Fading to Grow: Growing Preference Ratios via Preference Fading Discrete Diffusion for Recommendation

Guoqing Hu[‡] **An Zhang**[‡*] **Shuchang Liu**[§] **Wenyu Mao**[‡] **Jiancan Wu**[‡]

Xun Yang[‡] **Xiang Li**[§] **Lantao Hu**[§] **Han Li**[§] **Kun Gai**[§] **Xiang Wang**[‡]

[‡]University of Science and Technology of China
[§]Independent Researcher
{HugoChinn}@mail.ustc.edu.cn
{an_zhang,xyang21}@ustc.edu.cn
{wenyumao2, wujcan, xiangwang1223}@gmail.com
{xiangli.th11, hulantao, lee.han06}@gmail.com, gai.kun@qq.com

Abstract

Recommenders aim to rank items from a discrete item corpus in line with user interests, yet suffer from extremely sparse user preference data. Recent advances in diffusion models have inspired diffusion-based recommenders, which alleviate sparsity by injecting noise during a forward process to prevent the collapse of perturbed preference distributions. However, current diffusion-based recommenders predominantly rely on continuous Gaussian noise, which is intrinsically mismatched with the discrete nature of user preference data in recommendation. In this paper, building upon recent advances in discrete diffusion, we propose **PreferGrow**, a discrete diffusion-based recommender system that models preference ratios by fading and growing user preferences over the discrete item corpus. PreferGrow differs from existing diffusion-based recommenders in three core aspects: (1) Discrete modeling of preference ratios: PreferGrow models relative preference ratios between item pairs, rather than operating in the item representation or raw score simplex. This formulation aligns naturally with the discrete and ranking-oriented nature of recommendation tasks. (2) Perturbing via preference fading: Instead of injecting continuous noise, PreferGrow fades user preferences by replacing the preferred item with alternatives—physically akin to negative sampling—thereby eliminating the need for any prior noise assumption. (3) Preference reconstruction via growing: PreferGrow reconstructs user preferences by iteratively growing the preference signals from the estimated ratios. PreferGrow offers a well-defined matrix-based formulation with theoretical guarantees on Markovianity and reversibility, and it demonstrates consistent performance gains over state-of-the-art diffusion-based recommenders across five benchmark datasets, highlighting both its theoretical soundness and empirical effectiveness. Our codes are available at `https://github.com/Hugo-Chinn/PreferGrow`.

1 Introduction

Recommender systems aim to rank items from a discrete item set that align with user interests, where ones in the user-item interaction matrix denote observed interactions, and zeros indicate unobserved

*Corresponding authors.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

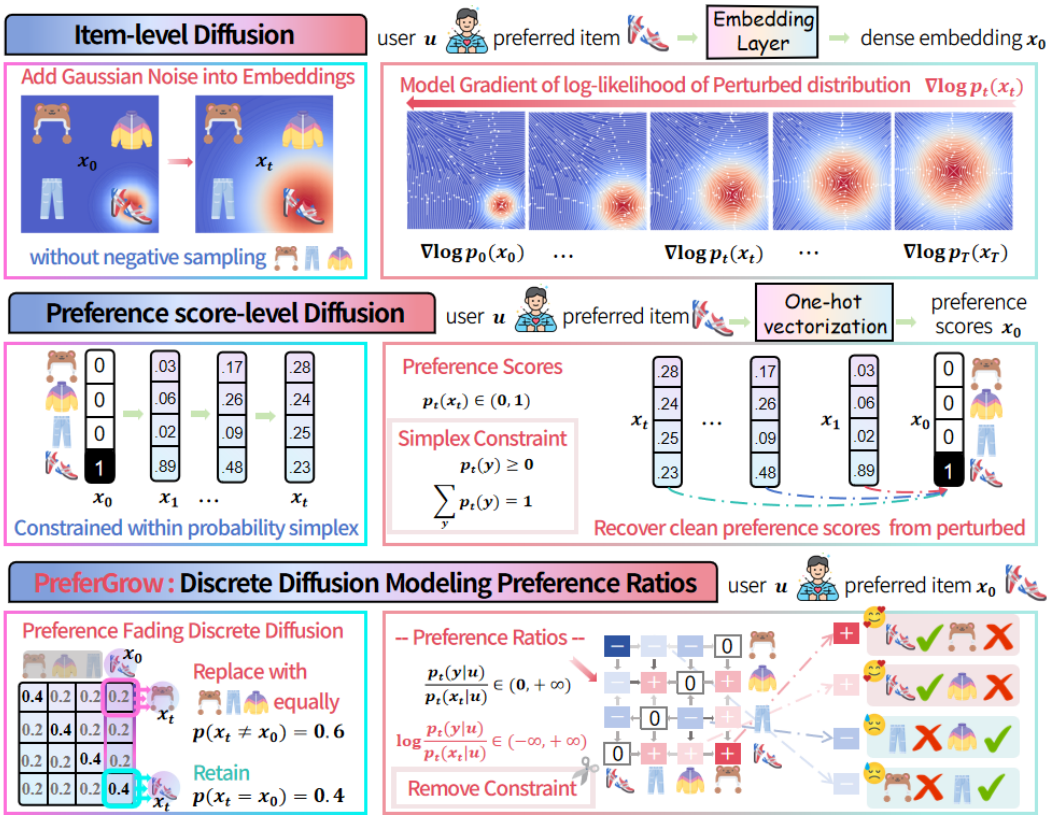


Figure 1: Comparison of diffusion-based recommenders in terms of modeling targets and perturbation strategies. Item-level diffusion recommenders (top) add Gaussian noise into dense item embeddings but overlook negative signals. Preference score-level diffusion recommenders (middle) perturb one-hot preference vectors under the constraints of the probability simplex. PreferGrow (bottom) directly models discrete preference ratios via preference fading.

or missing entries. In real-world scenarios, this interaction matrix is often extremely sparse [1, 2], which poses a significant challenge for recommenders in accurately modeling user preferences [3–5]. Recent advances in diffusion models (DMs) offer a promising solution: by injecting noise during the forward process, DMs transform sparse data into smoother, denser distributions while preventing collapse into isotropic zero values [6–8]. Motivated by these properties, diffusion-based recommenders [9–37] have proliferated in recent studies, showing strong potential in addressing data sparsity and improving preference modeling.

Current diffusion-based recommenders predominantly adopt continuous noise as the perturbation mechanism by adding noise into either dense item embeddings or one-hot preference score vectors. As shown in Figure 1, item-level diffusion recommenders [9–27] apply Gaussian noise to user-interacted item embeddings during the forward process, and learn a score function—the gradient of the perturbed distribution's log-likelihood—to recover item embeddings during the reverse process. However, these methods overlook the negative signals in recommendations, lacking mechanisms such as negative sampling to differentiate user preferences [9, 27]. In contrast, preference score-level diffusion recommenders [28–37] model users' preference scores among the full item corpus, *i.e.,* user interacted one-hot vectors, by perturbing them within the probability simplex in the forward process. While the reverse process attempts to reconstruct the preference scores, the simplex constraints—non-negativity and normalization—introduce additional optimization difficulties [38]. Worse still, both diffusion-based recommenders assume a prior noise in the sampling process, *e.g.,* Gaussian [9] or Bernoulli [29], which may poorly reflect the inherently discrete and sparse nature of the user preference data in recommendation scenarios.

Building on recent advances in discrete diffusion models [39, 38, 40], we propose **PreferGrow**, a discrete diffusion-based recommender that models the *preference ratios* by fading and growing user preferences through forward and backward processes. Unlike existing continuous diffusion-based

recommenders that model preference scores (*e.g.,* probabilities that users interacted with items), PreferGrow directly models relative preference ratios over a discrete item set. This formulation aligns more naturally with the discrete and ranking-oriented nature of recommendation tasks and avoids the strong constraints imposed by the probability simplex. In the forward perturbation, PreferGrow fades user preference by replacing the preferred item with alternatives, enabling explicit negative sampling and alleviating data sparsity without relying on predefined noise distribution. In the backward generation, PreferGrow reconstructs user preferences by iteratively growing the preference signal from the estimated ratios. We further provide a theoretical analysis showing that these forward and backward processes preserve key properties of discrete diffusion models.

PreferGrow offers a unified and theoretically grounded framework for discrete diffusion-based recommendation, characterized by a well-defined matrix-based formulation, flexible and interpretable preference ratio modeling aligned with diverse negative sampling strategies, and consistently superior empirical performance.

- **Theoretical foundation:** At the core of the formulation for PreferGrow is an idempotent fading matrix used to replace preferred items during the forward perturbation process. We provide a closed-form solution of this preference fading matrix and theoretically prove that its idempotent property is critical for ensuring both the Markov property and reversibility of the diffusion process.
- **Flexible modeling:** By parameterizing the preference fading matrix, PreferGrow flexibly supports point-wise, pair-wise, and hybrid preference ratios modeling. These variants correspond to distinct and physically interpretable negative sampling strategies, enabling the framework to adapt to diverse modeling targets in recommendation.
- **Empirical validation:** Extensive experiments on five benchmark datasets demonstrate that PreferGrow consistently outperforms existing diffusion-based recommenders, validating the practical effectiveness of its theoretical foundations.

2 Preliminaries

User preference data is represented as a pair (u, i), where u denotes the user and i is the preferred item. The preference score $p(i|u)$ indicates the probability that the user u interacts with item i. In sequential recommendation, u denotes the item sequence that the user has interacted with.

Diffusion-based Recommenders generally consist of three major components: a forward noise-adding process for perturbation, a generative modeling target, and a backward denoising process for reconstruction. Specifically, **item-level diffusion-based recommenders** encode a preferred item i into its dense embedding $\mathbf{x}_0$, and instantiate three components as follows [41]:

- *forward perturbation*: $\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1-\alpha_t}\epsilon_t$, where $\alpha_t \in [0,1], \epsilon_t \sim \mathcal{N}(0, I)$.
- *modeling target*: network $\mathbf{s}_\Theta(\mathbf{x}_t, t, u) \approx \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|u) = \frac{\sqrt{\alpha_t}\mathbf{x}_0 - \mathbf{x}_t}{1-\alpha_t} = -\frac{\epsilon_t}{\sqrt{1-\alpha_t}}$.
- *backward generation*: $\mathbf{x}_s = \sqrt{\alpha_s}\frac{\mathbf{x}_t + (1-\alpha_t)\mathbf{s}_\Theta(\mathbf{x}_t,t,u)}{\sqrt{\alpha_t}} - \sqrt{1-\alpha_s}\sqrt{1-\alpha_t}\mathbf{s}_\Theta(\mathbf{x}_t, t, u), s < t$.

Preference score–level diffusion-based recommenders represent the preferred item i as a one-hot preference score $\mathbf{x}_0$, modeling user preference over the entire item space. Early works [28, 30, 34] adopt Gaussian noise priors $\mathcal{N}(\cdot,\cdot)$; however, Gaussian perturbations are incompatible with the probability simplex constraints inherent to preference scores. To address this mismatch, subsequent studies [29, 35] introduce discrete priors to replace the Gaussian assumption that preserves the simplex structure throughout the diffusion process. For instance, RecFusion [29] adopts a Bernoulli prior $\mathcal{B}(\cdot;\cdot)$, resulting in a binomial diffusion formulation:

- *forward perturbation*: $\mathbf{x}_t \sim \mathcal{B}\left(\mathbf{x}_t; \alpha_t\mathbf{x}_0 + \frac{\alpha_t(1-\alpha_t)}{2}\right), \alpha_t \in [0,1]$.
- *modeling target*: $\mathcal{B}\left(\mathbf{x}_{t-1}; \mathbf{s}_\Theta(\mathbf{x}_t, t, u)\right) \approx p(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$.
- *backward generation*: $\mathbf{x}_{t-1} \sim \mathcal{B}\left(\mathbf{x}_{t-1}; \mathbf{s}_\Theta(\mathbf{x}_t, t, u)\right)$.

The other approach employs a Categorical prior $\text{Cat}(\cdot;\cdot)$:

- *forward perturbation:* $p_t(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \text{Cat}(\mathbf{x}_t; \overline{\mathbf{Q}}_t\mathbf{x}_0)$ where $\overline{\mathbf{Q}}_t = \prod_{i=1}^{t}\mathbf{Q}_i$.

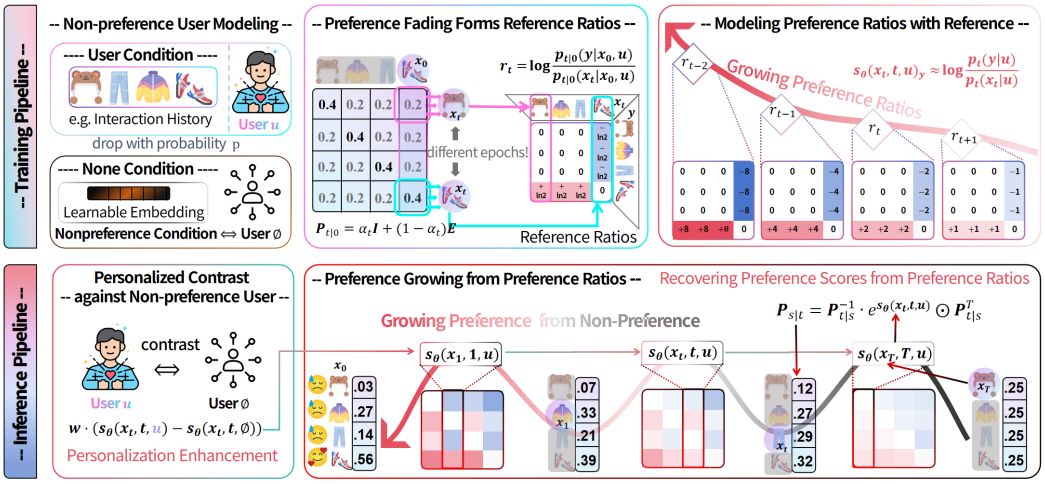


Figure 2: The overall training and inference pipeline of PreferGrow under the pair-wise setting.

- *modeling target:* $\mathbf{s}_\Theta(\mathbf{x}_t, t, u) \approx \mathbf{x}_0$.
- *backward generation:* $p_t\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0 = \mathbf{s}_\Theta(\mathbf{x}_t, t, u)\right) = \text{Cat}\left(\mathbf{x}_{t-1}; \frac{\mathbf{Q}_t^\top \mathbf{x}_t \odot \overline{\mathbf{Q}}_{t-1} \mathbf{s}_\Theta(\mathbf{x}_t, t, u)}{\mathbf{x}_t^\top \overline{\mathbf{Q}}_t \mathbf{s}_\Theta(\mathbf{x}_t, t, u)}\right)$.

Preference Ratios $\log \frac{p(i_p|u)}{p(i_d|u)}$ characterize the relative preference between items, and a positive value indicates a more preferred item i_p and a less preferred item i_d. Notably, preference modeling in RLHF [42–44] and DPO [45–48] is grounded in the Bradley–Terry model [49], which explicitly models preference ratios as in Equation (1). This underscores the effectiveness of preference ratios in more faithfully capturing user preferences.

$$p(i_p \succ i_d \mid u) = \frac{p(i_p \mid u)}{p(i_p \mid u) + p(i_d \mid u)} = \frac{1}{1 + \exp\left(-\log \frac{p(i_p|u)}{p(i_d|u)}\right)} = \sigma(\log \frac{p(i_p \mid u)}{p(i_d \mid u)}). \quad (1)$$

Discrete Diffusion Models previously are formulated based on the following Kolmogorov forward and backward equations [50, 51, 38]:

$$\frac{\partial \mathbf{P}_{t|s}}{\partial t} = \mathbf{Q}_t \mathbf{P}_{t|s}, \qquad \frac{\partial \mathbf{P}_{s|t}}{\partial t} = \mathbf{R}_s \mathbf{P}_{s|t}, \quad (2)$$

where $\mathbf{P}_{t|s}$ denotes the transition probability matrix from time s to time t, $\mathbf{Q}_t$ is the forward transition rate matrix at time t, and $\mathbf{R}_s$ is the reverse-time transition rate matrix at time s. Discrete diffusion models further model the ratios of data distributions through the following score entropy loss [38]:

$$\mathcal{L}_{SE} = \mathbb{E}_{x_0 \sim p_{data}} \mathbb{E}_{t \in U[0,T]} \mathbb{E}_{x_t \sim p_{t|0}(\cdot|x_0)} \left[\sum_{y \in \mathcal{X}} \mathbf{Q}_t(x_t, y) \cdot l_{SE}(x_0, x_t, y) \right], \quad (3)$$

$$l_{SE}(x_0, x_t, y) = e^{\mathbf{s}_\Theta(y, x_t)} - \frac{p_{t|0}(y|x_0)}{p_{t|0}(x_t|x_0)} \mathbf{s}_\Theta(y, x_t) + \frac{p_{t|0}(y|x_0)}{p_{t|0}(x_t|x_0)} [\log \frac{p_{t|0}(y|x_0)}{p_{t|0}(x_t|x_0)} - 1], \quad (4)$$

where $\mathcal{X}$ denotes the discrete state space, and $\mathbf{s}_\Theta(y, x_t)$ is the output of a neural network that estimates the preference ratio between y and x_t at timestep t.

3 Method

As illustrated in Figure 2, the pipeline of PreferGrow consists of three stages: the preference fading discrete diffusion process (Section 3.1), the modeling of preference ratios with reference via score entropy loss (Section 3.2), and the preference growing reverse generation process (Section 3.3). Additionally, PreferGrow models a non-preference user and achieves personalized enhancement (Section 3.4). Theoretical proofs are provided in Appendix C, while the computation of score entropy loss under different fading matrix settings is detailed in Appendix D.

3.1 Forward Perturbation: Preference Fading Discrete Diffusion Process

PreferGrow operates directly on the discrete item set $\mathcal{X}$, wherein x_0 represents the preferred item i of user u. PreferGrow first fades user preference based on whether to retain or not to retain the preferred item x_0. The whole fading process is further achieved by progressively decreasing the probability of retention α_t from timestep 0 to T, fading user preferences towards non-preference.

