Flow Matching Based Sequential Recommender Model

Feng Liu[1] , **Lixin Zou**[1*] , **Xiangyu Zhao**[2] , **Min Tang**[3] , **Liming Dong**[4] ,
Dan Luo[5] , **Xiangyang Luo**[6*] and **Chenliang Li**[1]

[1]Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education,
School of Cyber Science and Engineering, Wuhan University
[2]City University of Hong Kong,
[3]Monash University,
[4]National Defense University,
[5]Lehigh University
[6]State Key Lab of Mathematical Engineering and Advanced Computing
{liufeng.tanh, zoulixin}@whu.edu.cn, xianzhao@cityu.edu.hk, min.tang@monash.edu,
dlm14@tsinghua.org.cn, danluo.ir@gmail.com, xiangyangluo@126.com, cllee@whu.edu.cn

Abstract

Generative models, particularly diffusion model, have emerged as powerful tools for sequential recommendation. However, accurately modeling user preferences remains challenging due to the noise perturbations inherent in the forward and reverse processes of diffusion-based methods. Towards this end, this study introduces FMRec, a Flow matching based model that employs a straight flow trajectory and a modified loss tailored for the recommendation task. Additionally, from the diffusion-model perspective, we integrate a reconstruction loss to improve robustness against noise perturbations, thereby retaining user preferences during the forward process. In the reverse process, we employ a *deterministic reverse sampler*, specifically an ODE-based updating function, to eliminate unnecessary randomness, thereby ensuring that the generated recommendations closely align with user needs. Extensive evaluations on four benchmark datasets reveal that FMRec achieves an average improvement of **6.53%** over state-of-the-art methods. The replication code is available at https://github.com/FengLiu-1/FMRec.

1 Introduction

Diffusion model (DM), owned to their great ability to generate high-quality image [Nichol and Dhariwal, 2021; Song *et al.*, 2020], video [Ho *et al.*, 2022; Harvey *et al.*, 2022; Yu *et al.*, 2024] and text [Gong *et al.*, 2022; Wu *et al.*, 2023], has inspired the development of innovative adaptations in sequential recommendation systems (*e.g.*, DiffuRec [Li *et al.*, 2023] and DiffRec [Wang *et al.*, 2023]). Usually, the diffusion-based model consists of two main phases: the forward procedure and the reverse procedure. During the forward procedure of diffusion model, *i.e.*, the **training pro-**
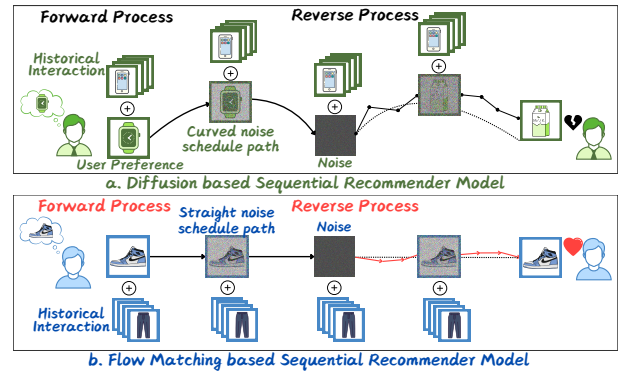

Figure 1: An illustration that highlights the differences between Diffusion-based (a) and Flow Matching based (b) sequential recommender models in both the forward and reverse processes.

cedure, the model progressively adds noise to the real data based on a predefined *noise schedule*, eventually transforming it into random noise resembling that drawn from a normal distribution. In contrast, the **reverse procedure**, or the inference stage, iteratively removes the noise from the sampled noise using a *reverse samplers*, *i.e.*, the SDE-based stochastic reverse sampler [Nakkiran *et al.*, 2024]. This process is typically conditioned on both the random noise and additional inputs, allowing for the generation of realistic samples.

Following this paradigm, methods such as DiffuRec [Li *et al.*, 2023], DreamRec [Yang *et al.*, 2024], and DimeRec [Li *et al.*, 2024b] have extended the diffusion model to the sequential recommendation. Specifically, these approaches generate next-item predictions by leveraging both random noise and user-item interactions. In the forward procedure, illustrated in Figure 1(a), these models progressively add noise to the target recommendation, transforming the actual next item into random noise. After integrating the random noise and historical interactions using a deep learning model, the reverse process utilizes a *stochastic reverse sampler* to progressively denoise the next item from the noise-perturbed user preferences.

Although effective, existing methods exhibit several limi-

---
*Corresponding Author.

tations: **(1) Inaccurate user preference modeling**: The integration of random noise during the forward process, along with user preferences, can indeed compromise the accuracy of user preference modeling. Additionally, these methods [Li *et al.*, 2023; Wang *et al.*, 2023] typically utilize a *curved noise schedule path* during the noise addition process, which can result in error accumulation due to their long-curved trajectory, as illustrated by the curve in Figure 1(a). Consequently, during the reverse inference process, existing methods need to fit these curved paths, necessitating a greater number of diffusion steps to counteract the impact of these errors effectively. While existing methods might be aware of this issue, they often address it by selecting hyperparameters that introduce minimal random noise into user preference modeling. However, this merely treats the symptoms rather than resolving the underlying problem. **(2) Randomness on recommendation generation**: The *stochastic reverse sampler* used in reverse procedures introduces randomness into the recommendation generation process, potentially resulting in irrelevant suggestions. These stochastic samplers typically introduce extra noise-perturbations during the sampling phase, yielding diverse and varied samples that are beneficial in tasks like image generation (*e.g.*, creating different cat breeds such as Ragdolls, Persians, and Folded-ear cats). However, in sequential recommendation systems, the primary goal is to accurately predict the next likely item while exploring diverse yet relevant topics. Unfortunately, these unintended perturbations often lead to irrelevant recommendations, which can ultimately degrade the user experience. In the example shown in Figure 1(a), such perturbations can shift recommendations from "watch" to "milk" and negatively impact users' preferences on the actual platform.

