introduces several stationary distributions, such as the uniform distribution (where the transition probability to any other state is uniform) and the absorbing state (where perturbation is applied by masking the token). Campbell et al. [2] first formulated discrete-state diffusion using stochastic differential equations (SDEs) in continuous time to enhance sampling flexibility. However, its discrete parameterization of the reverse process constrains empirical performance. Sun et al. [23] introduced the ratio matching algorithm to model marginal probabilities via maximum likelihood training, followed by methods such as CSM [17] and SEDD [16], which proposed alternative score matching approaches. However, the parameterization methods remain underdeveloped, hindered by requirements for specialized architectures, constraints on positive probabilities, and high computational cost.

5 Conclusion

In this paper, we propose CDRec, a framework that employs a discrete-state diffusion algorithm in continuous time to model the distribution of user–item interactions. CDRec leverages an absorbing state to perturb historical interactions via masking operations and learns a parameterized reverse process to generate personalized recommendations. To better capture real-world interaction generation, we propose a popularity-aware noise schedule that assigns lower absorption probabilities to items with higher interaction frequencies, enabling an easy-to-hard generation strategy for more accurate preference modeling. In the reverse process, we parameterize it with a consistency function that models user behavior patterns over masked historical interactions, thereby mitigating the limitations of existing parameterization methods in recommendation contexts. This framework balances sampling quality and efficiency by supporting both single-step and multi-step generation. Extensive experiments on real-world datasets show that CDRec consistently outperforms existing methods, validating its effectiveness.

6 Acknowledgments

The research described in this paper has been partially supported by the General Research Funds from the Hong Kong Research Grants Council (project no. PolyU 15207322, 15200023, 15206024, and 15224524), internal research funds from Hong Kong Polytechnic University (project no. P0042693, P0048625, and P0051361). This work was supported by computational resources provided by The Centre for Large AI Models (CLAIM) of The Hong Kong Polytechnic University.

7 Ethical Considerations

In this section, we discuss potential ethical issues associated with the proposed CDRec framework. Although designed to enhance recommendation accuracy and efficiency, CDRec may also pose risks related to data privacy, fairness, and bias.

Data Privacy. Recommendation systems inherently process user behavioral data, which may contain sensitive personal information. Although our framework does not require explicit personal identifiers, improper data handling or insufficient anonymization could still expose users to privacy risks, including re-identification and unauthorized data misuse. To address these concerns, we adhere to strict data usage policies, such as enforcing access controls and encryption, to ensure compliance with privacy regulations (e.g., GDPR, CCPA). Furthermore, robust anonymization techniques, such as k-anonymity and tokenization, are applied to safeguard sensitive attributes. These measures collectively mitigate privacy risks and protect user information throughout the training process.

Fairness and Bias. Since the model is trained on historical user–item interaction data, it may inherently encode societal biases or skewed popularity distributions. Without appropriate intervention, these biases could be perpetuated or even amplified, leading to unequal item exposure or unfair treatment of certain user groups. To mitigate such risks, regular bias audits should be performed to monitor disparate impacts using established fairness metrics. These audits, combined with fairness-aware model adjustments, can help ensure equitable outcomes while preserving recommendation performance.

References

[1] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. *Proc. Adv. Neural Inf. Process. Syst.* 34 (2021), 17981–17993.

[2] Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. 2022. A continuous time framework for discrete denoising models. *Proc. Adv. Neural Inf. Process. Syst.* 35 (2022), 28266–28279.

[3] Xiao Chen, Wenqi Fan, Jingfan Chen, Haochen Liu, Zitao Liu, Zhaoxiang Zhang, and Qing Li. 2023. Fairly adaptive negative sampling for recommendations. In *Proc. World Wide Web Conf.* 3723–3733.

[4] Jeongwhan Choi, Seoyoung Hong, Noseong Park, and Sung-Bae Cho. 2023. Blurring-sharpening process models for collaborative filtering. In *Proc. 46th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval.* 1096–1106.

[5] Wenqi Fan, Xiaorui Liu, Wei Jin, Xiangyu Zhao, Jiliang Tang, and Qing Li. 2022. Graph trend filtering networks for recommendation. In *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval.* 112–121.

[6] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *Proc. World Wide Web Conf.* 417–426.

[7] Zhengyang Geng, Ashwini Pokle, Weijian Luo, Justin Lin, and J Zico Kolter. 2025. Consistency Models Made Easy. In *Proc. 13th Int. Conf. Learn. Representations.* 96638–96666.

[8] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval.* 639–648.

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Proc. Adv. Neural Inf. Process. Syst.* 33 (2020), 6840–6851.

[10] Shiu-Li Huang and Yi-Hsien Lin. 2022. Exploring consumer online purchase and search behavior: An FCB grid perspective. *Asia Pacific Management Review* 27, 4 (2022), 245–256.

[11] Chaejeong Lee, Jeongwhan Choi, Hyowon Wi, Sung-Bae Cho, and Noseong Park. 2025. SCONE: A Novel Stochastic Sampling to Generate Contrastive Views and Hard Negative Samples for Recommendation. In *Proc. 18th ACM Int. Conf. Web Search Data Mining.* 419–428.

[12] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proc. World Wide Web Conf.* 689–698.