3.1.1 Preference Fading Forms Reference Ratios

Building upon recent advances in discrete DM [39, 38, 40], we introduce a preference fading discrete diffusion model tailored for recommendation. For a user preference data (u, x_0), we fade user preference by retaining the target item $x_t = x_0$ with probability α_t and replacing x_0 according to discrete distribution $\mathbf{E}(\mathcal{X}, x_0)$ as the following equation:

$$p_{t|0}(x_t|x_0) = \alpha_t \delta_{x_0}(x_t) + (1 - \alpha_t)\mathbf{E}(x_t, x_0), x_t \in \mathcal{X}. \tag{5}$$

where Dirac delta function $\delta_{x_0}(x_t)$ equals 1 when $x_t = x_0$, and 0 otherwise. Also, in matrix form:

$$\mathbf{P}_{t|0} = \alpha_t \mathbf{I} + (1 - \alpha_t)\mathbf{E}, \tag{6}$$

where $\mathbf{I}$ is the identity matrix, representing retention, and $\mathbf{E}$ is a matrix whose column sums equal 1, defining the replacing mode for perturbation. For simplicity, we refer to $\mathbf{E}$ as fading matrix. We further demonstrate that the idempotence of the fading matrix $\mathbf{E}$ ensures the Markov property and reversibility of preference fading discrete diffusion process, which is crucial for the reverse generation.

Theorem 1. *Suppose $\alpha_t : [0, T] \to [0, 1]$ is a strictly decreasing function with $\alpha_0 = 1$ and $\alpha_T = 0$. If $\mathbf{E} \in \mathbb{R}^{N \times N}$ is idempotent,* i.e., $\mathbf{E}^2 = \mathbf{E}$*, the following properties hold:*

- ***Markov property:*** *$\{\mathbf{P}_{t|0}\}_{t=0}^T$ is a Markov process and satisfies the Chapman-Kolmogorov equation:*

$$\mathbf{P}_{t|s} := \frac{\alpha_t}{\alpha_s}\mathbf{I} + \left(1 - \frac{\alpha_t}{\alpha_s}\right)\mathbf{E}, \tag{7}$$

$$\mathbf{P}_{t|r} = \mathbf{P}_{t|s}\mathbf{P}_{s|r}, \quad \textit{for all } 0 \le r \le s \le t \le 1. \tag{8}$$

- ***Invertibility:*** *Each $\mathbf{P}_{t|s}$ is invertible, and its inverse is given by:*

$$\mathbf{P}_{t|s}^{-1} = \frac{\alpha_s}{\alpha_t}\mathbf{I} + \left(1 - \frac{\alpha_s}{\alpha_t}\right)\mathbf{E}. \tag{9}$$

Upon obtaining x_t through preference fading, as illustrated in Figure 2, we can compute the reference ratios for all items $\mathcal{X}$ at timestep t with fading awareness $1 - \alpha_t$ as follows:

$$r_t(x_0, x_t, y) = \log \frac{p_{t|0}(y \mid x_0, u)}{p_{t|0}(x_t \mid x_0, u)} = \log \frac{\alpha_t \delta_{x_0}(y) + (1 - \alpha_t)\mathbf{E}(y, x_0)}{\alpha_t \delta_{x_0}(x_t) + (1 - \alpha_t)\mathbf{E}(x_t, x_0)}, \forall y \in \mathcal{X}. \tag{10}$$

As shown in Figure 2, when the preferred item x_0 is retained, *i.e.,* $x_t = x_0$, the reference ratio $r_t(x_0, x_t = x_0, y \neq x_0)$ is negative, reflecting a tendency to stay with the current item. Conversely, when x_0 is replaced, *i.e.,* $x_t \neq x_0$, the reference ratio $r_t(x_0, x_t \neq x_0, y = x_0)$ tends to be positive, highlighting the preference for the positive item x_0. In this case, the faded item x_t can be interpreted as a negative item, establishing a conceptual connection to the mechanism of negative sampling.

3.1.2 Design Paradigms of the Idempotent Fading Matrix

Given that designing such an idempotent fading matrix is crucial, we derive that, if the preference fading process converges to a unified state, the fading matrix $\mathbf{E}$ can be expressed in closed form.

Proposition 1. *Suppose the preference fading Markov process $\{\mathbf{P}_{t|0}\}_{t=0}^T$ converges to a unified non-preference state $\vec{p}_T$. Then, the fading matrix $\mathbf{E}$ can be expressed in closed form as:*

$$\mathbf{E} = \frac{\vec{p}_T \vec{1}^\top}{\vec{1}^\top \vec{p}_T}. \tag{11}$$

Equation (11) gives a rank-1 solution for the fading matrix $\mathbf{E}$. Most existing discrete diffusion models [38–40, 52–55] implicitly adopt such rank-1 instances (i.e., the absorbing and uniform cases). In Appendix D, examine several representative rank-1 configurations of $\mathbf{E}$ that, in a physical sense, correspond to distinct negative sampling strategies and thus induce different forms of preference ratios. Specifically, by parameterizing the rank-1 preference-fading matrix, PreferGrow flexibly supports point-wise, pair-wise, and hybrid preference-ratio modeling. These variants align with interpretable negative-sampling schemes, enabling the framework to adapt to diverse recommendation objectives. For each setting, we provide analytical solutions for the fading matrix, closed-form expressions for the reference ratios, and simplified training losses. We believe prior works on negative sampling [56, 57] offer valuable guidance for designing more effective rank-1 fading matrices; a systematic exploration is left for future work. Beyond the rank-1 case, Appendix E derives a general rank-r solution for $\mathbf{E}$. We present a closed-form characterization together with a discussion of its physical interpretation. Notably, a rank-r fading matrix induces a quantization of the item space — i.e., a partition into r components. A comprehensive study of this quantization mechanism and its algorithmic implications is deferred to our future work.

3.2 Modeling Target: Preference Ratios with Reference via Score Entropy

We begin by computing the reference ratios $r_t(x_0, x_t, y)$ induced by preference fading, as defined in Equation (10). These reference ratios are then employed to guide the modeling of the user-conditioned preference ratio $\log \frac{p_t(y|u)}{p_t(x_t|u)}$, where u denotes the user context. Concretely, given a training instance (u, x_0) in sequential recommendation, we encode the interaction history u with a sequential recommender to obtain a user embedding $\mathbf{u} = \mathrm{SeqRec}(u)$ (instantiated as SASRec [58] in our experiments). We then parameterize a learnable function s_Θ to estimate the preference ratio for each $y \in \mathcal{X}$ at time t:

$$s_\Theta(x_t, t, u)_y = \mathbf{y}^\top \mathbf{MLP}\big(\,\mathrm{concat}(\mathbf{x}_t, \mathbf{t}, \mathbf{u})\big), \quad y \in \mathcal{X}, \tag{12}$$

where $\mathbf{y}$ and $\mathbf{x}_t$ are the embeddings of item y and the faded item x_t, respectively, and $\mathbf{t}$ is the embedding of timestep t. For the training objective, we adopt the well-known score entropy loss for discrete diffusion modeling [38], as defined in Equations (3) and (4). To legitimately employ it, we first verify that PreferGrow satisfies the applicability conditions of score entropy loss.

Proposition 2. *The preference-fading discrete diffusion $\{\mathbf{P}_{t|s}\}_{0\le s\le t\le T}$ with an idempotent fading matrix $\mathbf{E}$ satisfies the Kolmogorov forward equation:*

$$\frac{\partial \mathbf{P}_{t|s}}{\partial t} \;=\; \mathbf{Q}_t\, \mathbf{P}_{t|s}. \tag{13}$$

Here the rate matrix $\mathbf{Q}_t$ and α_t are defined succinctly by

$$\mathbf{Q}_t := \lim_{s\to t} \frac{\partial \mathbf{P}_{t|s}}{\partial t} = \beta(t)\,(\mathbf{E} - \mathbf{I}), \quad \alpha_t := \exp\Big(-\int_0^t \beta(\tau)\,\mathrm{d}\tau\Big), \tag{14}$$

with $\beta(\tau) > 0$ and $\mathbf{I}$ the identity matrix.

To better explain the effectiveness of the Score Entropy (SE) loss in recommendation, we theoretically analyze its connection to the Binary Cross-Entropy (BCE) loss [56].

Proposition 3. *Define the soft label $\pi_{y\succ x_t|x_0} = p(y \succ x_t \mid x_0) = \sigma(r_t(x_0, x_t, y))$ as in Equation (1), where σ denotes the sigmoid function, and use it as the label for the BCE loss. Treating $\sigma(s_\Theta(x_t, t, u)_y)$ as the prediction for the BCE loss yields the soft-label BCE objective:*

$$\mathcal{L}_{sBCE} = -\pi_{y\succ x_t|x_0} \log \sigma(s_\Theta(x_t, t, u)_y) - (1 - \pi_{y\succ x_t|x_0}) \log(1 - \sigma(s_\Theta(x_t, t, u)_y)).$$

From a gradient perspective, the SE loss and the soft-label BCE loss are related by:

$$\nabla_{s_\Theta} \mathcal{L}_{SE} = (1 + e^{s_\Theta(x_t,t,u)_y})(1 + e^{r_t(x_0,x_t,y)}) \nabla_{s_\Theta} \mathcal{L}_{sBCE}. \tag{15}$$

Consequently, the SE loss and the soft-label BCE loss share the same descent direction (up to a positive scalar) and attain the same optima. Hence, although originating from discrete diffusion modeling, SE loss is equally well suited to ranking-oriented recommendation tasks as the BCE loss.

3.3 Backward Generation: Preference Growing from Preference Ratios

After estimating the preference ratios via score entropy, we reverse the preference fading process to grow user preferences from the non-preference state. We first demonstrate that this reverse process of preference fading, referred to as the preference growing process, is Markovian and satisfies the Kolmogorov backward equation. Together with Proposition 2, this implies that PreferGrow with an idempotent fading matrix inherits the key properties of previous discrete diffusion models [51, 38].

Theorem 2. *If the preference fading process* $\{\mathbf{P}_{t|0}\}_{t=0}^{T}$ *with an idempotent fading matrix* $\mathbf{E}$ *converges to a unified non-preference state* $\vec{p}_T$*, the reverse preference growing process* $\{\mathbf{P}_{s|T}\}_{s=T}^{0}$ *holds:*

- ***Markov property:*** $\{\mathbf{P}_{s|T}\}_{s=T}^{0}$ *is also a Markov process:*

$$\mathbf{P}_{s|t} = \mathbf{P}_{t|s}^{-1} \cdot \left[\vec{p}_t \cdot \left(\frac{1}{\vec{p}_t}\right)^{\top}\right] \odot \mathbf{P}_{t|s}^{\top}, \quad \forall\, 0 \le s \le t \le T. \tag{16}$$

- ***Kolmogorov backward equation:*** *Preference growing* $\{\mathbf{P}_{s|T}\}_{s=T}^{0}$ *satisfies:*

$$\frac{\partial \mathbf{P}_{s|t}}{\partial t} = \mathbf{R}_s \mathbf{P}_{s|t}, \quad \forall\, 0 \le s \le t \le T. \tag{17}$$

$$\mathbf{R}_s := \lim_{t \to s} \frac{\partial \mathbf{P}_{s|t}}{\partial s} = \mathbf{Q}_s^{\top} \odot \left[\vec{p}_s \cdot \left(\frac{1}{\vec{p}_s}\right)^{\top}\right] - \mathbf{Q}_s \cdot \left[\vec{p}_s \cdot \left(\frac{1}{\vec{p}_s}\right)^{\top}\right] \odot \mathbf{I}. \tag{18}$$

The matrix $\vec{p}_t \cdot \left(\frac{1}{\vec{p}_t}\right)^{\top}$ *denotes exponential preference ratios with entries given by* $\frac{p_t(y|u)}{p_t(x_t|u)} \forall x_t, y \in \mathcal{X}$.

At this point, we are equipped with the approximated preference ratios $\mathbf{s}_\Theta$ to grow user preferences. Given a user condition u, we first sample an item x_T from the non-preference state $\vec{p}_T$, and then compute the reverse transition matrix $\mathbf{P}_{s|t}$ to progressively grow user preference over time as:

$$p_{s|t}(x_s = y|x_t) = p_{t|s}(x_t|x_s = y) \cdot \sum_{z \in \mathcal{X}} p_{t|s}^{-1}(x_t = y|x_s = z) \cdot e^{\mathbf{s}_\Theta(x_t,t,u)_z}, \forall\, 0 \le s \le t \le T. \tag{19}$$

As illustrated in Figure 2, we iteratively grow user preferences backward until reaching x_0. Finally, based on the preference scores at timestep 0, we recommend the top-K items with the highest scores.

3.4 Modeling Non-preference User for Personalization

3.4.1 Non-preference User Modeling

To enhance the personalization of preference ratios, we introduce the modeling of non-preference users—*i.e.,* cold-start users with no interaction history—during training. This allows the model to perform personalized contrast against the non-preference user during inference, thereby reinforcing user-specific signals for personalized recommendations. As illustrated in Figure 2, we randomly drop the user condition u with a fixed probability p during training, and replace it with a learnable embedding that represents the non-preference user ϕ.

3.4.2 Personalized Contrast against Non-preference User

The non-preference user ϕ is defined such that $p_{t|0}(x_t|x_0, \phi) = p_{t|0}(x_t|x_0)$. According to Bayes' theorem $p_{t|0}(x_t|x_0, u) = \frac{p_{t|0}(x_t|x_0,\phi)}{p(u|x_0)} \cdot p(u|x_0, x_t)$, there holds the following ratio condition:

$$\frac{p_{t|0}(x_t = y|x_0, u)}{p_{t|0}(x_t = x|x_0, u)} = \frac{p_{t|0}(x_t = y|x_0)}{p_{t|0}(x_t = x|x_0)} \cdot \frac{p(u|x_0, x_t = y)}{p(u|x_0, x_t = x)}. \tag{20}$$

We then estimate the ratio of the likelihood with a personalization strength parameter w as [59]:

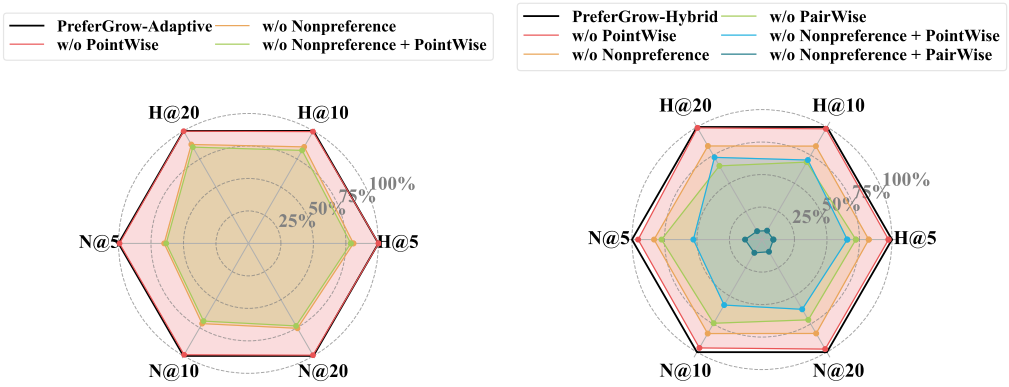
$$\frac{p(u|x_0, x_t = y)}{p(u|x_0, x_t = x)} \approx \left\{\frac{p_{t|0}(x_t = y|x_0, \phi)}{p_{t|0}(x_t = x|x_0, \phi)} \cdot \left[\frac{p_{t|0}(x_t = y|x_0, u)}{p_{t|0}(x_t = x|x_0, u)}\right]^{-1}\right\}^{w}. \tag{21}$$

Therefore, the personalization-enhanced preference ratios with strength w are computed as follows:

$$\hat{s}_\Theta(x_t, t, u)_y = w \cdot s_\Theta(x_t, t, u)_y + (1 - w) \cdot s_\Theta(x_t, t, \phi)_y. \tag{22}$$

Table 1: Performance comparison across different datasets and baselines.

Dataset		MoviesLens		Steam		Beauty		Toys		Sports	
Method		HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG
SASRec	@5	0.0905	0.0502	0.0321	0.0192	0.0326	0.0221	0.0340	0.0258	0.0154	0.0101
	@10	0.1709	0.0760	0.0572	0.0272	0.0438	0.0257	0.0438	0.0289	0.0230	0.0126
	@20	0.2899	0.1059	0.0957	0.0368	0.0595	0.0297	0.0499	0.0305	0.0298	0.0143
Caser	@5	0.0925	0.0576	0.0258	0.0163	0.0170	0.0106	0.0093	0.0072	0.0051	0.0030
	@10	0.1605	0.0794	0.0446	0.0223	0.0219	0.0122	0.0129	0.0083	0.0093	0.0043
	@20	0.2592	0.1042	0.0736	0.0295	0.0295	0.0141	0.0185	0.0098	0.0149	0.0057
GRURec	@5	0.0892	0.0551	0.0255	0.0156	0.0130	0.0082	0.0124	0.0078	0.0070	0.0048
	@10	0.1534	0.0757	0.0441	0.0216	0.0188	0.0101	0.0180	0.0092	0.0107	0.0059
	@20	0.2501	0.1000	0.0769	0.0298	0.0313	0.0132	0.0304	0.0117	0.0171	0.0076
DreamRec	@5	0.0676	0.0437	0.0109	0.0073	0.0300	0.0247	0.0381	0.0304	0.0095	0.0084
	@10	0.1083	0.0568	0.0157	0.0088	0.0353	0.0265	0.0412	0.0314	0.0112	0.0089
	@20	0.1610	0.0701	0.0218	0.0104	0.0402	0.0277	0.0463	0.0327	0.0149	0.0098
PreferDiff	@5	0.0538	0.0349	0.0167	0.0105	0.0335	0.0272	0.0386	0.0308	0.0168	0.0130
	@10	0.0852	0.0450	0.0297	0.0145	0.0434	0.0304	0.0494	0.0343	0.0211	0.0144
	@20	0.1270	0.0555	0.0526	0.0203	0.0577	0.0340	**0.0644**	0.0380	0.0256	0.0155
DiffRec	@5	0.0266	0.0121	0.0268	0.0143	0.0095	0.0055	0.0053	0.0028	0.0063	0.0035
	@10	0.0889	0.0320	0.0657	0.0268	0.0218	0.0094	0.0153	0.0059	0.0117	0.0052
	@20	0.1768	0.0541	0.1159	0.0394	0.0355	0.0128	0.0226	0.0078	0.0194	0.0072
DDSR	@5	0.0871	0.0533	0.0252	0.0157	0.0291	0.0217	0.0386	0.0302	0.0146	0.0109
	@10	0.1523	0.0742	0.0437	0.0216	0.0434	0.0262	0.0479	0.0332	0.0213	0.0130
	@20	0.2450	0.0975	0.0736	0.0291	0.0608	0.0306	0.0618	0.0367	0.0298	0.0151
PreferGrow	Impr.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Hybrid	@5	0.1409	0.0911	**0.0615**	**0.0406**	**0.0420**	**0.0322**	**0.0441**	**0.0331**	**0.0207**	**0.0142**
	@10	0.2229	0.1174	**0.0935**	**0.0508**	**0.0532**	**0.0358**	**0.0519**	**0.0356**	**0.0267**	**0.0162**
	@20	0.3355	0.1458	**0.1413**	**0.0628**	**0.0708**	**0.0402**	0.0625	**0.0383**	**0.0343**	**0.0181**
Adaptive	@5	**0.1413**	**0.0912**	0.0583	0.0375	0.0396	0.0310	0.0413	0.0304	0.0168	0.0127
	@10	**0.2240**	**0.1177**	0.0914	0.0481	0.0508	0.0347	0.0513	0.0337	0.0207	0.0140
	@20	**0.3362**	**0.1460**	0.1387	0.0600	0.0610	0.0373	0.0642	0.0370	0.0267	0.0155



(a) Abalation of Adaptive on Steam.