Towards this end, this work firstly adapts Flow Fatching, *i.e.*, a simplified diffusion model, for the sequential recommendation and proposes FMREC. Specifically, in the forward process, we utilize a straight flow trajectory and derive a noise-free equivalent learning target for the sequential recommendation (the theoretical analysis of straight trajectory's advantage is provided in Appendix C). Thereby, FMREC minimizes error accumulation and achieves more precise recommendations. Additionally, we introduce a decoder architecture to reconstruct users' historical preferences, along with a corresponding *interaction information reconstruction loss* during the training process to enhance the model's robustness against noise perturbations. To control randomness in item generation, we implement a *deterministic reverse sampler* using an ODE-based deterministic sampling method, effectively eliminating random perturbations during inference. Figure 1(b) illustrates our proposed method. Finally, we conduct extensive experiments on four benchmark datasets and compare FMREC with state-of-the-art (SOTA) approaches. An average improvement of **6.53%** over SOTA verifies the effectiveness of our proposed methods.

2 Related Work

2.1 Sequential Recommendation Systems

The rapid advancement of deep learning has significantly enhanced sequential recommendation systems [Zou *et al.*,

2019; Tang *et al.*, 2024] through various architectures. Early pioneering works employ Recurrent Neural Network (RNN) [Donkers *et al.*, 2017], such as GRU4Rec [Hidasi *et al.*, 2015], RRN [Wu *et al.*, 2017], and Convolutional Neural Networks (CNN) [de Souza Pereira Moreira *et al.*, 2018], including RCNN [Xu *et al.*, 2019] and Caser [Tang and Wang, 2018], which effectively capture user preferences and immediate interests. More recent approaches have enhanced sequence modeling using self-attention mechanisms [Zou *et al.*, 2020; Tang *et al.*, 2025; Yu *et al.*, 2025], particularly through the transformer architecture [Vaswani, 2017]. Models like SASRec [Kang and McAuley, 2018], BERT4Rec [Sun *et al.*, 2019], and STOSA [Fan *et al.*, 2022] utilize self-attention to enhance performance on user interaction data, where SASRec focuses on sequential user behavior, BERT4Rec employs bidirectional self-attention with the Cloze objective for richer feature representation, and STOSA introduces uncertainty in capturing dynamic preferences using Wasserstein Self-Attention.

These methods discussed above form the backbone of sequential recommendation and are orthogonal to our proposed approach. Our work also uses a transformer-based architecture as foundation. Ideally, our proposed method is designed to be seamlessly integrated with all of these techniques.

2.2 Generative Recommender Systems

Generative recommender systems have gained significant attention due to their ability to model complex user-item interactions and generate diverse, innovative recommendations. Early works like AutoRec [Sedhain *et al.*, 2015] apply autoencoder to collaborative filtering, while models like AutoSeqRec [Liu *et al.*, 2023] and MAERec [Ye *et al.*, 2023] enhance them with incremental learning and graph representations, boosting robustness to noisy, sparse data. Variational Autoencoder (VAE) introduces probabilistic latent space for modeling sparse user-item interactions, with innovations like dual disentanglement modules [Guo *et al.*, 2024] and hierarchical priors [Li *et al.*, 2024a] improving interpretability and addressing sparsity. Another popular direction is the use of Generative Adversarial Network (GAN), which enhances generative recommendation through adversarial training between generators and discriminators. Combined with traditional collaborative filtering [Dervishaj and Cremonesi, 2022], GAN captures complex user preferences and integrates them with techniques like Determinantal Point Processes (DPP)[Wu *et al.*, 2019] to improve recommendation diversity. To address challenges like mode collapse, a newer GAN model [Jiangzhou *et al.*, 2024] incorporates diffusion model for stable training and reliable recommendations.

While these methods advance generative recommendation by a large margin, our research focuses on tailoring more advanced diffusion model to generate even more diverse and innovative recommendations.

Diffusion-Based Recommendation The usage of diffusion model in sequential recommendation is still in its early stages but rapidly gaining attention due to their success in various generative tasks [Ho *et al.*, 2022; Gong *et al.*, 2022]. Pioneering efforts like [Li *et al.*, 2023; Yang *et al.*, 2024] add noise

to the target item and leverage user interaction history implicitly in the reverse process. Based on this, Huang *et al.* [2024] integrates both historical interactions and target items during noise addition, using both explicit sequence embeddings and implicit attention mechanisms to boost preference representation. Meanwhile, Wang *et al.* [2024] harnesses a Transformer as a conditional denoising decoder, embedding historical interactions into the model via cross-attention, thereby effectively guiding the denoising and enabling the model to focus on pertinent historical interactions.

Although these approaches are effective, they overlook the biases introduced by noise in diffusion model for recommendation task. In contrast, our model focuses on mitigating the distortion of user preferences caused by perturbations in both the forward and reverse diffusion processes.

3 Preliminaries

This section provides a brief overview of the Flow Matching to establish the necessary background. Particularly, Flow Matching can be considered a simplied diffusion model, designed to construct probabilistic path between two distinct distributions, thereby enabling the transformation from simple distribution, *e.g.,* a simple normal distribution $p_n = \mathcal{N}(0, I)$, to the complex and unknown distribution, denoted as p_c [Esser *et al.*, 2024; Nakkiran *et al.*, 2024]. From the perspective of diffusion model, Flow Matching can also divide into the forward and reverse procedure.