[13] Xiao Lin, Xiaokai Chen, Chenyang Wang, Hantao Shu, Linfeng Song, Biao Li, and Peng Jiang. 2024. Discrete conditional diffusion for reranking in recommendation. In *Companion Proceedings of the ACM Web Conference 2024.* 161–169.

[14] Chengyi Liu, Jiahao Zhang, Shijie Wang, Wenqi Fan, and Qing Li. 2025. Score-based generative diffusion models for social recommendations. *IEEE Trans. Knowl. Data Eng.* 37 (2025), 6666–6679.

[15] Shuo Liu, An Zhang, Guoqing Hu, Hong Qian, and Tat-Seng Chua. 2025. Preference Diffusion for Recommendation. In *Proc. 13th Int. Conf. Learn. Representations.* 11043–11081.

[16] Aaron Lou, Chenlin Meng, and Stefano Ermon. 2024. Discrete Diffusion Modeling by Estimating the Ratios of the Data Distribution. In *Proc. Int. Conf. Mach. Learn.*

[17] Chenlin Meng, Kristy Choi, Jiaming Song, and Stefano Ermon. 2022. Concrete score matching: Generalized score matching for discrete data. *Proc. Adv. Neural Inf. Process. Syst.* 35 (2022), 34532–34545.

[18] Mang Ning et al. 2024. Elucidating the Exposure Bias in Diffusion Models. In *Proc. 12th Int. Conf. Learn. Representations.* 21641–21664.

[19] Bernt Øksendal. 2003. Stochastic differential equations. In *Stochastic differential equations: an introduction with applications.* Springer, 38–50.

[20] Haohao Qu, Wenqi Fan, and Shanru Lin. 2025. Generative Recommendation with Continuous-Token Diffusion. *arXiv preprint arXiv:2504.12007* (2025).

[21] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. 2023. Consistency Models. In *Proc. Int. Conf. Mach. Learn.* PMLR, 32211–32252.

[22] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2021. Score-Based Generative Modeling through Stochastic Differential Equations. In *Proc. 9th Int. Conf. Learn. Representations.* 240–276.

[23] Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. 2022. Score-based continuous-time discrete diffusion models. *arXiv preprint arXiv:2211.16750* (2022).

[24] Shijie Wang, Wenqi Fan, Yue Feng, Lin Shanru, Xinyu Ma, Shuaiqiang Wang, and Dawei Yin. 2025. Knowledge Graph Retrieval-Augmented Generation for LLM-based Recommendation. In *Proc. Annu. Meet. Assoc. Comput. Linguist.* Association for Computational Linguistics, 27152–27168.

[25] Wenjie Wang, Xinyu Lin, Fuli Feng, Xiangnan He, and Tat-Seng Chua. 2023. Generative recommendation: Towards next-generation recommender paradigm. *arXiv preprint arXiv:2304.03516* (2023).

[26] Wenjie Wang, Yiyan Xu, Fuli Feng, Xinyu Lin, Xiangnan He, and Tat-Seng Chua. 2023. Diffusion recommender model. In *Proc. 46th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval.* 832–841.

[27] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval.* 165–174.

[28] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval.* 726–735.

[29] Wenjia Xie, Hao Wang, Luankang Zhang, Rui Zhou, Defu Lian, and Enhong Chen. 2024. Breaking determinism: Fuzzy modeling of sequential recommendation using discrete state space diffusion model. *Proc. Adv. Neural Inf. Process. Syst.* 37 (2024), 22720–22744.

[30] Zhe Xu, Ruizhong Qiu, Yuzhong Chen, Huiyuan Chen, Xiran Fan, Menghai Pan, Zhichen Zeng, Mahashweta Das, and Hanghang Tong. 2024. Discrete-state continuous-time diffusion for graph generation. *Proc. Adv. Neural Inf. Process. Syst.* 37 (2024), 79704–79740.

[31] Yonghui Yang, Zhengwei Wu, Le Wu, Kun Zhang, Richang Hong, Zhiqiang Zhang, Jun Zhou, and Meng Wang. 2023. Generative-contrastive graph learning for recommendation. In *Proc. 46th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval.* 1117–1126.

[32] Zhengyi Yang, Jiancan Wu, Zhicai Wang, Xiang Wang, Yancheng Yuan, and Xiangnan He. 2023. Generate what you prefer: Reshaping sequential recommendation via guided diffusion. *Proc. Adv. Neural Inf. Process. Syst.* 36 (2023), 24247–24261.

[33] Junyu Zhang, Daochang Liu, Shichao Zhang, and Chang Xu. 2023. Contrastive sampling chains in diffusion models. *Proc. Adv. Neural Inf. Process. Syst.* 36 (2023), 73524–73542.

[34] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. Causal intervention for leveraging popularity bias in recommendation. In *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval.* 11–20.

[35] Jujia Zhao, Wang Wenjie, Yiyan Xu, Teng Sun, Fuli Feng, and Tat-Seng Chua. 2024. Denoising diffusion recommender model. In *Proc. 47th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval.* 1370–1379.

[36] Zihuai Zhao, Wenqi Fan, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, et al. 2024. Recommender systems in the era of large language models (llms). *IEEE Trans. Knowl. Data Eng.* (2024).

[37] Yunqin Zhu, Chao Wang, Qi Zhang, and Hui Xiong. 2024. Graph signal diffusion model for collaborative filtering. In *Proc. 47th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval.* 1380–1390.