(b) Ablation of Hybrid on Steam.

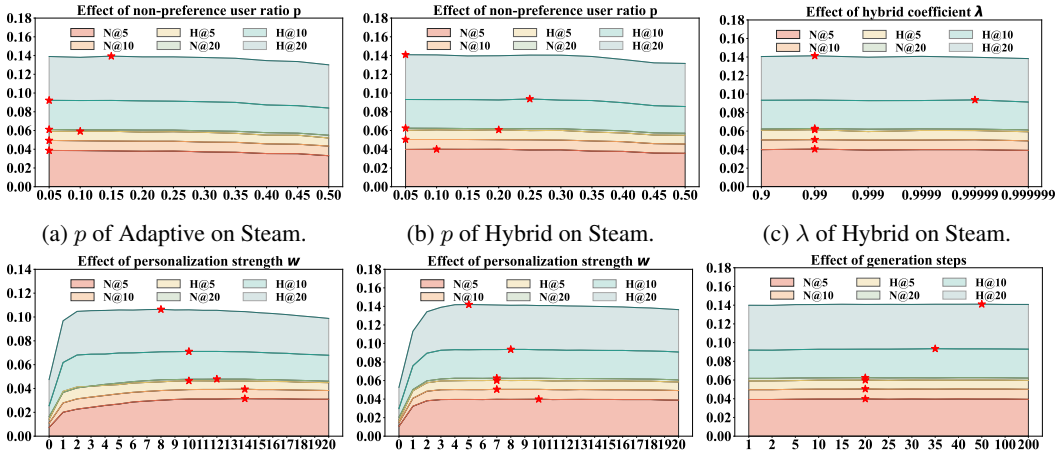
Figure 3: Ablation Study on Steam, showing the percentage of the relative effectiveness.

4 Experiment

We compare PreferGrow with a variety of baselines under the all-ranking evaluation protocol, including classical recommenders (SASRec [58], Caser [60], GRURec [61]), item-level diffusion-based recommenders (DreamRec [9], PreferDiff [27]), and preference score-level diffusion-based recommenders (DiffRec [28], DDSR [35]) across five benchmark datasets. Details of the datasets are provided in Appendix G.1, discussions of the baselines are presented in Appendix G.2, and the training and evaluation settings are described in Appendix G.3.

4.1 Overall Comparsion

From the results in Table 1, we observe that PreferGrow consistently outperforms all baselines. We attribute the effectiveness of PreferGrow to three factors: 1) its discrete diffusion process aligns with the discrete nature of the recommendation scenario; 2) the preference fading noise perturbation mechanism, akin to negative sampling; and 3) preference ratios modeling without the simplex



(a) p of Adaptive on Steam. (b) p of Hybrid on Steam. (c) λ of Hybrid on Steam.

(d) w of Adaptive on Steam. (e) w of Hybrid on Steam. (f) Steps of Hybrid on Steam.

Figure 4: Hyperparameter analysis of PreferGrow on Steam. Red stars denote the best results.

constraints, which better captures user preferences. Additionally, the hybrid and adaptive fading matrix settings, inspired by the negative sampling strategy, also demonstrate the flexibility.

4.2 Ablation Study

As presented in Figure 3, we perform a thorough analysis and evaluation of each key component within PreferGrow to assess their individual significance. The ablation study is conducted using the following three variations: (1) *w/o-PointWise*, PreferGrow without the general hard negative item x_{-1}; (2) *w/o-PairWise*, Point-Wise PreferGrow with $\vec{p}_T = \vec{e}_{-1}$; (3) *w/o-Nonpreference*, PreferGrow without modeling the non-preference user ϕ, resulting in no personalized enhancement during the backward generation, as well as the combination of the aforementioned ablations. Overall, PairWise preference ratios modeling and non-preference user modeling are crucial for the effectiveness of PreferGrow, while PairWise preference ratios modeling adds an additional refinement.

4.3 Hyper-parameter Analysis

We further investigate the effect of key hyperparameters on the performance of PreferGrow (Figure 4). Overall, PreferGrow is fairly robust to the non-preference user proportion p, the hybrid coefficient λ, and the number of sampling steps, while exhibiting relatively higher sensitivity to the personalization strength w. Importantly, w is chosen at inference time, so no retraining is required. On the one hand, w is locally stable within a reasonable range (see Figure 4); on the other hand, we show that the optimal w is highly consistent across data splits (Appendix G.4). Taken together, these properties ensure that tuning w is practical and does not introduce significant overhead.

5 Limitations

While effective, PreferGrow still has several aspects that warrant further optimization.

- **Higher modeling complexity.** As shown in Table 8, targeting *preference ratios* incurs substantially higher complexity than prior diffusion-based recommenders. On the one hand, preference ratios are more expressive and thus raise the potential ceiling of the model; on the other hand, they also increase the learning difficulty under finite model capacity—for example, leading to longer training time than previous diffusion-based methods (Appendix G.5). A principled remedy is to extend PreferGrow to a rank-r fading matrix. By inducing a quantization structure, this formulation can markedly reduce the complexity of preference ratios while preserving their expressive power, thus achieving a better expressiveness–redundancy trade-off. We identify this as a primary avenue for future work. In addition, we observe that the final 50% of training yields only about a 5% gain in NDCG@5, suggesting substantial redundancy in the training process. A likely cause is uniform timestep sampling: different timesteps reflect different degrees of user preference fading and thus vary in the difficulty of modeling preference ratios. Incorporating this difficulty — by

beginning with easier timesteps (mild fading) and gradually advancing to harder ones — may further accelerate convergence.

- **High computational complexity.** As shown in Table 8, PreferGrow has $\mathcal{O}(N)$ complexity for both loss computation and inference, where N represents the size of the item corpus. While this is comparable to prior diffusion-based recommenders, it becomes impractical when scaling to extremely large item sets (e.g., billions of items), where even $\mathcal{O}(N)$ is prohibitive. A promising next step is to incorporate a quantization structure such as *Semantic IDs (SIDs)* to reduce the cost from $\mathcal{O}(N)$ to $\mathcal{O}(mc)$. SIDs represent each item with m codebooks of size c, enabling up to c^m items while maintaining only $\mathcal{O}(mc)$ computation. Extending PREFERGROW with rank-r fading matrix to operate directly on SIDs is our future work.

6 Conclusion

Building upon the previous diffusion-based recommenders and discrete diffusion models discussed in Appendix A, this paper introduce a new discrete diffusion-based recommender tailored for discrete and sparse recommendation scenarios, named PreferGrow. In summary, PreferGrow distinguishes itself from existing diffusion-based recommenders in three key aspects: (1) **Discrete Diffusion:** It operates directly on the discrete item set, fully aligning with the discrete nature of recommendation. (2) **Preference Fading:** It fades user preferences by replacing the preferred item with others, akin to negative sampling, thus removing the need for prior noise assumptions. (3) **Preference Ratios:** It estimates preference ratios by modeling the logarithmic ratios of user–item interaction probabilities, circumventing the constraints of the probability simplex. PreferGrow features a well-defined theoretical formulation and demonstrates superior performance in experiments. We further discuss the broader impacts of PreferGrow in Appendix B.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (62572449).

References

[1] Nouhaila Idrissi and Ahmed Zellou. A systematic literature review of sparsity issues in recommender systems. *Social Network Analysis and Mining*, 2020.

[2] Monika Singh. Scalability and sparsity issues in recommender datasets: a survey. *Knowledge and Information Systems*, 2020.

[3] Jinming Cui, Chaochao Chen, Lingjuan Lyu, Carl Yang, and Wang Li. Exploiting data sparsity in secure cross-platform social recommendation. *NeurIPS*, 2021.

[4] Hongzhi Yin, Qinyong Wang, Kai Zheng, Zhixu Li, and Xiaofang Zhou. Overcoming data sparsity in group recommendation. *TKDE*, 2020.

[5] Wenyu Mao, Zhengyi Yang, Jiancan Wu, Haozhe Liu, Yancheng Yuan, Xiang Wang, and Xiangnan He. Addressing missing data issue for diffusion-based recommendation. In *SIGIR*, 2025.

[6] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *NeurIPS*, 2019.

[7] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[9] Zhengyi Yang, Jiancan Wu, Zhicai Wang, Xiang Wang, Yancheng Yuan, and Xiangnan He. Generate what you prefer: Reshaping sequential recommendation via guided diffusion. In *NeurIPS*, 2023.

[10] Zihao Li, Aixin Sun, and Chenliang Li. Diffurec: A diffusion model for sequential recommendation. *TOIS*, 2023.

[11] Hanwen Du, Huanhuan Yuan, Zhen Huang, Pengpeng Zhao, and Xiaofang Zhou. Sequential recommendation with diffusion models. *arXiv*, 2023.

[12] Yu Wang, Zhiwei Liu, Liangwei Yang, and Philip S Yu. Conditional denoising diffusion for sequential recommendation. In *PAKDD*, 2024.

[13] Qidong Liu, Fan Yan, Xiangyu Zhao, Zhaocheng Du, Huifeng Guo, Ruiming Tang, and Feng Tian. Diffusion augmentation for sequential recommendation. In *CIKM*, 2023.

[14] Jujia Zhao, Wang Wenjie, Yiyan Xu, Teng Sun, Fuli Feng, and Tat-Seng Chua. Denoising diffusion recommender model. In *SIGIR*, 2024.

[15] Yuner Xuan. Diffusion cross-domain recommendation. *arXiv*, 2024.

[16] Zixuan Yi, Xi Wang, and Iadh Ounis. A directional diffusion graph transformer for recommendation. *arXiv*, 2024.

[17] Yu Hou, Jin-Duk Park, and Won-Yong Shin. Collaborative filtering based on diffusion models: Unveiling the potential of high-order connectivity. In *SIGIR*, 2024.

[18] Ziqiang Cui, Haolun Wu, Bowei He, Ji Cheng, and Chen Ma. Context matters: Enhancing sequential recommendation with context-aware diffusion-based contrastive learning. In *CIKM*, 2024.

[19] Federico Tomasi, Francesco Fabbri, Mounia Lalmas, and Zhenwen Dai. Diffusion model for slate recommendation. *arXiv*, 2024.

[20] Fan Huang and Wei Wang. Diffusion-augmented graph contrastive learning for collaborative filter. *arXiv*, 2025.

[21] Xiaodong Li, Hengzhu Tang, Jiawei Sheng, Xinghua Zhang, Li Gao, Suqi Cheng, Dawei Yin, and Tingwen Liu. Exploring preference-guided diffusion model for cross-domain recommendation. *arXiv*, 2025.

[22] Jin Li, Shoujin Wang, Qi Zhang, Shui Yu, and Fang Chen. Generating with fairness: A modality-diffused counterfactual framework for incomplete multimodal recommendations. *arXiv*, 2025.

[23] Wenyu Mao, Shuchang Liu, Haoyang Liu, Haozhe Liu, Xiang Li, and Lantao Hu. Distinguished quantized guidance for diffusion-based sequence recommendation. *WWW*, 2025.

[24] Chu Zhao, Enneng Yang, Yuliang Liang, Jianzhe Zhao, Guibing Guo, and Xingwei Wang. Distributionally robust graph out-of-distribution recommendation via diffusion model. *arXiv*, 2025.

[25] Sharare Zolghadr, Ole Winther, and Paul Jeha. Generative diffusion models for sequential recommendations. *arXiv*, 2024.

[26] Guoqing Hu, Zhengyi Yang, Zhibo Cai, An Zhang, and Xiang Wang. Generate and instantiate what you prefer: Text-guided diffusion for sequential recommendation. In *arXiv*, 2024.

[27] Shuo Liu, An Zhang, Guoqing Hu, Hong Qian, and Tat seng Chua. Preference diffusion for recommendation. In *ICLR*, 2025.

[28] Wenjie Wang, Yiyan Xu, Fuli Feng, Xinyu Lin, Xiangnan He, and Tat-Seng Chua. Diffusion recommender model. In *SIGIR*, 2023.

[29] Gabriel Bénédict, Olivier Jeunen, Samuele Papa, Samarth Bhargav, Daan Odijk, and Maarten de Rijke. Recfusion: A binomial diffusion process for 1d data for recommendation. *arXiv*, 2023.

[30] Penghang Yu, Zhiyi Tan, Guanming Lu, and Bing-Kun Bao. Ld4mrec: Simplifying and powering diffusion model for multimedia recommendation. *arXiv*, 2023.

[31] Haokai Ma, Ruobing Xie, Lei Meng, Xin Chen, Xu Zhang, Leyu Lin, and Zhanhui Kang. Plug-in diffusion model for sequential recommendation. In *AAAI*, 2024.

[32] Utkarsh Priyam, Hemit Shah, and Edoardo Botta. Edge-rec: Efficient and data-guided edge diffusion for recommender systems graphs. *arXiv*, 2024.

[33] Zhenhao Jiang and Jicong Fan. Diffairec: Generative fair recommender with conditional diffusion model. *arXiv*, 2024.

[34] Gwangseok Han, Wonbin Kweon, Minsoo Kim, and Hwanjo Yu. Controlling diversity at inference: Guiding diffusion recommender models with targeted category preferences. *arXiv preprint arXiv:2411.11240*, 2024.

[35] Wenjia Xie, Hao Wang, Luankang Zhang, Rui Zhou, Defu Lian, and Enhong Chen. Breaking determinism: Fuzzy modeling of sequential recommendation using discrete state space diffusion model. *NeurIPS*, 2024.

[36] Yangqin Jiang, Lianghao Xia, Wei Wei, Da Luo, Kangyi Lin, and Chao Huang. Diffmm: Multi-modal diffusion model for recommendation. In *MM*, 2024.

[37] Rui Xia, Yanhua Cheng, Yongxiang Tang, Xiaocheng Liu, Xialong Liu, Lisong Wang, and Peng Jiang. S-diff: An anisotropic diffusion model for collaborative filtering in spectral domain. In *WSDM*, 2025.

[38] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *ICML*, 2024.

[39] Chenlin Meng, Kristy Choi, Jiaming Song, and Stefano Ermon. Concrete score matching: Generalized score matching for discrete data. *NeurIPS*, 2022.

[40] Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. *ICLR*, 2025.

[41] Calvin Luo. Understanding diffusion models: A unified perspective. *arXiv*, 2022.

[42] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv*, 2022.

[43] Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. RLHF workflow: From reward modeling to online RLHF. *TMLR*, 2024.

[44] Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. Iterative preference learning from human feedback: Bridging theory and practice for RLHF under kl-constraint. In *ICML*, 2024.

[45] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, 2023.

[46] Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. β-dpo: Direct preference optimization with dynamic β. In *NeurIPS*, 2024.

[47] Yuxin Chen, Junfei Tan, An Zhang, Zhengyi Yang, Leheng Sheng, Enzhi Zhang, Xiang Wang, and Tat-Seng Chua. On softmax direct preference optimization for recommendation. In *NeurIPS*, 2024.

[48] Junkang Wu, Xue Wang, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. α-dpo: Adaptive reward margin is what direct preference optimization needs. *ICML*, 2025.

[49] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 1952.

[50] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *NeurIPS*, 2021.

[51] Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. In *NeurIPS*, 2022.

[52] Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *NeurIPS*, 2024.

[53] Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling. *ICLR*, 2025.

[54] Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. *NeurIPS*, 2022.

[55] Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. Score-based continuous-time discrete diffusion models. *ICLR*, 2023.

[56] 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 *CIKM*, 2019.

[57] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, 2018.

[58] Wang-Cheng Kang and Julian J. McAuley. Self-attentive sequential recommendation. In *ICDM*, 2018.

[59] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv*, 2022.

[60] Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *WSDM*, pages 565–573. ACM, 2018.

[61] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In *ICLR (Poster)*, 2016.

[62] Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. Text is all you need: Learning language representations for sequential recommendation. In *KDD*, pages 1258–1267, 2023.

[63] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *UAI*, 2009.

[64] Wuchao Li, Rui Huang, Haijun Zhao, Chi Liu, Kai Zheng, Qi Liu, Na Mou, Guorui Zhou, Defu Lian, Yang Song, et al. Dimerec: A unified framework for enhanced sequential recommendation via generative diffusion models. In *WSDM*, 2025.

[65] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *NeurIPS*, 2021.

[66] Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: enabling multimodal flows with applications to protein co-design. In *ICML*, 2024.

[67] F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 2016.

[68] Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. Image-based recommendations on styles and substitutes. In *SIGIR*, 2015.

[69] Yitong Ji, Aixin Sun, Jie Zhang, and Chenliang Li. A critical study on data leakage in recommender system offline evaluation. *TOIS*, 2023.

[70] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 2024.

[71] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*, 2020.

[72] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. Self-supervised graph learning for recommendation. *SIGIR*, 2021.

[73] Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. Representation learning with large language models for recommendation. In *WWW*, 2024.

[74] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. Recommender systems with generative retrieval. *NeurIPS*, 2023.

[75] Yupeng Hou, Jianmo Ni, Zhankui He, Noveen Sachdeva, Wang-Cheng Kang, Ed H Chi, Julian McAuley, and Derek Zhiyuan Cheng. Actionpiece: Contextually tokenizing action sequences for generative recommendation. In *ICML*, 2025.