Forward Procedure In the forward procedure, the model is required to map a real data point $\boldsymbol{x}_c \sim p_c$ to a noise data point $\boldsymbol{x}_n \sim p_n$. It is defined as a time-dependent flow ϕ as

$$\phi_t(\boldsymbol{x}_c|\boldsymbol{x}_n) : \boldsymbol{x}_c \mapsto a_t\boldsymbol{x}_c + b_t\boldsymbol{x}_n \qquad (1)$$

where t is a random variable uniformly sampled from the interval $[0, 1]$. The time-dependent hyperparameters a_t and b_t follow two constraints: if $a_0 = 1$, then $b_0 = 0$, ensuring $\phi_0(\boldsymbol{x}_c|\boldsymbol{x}_n) = \boldsymbol{x}_c$ at $t = 0$, and if $a_1 = 0$, then $b_1 = 1$, ensuring $\phi_1(\boldsymbol{x}_c|\boldsymbol{x}_n) = \boldsymbol{x}_n$ at $t = 1$. These ensure a transition from the target distribution $\boldsymbol{x}_c$ to the normal distribution $\boldsymbol{x}_n$ over time. $\phi_t(\boldsymbol{x}_c|\boldsymbol{x}_n)$ provides a concise representation of the Flow Trajectory, illustrating the manner in which states change during the Flow process. To simplify the notation, we denote the intermediate variable perturbed by noise as

$$\boldsymbol{z}_t = a_t\boldsymbol{x}_c + b_t\boldsymbol{x}_n. \qquad (2)$$

To characterize the flow $\phi_t(\cdot|\boldsymbol{x}_n)$, a vector field u_t is employed to construct the time-dependent diffeomorphic map ϕ_t as follows:

$$\boldsymbol{u}_t(\boldsymbol{z}_t|\boldsymbol{x}_n) := \phi_t'(\phi_t^{-1}(\boldsymbol{z}_t|\boldsymbol{x}_n)|\boldsymbol{x}_n), \qquad (3)$$

where $\phi_t^{-1}(\boldsymbol{z}_t|\boldsymbol{x}_n)$ represents the inverse function, which computes $\boldsymbol{x}_c$ based on the perturbed noise. The notation ϕ_t' denotes the differential of ϕ_t.

The objective of the training process is to learn and predict this vector field with a Θ **parameterized vector field** $\boldsymbol{v}_\Theta(\boldsymbol{z}_t, t)$ as

$$\mathcal{L}_{FM} = \mathbb{E}_{t, p_t(\boldsymbol{z}|\boldsymbol{x}_c), p(\boldsymbol{x}_c)} ||\boldsymbol{v}_\Theta(\boldsymbol{z}_t, t) - u_t(\boldsymbol{z}_t|\boldsymbol{x}_n)||_2^2, \quad (4)$$

where $|| \cdot ||_2^2$ is the L2-norm.

Reverse Procedure In the reverse process, *i.e.*, the inference procedure, the model reconstructs $\boldsymbol{x}_c$ by solving the following ordinary differential equation (ODE):

$$d\boldsymbol{z}_t = -\boldsymbol{v}_\Theta(\boldsymbol{z}_t, t)\, dt,$$

where $\boldsymbol{z}_t$ is the linear combination of $\boldsymbol{x}_c$ and $\boldsymbol{x}_n$. In this work, we employ a *deterministic reverse sampler*, *i.e.*, the Euler method, to solve this ODE.

Remarkably, the diffusion-based recommendation models [Li *et al.*, 2023] and [Wang *et al.*, 2023], typically employ an SDE-based *stochastic reverse sampler*, which introduces the stochastic disturbances, *i.e.*, the variance, to the reverse diffusion process. However, this stochasticity deviates from the objective of sequential recommendation, potentially resulting in irrelevant predictions when attempting to accurately identify a user's next interaction item, which ultimately undermines the user experience. Contrarily, the ODE-based *deterministic reverse sampler* does not introduce any random perturbations during the generation process, ensuring that the generation results meet the user's personalized preferences.

4 Methodology

4.1 Problem Statement

For the sequential recommendation task, we define a set of users $\mathcal{U} = \{u_1, u_2, \ldots, u_{|\mathcal{U}|}\}$ and a set of items $\mathcal{I} = \{i_1, i_2, \ldots, i_{|\mathcal{I}|}\}$, where $|\mathcal{U}|$ and $|\mathcal{I}|$ represent the total number of users and items, respectively. Each user $u \in \mathcal{U}$ has an interaction history represented as a chronological sequence of items $\mathcal{S} = (i_1, i_2, \ldots, i_m)$, where i_m corresponds to the m-th item interacted with by the user. Here, m is the length of the interaction sequence. Formally, we aim at generating the next recommendation i_{m+1}, maximizing a specific metric Θ as

$$i_{m+1} = \underset{i_{m+1}}{\arg\max} \, \Theta(i_{m+1}|\mathcal{S}). \qquad (5)$$

4.2 Flow Matching based Sequential Recommender Model

This section offers a detailed explanation of both the forward and reverse processes in FMREC. In the forward process, our designs include the development of flow trajectories, modification of the learning target, design of the parameterized vector field, and the corresponding learning loss. In the reverse process, we discuss the specific selection and implementation of the reverse sampler.

Forward Process

In the context of sequence recommendation task, we adopt the target distribution p_c in Flow Matching as follows:

$$p(\boldsymbol{x}_c|\boldsymbol{e}_{m+1}) = \mathcal{N}(\boldsymbol{x}_c; \boldsymbol{e}_{m+1}, 0), \qquad (6)$$

where $\boldsymbol{e}_{m+1}$ represents the embedding of the target item, generated using the equation:

$$\boldsymbol{e}_{m+1} = \texttt{Embedding}(i_{m+1}). \qquad (7)$$

Here, $\texttt{Embedding}$ denotes an embedding module that transforms discrete next item ID i_{m+1} into dense vector representation. Notably, the parameter $\boldsymbol{e}_{m+1}$ is trainable, which is different from the traditional Flow Matching settings.
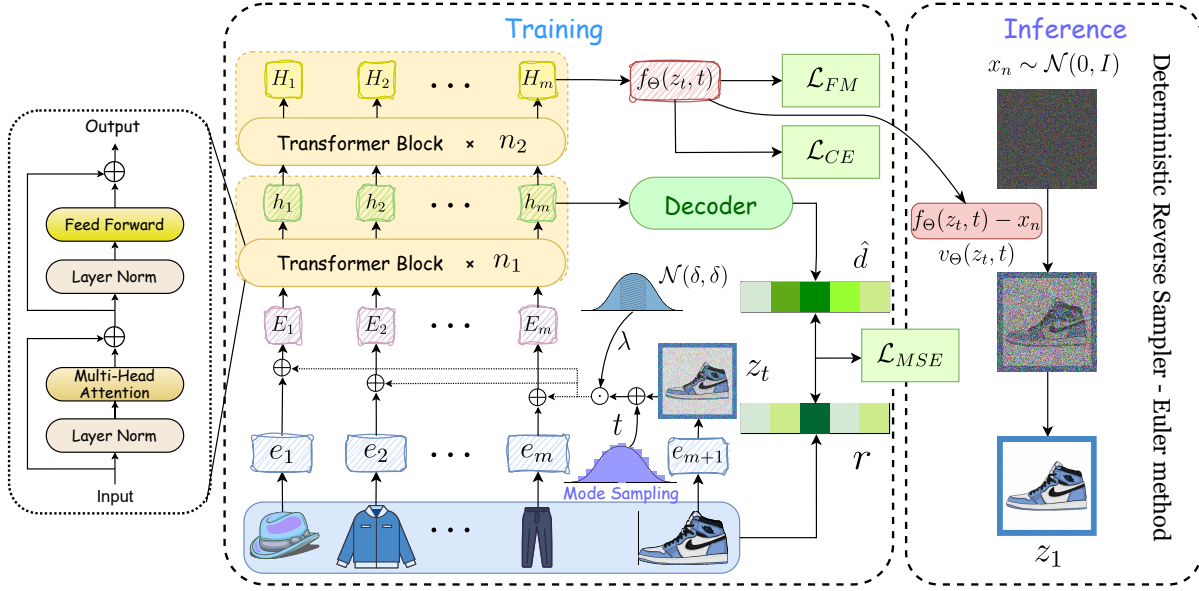
Figure 2: The framework of the FMREC. In the training process, our design incorporates the development of straight flow trajectories, modifications to the learning target $\mathcal{L}_{FM}$, the design of a decoder-based model, and the implementation of regularized loss functions $\mathcal{L}_{CE}$ and $\mathcal{L}_{MSE}$. In the inference process, we present a deterministic reverse sampler that generates recommendations.

Straight Trajectory Flow and Modified Loss The time-dependent hyper-parameters a_t and b_t in the forward process define the trajectory of the generated flow, enabling the flexible selection of flow paths to control the flow process. The Straight Trajectory, characterized by $a_t = (1 - t)$ and $b_t = t$, is known for its simplicity and computational efficiency, making it widely employed in various studies [Lipman *et al.*, 2022; Liu *et al.*, 2022]. Following this setting, we define the time-dependent flow of the next item recommendation as follows:

$$\boldsymbol{z}_t = (1 - t)\boldsymbol{x}_c + t\boldsymbol{x}_n, \quad \text{where } \boldsymbol{x}_n \sim \mathcal{N}(0, I), \quad (8)$$

where I denotes the identity matrix. Further discussions regarding its effectiveness are provided in Section 5.3. The variable t is sampled using the Mode Sampling with Heavy Tails [Esser *et al.*, 2024] method as

$$t = g(k; s) = 1 - k - s \times \left(\cos^2 \left(\frac{\pi}{2} k \right) - 1 + k \right), \quad (9)$$

where the parameter s represents a scaling factor that governs the density distribution of the time step sampling. $k \in \mathcal{U}(0, 1)$ is a random variable. Further discussions on the sampling methods and the impact of the parameter s on model performance are provided in Appendix B.

To characterize the flow $\phi_t(\cdot|\boldsymbol{x}_n)$, a vector field can be defined in the following form:

$$\boldsymbol{u}_t(\boldsymbol{z}_t|\boldsymbol{x}_n) = -\phi_t^{-1}(\boldsymbol{z}_t|\boldsymbol{x}_n) + \boldsymbol{x}_n, \quad (10)$$

where $\phi_t^{-1}(\boldsymbol{z}_t|\boldsymbol{x}_n)$ represents the inverse of the flow function at time t, conditioned on $\boldsymbol{x}_n$. By substituting into the Flow Matching loss defined in Equation (4), we can reformulate the training loss as follows:

$$\mathcal{L}_{FM} = \mathbb{E}_{t, p_t(\boldsymbol{z}|\boldsymbol{x}_c), p(\boldsymbol{x}_c)} ||\boldsymbol{v}_\Theta(\boldsymbol{z}_t, t) - (-\phi_t^{-1}(\boldsymbol{z}_t|\boldsymbol{x}_n) + \boldsymbol{x}_n)||_2^2 \quad (11)$$

$$= \mathbb{E}_{t, p_t(\boldsymbol{z}|\boldsymbol{x}_c), p(\boldsymbol{x}_c)} ||(-f_\Theta(\boldsymbol{z}_t, t) + \boldsymbol{x}_n) - (-\boldsymbol{x}_c + \boldsymbol{x}_n)||_2^2 \quad (12)$$

$$= \mathbb{E}_{t, p_t(\boldsymbol{z}|\boldsymbol{x}_c), p(\boldsymbol{x}_c)} ||f_\Theta(\boldsymbol{z}_t, t) - \boldsymbol{x}_c||_2^2. \quad (13)$$