[76] Guorui Zhou, Jiaxin Deng, Jinghao Zhang, Kuo Cai, Lejian Ren, Qiang Luo, Qianqian Wang, Qigen Hu, Rui Huang, Shiyao Wang, et al. Onerec technical report. *arXiv*, 2025.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We accurately reflect that this paper introduces a new discrete diffusion-based recommender growing preference ratios via preference fading.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our work in Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide the full set of assumptions and a complete (and correct) proof in Appendix C.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We fully disclose all the information needed to reproduce the main experimental results in Appendix G.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide open access to our data and code.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify all the training and test details in Appendix G.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: To ensure determinism and due to computational resource constraints, we fix all random seeds to a random number.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide sufficient information in Appendix G.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We fully respect the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss both potential positive societal impacts and negative societal impacts in Appendix B.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All assets used in this paper are properly credited and respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: New assets introduced in this paper are well documented.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: We do not user LLMs as the core method.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Related Work

Table 2: Comparison of Diffusion-based Recommenders

	Modeling Target	*Forward Perturbation*	*Negative Sampling*	*Representative Work*
Continuous	Score Function	Gaussian on Embedding	✗	DreamRec [9]
	Score Function	Gaussian on Embedding	✓	PreferDiff [27]
	Preference Scores	Gaussian on One-hot	✓	DiffRec [28]
Discrete	Preference Scores	Bernoulli on One-hot	✓	RecFusion [62]
	Preference Scores	Categorical on One-hot	✓	DDSR [35])
	Preference Ratios	Fading on Items	✓	**PreferGrow** (Ours)

Diffusion-based recommenders [9–37] utilize forward perturbation in diffusion models to address data sparsity [6], thereby better adapting to sparse recommendation scenarios. They typically consist of three core components: the modeling objective, the forward noise addition, and the corresponding backward generation process, which are summarized in Table 2. Current research on diffusion-based recommenders can be classified into two primary approaches: one involves adding noise to dense item embeddings at the item level, and the other focuses on perturbing the one-hot interaction probability vector for all items, which is referred to as preference scores. The first research line, pioneered by DreamRec [9] and DiffuRec [10], encodes user-preferred items as dense embeddings and adds Gaussian noise to these item embeddings. DreamRec [9] models the score functions (the gradient of the log-likelihood of the perturbed distribution) without negative sampling, whereas DiffuRec [10] incorporates a recommendation loss that includes negative sampling. Building on them, PreferDiff [27] introduced an optimization objective derived from BPR loss [63], which integrates multiple negative samples into the generative modeling framework. However, the application of continuous Gaussian noise to positive preferred items, in contrast to the discrete nature of negative samples, creates an inherent mismatch, making it difficult to optimize both simultaneously during training, leading to a trade-off [10, 64, 27]. [25] Additionally, other works have introduced more sophisticated module designs [11, 12, 14, 16–18, 20, 23, 25, 26] or applied them to different recommendation tasks [13–22, 24]. For instance, DimeRec [64] incorporates multi-interest models, DiQDiff [23] introduces semantic vector quantization, and DiffuASR [13] applies item-level diffusion to sequential recommendation data augmentation. The second research line [28–37] involves converting user preference data into one-hot vectors, which are then mapped to preference scores within the probability simplex. DiffRec [28] add continuous Gaussian noise to the preference scores, and then learn to recover the clean preference scores from the perturbed ones. Consecutively, LD4MRec [30] refines the design for efficient multimedia recommendation, and D3Rec [34] introduces targeted category preferences to control diversity during inference. However, the constraints of the probability simplex—non-negativity and normalization—pose significant challenges in accurately estimating the preference scores [38]. To address these challenges while considering the probability simplex constraints, RecFusion [62] assumes a Bernoulli noise prior, completing the binomial diffusion process and subsequently modeling the parameters of the reverse binomial distribution to facilitate the reverse generation of preference scores. On the other hand, DDSR [35] adopts a categorical noise prior, as proposed in [65], and directly recovers clean preference scores from the perturbed ones. Nonetheless, the constraints of the probability simplex may limit the effectiveness of preference score modeling. Moreover, both diffusion-based recommenders rely on prior noise assumptions, such as Gaussian [9] or Bernoulli noise [29], which may not be optimal for recommendation scenarios where user preference data is inherently discrete.

Discrete Diffusion Models [65, 54, 39, 55, 38, 40, 66] have made substantial advances recently. Initially, D3PM [65] proposed a discrete diffusion framework based on an arbitrary probability transition matrix, trained with the evidence lower bound of the log-likelihood. Subsequently, LDR [54] extended this framework to a continuous-time setting using the Kolmogorov forward and backward equations. However, modeling score functions in such models presents challenges, as the gradient of the data distribution is undefined. To address this, CSM [39] introduced Concrete Score—discretization of score functions and the ratios of data distributions—as modeling objectives

for discrete diffusion models. Building upon these advances, SEDD [38] further bridges discrete diffusion models and ConcreteScore by introducing the score entropy loss. Expanding on these developments, we present PreferGrow, a matrix-based discrete diffusion framework which perturbing data by retaining or replacing items within a discrete corpus. The idempotent property of the replacement matrix (or fading matrix) is central to PreferGrow, and we demonstrate that it satisfies the Kolmogorov forward and backward equations as LDR [54], aligning it with prior works. Additionally, we introduce a design paradigm for the idempotent replacement matrix, which unifies previous approaches, including absorbing and uniform settings.

B Broader Impacts

Theoretically, PreferGrow introduces a well-defined discrete diffusion model building upon prior work. While designed for recommendation, PreferGrow is also applicable to other discrete domains, such as molecular design in chemistry and protein structure prediction. There are many potential societal consequences of our work, none of which we believe warrant specific attention at this time.

C Proofs of Main Results

Proof of Theorem 1: $\alpha_t : [0, T] \to [0, 1]$ is a strictly decreasing function with $\alpha_0 = 1$ and $\alpha_T = 0$ and the preference fading discrete diffusion process is denoted as $\mathbf{P}_{t|0} = \alpha_t \mathbf{I} + (1-\alpha_t)\mathbf{E}, \forall t \in [0, T]$. Then we can rewrite preference fading discrete diffusion process as for $\alpha_0 = 1$:

$$\mathbf{P}_{t|0} = \frac{\alpha_t}{\alpha_0}\mathbf{I} + (1 - \frac{\alpha_t}{\alpha_0})\mathbf{E}, \forall t \in [0, T]. \tag{23}$$

For any well-defined $\mathbf{P}_{t|s,0}$ with $0 < s < t$, the following limiting distribution constraint must hold:

$$p_{t|0}(x_t|x_0) = \sum_{x_s \in \mathcal{X}} p_{t|s,0}(x_t|x_s, x_0) p_{s|0}(x_s|x_0), \forall x_t, x_0 \in \mathcal{X}. \tag{24}$$

In matrix form, this constraint reduces to $\mathbf{P}_{t|0} = \mathbf{P}_{t|s,0}\,\mathbf{P}_{s|0}$ for all $0 < s < t$. Note that one possible particular solution of this equation is:

$$\mathbf{P}_{t|s,0} := \mathbf{P}_{t|s} = \frac{\alpha_t}{\alpha_s}\mathbf{I} + (1 - \frac{\alpha_t}{\alpha_s})\mathbf{E}, \forall 0 \le s \le t \le T. \tag{25}$$

We will show that this indeed satisfies the constraint under the condition that $\mathbf{E}$ is idempotent.

$$\begin{aligned}
\mathbf{P}_{t|s}\mathbf{P}_{s|0} &= \left[\frac{\alpha_t}{\alpha_s}\mathbf{I} + (1 - \frac{\alpha_t}{\alpha_s})\mathbf{E}\right] \cdot \left[\frac{\alpha_s}{\alpha_0}\mathbf{I} + (1 - \frac{\alpha_s}{\alpha_0})\mathbf{E}\right] \\
&= \frac{\alpha_t}{\alpha_0}\mathbf{I} + \left(\frac{\alpha_t}{\alpha_s} - 2\frac{\alpha_t}{\alpha_0} + \frac{\alpha_s}{\alpha_0}\right)\mathbf{E} + \left(1 + \frac{\alpha_t}{\alpha_0} - \frac{\alpha_s}{\alpha_0} - \frac{\alpha_t}{\alpha_s}\right)\mathbf{E}^2 \\
&= \frac{\alpha_t}{\alpha_0}\mathbf{I} + \mathbf{E}^2 - \frac{\alpha_t}{\alpha_0}\mathbf{E} + \left(\frac{\alpha_t}{\alpha_s} - \frac{\alpha_t}{\alpha_0} + \frac{\alpha_s}{\alpha_0}\right)(\mathbf{E} - \mathbf{E}^2) \\
&\quad \text{the fading matrix is idempotent} \Rightarrow \mathbf{E}^2 = \mathbf{E} \\
&= \frac{\alpha_t}{\alpha_0}\mathbf{I} + (1 - \frac{\alpha_t}{\alpha_0})\mathbf{E} \\
&= \mathbf{P}_{t|0}.
\end{aligned} \tag{26}$$

Similarly, for all $0 \le r \le s \le t \le T$, there holds the Chapman-Kolmogorov equation:

$$\begin{aligned}
\mathbf{P}_{t|s}\mathbf{P}_{s|r} &= \left[\frac{\alpha_t}{\alpha_s}\mathbf{I} + (1 - \frac{\alpha_t}{\alpha_s})\mathbf{E}\right] \cdot \left[\frac{\alpha_s}{\alpha_r}\mathbf{I} + (1 - \frac{\alpha_s}{\alpha_r})\mathbf{E}\right] \\
&= \frac{\alpha_t}{\alpha_r}\mathbf{I} + \mathbf{E}^2 - \frac{\alpha_t}{\alpha_r}\mathbf{E} + \left(\frac{\alpha_t}{\alpha_s} - \frac{\alpha_t}{\alpha_r} + \frac{\alpha_s}{\alpha_r}\right)(\mathbf{E} - \mathbf{E}^2) \\
&\quad \text{the fading matrix is idempotent} \Rightarrow \mathbf{E}^2 = \mathbf{E} \\
&= \frac{\alpha_t}{\alpha_r}\mathbf{I} + (1 - \frac{\alpha_t}{\alpha_r})\mathbf{E} \\
&= \mathbf{P}_{t|r}.
\end{aligned} \tag{27}$$

We further note that $\mathbf{P}_{t|s}, \forall 0 \leq s \leq t \leq T$ is invertible, with inverse given by:

$$\mathbf{P}_{t|s}^{-1} = \frac{\alpha_s}{\alpha_t}\mathbf{I} + (1 - \frac{\alpha_s}{\alpha_t})\mathbf{E}, \forall 0 \leq s \leq t \leq T. \tag{28}$$

$$\begin{aligned}
\mathbf{P}_{t|s}^{-1}\mathbf{P}_{t|s} &= \left[\frac{\alpha_s}{\alpha_t}\mathbf{I} + (1 - \frac{\alpha_s}{\alpha_t})\mathbf{E}\right] \cdot \left[\frac{\alpha_t}{\alpha_s}\mathbf{I} + (1 - \frac{\alpha_t}{\alpha_s})\mathbf{E}\right] \\
&= \mathbf{I} + \left(\frac{\alpha_t}{\alpha_s} + \frac{\alpha_s}{\alpha_t} - 2\right) \cdot (\mathbf{E} - \mathbf{E}^2) \\
&\quad \text{the fading matrix is idempotent} \Rightarrow \mathbf{E}^2 = \mathbf{E} \\
&= \mathbf{I}.
\end{aligned} \tag{29}$$

$\forall 0 \leq r \leq s \leq t \leq T$, combined with limiting distribution constraint $\mathbf{P}_{t|r} = \mathbf{P}_{t|s,r}\mathbf{P}_{s|r}$, there are:

$$\begin{aligned}
\mathbf{P}_{t|s,r} &= \mathbf{P}_{t|r}\mathbf{P}_{s|r}^{-1} \\
&= \left[\frac{\alpha_t}{\alpha_r}\mathbf{I} + (1 - \frac{\alpha_t}{\alpha_r})\mathbf{E}\right] \cdot \left[\frac{\alpha_r}{\alpha_s}\mathbf{I} + (1 - \frac{\alpha_r}{\alpha_s})\mathbf{E}\right] \\
&= \frac{\alpha_t}{\alpha_s}\mathbf{I} + \mathbf{E}^2 - \frac{\alpha_t}{\alpha_s}\mathbf{E} + \left(\frac{\alpha_t}{\alpha_r} - \frac{\alpha_t}{\alpha_s} + \frac{\alpha_r}{\alpha_s}\right)(\mathbf{E} - \mathbf{E}^2) \\
&\quad \text{the fading matrix is idempotent} \Rightarrow \mathbf{E}^2 = \mathbf{E} \\
&= \frac{\alpha_t}{\alpha_s}\mathbf{I} + (1 - \frac{\alpha_t}{\alpha_s})\mathbf{E} \\
&= \mathbf{P}_{t|s}.
\end{aligned} \tag{30}$$

$\mathbf{P}_{t|s,r} = \mathbf{P}_{t|s}$ indicates $p_{t|s,r}(x_t|x_s, x_r) = p_{t|s}(x_t|x_s), \forall 0 \leq r \leq s \leq t \leq T$. In summary, the preference fading discrete diffusion process is Markovian but not time-homogeneous, satisfies the Chapman–Kolmogorov equation, and is reversible. □

Proof of Proposition 1: $\alpha_t : [0, T] \to [0, 1]$ is a strictly decreasing function with $\alpha_0 = 1$ and $\alpha_T = 0$. α_t is further defined as $e^{-\int_0^t \beta(\tau)\mathrm{d}\tau}$ with $\beta(\tau) > 0$. The preference fading discrete Markov diffusion process is denoted as $\mathbf{P}_{t|s} = \frac{\alpha_t}{\alpha_s}\mathbf{I} + (1 - \frac{\alpha_t}{\alpha_s})\mathbf{E}, \forall 0 \leq s \leq t \leq T$. We first show that the preference fading discrete Markov diffusion process converges to a unified non-preference state $\vec{p}_T$ is well-defined. The transition rate matrix $\mathbf{Q}_t$ is computed as follows:

$$\begin{aligned}
\mathbf{Q}_t &= \lim_{s \to t} \frac{\partial \mathbf{P}_{t|s}}{\partial t} \\
&= \lim_{s \to t} \frac{\partial}{\partial t}\left(\frac{\alpha_t}{\alpha_s}\mathbf{I} + (1 - \frac{\alpha_t}{\alpha_s})\mathbf{E}\right) \\
&= \lim_{s \to t} \frac{\partial}{\partial \alpha_t}\left(\frac{\alpha_t}{\alpha_s}\mathbf{I} + (1 - \frac{\alpha_t}{\alpha_s})\mathbf{E}\right) \cdot \frac{\partial \alpha_t}{\partial t} \\
&= \lim_{s \to t}\left(\frac{1}{\alpha_s}\mathbf{I} - \frac{1}{\alpha_s}\mathbf{E}\right) \cdot (-\beta(t)) \cdot \alpha_t \\
&= \beta(t) \cdot (\mathbf{E} - \mathbf{I}).
\end{aligned} \tag{31}$$

The rate matrix $\mathbf{Q}_t$ characterizes the velocity of probability transitions at time t, encompassing both the direction and rate of transition. As shown in Equation (32), $\mathbf{Q}_t$ at any time $t \in [0, T]$ shares a consistent transition direction $\mathbf{I} - \mathbf{E}$, while the transition rate $\beta(t)$ varies over time. This time-dependent rate results in a non-homogeneous preference fading process. However, the shared transition direction ensures that all diffusion paths converge to the same non-preference state, making the process well-defined. Specifically, the stationary distribution $\vec{\pi}_t$ at time t satisfies the equilibrium condition $\mathbf{Q}_t\vec{\pi}_t = \vec{0}$. Since different $\mathbf{Q}_t$ matrices differ only by a scalar factor $\beta(t)$, they yield the same stationary solution $\vec{\pi}$. Consequently, the Markov process converges to a common steady-state distribution $\vec{\pi}$, *i.e.,* the non-preference state $\vec{p}_T$:

$$\mathbf{Q}_t\vec{\pi} = \vec{0} \Rightarrow (\mathbf{I} - \mathbf{E})\,\vec{p}_T = \vec{0}. \tag{32}$$

Given that $(\mathbf{I}-\mathbf{E})\mathbf{E}=\mathbf{0}$, each column of $\mathbf{E}$ satisfies the non-preference state equation $(\mathbf{I}-\mathbf{E})\,\vec{p}_T = \vec{0}$. To ensure a unique solution $\vec{p}_T$, we therefore assume that all columns of $\mathbf{E}$ are identical, i.e. $\mathbf{E} \propto \vec{p}_T \cdot \vec{1}^\top$. Considering the idempotence constraint $\mathbf{E}^2 = \mathbf{E}$, we then have:

$$\mathbf{E} = \frac{\vec{p}_T \cdot \vec{1}^\top}{\vec{1}^\top \vec{p}_T}. \tag{33}$$

$$\begin{aligned}\mathbf{E}^2 &= \frac{\vec{p}_T \cdot \vec{1}^\top \cdot \vec{p}_T \cdot \vec{1}^\top}{(\vec{1}^\top \vec{p}_T)^2} \\ &= \frac{\vec{p}_T \cdot (\vec{1}^\top \vec{p}_T) \cdot \vec{1}^\top}{(\vec{1}^\top \vec{p}_T)^2} \\ &= \frac{\vec{p}_T \cdot \vec{1}^\top}{\vec{1}^\top \vec{p}_T} \\ &= \mathbf{E}.\end{aligned} \tag{34}$$

□

Proof of Proposition 2: We have $\mathbf{P}_{t|s} = \frac{\alpha_t}{\alpha_s}\mathbf{I} + (1-\frac{\alpha_t}{\alpha_s})\mathbf{E}$ and $\mathbf{Q}_t = \beta(t)\cdot(\mathbf{E}-\mathbf{I})$. Then we compute $\frac{\partial \mathbf{P}_{t|s}}{\partial t}$ as follows:

$$\begin{aligned}\frac{\partial \mathbf{P}_{t|s}}{\partial t} &= \frac{\partial}{\partial t}\left(\frac{\alpha_t}{\alpha_s}\mathbf{I} + (1-\frac{\alpha_t}{\alpha_s})\mathbf{E}\right) \\ &= \frac{\partial}{\partial \alpha_t}\left(\frac{\alpha_t}{\alpha_s}\mathbf{I} + (1-\frac{\alpha_t}{\alpha_s})\mathbf{E}\right)\cdot\frac{\partial \alpha_t}{\partial t} \\ &= \left(\frac{1}{\alpha_s}\mathbf{I} - \frac{1}{\alpha_s}\mathbf{E}\right)\cdot(-\beta(t))\cdot\alpha_t \\ &= \beta(t)\cdot\frac{\alpha_t}{\alpha_s}\cdot(\mathbf{E}-\mathbf{I}).\end{aligned} \tag{35}$$

There holds the Kolmogorov forward equation:

$$\begin{aligned}\mathbf{Q}_t\mathbf{P}_{t|s} &= \beta(t)\cdot(\mathbf{E}-\mathbf{I})\cdot\left[\frac{\alpha_t}{\alpha_s}\mathbf{I} + (1-\frac{\alpha_t}{\alpha_s})\mathbf{E}\right] \\ &= \beta(t)\cdot\left[-\frac{\alpha_t}{\alpha_s}\mathbf{I} + (2\frac{\alpha_t}{\alpha_s}-1)\mathbf{E} + (1-\frac{\alpha_t}{\alpha_s})\mathbf{E}^2\right] \\ &\quad \text{the fading matrix is idempotent} \Rightarrow \mathbf{E}^2 = \mathbf{E} \\ &= \beta(t)\cdot\frac{\alpha_t}{\alpha_s}\cdot(\mathbf{E}-\mathbf{I}) \\ &= \frac{\partial \mathbf{P}_{t|s}}{\partial t}.\end{aligned} \tag{36}$$

□

Proof of Proposition 3: We begin by considering the expression for the loss function $\mathcal{L}_{SE}$, which is given as:

$$\mathcal{L}_{SE}(x_0, x_t, y) = e^{s_\Theta(x_t,t,u)_y} - s_\Theta(x_t,t,u)_y \cdot e^{r_t(x_0,x_t,y)} + e^{r_t(x_0,x_t,y)}(r_t(x_0,x_t,y)-1). \tag{37}$$

Then, we compute the gradient of this loss with respect to s_Θ, which will help establish a link between this and the binary cross-entropy loss.

$$\begin{aligned}\nabla_{s_\Theta}\mathcal{L}_{SE} &= e^{s_\Theta(x_t,t,u)_y} - e^{r_t(x_0,x_t,y)} \\ &= e^{s_\Theta(x_t,t,u)_y}(1+e^{r_t(x_0,x_t,y)}) - (1+e^{s_\Theta(x_t,t,u)_y})e^{r_t(x_0,x_t,y)} \\ &= (1+e^{s_\Theta(x_t,t,u)_y})(1+e^{r_t(x_0,x_t,y)})[\sigma(s_\Theta(x_t,t,u)_y) - \sigma(r_t(x_0,x_t,y))].\end{aligned} \tag{38}$$

Next, we consider the soft binary cross-entropy loss $\mathcal{L}_{sBCE}$, which is defined as:

$$\mathcal{L}_{sBCE}(x_0, x_t, y) = -\pi_{y\succ x_t|x_0} \log \sigma(s_\Theta(x_t, t, u)_y) - (1-\pi_{y\succ x_t|x_0}) \log(1-\sigma(s_\Theta(x_t, t, u)_y)). \quad (39)$$

where the soft label $\pi_{y\succ x_t|x_0} = p(y \succ x_t \mid x_0) = \sigma(r_t(x_0, x_t, y))$ is the probability of the preference of y over x_t, and $\sigma(\cdot)$ is the sigmoid function.

Now, we compute the gradient of $\mathcal{L}_{sBCE}$ with respect to s_Θ:

$$\begin{aligned}\nabla_{s_\Theta}\mathcal{L}_{sBCE} &= -\pi_{y\succ x_t|x_0}(1-\sigma(s_\Theta(x_t,t,u)_y)) + (1-\pi_{y\succ x_t|x_0})\sigma(s_\Theta(x_t,t,u)_y) \quad (40)\\ &= \sigma(s_\Theta(x_t,t,u)_y) - \pi_{y\succ x_t|x_0}\\ &= \sigma(s_\Theta(x_t,t,u)_y) - \sigma(r_t(x_0,x_t,y)).\end{aligned}$$

We can now relate the gradients of both loss functions:

$$\nabla_{s_\Theta}\mathcal{L}_{SE} = (1+e^{s_\Theta(x_t,t,u)_y})(1+e^{r_t(x_0,x_t,y)})\nabla_{s_\Theta}\mathcal{L}_{sBCE}. \quad (41)$$

□

Proof of Theorem 2: The reverse preference growing process is denoted as $\Omega = \{\mathbf{P}_{s|T}\}_{s=T}^{0}$. A collection $\mathcal{F}$ of subsets of Ω is called a σ-algebra on Ω if it satisfies: 1) $\Omega \in \mathcal{F}$. 2) If $A \in \mathcal{F}$ then its complement $A^c = \Omega \setminus A$ also belongs to $\mathcal{F}$. 3) If $\{A_n\}_{n=1}^{\infty} \subseteq \mathcal{F}$, then the countable union $\bigcup_{n=1}^{\infty} A_n \in \mathcal{F}$. We first show that the preference growing process satisfies the Markov property. $\mathcal{F}_t$ is a σ-algebra of the preference growing process $\Omega_t = \{\mathbf{P}_{s|T}\}_{s=T}^{t}$. $\forall A \in \mathcal{F}_t$ and $0 \le s \le t \le T$:

$$\begin{aligned}p_{s|\ge t}(x_s|x_t, A) &= \frac{p_{s,t|>t}(x_s, x_t|A)}{p_{t|>t}(x_t|A)} \cdot \frac{p(A)}{p(A)} \quad (42)\\ &= \frac{p_{s,\ge t}(x_s, x_t, A)}{p_{\ge t}(x_t, A)}\\ &= \frac{p_{>t|t,s}(A|x_t, x_s)}{p_{>t|t}(A|x_t)} \cdot \frac{p_{t|s}(x_t|x_s)p_s(x_s)}{p_t(x_t)}\\ &\quad \text{the preference fading process is Markovian} \Rightarrow \frac{p_{>t|t,s}(A|x_t,x_s)}{p_{>t|t}(A|x_t)} = 1\\ &= \frac{p_s(x_s)}{p_t(x_t)} p_{t|s}(x_t|x_s)\\ &\quad \text{the Bayes' theorem} \Rightarrow \frac{p_s(x_s)}{p_t(x_t)} p_{t|s}(x_t|x_s) = p_{s|t}(x_s|x_t)\\ &= p_{s|t}(x_s|x_t).\end{aligned}$$

We rewrite the equation $p_{s|t}(x_s|x_t) = \frac{p_s(x_s)}{p_t(x_t)} \cdot p_{t|s}(x_t|x_s)$ in matrix form $\forall 0 \le s \le t \le T$:

$$\begin{aligned}p_{s|t}(x_s|x_t) &= \frac{p_s(x_s)}{p_t(x_t)} \cdot p_{t|s}(x_t|x_s). \quad (43)\\ &\quad \vec{p}_s \cdot \left(\frac{1}{\vec{p}_t}\right)^{\top} \text{ denotes the matrix with entries} \frac{p_s(x_s=y)}{p_t(x_t=x)}, \forall x, y \in \mathcal{X}\\ \mathbf{P}_{s|t} &= \left[\vec{p}_s \cdot \left(\frac{1}{\vec{p}_t}\right)^{\top}\right] \odot \mathbf{P}_{t|s}^{\top}\\ &\quad \vec{p}_t = \mathbf{P}_{t|s} \cdot \vec{p}_s \Rightarrow \vec{p}_s = \mathbf{P}_{s|t}^{-1} \cdot \vec{p}_t\\ &= \mathbf{P}_{t|s}^{-1} \cdot \left[\vec{p}_t \cdot \left(\frac{1}{\vec{p}_t}\right)^{\top}\right] \odot \mathbf{P}_{t|s}^{\top}.\end{aligned}$$

The reverse-time preference growing process progresses from $s = T$ to $s = 0$, and thus α_s is an increasing function, transitioning from $\alpha_T = 0$ to $\alpha_0 = 1$. We then compute the reverse transition rate matrix $\mathbf{R}_s$ as follows:

$$
\begin{aligned}
\mathbf{R}_s &= \lim_{t\to s} \frac{\partial \mathbf{P}_{s|t}}{\partial s} \\
&= \lim_{t\to s} \frac{\partial \mathbf{P}_{s|t}}{\partial \alpha_s} \cdot \frac{\partial \alpha_s}{\partial s} \\
&= \lim_{t\to s} \frac{\partial}{\partial \alpha_s} \left(\mathbf{P}_{t|s}^{-1} \cdot \left[\vec{p}_t \cdot \left(\frac{1}{\vec{p}_t} \right)^\top \right] \odot \mathbf{P}_{t|s}^\top \right) \cdot \frac{\partial \alpha_s}{\partial s} \\
&= \lim_{t\to s} \left(\frac{\partial \mathbf{P}_{t|s}^{-1}}{\partial \alpha_s} \cdot \left[\vec{p}_t \cdot \left(\frac{1}{\vec{p}_t} \right)^\top \right] \odot \mathbf{P}_{t|s}^\top + \mathbf{P}_{t|s}^{-1} \cdot \left[\vec{p}_t \cdot \left(\frac{1}{\vec{p}_t} \right)^\top \right] \odot \frac{\partial \mathbf{P}_{t|s}^\top}{\partial \alpha_s} \right) \cdot \frac{\partial \alpha_s}{\partial s} \\
&\quad \frac{\partial \mathbf{P}_{t|s}^{-1}}{\partial \alpha_s} = \frac{\partial}{\partial \alpha_s} \left(\frac{\alpha_s}{\alpha_t}\mathbf{I} + (1 - \frac{\alpha_s}{\alpha_t})\mathbf{E} \right) = \frac{1}{\alpha_t}(\mathbf{I} - \mathbf{E}) \\
&\quad \lim_{t\to s} \mathbf{P}_{t|s}^{-1} = \lim_{t\to s} \left(\frac{\alpha_s}{\alpha_t}\mathbf{I} + (1 - \frac{\alpha_s}{\alpha_t})\mathbf{E} \right) = \mathbf{I} \\
&\quad \frac{\partial \mathbf{P}_{t|s}^\top}{\partial \alpha_s} = \frac{\partial}{\partial \alpha_s} \left(\frac{\alpha_t}{\alpha_s}\mathbf{I} + (1 - \frac{\alpha_t}{\alpha_s})\mathbf{E}^\top \right) = -\frac{\alpha_t}{\alpha_s^2}(\mathbf{I} - \mathbf{E}^\top) \\
&\quad \lim_{t\to s} \mathbf{P}_{t|s}^\top = \lim_{t\to s} \left(\frac{\alpha_t}{\alpha_s}\mathbf{I} + (1 - \frac{\alpha_t}{\alpha_s})\mathbf{E}^\top \right) = \mathbf{I} \\
&= \lim_{t\to s} \left(\frac{1}{\alpha_t}(\mathbf{I} - \mathbf{E}) \cdot \left[\vec{p}_t \cdot \left(\frac{1}{\vec{p}_t} \right)^\top \right] \odot \mathbf{I} - \left[\vec{p}_t \cdot \left(\frac{1}{\vec{p}_t} \right)^\top \right] \odot \frac{\alpha_t}{\alpha_s^2}(\mathbf{I} - \mathbf{E}^\top) \right) \cdot \frac{\partial \alpha_s}{\partial s} \\
&\quad \mathbf{Q}_t = \beta(t) \cdot (\mathbf{E} - \mathbf{I}) \, , \mathbf{Q}_t^\top = \beta(t) \cdot \left(\mathbf{E}^\top - \mathbf{I} \right) \\
&\quad \text{Note that from time } t \text{ to time } s < t, \alpha_s \text{ is increasing. } \Rightarrow \frac{\partial \alpha_s}{\partial s} = \alpha_s \beta(s) > 0 \\
&= \beta(s) \cdot (\mathbf{E}^\top - \mathbf{I}) \odot \left[\vec{p}_s \cdot \left(\frac{1}{\vec{p}_s} \right)^\top \right] - \beta(s) \cdot (\mathbf{E} - \mathbf{I}) \cdot \left[\vec{p}_s \cdot \left(\frac{1}{\vec{p}_s} \right)^\top \right] \odot \mathbf{I} \\
&= \mathbf{Q}_s^\top \odot \left[\vec{p}_s \cdot \left(\frac{1}{\vec{p}_s} \right)^\top \right] - \mathbf{Q}_s \cdot \left[\vec{p}_t \cdot \left(\frac{1}{\vec{p}_t} \right)^\top \right] \odot \mathbf{I}.
\end{aligned}
\tag{44}
$$

Moreover, we compute $\frac{\partial \mathbf{P}_{t|s}}{\partial s}$ and $\frac{\partial \mathbf{P}_{s|t}}{\partial s}$ as follows:

$$
\begin{aligned}
\frac{\partial \mathbf{P}_{t|s}}{\partial s} &= \frac{\partial}{\partial \alpha_s} \left(\frac{\alpha_t}{\alpha_s}\mathbf{I} + (1 - \frac{\alpha_t}{\alpha_s})\mathbf{E} \right) \cdot \frac{\partial \alpha_s}{\partial s} \\
&\quad \text{Note that from time } t \text{ to time } s < t, \alpha_s \text{ is increasing. } \Rightarrow \frac{\partial \alpha_s}{\partial s} = \alpha_s \beta(s) > 0 \\
&= \left(-\frac{\alpha_t}{\alpha_s^2}\mathbf{I} + \frac{\alpha_t}{\alpha_s^2}\mathbf{E} \right) \cdot \alpha_s \beta(s) \\
&= \beta(s) \cdot \frac{\alpha_t}{\alpha_s} \cdot (\mathbf{E} - \mathbf{I}) \\
&\quad (\mathbf{E} - \mathbf{I})^2 = \mathbf{E}^2 - 2\mathbf{E} + \mathbf{I} = -(\mathbf{E} - \mathbf{I}) \, , (\mathbf{E} - \mathbf{I})\, \mathbf{E} = \mathbf{0} \\
&= \beta(s) \cdot \left[-\frac{\alpha_t}{\alpha_s} \cdot (\mathbf{E} - \mathbf{I})^2 \right] + \beta(s) \cdot (\mathbf{E} - \mathbf{I})\, \mathbf{E} \\
&= \left[\frac{\alpha_t}{\alpha_s}\mathbf{I} + (1 - \frac{\alpha_t}{\alpha_s})\mathbf{E} \right] \cdot [\beta(s) \cdot (\mathbf{E} - \mathbf{I})] \\
&= \mathbf{P}_{t|s}\mathbf{Q}_s.
\end{aligned}
\tag{45}
$$

$$
\begin{aligned}
\frac{\partial \mathbf{P}_{s|t}}{\partial s} &= \frac{\partial}{\partial s}\left(\mathbf{P}_{t|s}^{-1}\cdot\left[\vec{p}_t\cdot\left(\frac{1}{\vec{p}_t}\right)^\top\right]\odot\mathbf{P}_{t|s}^\top\right) \\
&= \frac{\partial}{\partial s}\left(\left[\vec{p}_s\cdot\left(\frac{1}{\vec{p}_t}\right)^\top\right]\odot\mathbf{P}_{t|s}^\top\right) \\
&= \left[\frac{\partial\vec{p}_s}{\partial s}\cdot\left(\frac{1}{\vec{p}_t}\right)^\top\right]\odot\mathbf{P}_{t|s}^\top+\left[\vec{p}_s\cdot\left(\frac{1}{\vec{p}_t}\right)^\top\right]\odot\frac{\partial\mathbf{P}_{t|s}^\top}{\partial s} \\
&\quad \text{The reverse time } s \text{ begins with } s=T.\Rightarrow \frac{\partial\mathbf{P}_{s|0}}{\partial s}=-\mathbf{Q}_s\cdot\mathbf{P}_{s|0} \\
&\quad \frac{\partial\vec{p}_s}{\partial s}=\frac{\partial\mathbf{P}_{s|0}\cdot\vec{p}_0}{\partial s}=-\mathbf{Q}_s\cdot\mathbf{P}_{s|0}\cdot\vec{p}_0=-\mathbf{Q}_s\cdot\vec{p}_s \\
&\quad \frac{\partial\mathbf{P}_{t|s}}{\partial s}=\mathbf{P}_{t|s}\cdot\mathbf{Q}_s\Rightarrow\frac{\partial\mathbf{P}_{t|s}^\top}{\partial s}=\left(\mathbf{P}_{t|s}\cdot\mathbf{Q}_s\right)^\top=\mathbf{Q}_s^\top\cdot\mathbf{P}_{t|s}^\top \\
&= \left[\vec{p}_s\cdot\left(\frac{1}{\vec{p}_t}\right)^\top\right]\odot(\mathbf{Q}_s^\top\cdot\mathbf{P}_{t|s}^\top)-\left[\mathbf{Q}_s\cdot\vec{p}_s\cdot\left(\frac{1}{\vec{p}_t}\right)^\top\right]\odot\mathbf{P}_{t|s}^\top.
\end{aligned}
\tag{46}
$$

Finally, we prove that the preference growing process satisfies the Kolmogorov backward equation:

$$
\mathbf{R}_s\mathbf{P}_{s|t}=\left\{\mathbf{Q}_s^\top\odot\left[\vec{p}_s\cdot\left(\frac{1}{\vec{p}_s}\right)^\top\right]-\mathbf{Q}_s\left[\vec{p}_t\cdot\left(\frac{1}{\vec{p}_t}\right)^\top\right]\odot\mathbf{I}\right\}\cdot\left\{\left[\vec{p}_s\cdot\left(\frac{1}{\vec{p}_t}\right)^\top\right]\odot\mathbf{P}_{t|s}^\top\right\} \tag{47}
$$

The idea behind this step is to prove that the elements at each position of the matrices are identical, thereby establishing their equality.