The first derivation relies on the fact $\boldsymbol{x}_n = \phi_t^{-1}(\boldsymbol{z}_t|\boldsymbol{x}_n)$. The second equation replaces the learning target $\boldsymbol{v}_\Theta(\boldsymbol{z}_t, t)$ with $(-f_\Theta(\boldsymbol{z}_t, t) + \boldsymbol{x}_n)$, leading to the modified target in Equation (13). The reason for replacing the learning target is that predicting the intricate combination of real data and noise can be difficult and detrimental to recommendation systems, leading to the recommendation of diverse but irrelevant items.

Parameterized Vector Field To parameterize $f_\Theta(\boldsymbol{z}_t, t)$ for predicting real data, we first integrate the historical interaction sequence with the noised data and then model the noised historical interactions using a robust transformer decoder. Specifically, we combine the historical interaction sequence with the noised data as follows [Li *et al.*, 2023]:

$$\boldsymbol{E}_i^t = \boldsymbol{e}_i + \lambda_i \odot (\boldsymbol{z}_t + t), \quad \lambda_i \sim \mathcal{N}(\delta, \delta), \quad (14)$$

where $\odot$ denotes element-wise multiplication, λ_i is sampled from a Gaussian distribution, and δ is a hyperparameter representing both the mean and variance of the distribution. The term λ_i is instrumental in balancing the fusion ratio between the historical interaction sequence and the noised data.

Afterwards, we employ a decoder, denoted as $\texttt{Decoder}_1$, which consists of n_1 layers of unidirectional self-attention based transformer, to obtain the hidden states as follows:

$$[\boldsymbol{h}_1, \dots, \boldsymbol{h}_m] = \texttt{Decoder}_1([\boldsymbol{E}_1^t, \dots, \boldsymbol{E}_m^t]), \quad (15)$$

where $\boldsymbol{h}_1$ to $\boldsymbol{h}_m$ are the intermediate outputs that will be utilized for reconstructing the interaction matrix and calculating

the reconstruction loss. Next, we model $f_\Theta(\boldsymbol{z}_t, t)$ by taking the `LAST` token output from an additional n_2-layer decoder, denoted as `Decoder`$_2$, as follows:

$$f_\Theta(\boldsymbol{z}_t, t) = \texttt{LAST}(\texttt{Decoder}_2([\boldsymbol{h}_1, \ldots, \boldsymbol{h}_m])). \quad (16)$$

Regularized Losses The modified Flow Matching loss presented in Equation (13) is not entirely suitable for recommendation task because the vector $\boldsymbol{x}_c$ is trainable rather than fixed. This might lead to the problem that different $\boldsymbol{x}_c$ converge to a single embedding, which is obviously meaningless. To mitigate this issue, we introduce a cross-entropy loss $\mathcal{L}_{CE}$ as a regularizer that differentiates between various item embeddings, thereby preventing the aforementioned problem as:

$$\mathcal{L}_{CE} = -\log \hat{y}_{m+1} \quad (17)$$

$$\hat{y}_{m+1} = \frac{\exp(f_\Theta(\boldsymbol{z}_t, t) \cdot \boldsymbol{e}_{m+1})}{\sum_{i \in \mathcal{I}} \exp(f_\Theta(\boldsymbol{z}_t, t) \cdot \boldsymbol{e}_i)}. \quad (18)$$

where $\hat{y}_{m+1}$ is the normalized score of recommending i_{m+1}.

Furthermore, to mitigate the detrimental effects of noise perturbation on model performance, we incorporate an additional reconstruction loss that interprets the user's history from the final token of the hidden representation $\boldsymbol{h}_m$ as follows:

$$\hat{\boldsymbol{d}} = \texttt{Decoder}_\omega(\boldsymbol{h}_m), \quad (19)$$

where $\hat{\boldsymbol{d}} \in \mathbb{R}^{|\mathcal{I}|}$ is the predicted user-item interaction information. ω represents the parameters of the MLP-based `Decoder`$_\omega$. We optimize these parameters using a Mean Squared Error (MSE) loss, enabling the generation of more accurate and personalized recommendations as follows:

$$\mathcal{L}_{MSE} = \|\hat{\boldsymbol{d}} - \boldsymbol{r}\|^2, \quad (20)$$

where $\boldsymbol{r} \in \mathbb{R}^{|\mathcal{I}|}$ is a binary vector, with 1 indicating an interaction between the user and the item, and 0 indicating no interaction.

Finally, the loss function utilized during training is formulated by integrating three distinct components, represented as follows:

$$\mathcal{L} = \mathcal{L}_{FM} + \alpha \mathcal{L}_{CE} + \beta \mathcal{L}_{MSE}, \quad (21)$$

where α and β are hyperparameters that govern their relative importance. A more detailed training procedure of FMREC is presented in Appendix A.