The details are provided in Equation (48), (49) and (50).

$$
\begin{aligned}
&= \left\{\left[\vec{p}_s\cdot\left(\frac{1}{\vec{p}_t}\right)^\top\right]\odot(\mathbf{Q}_s^\top\cdot\mathbf{P}_{t|s}^\top)\right\}-\left\{\left[\mathbf{Q}_s\cdot\vec{p}_s\cdot\left(\frac{1}{\vec{p}_t}\right)^\top\right]\odot\mathbf{P}_{t|s}^\top\right\} \\
&= \frac{\partial\mathbf{P}_{s|t}}{\partial s}.
\end{aligned}
$$

$$
\begin{aligned}
&\left\{\mathbf{Q}_s^\top\odot\left[\vec{p}_s\cdot\left(\frac{1}{\vec{p}_s}\right)^\top\right]\right\}\cdot\left\{\left[\vec{p}_s\cdot\left(\frac{1}{\vec{p}_t}\right)^\top\right]\odot\mathbf{P}_{t|s}^\top\right\}(x,y) \\
&= \sum_{z\in\mathcal{X}} q_s(z,x)\frac{p_s(x)}{p_s(z)}\cdot\frac{p_s(z)}{p_t(y)}p_{t|s}(y|z) \\
&= \frac{p_s(x)}{p_t(y)}\cdot\sum_{z\in\mathcal{X}} q_s(z,x)p_{t|s}(y|z) \\
&= \left\{\left[\vec{p}_s\cdot\left(\frac{1}{\vec{p}_t}\right)^\top\right]\odot(\mathbf{Q}_s^\top\cdot\mathbf{P}_{t|s}^\top)\right\}(x,y),\forall x,y\in\mathcal{X}.
\end{aligned}
\tag{48}
$$

$$
\begin{aligned}
&\left\{\mathbf{Q}_s\cdot\left[\vec{p}_s\cdot\left(\frac{1}{\vec{p}_s}\right)^\top\right]\odot\mathbf{I}\right\}(x,y) \\
&= \sum_{z\in\mathcal{X}} q_s(x,z)\frac{p_s(z)}{p_s(y)}\cdot\delta_x(y) \\
&= \delta_x(y)\cdot\sum_{l\in\mathcal{X}} q_s(x,l)\frac{p_s(l)}{p_s(x)}.
\end{aligned}
\tag{49}
$$

$$\left\{\mathbf{Q}_s \cdot \left[\vec{p}_s \cdot \left(\frac{1}{\vec{p}_s}\right)^\top\right] \odot \mathbf{I}\right\} \cdot \left\{\left[\vec{p}_s \cdot \left(\frac{1}{\vec{p}_t}\right)^\top\right] \odot \mathbf{P}_{t|s}^\top\right\}(x,y) \tag{50}$$

$$= \sum_{z\in\mathcal{X}} \delta_x(z) \cdot \sum_{l\in\mathcal{X}} q_s(x,l)\frac{p_s(l)}{p_s(x)} \cdot \frac{p_s(z)}{p_t(y)} p_{t|s}(y|z)$$

$$= \sum_{l\in\mathcal{X}} q_s(x,l)\frac{p_s(l)}{p_s(x)} \cdot \frac{p_s(x)}{p_t(y)} p_{t|s}(y|x)$$

$$= p_{t|s}(y|x) \cdot \sum_{l\in\mathcal{X}} q_s(x,l)\frac{p_s(l)}{p_t(y)}$$

$$= \left\{\left[\mathbf{Q}_s \cdot \vec{p}_s \cdot \left(\frac{1}{\vec{p}_t}\right)^\top\right] \odot \mathbf{P}_{t|s}^\top\right\}(x,y), \forall x,y \in \mathcal{X}.$$

□

D Details of Different Fading Matrix Setting

D.1 Point-Wise Setting

In the setting of masked discrete diffusion models [40, 52, 53], we can model point-wise preference ratios. Specifically, we introduce an auxiliary general hard negative item x_{-1}, which is represented as a learnable embedding. The unified non-preference state corresponds to the general hard negative item x_{-1}, that is, $\vec{p}_T = \vec{e}_{-1} \in \mathbb{R}^{N+1}$, where $\vec{e}_{-1}$ denotes the one-hot vector associated with x_{-1}. In this case, the reference ratios $r_t(x_0, x_t \in \{x_0, x_{-1}\}, y \in \{x_0, x_{-1}\})$ capture only the relative preference between the positive item x_0 and the general hard negative x_{-1}. Thus, by using x_{-1} as a common reference, we derive the point-wise preference ratios.

Let $\vec{p}_T = \vec{e}_{-1} \in \mathbb{R}^{N+1}$. Then, the rank-1 fading matrix $\mathbf{E}$ is defined as follows:

$$\mathbf{E} = \begin{pmatrix} 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \\ 1 & \cdots & 1 & 1 \end{pmatrix}$$

In this case, the reference ratios $r_t(x_0, x_t \in \{x_0, x_{-1}\}, y \in \{x_0, x_{-1}\})$ model only the ratios between the general hard negative x_{-1} and real items $\mathcal{X}$.

$$r_t(x_0, x_t, y) = \begin{cases} r_t(x_0, x_t, x_t) = 0 & \text{if } x_t = y \\ r_t(x_0, x_0, x_{-1}) = \log\frac{1-\alpha_t}{\alpha_t} & \text{if } y = x_{-1} \text{ and } x_t = x_0 \\ r_t(x_0, x_{-1}, x_0) = \log\frac{\alpha_t}{1-\alpha_t} & \text{if } y = x_0 \text{ and } x_t = x_{-1} \end{cases}$$

$$\mathbf{Q}_t(x,y) = \beta(t) \cdot (\mathbf{E} - \mathbf{I}) = \begin{cases} -\beta(t) & \text{if } x = y \neq x_{-1} \\ \beta(t) & \text{if } x = x_{-1} \text{ and } x_t \neq x_{-1} \\ 0 & \text{otherwise} \end{cases}$$

We eastimates the preference ratios $\log\frac{p_t(y|u)}{p_t(x_t|u)}$ with $s_\Theta(x_t, t, u)_y$:

$$l_{SE}(x_0, x_t, y|u) = e^{s_\Theta(x_t,t,u)_y} - e^{r_t(x_0,x_t,y)} s_\Theta(x_t, t, u)_y + e^{r_t(x_0,x_t,y)}[r_t(x_0, x_t, y) - 1].$$

For one user preference data (u, x_0), with faded item x_t, we compute $\mathcal{L}_{SE}(x_0, x_t, y|u)$:

$$\mathcal{L}_{SE} = \sum_{y\in\{x_0,x_{-1}\}} \mathbf{Q}_t(x_t, y) \cdot l_{SE}(x_0, x_t, y|u)$$

$$
\begin{aligned}
&= \mathbf{Q}_t(x_t, x_t) \cdot l_{SE}(x_0, x_t, x_t|u) + \sum_{y \in \{x_0, x_{-1}\} \setminus \{x_t\}} \mathbf{Q}_t(x_t, y) \cdot l_{SE}(x_0, x_t, y|u) \\
&\quad s_\Theta(x_t, t, u)_{x_t} = r_t(x_0, x_t, x_t) = 0 \Rightarrow l_{SE}(x_0, x_t, x_t|u) = 0 \\
&= \sum_{y \in \{x_0, x_{-1}\} \setminus \{x_t\}} \mathbf{Q}_t(x_t, y) \cdot l_{SE}(x_0, x_t, y|u) \\
&\quad \mathbf{Q}_t(x_t \neq x_{-1}, y \neq x_t) = 0 \Rightarrow \mathcal{L}_{SE}(x_0, x_0, y|u) = 0 \Rightarrow x_t = x_{-1} \\
&= \mathbf{Q}_t(x_{-1}, x_0) \cdot l_{SE}(x_0, x_{-1}, x_0|u) \\
&= \beta(t) \cdot \left[e^{s_\Theta(x_t, t, u)_y} - \frac{\alpha_t}{1 - \alpha_t} \cdot s_\Theta(x_t, t, u)_y + \frac{\alpha_t}{1 - \alpha_t} [\log \frac{\alpha_t}{1 - \alpha_t} - 1] \right].
\end{aligned}
$$

D.2 Pair-Wise Setting

In the field of natural language processing, uniform discrete diffusion models are generally considered inferior to masked discrete diffusion models [38, 40]. However, in the context of recommendation, uniform discrete diffusion models correspond to pair-wise preference ratios, which resemble random negative sampling, and are in fact superior to point-wise masked discrete diffusion models. Specifically, we define the unified non-preference state as having equal probability over all items, *i.e.,* $\vec{p}_T = \vec{1} \in \mathbb{R}^N$, where $\vec{1}$ denotes the all-ones vector. In this setting, the reference ratios $r_t(x_0, x_t \in \mathcal{X}, y \in \mathcal{X})$ capture the relative preferences among all item pairs.

Let $\vec{p}_T = \vec{1} \in \mathbb{R}^N$, we have fading matrix $\mathbf{E}$ as follows:

$$
\mathbf{E} = \begin{pmatrix} \frac{1}{N} & \cdots & \frac{1}{N} \\ \vdots & \ddots & \vdots \\ \frac{1}{N} & \cdots & \frac{1}{N} \end{pmatrix}
$$

In this case, the reference ratios $r_t(x_0, x_t \in \mathcal{X}, y \in \mathcal{X})$ model the ratios of all item pairs.

$$
r_t(x_0, x_t, y) = \begin{cases} r_t(x_0, x_t, x_t) = 0 & \text{if } x_t = y \\ r_t(x_0, x_0, y \neq x_0) = -\log(1 + N \cdot \frac{\alpha_t}{1 - \alpha_t}) & \text{if } y \neq x_0 \text{ and } x_t = x_0 \\ r_t(x_0, x_t \neq x_0, x_0) = \log(1 + N \cdot \frac{\alpha_t}{1 - \alpha_t}) & \text{if } y = x_0 \text{ and } x_t \neq x_0 \\ r_t(x_0, x_t \neq x_0, y \neq x_0) = 0 & \text{otherwise} \end{cases}
$$

$$
\mathbf{Q}_t(x, y) = \beta(t) \cdot (\mathbf{E} - \mathbf{I}) = \begin{cases} \beta(t)(\frac{1}{N} - 1) & \text{if } x = y \\ \beta(t) \cdot \frac{1}{N} & \text{otherwise} \end{cases}
$$

We eastimates the preference ratios $\log \frac{p_t(y|u)}{p_t(x_t|u)}$ with $s_\Theta(x_t, t, u)_y$:

$$
l_{SE}(x_0, x_t, y|u) = e^{s_\Theta(x_t, t, u)_y} - e^{r_t(x_0, x_t, y)} s_\Theta(x_t, t, u)_y + e^{r_t(x_0, x_t, y)} [r_t(x_0, x_t, y) - 1].
$$

For one user preference data (u, x_0), with faded item x_t, we compute $\mathcal{L}_{SE}(x_0, x_t, y|u)$:

$$
\begin{aligned}
\mathcal{L}_{SE} &= \sum_{y \in \mathcal{X}} \mathbf{Q}_t(x_t, y) \cdot l_{SE}(x_0, x_t, y|u) \\
&= \mathbf{Q}_t(x_t, x_t) \cdot l_{SE}(x_0, x_t, x_t|u) + \sum_{y \in \mathcal{X} \setminus \{x_t\}} \mathbf{Q}_t(x_t, y) \cdot l_{SE}(x_0, x_t, y|u) \\
&\quad s_\Theta(x_t, t, u)_{x_t} = r_t(x_0, x_t, x_t) = 0 \Rightarrow l_{SE}(x_0, x_t, x_t|u) = 0 \\
&= \sum_{y \in \mathcal{X} \setminus \{x_t\}} \mathbf{Q}_t(x_t, y) \cdot l_{SE}(x_0, x_t, y|u) \\
&\quad \mathbf{Q}_t(x_t, y \neq x_t) = \beta(t) \cdot \frac{1}{N} \\
&= \beta(t) \cdot \frac{1}{N} \cdot \sum_{y \in \mathcal{X} \setminus \{x_t\}} l_{SE}(x_0, x_{-1}, x_0|u)
\end{aligned}
$$

$$e^{\bar{s}_\Theta(x_t,t,u)} = \frac{1}{N} \cdot \sum_{y \in \mathcal{X} \setminus \{x_t\}} e^{s_\Theta(x_t,t,u)_y} = \frac{1}{N} \cdot \left[\sum_{y \in \mathcal{X}} e^{s_\Theta(x_t,t,u)_y} - 1 \right]$$

$$\bar{s}_\Theta(x_t,t,u) = \frac{1}{N} \cdot \sum_{y \in \mathcal{X} \setminus \{x_t\}} s_\Theta(x_t,t,u)_y = \frac{1}{N} \cdot \sum_{y \in \mathcal{X}} s_\Theta(x_t,t,u)_y$$

if $x_t = x_0$, denote$\Delta = N \cdot \frac{\alpha_t}{1-\alpha_t}$:

$$= \beta(t) \cdot \left[e^{\bar{s}_\Theta(x_t,t,u)} - \frac{1}{1+\Delta} \cdot \bar{s}_\Theta(x_t,t,u) - \frac{1}{1+\Delta}(\log(1+\Delta)+1) \right]$$

if $x_t \neq x_0$, denote$\Delta = N \cdot \frac{\alpha_t}{1-\alpha_t}$:

$$= \beta(t) \cdot \left[e^{\bar{s}_\Theta(x_t,t,u)} - \bar{s}_\Theta(x_t,t,u) - \Delta \cdot s_\Theta(x_t,t,u)_{x_0} + (1+\Delta)(\log(1+\Delta)-1) \right].$$

D.3 Hybrid-Wise Setting

Note that the point-wise and pair-wise settings are independent of each other. Therefore, we can define a hybrid non-preference state as $\vec{p}_T = \lambda(\vec{1} \in \mathbb{R}^N, 0) + (1-\lambda)\vec{e}_{-1} \in \mathbb{R}^{n+1}$, simultaneously modeling point-wise and pair-wise preference ratios. Since $|\mathcal{X}|$ is typically large, the hybrid coefficient λ should be designed as $1 - 10^{-n_\lambda}$, where $n_\lambda \in \mathbb{Z}^+$.

Let $\vec{p}_1 = \lambda(\vec{1}, 0) + (1-\lambda)\vec{e}_{-1} \in \mathbb{R}^{n+1}$, we have fading matrix $\mathbf{E}$ as follows:

$$\mathbf{E} = \begin{pmatrix} \frac{\lambda}{N} & \cdots & \frac{\lambda}{N} & \frac{\lambda}{N} \\ \vdots & \ddots & \vdots & \vdots \\ \frac{\lambda}{N} & \cdots & \frac{\lambda}{N} & \frac{\lambda}{N} \\ 1-\lambda & \cdots & 1-\lambda & 1-\lambda \end{pmatrix}$$

In this case, the reference ratios $r_t(x_0, x_t \in \mathcal{X}, y \in \mathcal{X})$ model the ratios of all item pairs.

$$r_t(x_0,x_t,y) = \begin{cases} 0 & \text{if } x_t = y \\ \log \frac{\alpha_t + (1-\alpha_t)\frac{\lambda}{N}}{(1-\alpha_t)(1-\lambda)} & \text{else if } y = x_0 \text{ and } x_t = x_{-1} \\ \log \frac{\frac{\lambda}{N}}{(1-\lambda)} & \text{else if } y \neq x_0 \text{ and } x_t = x_{-1} \\ \log \frac{(1-\alpha_t)(1-\lambda)}{\alpha_t + (1-\alpha_t)\frac{\lambda}{N}} & \text{else if } y = x_{-1} \text{ and } x_t = x_0 \\ \log \frac{(1-\alpha_t)\frac{\lambda}{N}}{\alpha_t + (1-\alpha_t)\frac{\lambda}{N}} & \text{else if } y \neq x_{-1} \text{ and } x_t = x_0 \\ \log \frac{(1-\alpha_t)(1-\lambda)}{(1-\alpha_t)\frac{\lambda}{N}} & \text{else if } y = x_{-1} \text{ and } x_t \neq \{x_{-1}, x_0\} \\ \log \frac{\alpha_t + (1-\alpha_t)\frac{\lambda}{N}}{(1-\alpha_t)\frac{\lambda}{N}} & \text{else if } y = x_0 \text{ and } x_t \neq \{x_{-1}, x_0\} \\ 0 & \text{else if } y \neq \{x_{-1}, x_0\} \text{ and } x_t \neq \{x_{-1}, x_0\} \end{cases}$$

$$\mathbf{Q}_t(x,y) = \beta(t) \cdot (\mathbf{E} - \mathbf{I}) = \begin{cases} \beta(t)(-\lambda) & \text{if } x = y = x_{-1} \\ \beta(t)(\frac{\lambda}{N} - 1) & \text{if } x = y \neq x_{-1} \\ \beta(t)(1-\lambda) & \text{if } x \neq y \text{ and } x = x_{-1} \\ \beta(t)(\frac{\lambda}{N}) & \text{if } x \neq y \text{ and } x \neq x_{-1} \end{cases}$$

We estimate the preference ratios $\log \frac{p_t(y|u)}{p_t(x_t|u)}$ with $s_\Theta(x_t,t,u)_y$:

$$l_{SE}(x_0,x_t,y|u) = e^{s_\Theta(x_t,t,u)_y} - e^{r_t(x_0,x_t,y)} s_\Theta(x_t,t,u)_y + e^{r_t(x_0,x_t,y)}[r_t(x_0,x_t,y) - 1].$$