Reverse Process

In the reverse process, we employ a *deterministic reverse sampler* to model the generative process, thereby mitigating the errors introduced by the *stochastic reverse sampler* in the diffusion-based recommendation system. Specifically, the *deterministic reverse sampler* is defined as the updating function for the following ordinary differential equation:

$$d\boldsymbol{z}_t = -u_t(\boldsymbol{z}_t | \boldsymbol{x}_n) dt = (f_\Theta(\boldsymbol{z}_t, t) - \boldsymbol{x}_n) dt. \quad (22)$$

To solve this equation, we iteratively apply the Euler method to compute the point transformations guided by the vector field, as given by

$$\boldsymbol{z}_{t+\Delta t} = \boldsymbol{z}_t + (f_\Theta(\boldsymbol{z}_t, t) - \boldsymbol{x}_n)\Delta t, \quad (23)$$

Dataset	# Users	# Items	# Actions	Sparsity
Beauty	22,361	12,101	198,502	99.93%
Steam	281,428	13,044	3,485,022	99.90%
Movielens-100k	943	1,682	100,000	93.70%
Yelp	28,298	59,951	1,764,589	99.90%

Table 1: The statistics of the datasets.

where Δt is determined by a custom number of Euler method steps, denoted as q, with $\Delta t = \frac{1}{q}$. This process is iteratively calculated from $t = 0$ to $t = 1$, resulting in $\boldsymbol{z}_1$ as the fully denoised generation $\hat{\boldsymbol{x}}_c$, which represents the corresponding item embedding. The inference procedure of FMREC is presented in Appendix A.

5 Experiment

This section presents comprehensive experiments to demonstrate the effectiveness of FMREC.

5.1 Experimental Settings

Dataset We evaluate FMREC's effectiveness using four widely recognized publicly available datasets: **(1)** *Amazon Beauty* [Ni *et al.*, 2019] contains global purchasing interactions and user reviews for beauty products on the Amazon platform, documenting the purchasing history of 22,361 users across 12,101 products. **(2)** *Steam* is a leading PC game distribution platform with 3,480,000 interaction records from 280,000 gamers; **(3)** *Movielens-100k* [Harper and Konstan, 2015] is a widely used benchmark dataset in sequential recommendation research, providing ratings from 943 users on 1,682 movies from the Movielens platform; and **(4)** *Yelp* is a popular review site featuring user reviews and 1,764,589 ratings for various businesses, including restaurants, entertainment venues, and hotels. The statistics of the dataset are provided in Table 1.

Baselines We compare FMREC against both widely adopted and recently introduced baseline models: **(1) GRU4Rec** utilizes GRU to capture in-session behavioral patterns and predict subsequent user item preferences [Hidasi *et al.*, 2015]; **(2) Caser** leverages convolutional neural networks to map user action sequences into both temporal and latent spaces [Tang and Wang, 2018]; **(3) SASRec** introduces a decoder architecture for sequential recommender model that effectively captures long-term dependencies in user behavior [Kang and McAuley, 2018]; **(4) BERT4Rec** employs a bidirectional Transformer architecture, coupled with a Cloze task for training, to effectively learn users' dynamic preferences [Sun *et al.*, 2019]; **(5) STOSA** utilizes Wasserstein attention to introduce a degree of uncertainty within its model, allowing for the accurate representation of evolving user preferences [Fan *et al.*, 2022]; **(6) AutoSeqRec** leverages a multi-autoencoder framework to fuse long-term user preferences and short-term interests for sequential recommendation [Liu *et al.*, 2023]; **(7) DreamRec** achieves direct generation of personalized oracle item embeddings through guided diffusion model [Yang *et al.*, 2024]; **(8) DiffuRec** models items as distributions using diffusion model, capturing multi-faceted content and user preferences [Li *et al.*, 2023].

Dataset	Metric	GRU4Rec	Caser	SASRec	BERT4Rec	STOSA	AutoSeqRec	DreamRec	DiffuRec	FM4Rec	$\Delta\%$
Beauty	HR@5	1.0112	1.6188	3.2688	2.1326	3.8814	4.9628	4.9833	5.3943	5.8373	8.21%
	HR@10	1.9370	2.8166	6.2648	3.7160	6.1262	7.1016	6.9821	7.8374	8.2693	5.51%
	HR@20	3.8531	4.4048	8.9791	5.7922	9.0954	9.3342	9.4531	10.9358	11.626	6.31%
	NDCG@5	0.6084	0.9758	2.3989	1.3207	2.4859	3.3186	3.2507	3.9153	4.1631	6.33%
	NDCG@10	0.9029	1.3602	3.2305	1.8291	3.2053	4.0157	3.9769	4.6971	4.9461	5.30%
	NDCG@20	1.3804	1.7595	3.6563	2.3541	3.9491	5.0133	4.9860	5.4784	5.7876	5.64%
Steam	HR@5	3.0124	3.6053	4.7428	4.7391	4.8546	5.0021	5.1267	6.0073	6.5254	8.62%
	HR@10	5.4257	6.4940	8.3763	7.9448	8.5870	8.7741	8.9875	9.8437	10.5908	7.59%
	HR@20	9.2319	10.9633	13.6060	12.7322	14.1107	14.6752	15.0871	15.3817	16.4669	7.06%
	NDCG@5	1.8293	2.1586	2.8842	2.9708	2.9220	3.0912	3.1507	3.8109	4.1878	9.89%
	NDCG@10	2.6033	3.0846	4.0489	4.0002	4.1191	4.4729	4.6416	5.0429	5.4925	8.91%
	NDCG@20	3.5572	4.2043	5.3630	5.2027	5.5072	5.9823	5.9701	6.4340	6.9689	8.31%
Movielens-100k	HR@5	7.7412	6.0438	6.5748	5.0901	8.0148	8.7320	7.3692	7.5209	9.3338	6.89%
	HR@10	12.1951	11.2426	13.5737	9.3319	13.6542	14.6641	12.4377	12.8501	15.4934	5.66%
	HR@20	21.8451	19.5189	22.6935	16.8611	21.7761	22.8724	20.8357	19.4127	24.3146	6.31%
	NDCG@5	4.5982	3.3721	4.1333	3.0850	4.9721	5.5010	4.2503	4.6969	5.6848	3.34%
	NDCG@10	6.0326	5.0683	6.3427	4.4568	5.2159	7.3955	5.9837	6.4136	7.6571	3.53%
	NDCG@20	8.4727	7.1439	8.6340	6.3442	8.3302	9.4584	7.8234	8.0459	9.8158	3.78%
Yelp	HR@5	2.4560	2.0956	2.8389	2.2465	1.9360	OOM	1.7486	3.0390	3.3084	8.86%
	HR@10	4.2335	3.7140	4.8569	4.0581	3.3858	OOM	1.9362	5.075	5.4421	7.23%
	HR@20	7.4952	6.6189	8.2656	7.0433	5.7285	OOM	3.5873	8.5447	9.0631	6.07%
	NDCG@5	1.5588	1.3108	1.8301	1.4027	1.2100	OOM	1.1740	1.9868	2.1174	6.57%
	NDCG@10	2.1269	1.8311	2.5466	1.9732	1.6728	OOM	1.5268	2.6352	2.7855	5.71%
	NDCG@20	2.9431	2.5607	3.3144	2.7233	2.2584	OOM	2.3641	3.4395	3.618	5.19%

Table 2: Overall performance comparison across four benchmark datasets, presented as percentages (%). We highlight the highest-performing metric values in **bold** and the second-best values in underlined. The symbol Δ indicates the relative performance improvement of FMREC compared to the best baseline model. OOM refers to the out-of-memory problem.

Evaluation Protocol Following the procedures in [Sun *et al.*, 2019; Li *et al.*, 2023], we split user interaction sequences into three parts: the first $m-2$ sequences formed the training set, while i_{m-1} and i_m served as targets for the validation and test sets, respectively. We evaluated performance using the metrics HR@K (Hit Rate@K) and NDCG@K (Normalized Discounted Cumulative Gain@K). Each baseline model generates a ranked list of items predicted for the next interaction based on user history, considering all dataset items as candidates, with K values set at $\{5, 10, 20\}$.

Implementation Details The implementation details are as follows: both Decoder_1 and Decoder_2 consist of 2 layers, with a hidden dimension of 128 and 4 attention heads. The item embedding dimension is also set to 128. The Decoder_w is a three-layer MLP with tanh activations, featuring layer sizes of $\{128, 512, 2048, |\mathcal{I}|\}$, mapping 128-dimensional decoder outputs to the number of items. Hyperparameters include a batch size of 512, a learning rate of 0.001, and a maximum user interaction sequence length of 50. The loss weighting parameters α and β are set to 0.2 and 0.4, respectively. The scaling parameter s in the timestep schedule is set to 1.0. Besides, we use 30 Euler integration steps for generation. All experiments are conducted on a server with two Intel XEON 6271C processors, 256 GB of memory, and four NVIDIA RTX 3090 Ti GPUs.

5.2 Overall Performance Comparison

The experimental results and performance comparisons with baseline models are presented in detail in Table 2. From the table, we have the following observations: **(1) The FMREC model shows a notable advantage over existing SOTA methods across four benchmark datasets.** Significant improvements are consistently observed in the Beauty, Yelp, Movielens-100k, and Steam datasets with HR increasing by $5.5 - 8.9\%$ and NDCG rising by $3.3 - 9.9\%$; **(2) FM-REC can effectively eliminate the negative influence of noise perturbations associated with diffusion-based recommendation methods.** Specifically, we observe that FM-REC outperforms both DreamRec and DiffuRec, achieving superior results with an average improvement of 12.44% on HR@5, which confirms that the *deterministic reverse sampler* and regularized loss are beneficial for generating more accurate recommendation. **(3) Generative models demonstrate greater effectiveness in capturing user preferences and providing enhanced recommendations.** Particularly, DreamRec, DiffuRec, and FMREC show marked improvements in performance over SOTA traditional sequential recommender models by large margins of 13.7%, 27.93%, and 34.98% on HR@5.

5.3 Analytical Experiment

This section evaluates the effectiveness of each design option in FMREC and analyzes the impact of hyperparameter configurations on model performance. We present results of *Beauty* and *Movielens-100k* as examples. Additional results are available in Appendix B.

Influence of Flow Matching Loss In this work, we have modified the Flow Matching loss from directing predicting the overall vector field, *i.e.*, Equation (11), to the modified vector field, *i.e.*, Equation (13). To demonstrate its effectiveness, Table 3 presents the comparison between the model using the modified loss and the naive loss (v-prediction in Table 3) from the Flow Matching loss. From the table, we observe a notable performance drop when using the naive Flow Matching loss. This suggests that **directly predicting the vector field results in inaccurate modeling of user pref-**

	HR@5	HR@10	HR@20	NDCG@5	NDCG@10	NDCG@20
Beauty						
v-prediction	0.5753	1.1962	2.0276	0.3138	0.5104	0.7948
FMREC	**5.8373**	**8.2693**	**11.626**	**4.1631**	**4.9461**	**5.7876**
Movielens-100k						
	HR@5	HR@10	HR@20	NDCG@5	NDCG@10	NDCG@20
v-prediction	0.8251	1.5239	2.2085	0.4637	0.7524	0.9896
FMREC	**9.3338**	**15.4934**	**24.3146**	**5.6848**	**7.6571**	**9.8158**

Table 3: Performance comparison using different Flow Matching losses, presented as percentages (%). The v-prediction approach employs the naive training loss. FMREC utilizes the modified Flow Matching loss.

erences, which significantly undermines the model's performance.
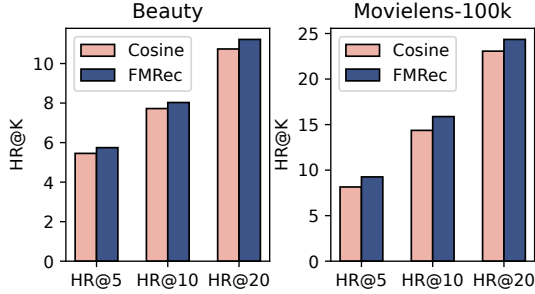
Figure 3: Performance comparison based on different flow trajectories, measured as percentages(%): "Cosine" represents the results obtained using the Cosine trajectory, while "FMREC" denotes the use of the straight trajectory.