For one user preference data (u, x_0), with faded item x_t, we compute $\mathcal{L}_{SE}(x_0,x_t,y|u)$:

$$\mathcal{L}_{SE} = \sum_{y \in \mathcal{X}} \mathbf{Q}_t(x_t,y) \cdot l_{SE}(x_0,x_t,y|u)$$

$$= \mathbf{Q}_t(x_t, x_t) \cdot l_{SE}(x_0, x_t, x_t|u) + \sum_{y \in \mathcal{X} \setminus \{x_t\}} \mathbf{Q}_t(x_t, y) \cdot l_{SE}(x_0, x_t, y|u)$$

$$s_\Theta(x_t, t, u)_{x_t} = r_t(x_0, x_t, x_t) = 0 \Rightarrow l_{SE}(x_0, x_t, x_t|u) = 0$$

$$= \sum_{y \in \mathcal{X} \setminus \{x_t\}} \mathbf{Q}_t(x_t, y) \cdot l_{SE}(x_0, x_t, y|u)$$

$$\mathbf{Q}_t(x_t = x_{-1}, y \neq x_t) = \beta(t) \cdot (1 - \lambda), \mathbf{Q}_t(x_t \neq x_{-1}, y \neq x_t) = \beta(t) \cdot (\frac{\lambda}{N})$$

$$= \mathbf{Q}_t(x_t, y \neq x_t) \cdot \sum_{y \in \mathcal{X} \setminus \{x_t\}} l_{SE}(x_0, x_t, y|u)$$

$$e^{\bar{\mathbf{s}}_\Theta(x_t,t,u)} = \frac{1}{N} \cdot \sum_{y \in \mathcal{X} \setminus \{x_t\}} e^{s_\Theta(x_t,t,u)_y} = \frac{1}{N} \cdot \left[\sum_{y \in \mathcal{X}} e^{s_\Theta(x_t,t,u)_y} - 1 \right]$$

$$\bar{\mathbf{s}}_\Theta(x_t, t, u) = \frac{1}{N} \cdot \sum_{y \in \mathcal{X} \setminus \{x_t\}} s_\Theta(x_t, t, u)_y = \frac{1}{N} \cdot \sum_{y \in \mathcal{X}} s_\Theta(x_t, t, u)_y$$

if $x_t = x_{-1}$, denote $\Delta = N \cdot \frac{\alpha_t}{1 - \alpha_t}$ and $\Lambda = \frac{\lambda \cdot \Delta}{\lambda \cdot \Delta + N}$:

$$= \beta(t) \cdot \left[N(1 - \lambda) \cdot e^{\bar{\mathbf{s}}_\Theta(x_t,t,u)} - \lambda \cdot \bar{\mathbf{s}}_\Theta(x_t, t, u) - \frac{\frac{1}{\Lambda} - \lambda}{N} \cdot s_\Theta(x_t, t, u)_{x_0} \right]$$

$$+\beta(t) \cdot \left\{ \lambda \cdot \left[1 + \frac{1}{N} \left(\frac{1}{\Lambda} - 1 \right) \right] \cdot \left[\log \left(\frac{1}{N} \cdot \frac{\lambda}{1 - \lambda} \right) - 1 \right] - \frac{\log \Lambda}{\Lambda} \right\}$$

if $x_t = x_0$, denote $\Delta = N \cdot \frac{\alpha_t}{1 - \alpha_t}$ and $\Lambda = \frac{\lambda \cdot \Delta}{\lambda \cdot \Delta + N}$:

$$= \beta(t) \cdot \left[\lambda \cdot e^{\bar{\mathbf{s}}_\Theta(x_t,t,u)} - \lambda \cdot \Lambda \cdot \bar{\mathbf{s}}_\Theta(x_t, t, u) - (1 - \lambda - \frac{\lambda}{N}) \cdot \Lambda \cdot s_\Theta(x_t, t, u)_{x_{-1}} \right]$$

$$+\beta(t) \cdot \left\{ (1 - \frac{\lambda}{N}) \cdot \Lambda \cdot (\log \Lambda - 1) - (1 - \lambda) \cdot \Lambda \cdot (\log(\frac{1}{N} \frac{\lambda}{1 - \lambda})) \right\}$$

if $x_t \neq \{x_0, x_{-1}\}$, denote $\Delta = N \cdot \frac{\alpha_t}{1 - \alpha_t}$ and $\Lambda = \frac{\lambda \cdot \Delta}{\lambda \cdot \Delta + N}$:

$$= \beta(t) \cdot \left[\lambda \cdot e^{\bar{\mathbf{s}}_\Theta(x_t,t,u)} - \lambda \cdot \bar{\mathbf{s}}_\Theta(x_t, t, u) \right]$$

$$-\beta(t) \cdot \left[(\frac{1}{N \cdot \Lambda} - \frac{\lambda}{N}) \cdot \Lambda \cdot s_\Theta(x_t, t, u)_{x_0}) + (1 - \lambda - \frac{\lambda}{N}) \cdot s_\Theta(x_t, t, u)_{x_0} \right]$$

$$+\beta(t) \cdot \left\{ \frac{1}{N \cdot \Lambda} \cdot (\log \Lambda + \log \lambda - 1) - (1 - \lambda) \cdot (1 + \log(\frac{1}{N} \frac{\lambda}{1 - \lambda})) - (1 - \frac{2}{N}) \cdot \lambda \right\}.$$

D.4 Adative Setting

Furthermore, we can adaptively update the non-preference state $\vec{p}_T$ initialized with above settings. Under adaptive setting, $\vec{p}_T = \vec{\mu} = \text{softmax}(\vec{\theta})$, where $\vec{\theta}$ are learnable parameters.

Let $\vec{p}_T = \vec{\mu} \in \mathbb{R}^N$ or $\mathbb{R}^{N+1}$, with $\sum_{x \in \mathcal{X}} \mu_x = 1$, we have fading matrix $\mathbf{E}$ as follows:

$$\mathbf{E} = \begin{pmatrix} \mu_1 & \cdots & \mu_1 \\ \vdots & \ddots & \vdots \\ \mu_{|\mathcal{X}|} & \cdots & \mu_{|\mathcal{X}|} \end{pmatrix}$$

By arbitrarily specifying a distribution $\vec{\mu}$ satisfying $\sum_{x \in \mathcal{X}} \mu_x = 1$, we can instantiate any desired non-preference state, which is physically associated with different negative sampling strategies. In

this case, the reference ratios $r_t(x_0, x_t \in \mathcal{X}, y \in \mathcal{X})$ are computed as follows:

$$r_t(x_0, x_t, y) = \begin{cases} r_t(x_0, x_t, x_t) = 0 & \text{if } x_t = y \\ r_t(x_0, x_0, y \neq x_0) = -\log(\frac{\alpha_t + (1-\alpha_t)\cdot\mu_{x_t}}{(1-\alpha_t)\cdot\mu_y}) & \text{if } y \neq x_0 \text{ and } x_t = x_0 \\ r_t(x_0, x_t \neq x_0, x_0) = \log(\frac{\alpha_t + (1-\alpha_t)\cdot\mu_{x_t}}{(1-\alpha_t)\cdot\mu_y}) & \text{if } y = x_0 \text{ and } x_t \neq x_0 \\ r_t(x_0, x_t \neq x_0, y \neq x_0) = \log\frac{\mu_y}{\mu_{x_t}} & \text{otherwise} \end{cases}$$

$$\mathbf{Q}_t(x, y) = \beta(t) \cdot (\mathbf{E} - \mathbf{I}) = \begin{cases} \beta(t)(\frac{1}{N} - 1) & \text{if } x = y \\ \beta(t) \cdot \frac{1}{N} & \text{otherwise} \end{cases}$$

We eastimates the preference ratios $\log \frac{p_t(y|u)}{p_t(x_t|u)}$ with $s_\Theta(x_t, t, u)_y$:

$$l_{SE}(x_0, x_t, y|u) = e^{s_\Theta(x_t,t,u)_y} - e^{r_t(x_0,x_t,y)} s_\Theta(x_t, t, u)_y + e^{r_t(x_0,x_t,y)}[r_t(x_0, x_t, y) - 1].$$

For one user preference data (u, x_0), with faded item x_t, we compute $\mathcal{L}_{SE}(x_0, x_t, y|u)$:

$$\begin{aligned}
\mathcal{L}_{SE} &= \sum_{y\in\mathcal{X}} \mathbf{Q}_t(x_t, y) \cdot l_{SE}(x_0, x_t, y|u) \\
&= \mathbf{Q}_t(x_t, x_t) \cdot l_{SE}(x_0, x_t, x_t|u) + \sum_{y\in\mathcal{X}\setminus\{x_t\}} \mathbf{Q}_t(x_t, y) \cdot l_{SE}(x_0, x_t, y|u) \\
&\quad s_\Theta(x_t, t, u)_{x_t} = r_t(x_0, x_t, x_t) = 0 \Rightarrow l_{SE}(x_0, x_t, x_t|u) = 0 \\
&= \sum_{y\in\mathcal{X}\setminus\{x_t\}} \mathbf{Q}_t(x_t, y) \cdot l_{SE}(x_0, x_t, y|u) \\
&\quad \mathbf{Q}_t(x_t, y \neq x_t) = \beta(t) \cdot \mu_{x_t} \\
&= \beta(t) \cdot \frac{1}{N} \cdot \sum_{y\in\mathcal{X}\setminus\{x_t\}} l_{SE}(x_0, x_{-1}, x_0|u) \\
&\quad e^{\bar{\mathbf{s}}_\Theta(x_t,t,u)} = \frac{1}{N} \cdot \sum_{y\in\mathcal{X}\setminus\{x_t\}} e^{s_\Theta(x_t,t,u)_y} = \frac{1}{N} \cdot \left[\sum_{y\in\mathcal{X}} e^{s_\Theta(x_t,t,u)_y} - 1\right] \\
&\quad \tilde{\mathbf{s}}_\Theta(x_t, t, u) = \sum_{y\in\mathcal{X}\setminus\{x_t\}} \mu_y \cdot s_\Theta(x_t, t, u)_y = \sum_{y\in\mathcal{X}} \mu_y \cdot s_\Theta(x_t, t, u)_y \\
&\quad \hat{\mu} = \sum_{y\in\mathcal{X}} \mu_y \cdot \log \mu_y \\
&\text{if } x_t = x_0, \text{ denote } \Sigma = \frac{\alpha_t}{1-\alpha_t} : \\
&= \beta(t) \cdot \left[\mu_{x_t} \cdot e^{\bar{\mathbf{s}}_\Theta(x_t,t,u)} - \frac{1}{\Sigma + \mu_{x_t}} \cdot \tilde{\mathbf{s}}_\Theta(x_t, t, u)\right] \\
&+\beta(t) \cdot \frac{\mu_{x_t}}{\mu_{x_t} + \Sigma} \cdot [\hat{\mu} + (\mu_{x_t} - 1) \cdot (\log(\mu_{x_t} + \Sigma) - 1) - \mu_{x_t} \cdot \log \mu_{x_t}] \\
&\text{if } x_t \neq x_0, \text{ denote } \Sigma = \frac{\alpha_t}{1-\alpha_t} : \\
&= \beta(t) \cdot \left[\mu_{x_t} \cdot e^{\bar{\mathbf{s}}_\Theta(x_t,t,u)} - \tilde{\mathbf{s}}_\Theta(x_t, t, u) - \Sigma \cdot s_\Theta(x_t, t, u)_{x_0}\right] \\
&+\beta(t) \cdot [\hat{\mu} + \mu_{x_t} - (1 + \Sigma)(1 + \log(\mu_{x_t})) + (\Sigma + \mu_{x_0}) \log(\Sigma + \mu_{x_0}) - (\mu_{x_0}) \cdot \log(\mu_{x_0})] .
\end{aligned}$$

E Rank-r Solution of Fading Matrix E

Theorem 3 (Nonnegative Idempotent Decomposition)**.** *Let* $\mathbf{E} \in \mathbb{R}^{N\times N}$ *be entrywise nonnegative and idempotent* ($\mathbf{E}^2 = \mathbf{E}$) *with* $\operatorname{rank}(\mathbf{E}) = r$*. Then the following statements are equivalent:*

(A) $\mathbf{E}$ *is nonnegative and idempotent with* $\operatorname{rank}(\mathbf{E}) = r$.

(B) *There exist r nonzero, pairwise disjoint*[2] *nonnegative vectors $\vec{p}^1, \ldots, \vec{p}^r \in \mathbb{R}^N_{\geq 0}$ and the corresponding support-indicator vectors $\vec{s}^i := \mathbf{1}_{\mathrm{supp}(\vec{p}^i)} \in \{0,1\}^N$ such that*

$$\mathbf{E} = \sum_{i=1}^{r} \mathbf{E}_i, \qquad \mathbf{E}_i := \frac{\vec{p}^i (\vec{s}^i)^\top}{(\vec{s}^i)^\top \vec{p}^i}, \tag{51}$$

where $(\vec{s}^i)^\top \vec{p}^i > 0$ for each i and $\mathrm{supp}(\vec{p}^i) := \{\, k \in \{1, \ldots, N\} : p^i_k > 0 \,\}$. Equivalently, for any column index $j \in \{1, \ldots, N\}$,

$$\mathbf{E}_{:j} = \begin{cases} \vec{p}^i, & \text{if } j \in \mathrm{supp}(\vec{p}^i) \text{ for some } i \in \{1, \ldots, r\}, \\ \vec{0}, & \text{if } j \notin \bigcup_{i=1}^r \mathrm{supp}(\vec{p}^i). \end{cases}$$

Moreover, writing the support indicator as $\vec{1}$ restricted to $\mathrm{supp}(\vec{p}^i)$, the decomposition (51) *admits the compact form*

$$\mathbf{E} = \sum_{i=1}^{r} \frac{\vec{p}^i \, \vec{1}^\top_{\mathrm{supp}(\vec{p}^i)}}{\vec{1}^\top_{\mathrm{supp}(\vec{p}^i)} \vec{p}^i}, \qquad \vec{p}^i \geq 0, \quad (\vec{p}^i)^\top \vec{p}^j = 0 \;\; (i \neq j).$$

Proof (Sufficiency (B) $\Rightarrow$ (A)**).** Entrywise nonnegativity is immediate from $\vec{p}^i \geq 0$ and $\vec{s}^i \geq 0$. Idempotence follows from disjoint supports: for $i \neq j$ we have $(\vec{s}^i)^\top \vec{p}^j = 0$, hence

$$\mathbf{E}_i \mathbf{E}_j = \frac{\vec{p}^i (\vec{s}^i)^\top}{(\vec{s}^i)^\top \vec{p}^i} \cdot \frac{\vec{p}^j (\vec{s}^j)^\top}{(\vec{s}^j)^\top \vec{p}^j} = \frac{\vec{p}^i \, (\vec{s}^i)^\top \vec{p}^j \, (\vec{s}^j)^\top}{(\vec{s}^i)^\top \vec{p}^i \, (\vec{s}^j)^\top \vec{p}^j} = \mathbf{0}.$$

For $i = j$ we obtain

$$\mathbf{E}_i^2 = \frac{\vec{p}^i (\vec{s}^i)^\top \vec{p}^i (\vec{s}^i)^\top}{\left((\vec{s}^i)^\top \vec{p}^i\right)^2} = \frac{\vec{p}^i (\vec{s}^i)^\top}{(\vec{s}^i)^\top \vec{p}^i} = \mathbf{E}_i,$$

so $\mathbf{E}^2 = (\sum_i \mathbf{E}_i)^2 = \sum_i \mathbf{E}_i^2 = \sum_i \mathbf{E}_i = \mathbf{E}$. To compute the rank, observe that the column space of $\mathbf{E}_i$ is $\mathrm{span}\{\vec{p}^i\}$ and $\mathrm{supp}(\vec{p}^i)$ are disjoint, hence the r one-dimensional subspaces are independent; therefore $\mathrm{rank}(\mathbf{E}) = \sum_i \mathrm{rank}(\mathbf{E}_i) = r$.

Proof (Necessity (A) $\Rightarrow$ (B)**).** Since $\mathbf{E}$ is idempotent ($\mathbf{E}^2 = \mathbf{E}$), its eigenvalues are 0 or 1, so the image $\mathrm{Im}(\mathbf{E})$ has dimension r. Because $\mathbf{E} \geq 0$, every column of $\mathbf{E}$ is either the zero vector or a nonnegative fixed point of $\mathbf{E}$ (indeed, for the j-th standard basis vector $\vec{e}_j$, we have $\mathbf{E}(\mathbf{E}\vec{e}_j) = \mathbf{E}\vec{e}_j \geq 0$). Group the nonzero columns of $\mathbf{E}$ by proportionality: put indices j, k in the same class if the j-th and k-th nonzero columns are positive multiples of each other. This is an equivalence relation, and it produces exactly r classes, say $S_1, \ldots, S_r$, because each class contributes one linearly independent direction in $\mathrm{Im}(\mathbf{E})$. Pick one representative nonzero column from each class and denote it by $\vec{p}^i \geq 0$ ($i = 1, \ldots, r$). By construction, for every $j \in S_i$ the j-th column of $\mathbf{E}$ equals $\vec{p}^i$, and for $j \notin \bigcup_i S_i$ the column is zero. The classes are disjoint in support by definition. Let $\vec{s}^i := \mathbf{1}_{S_i}$ be the indicator of S_i. The matrix that places column $\vec{p}^i$ on S_i and zeros elsewhere is

$$\mathbf{E}_i \;=\; \frac{\vec{p}^i (\vec{s}^i)^\top}{(\vec{s}^i)^\top \vec{p}^i}, \quad \text{with } (\vec{s}^i)^\top \vec{p}^i > 0.$$

Therefore

$$\mathbf{E} \;=\; \sum_{i=1}^{r} \mathbf{E}_i,$$

which is exactly the Equation (51). Disjoint supports immediately give $(\vec{p}^i)^\top \vec{p}^j = 0$ for $i \neq j$. □

Physical intuition:

- **Clustering.** Partition the item set $\mathcal{X}$ ($|\mathcal{X}| = N$) into r clusters $\mathcal{C}_1, \ldots, \mathcal{C}_r$, where items in the same cluster are more similar. In practice, one may obtain $\{\mathcal{C}_i\}$ via vector-based methods (e.g., k-means on semantic embeddings) or via r-way classification on the user–item bipartite/weighted sequential graph. Since these constructions require side information beyond our core formulation, a full exploration is deferred to future work.

[2] Disjointness means the supports do not overlap: $\mathrm{supp}(\vec{p}^i) \cap \mathrm{supp}(\vec{p}^j) = \varnothing$ for $i \neq j$.

- **Cluster-wise target distributions.** Define for each cluster a nonnegative target vector

$$\vec{p}_T^{\,i}(x) > 0 \text{ if } x \in \mathcal{C}_i, \qquad \vec{p}_T^{\,i}(x) = 0 \text{ otherwise}, \quad i = 1, \dots, r.$$

Because $\{\vec{p}_T^{\,i}\}_{i=1}^r$ have disjoint supports, the resulting fading matrix $\mathbf{E} = \sum_{i=1}^r \frac{\vec{p}_T^{\,i}(\vec{s}^{\,i})^\top}{(\vec{s}^{\,i})^\top \vec{p}_T^{\,i}}$ is idempotent with $\mathrm{rank}(\mathbf{E}) = r$ (Theorem 3), where $\vec{s}^{\,i} = \mathbf{1}_{\mathrm{supp}(\vec{p}_T^{\,i})}$.

- **Effect.** During fading, each preferred item is replaced only by items within the same cluster, which filters out cross-cluster candidates and better respects item heterogeneity. This rank-r structure also improves efficiency. Using the decomposition, for any vector $\mathbf{x} \in \mathbb{R}^N$,

$$\mathbf{E}\mathbf{x} = \sum_{i=1}^r \vec{p}_T^{\,i} \frac{(\vec{s}^{\,i})^\top \mathbf{x}}{(\vec{s}^{\,i})^\top \vec{p}_T^{\,i}},$$

so computing $(\vec{s}^{\,i})^\top \mathbf{x}$ costs $\mathcal{O}(|\mathcal{C}_i|)$ and the combination step costs $\mathcal{O}(rN)$, reducing the overall complexity from the naive $\mathcal{O}(N^2)$ to $\mathcal{O}(rN)$.

- **Takeaway.** PreferGrow equipped with a *rank-r* fading matrix — i.e., cluster-wise replacement — combines theoretical elegance with practical realism: it confines replacements within semantically coherent groups while offering a scalable $\mathcal{O}(rN)$ implementation.

F Algorithms for Training and Inference

Algorithm 1 Training Algorithm of PreferGrow

Input: user preference data $\mathcal{D} = \{(u, x_0)\}$, non-preference user ratio p, retention probability $\alpha_t = e^{-\int_0^t \beta(\tau)\mathrm{d}\tau}$ with $\int_0^t \beta(\tau)\mathrm{d}\tau = (\beta_{\min})^{1-t}(\beta_{\max})^t$ or $\int_0^t \beta(\tau)\mathrm{d}\tau = \log(1-(1-\beta_{\mathrm{scale}}\cdot t))$, and the non-preference state $\vec{p}_T$ for preference fading.
Output: estimated *Preference Ratios* $\mathbf{s}_\Theta(x_t, t, u)$ and the non-preference user ϕ.
repeat
  $(u, x_0) \sim \mathcal{D}$ ▷ preference user-item pair.
  $t \sim \mathrm{Uniform}(\{1, \dots, T\})$ ▷ sampling timestep t uniformly.
  $u = \phi$ with probability p ▷ non-preference user modeling.
  $x_t \sim p_{t|0}(\cdot|x_0) = \alpha_t \vec{e}_{x_0} + (1-\alpha_t)\vec{p}_T, x_t \in \mathcal{X}$ ▷ retain or replace for preference fading.
  Compute the score entropy loss $\mathcal{L}_{SE} = \sum_{y\in\mathcal{X}} \mathbf{Q}_t(x_t, y) \cdot l_{SE}(x_0, x_t, y|u)$ as Appendix D.
  Take gradient descent step on $\nabla_\theta \mathcal{L}_{SE}$ and update parameters.
until converged

Algorithm 2 Inference Algorithm of PreferGrow

Input: user condition u, sampling timesteps $S_\tau = \{\tau_i\}_{i=0}^S$ with $\tau_S = T$ and $\tau_0 = 0$, personalization strength w, estimated *Preference Ratios* $\mathbf{s}_\theta(x_t, t, u)$ and the non-preference user ϕ.
Output: grown preference scores $p(x_0|u), x_0 \in \mathcal{X}$ of user u.
$x_T = x_{\tau_S} \sim \vec{p}_T, x_T \in \mathcal{X}$ ▷ the non-preference state.
for $s = S$ to 1 **do**
  $\hat{\mathbf{s}}_\Theta(x_{\tau_s}, \tau_s, u) = (1+w)\mathbf{s}_\Theta(x_{\tau_s}, \tau_s, u) - w \cdot \mathbf{s}_\Theta(x_{\tau_s}, \tau_s, \phi)$ ▷ personalization enhancement.
  $p_{\tau_{s-1}|\tau_s}(x_{\tau_{s-1}}|x_{\tau_s}, u) = p_{\tau_s|\tau_{s-1}}(x_{\tau_s}|x_{\tau_{s-1}}) \cdot \sum_{z\in\mathcal{X}} p^{-1}_{\tau_s|\tau_{s-1}}(x_{\tau_{s-1}}|z) \cdot e^{\hat{\mathbf{s}}_\Theta(x_{\tau_s}, \tau_s, u)_z}$
    ▷ $p_{\tau_s|\tau_{s-1}}(x_{\tau_s}|x_{\tau_{s-1}}) = \frac{\alpha_{\tau_s}}{\alpha_{\tau_{s-1}}}\delta_{x_{\tau_s}}(x_{\tau_{s-1}}) + (1 - \frac{\alpha_{\tau_s}}{\alpha_{\tau_{s-1}}}) \cdot \vec{p}_T(x_{\tau_s})$ as Equation (7).
    ▷ $p^{-1}_{\tau_s|\tau_{s-1}}(x_{\tau_{s-1}}|z) = \frac{\alpha_{\tau_{s-1}}}{\alpha_{\tau_s}}\delta_{x_{\tau_{s-1}}}(z) + (1 - \frac{\alpha_{\tau_{s-1}}}{\alpha_{\tau_s}}) \cdot \vec{p}_T(x_{\tau_{s-1}})$ as Equation (9).
    ▷ $\sum_{z\in\mathcal{X}} p^{-1}_{\tau_s|\tau_{s-1}}(x_{\tau_{s-1}}|z) \cdot e^{\hat{\mathbf{s}}_\Theta(x_{\tau_s}, \tau_s, u)_z} = \frac{\alpha_{\tau_{s-1}}}{\alpha_{\tau_s}} e^{\hat{\mathbf{s}}_\Theta(x_{\tau_s}, \tau_s, u)_{x_{\tau_{s-1}}}}$
      $+(1 - \frac{\alpha_{\tau_{s-1}}}{\alpha_{\tau_s}}) \cdot \vec{p}_T(x_{\tau_{s-1}}) \cdot \sum_{z\in\mathcal{X}} e^{\hat{\mathbf{s}}_\Theta(x_{\tau_s}, \tau_s, u)_z}$.
  $x_{\tau_{s-1}} \sim p_{\tau_{s-1}|\tau_s}(x_{\tau_{s-1}}|x_{\tau_s}, u), x_{\tau_{s-1}} \in \mathcal{X}$. ▷ reverse preference growing.
end for
return $p(x_0|u) = p_{\tau_0|\tau_1}(x_{\tau_0}|x_{\tau_1}, u), x_0 \in \mathcal{X}$ ▷ grown preference scores of user u.

Table 3: Statistics of datasets after preprocessing.

Dataset	# users	# items	# Interactions	sparsity
Movies	6040	3883	1001456	04.27%
Steam	39795	9265	2949605	00.80%
Beauty	22,363	12,101	198,502	00.07%
Toys	19,412	11,924	138,444	00.06%
Sports	35,598	18,357	256,598	00.04%

G Experiments Details

G.1 Datasets

We evaluate PreferGrow on five real-world benchmark datasets:

- **MoviesLens** [67] is a commonly used movie recommendation dataset that contains user ratings, movie titles, and movie genres.
- **Steam** [58] encompasses user reviews for video games on the Steam Store.
- **Beauty** [68] contains movie details and user reviews from Jun 1996 to Sep 2023.
- **Toys** [68] includes user reviews and metadata for toys and games from Jun 1996 to Jul 2014.
- **Sports** [68] comprises reviews and metadata for sports and outdoor products from 1996 to 2014.

Following prior works [9, 27], we adopt the user-splitting strategy, which has been shown to effectively prevent information leakage in test sets [69]. Specifically, we sort all sequences chronologically for each dataset and then split the data into training, validation, and test sets with an 8:1:1 ratio, while preserving the last 10 interactions as the historical sequence. The statistical characteristics of the processed dataset are shown in Table 3. As observed from the table, the recommendation datasets face a significant challenge of severe data sparsity.

G.2 Baselines

We compare PreferGrow with both traditional discriminative recommenders using negative sampling and diffusion-based generative recommenders, including classical recommenders (SASRec [58], Caser [60], GRURec [61]), item-level diffusion-based recommenders (DreamRec [9], PreferDiff [27]), and preference score-level diffusion-based recommenders (DiffRec [28], DDSR [35]):

- **SASRec** [58] leverages the self-attention mechanism in Transformer to model user preference scores from interaction histories, addressing data sparsity through negative sampling.
- **Caser** [60] utilizes horizontal and vertical convolutional filters to capture sequential patterns at the point-level and union-level, allowing for skip behaviors, and models user preference scores while addressing data sparsity through negative sampling.
- **GRURec** [61] adopts RNNs to model user preference scores from interaction histories, mitigating data sparsity through negative sampling.
- **DreamRec** [9] reshapes sequential recommendation as oracle item generation, addressing data sparsity by adding Gaussian noise to dense item embeddings.
- **PreferDiff** [27] introduces an optimization objective specifically designed for item-level DM-based recommenders, which can integrate multiple negative samples, addressing data sparsity by adding noise to dense item embeddings and using negative sampling both.
- **DiffRec** [28] is a preference score-level diffusion-based generative recommender assuming a Gaussian prior, addressing data sparsity by adding Gaussian noise to preference scores, without considering the constraints of the probability simplex.
- **DDSR** [35] is a preference score-level diffusion-based generative recommender assuming a categorical prior, addressing data sparsity by adding discrete noise to preference scores while respecting the constraints of the probability simplex.

G.3 Implementation Details

Training settings: We implement all models using Python 3.7 and PyTorch 1.12.1 on an Nvidia GeForce RTX 3090. During training, all methods are trained with a fixed batch size of 256 using the Adam optimizer. Additionally, we apply early stopping based on the model's performance on the validation set. To ensure reproducibility, we fix all random seeds to 100 in our main experiments, a randomly chosen value. For the classic recommenders with negative sampling, we employ the binary cross-entropy (BCE) loss. PreferGrow uses SASRec as the encoding model for the user's historical sequence, and we adopt both Hybrid-Wise and Adaptive settings for the fading matrix. For all SASRec modules, we apply RoPE position encoding [70]. The search space of hyperparameters for the baselines is shown in Table 5, and the optimal parameters for our PreferGrow under both hybrid and adaptive settings are presented in Table 6. To ensure the reliability of our findings, we have conducted a comprehensive re-evaluation of our experiments under multiple random seeds $\{100, 200, 300, 400, 500\}$ and statistical significance testing ($p < 0.001$). The results shown in Table 4 consistently confirm the significant performance gains of PreferGrow over baselines.

Table 4: Re-evaluation results (NDCG@5) under multiple random seeds.

Dataset	Beauty	Toys	Sports	Steam	MovieLens
SASRec	0.0229±.0020	0.0258±.0023	0.0104±.0016	0.0193±.0002	0.0507±.0004
PreferDiff	0.0224±.0031	0.0312±.0009	0.0122±.0007	0.0104±.0004	0.0348±.0005
PreferGrow	**0.0315±.0010**	**0.0326±.0007**	**0.0146±.0004**	**0.0399±.0004**	**0.0913±.0003**

Evaluation Protocols and Metrics: To ensure a comprehensive evaluation and mitigate potential biases, we adopt the all-rank protocol [71–73, 9, 27], which evaluates recommendations across all items. We employ two widely used ranking-based metrics: *Normalized Discounted Cumulative Gain* ($\mathbf{N}@K$) and *Mean Reciprocal Rank* ($\mathbf{M}@K$), to assess the effectiveness of the models.

G.4 Hyper-parameter Analysis

The personalization strength w is locally stable within a reasonable range and highly consistent across data splits, thus posing no practical challenge for tuning. As shown in Figure 4 of our paper, PreferGrow is sensitive to large-magnitude changes in the personalization strength parameter w (e.g., from 0 to 20). On datasets like *Steam*, we observe that performance remains locally stable within a reasonable range (e.g., $w \in [5, 15]$), but drops notably when w is set too small (*e.g.,* $w = 1$). This highlights the need to set w appropriately for each dataset. To assess sensitivity more systematically, we tested PreferGrow under $w \in \{0, 2, 5, 10\}$ and evaluated the mean absolute error between the optimal w values selected on the training, validation, and test sets. These optimal values are determined by uniformly sampling performance across 40 checkpoints throughout the training process and optimality is defined by maximizing the combined metric $HR@5 + HR@10 + NDCG@5 + NDCG@10$. Our findings in Table 7 show that the optimal w is highly consistent across data splits, indicating that w can be tuned on a validation (or even training) set like other standard hyperparameters. This ensures that tuning w does not pose a practical challenge.

G.5 Efficiency Analysis

As summarized in Table 9, we report each model's trainable parameters, the number of training epochs, GFLOPs per model output (recall that diffusion models require multiple outputs for denoising), and the total number of outputs.

Quantitative scalability analysis. We also provide a quantitative analysis of PreferGrow and clarify the solution outlined in the Limitations. PreferGrow introduces only $\mathcal{O}(N)$ complexity in the network, loss, and inference paths — comparable to contemporary diffusion-based recommenders. Table 8 contrasts model, loss, and inference complexities with representative baselines, where L is the history length, N the item set size, B the number of negatives, d the hidden dimension, and T the number of diffusion steps.

- **Model parameters.** We never materialize the full N^2 preference-ratio matrix. Equation (12) requires only N scores; the additional computation cost is thus $\mathcal{O}(Nd)$.

Table 5: Hyperparameters Search Space for Baselines.

Method	Hyperparameter Search Space
Shared	lr $\sim$ {1e-2, 1e-3, 1e-4, 1e-5} with decay 0, embedding size $d \sim$ {128, 256, 512} the number of negative sampling (if using) $\sim$ {64, 128, 256}, bath size = 256
DreamRec	$w \sim$ {0, 1, 2, 5, 10}, $T \sim$ {500, 1000, 2000,3000}, $p \sim$ {0.05, 0.1, 0.15, 0.2, 0.25, 0.3}
PreferDiff	$\lambda \sim$ {0.2, 0.4, 0.6, 0.8}, $w \sim$ {0, 1, 2, 5, 10}, $T \sim$ {500, 1000, 2000,3000}
DiffRec	noise scale $\sim$ {1e-1, 1e-2, 1e-3, 1e-4, 1e-5}, $T \sim$ {2, 5, 20, 50, 100}
DDSR	$T \sim$ {500, 1000, 2000,3000}
PreferGrow	$T \sim$ {5, 10, 20, 30, 40}, $p \sim$ {0.05, 0.1, 0.15, 0.2, 0.25, 0.3} $w \sim$ {0, 1, 2, 5, 10}, $\lambda \sim$ {0.9, 0.99, 0.999, 0.9999, 0.99999}

Table 6: Best Hyperparameters for PreferGrow on five datasets.

Variant	Dataset	lr	d	p	T	w	λ
Hybrid	**MovieLens**	1e-4	256	0.1	20	10	0.9999
	Steam	1e-3	256	0.1	20	10	0.99999
	Beauty	1e-4	256	0.1	20	5	0.999
	Toys	1e-3	256	0.2	20	10	0.9999
	Sports	1e-3	256	0.2	20	1	0.9999
Variant	**Method**	lr	d	p	T	w	x_{-1}
Adaptive	**Dataset**	1e-4	256	0.2	20	10	True
	Steam	1e-3	256	0.05	20	10	True
	Beauty	1e-4	256	0.1	20	2	True
	Toys	1e-4	256	0.2	20	5	True
	Sports	1e-4	256	0.2	20	5	True

Table 7: Mean absolute error of w across data splits on five datasets.

Dataset	MovieLens	Steam	Beauty	Sports	Toys
Mean abs. error of w	0.000	0.000	0.778	0.050	0.775

- **Loss computation.** With the idempotent fading matrix $\mathbf{E}$, the objective in Eq. (4) simplifies (Appendix C.1–C.4) to element-wise means, indexing, and a precomputable constant—overall $\mathcal{O}(N)$. The score-entropy loss contains three parts: (i) a positive term and (ii) a negative term (both simple mean/index operations, $\mathcal{O}(N)$), and (iii) a constant term ensuring non-negativity that depends only on t and can be precomputed in $\mathcal{O}(T)$ with T=20.
- **Inference cost.** Idempotency yields

$$\mathbf{E}\, s_\Theta(x_t, t, u) \;=\; \Big(\frac{\mathbf{p}_T^\top s_\Theta}{\mathbf{1}^\top \mathbf{p}_T}\Big)\mathbf{1},$$

  reducing Equation (19) from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$.

Where $\mathcal{O}(N^2)$ comes from. The apparent $\mathcal{O}(N^2)$ arises only from the *modeling target*: preference ratios are more expressive than conventional logits, which raises PreferGrow's performance ceiling but does not inflate the network/algorithmic path beyond $\mathcal{O}(N)$.

Scaling to industry scale. When N is extremely large (*e.g.,* billions of items), even $\mathcal{O}(N)$ becomes impractical. As discussed in the Limitations, Semantic IDs (SIDs) offer a principled route to reduce complexity. SIDs encode each item with m codebooks of size c, enabling up to c^m distinct items while keeping computation at $\mathcal{O}(mc)$; since $c^m \gg N \gg mc$, this yields a compact yet expressive representation space and allows PreferGrow-on-SIDs to reduce cost from $\mathcal{O}(N)$ to $\mathcal{O}(mc)$. A practical blocker is that current SID pipelines (e.g., Tiger [74], ActionPiece [75], DDSR [35], OneRec [76]) are not open-sourced. Truly deploying PreferGrow at industrial scale therefore requires

Table 8: Comparison of model complexities.

Complexity	Model Parameters	Loss Computation	Modeling Target	Inference
SASRec	$\mathcal{O}(Ld^2 + L^2d)$	$\mathcal{O}(Bd)$	$\mathcal{O}(N)$	$\mathcal{O}(Ld^2 + L^2d + Nd)$
DreamRec	$\mathcal{O}((L+3)d^2 + L^2d)$	$\mathcal{O}(d)$	$\mathcal{O}(Td)$	$\mathcal{O}(T(L+3)d^2 + TL^2d + Nd)$
PreferDiff	$\mathcal{O}((L+3)d^2 + L^2d)$	$\mathcal{O}(Bd)$	$\mathcal{O}(Td)$	$\mathcal{O}(T(L+3)d^2 + TL^2d + Nd)$
DiffRec	$\mathcal{O}(LN^2)$	$\mathcal{O}(N)$	$\mathcal{O}(TN)$	$\mathcal{O}(TLN^2)$
DDSR	$\mathcal{O}(Ld^2 + L^2d)$	$\mathcal{O}(N)$	$\mathcal{O}(TN)$	$\mathcal{O}(TLd^2 + TL^2d + TNd)$
PreferGrow	$\mathcal{O}((L+3)d^2 + L^2d)$	$\mathcal{O}(N+T)$	$\mathcal{O}(TN^2)$	$\mathcal{O}(T(L+3)d^2 + TL^2d + TN(d+3))$

Table 9: Comparison of efficiency on Steam.

Models	# Trainable Parameters	# training epochs	Inference GFLOPs	Inference steps
SASRec	2.70M	61	0.85	1
Caser	2.49M	58	0.38	1
GRURec	2.77M	55	4.06	1
DreamRec	7.25M	52	0.85	20
PreferDiff	7.25M	55	0.85	20
DiffRec	18.55M	1000	/	20
DDSR	3.03M	142	1.77	20
PreferGrow	3.03M	480	0.93	20

addressing the open challenge of *large-scale multimodal SID pretraining*. Extending PreferGrow to operate over SIDs is our ongoing work.