Effectiveness of Straight Trajectories To demonstrate the effectiveness of straight trajectories, we use the Cosine trajectory as a baseline for comparison. This trajectory has been employed in IDDPM [Nichol and Dhariwal, 2021] to achieve superior performance compared to DDPM [Ho *et al.*, 2020]. In Figure 3, we present the results for both the Cosine trajectory (denoted as "Cosine") and straight trajectories (FMREC) on the *Beauty* and *Movielens-100k* datasets. The figure reveals a noticeable performance drop when using the Cosine trajectory, highlighting that straight trajectories improve robustness against error propagation. This robustness facilitates faster convergence to optimal results with fewer iterations.
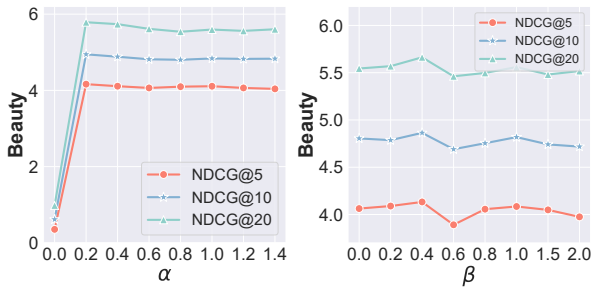

Figure 4: Comparison of model performance across various α and β on the *Beauty* dataset, measured as percentages(%). These parameters affect the importance of the cross-entropy loss and the reconstruction loss.

Effectiveness of Regularized Lossesd FMREC incorporates two regularized loss functions to enhance the model's

ability for recommendation. This section analyzes its effectiveness by examining the influence of the parameters α and β, as illustrated in Figure 4. Specifically, the figure displays the model's performance when trained with various combinations of α and β on the *Beauty* dataset. From Figure 4, we observe that an optimal setting of $\alpha = 0.2$ and $\beta = 0.4$ yields the best performance. Low values of α can significantly impact model performance, while excessively high values of α can also adversely affect it to some extent. In contrast, choosing an appropriate β can enhance the overall performance of the model. These results indicate that **both regularized loss functions are crucial for maintaining the effectiveness of the proposed FMREC model.**
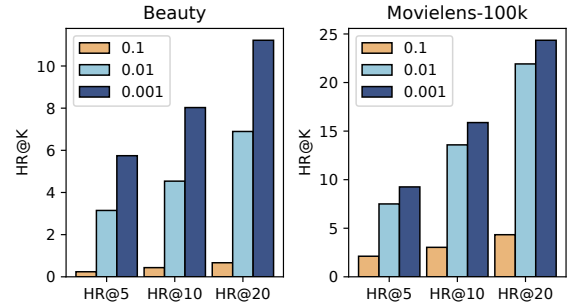

Figure 5: Performance comparison across different values of δ, measured as percentages(%), where δ controls the weight of noise perturbation in the user's sequential interactions fed into the model.

Influence of Noise Perturbation Before feeding the user's sequential interactions into the model, FMREC incorporates noise to meet the requirements of Flow Matching, where δ controls the fusion ratio's influence. To examine the impact of noise perturbation, we conduct experiments with $\delta \in \{0.1, 0.01, 0.001\}$. Figure 5 presents the performance comparison across different δ values. From the figure, we observe that the best performance is achieved when δ was set to 0.001, demonstrating that an appropriate level of noise perturbation enhances performance. Conversely, too large noise perturbation might lead to deviations in the calculation of the vector field during the reverse process and negatively impact the generation performance.

6 Conclusion

In conclusion, this work introduces a sequential recommendation model using Flow Matching, a simplified diffusion model that effectively mitigates noise perturbations of the diffusion-based model. By employing a straight flow trajectory and deriving a noise-free training target, our method reduces error accumulation in modeling user preferences. Additionally, the integration of a decoder architecture with an interaction reconstruction loss increases robustness against noise, ensuring precise user preference modeling. Furthermore, our deterministic reverse sampler, utilizing the Euler method, removes random perturbations during recommendation generation. Extensive experiments on four benchmark datasets demonstrate that our method, FMREC, achieves an average improvement of **6.53%** over SOTA techniques.

Acknowledgments

References

[de Souza Pereira Moreira *et al.*, 2018] Gabriel de Souza Pereira Moreira, Felipe Ferreira, and Adilson Marques Da Cunha. News session-based recommendations using deep neural networks. In *Proceedings of the 3rd workshop on deep learning for recommender systems*, pages 15–23, 2018.

[Dervishaj and Cremonesi, 2022] Ervin Dervishaj and Paolo Cremonesi. Gan-based matrix factorization for recommender systems. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, pages 1373–1381, 2022.

[Donkers *et al.*, 2017] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. Sequential user-based recurrent neural network recommendations. In *Proceedings of the eleventh ACM conference on recommender systems*, pages 152–160, 2017.

[Esser *et al.*, 2024] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.

[Fan *et al.*, 2022] Ziwei Fan, Zhiwei Liu, Yu Wang, Alice Wang, Zahra Nazari, Lei Zheng, Hao Peng, and Philip S Yu. Sequential recommendation via stochastic self-attention. In *Proceedings of the ACM web conference 2022*, pages 2036–2047, 2022.

[Gong *et al.*, 2022] Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and LingPeng Kong. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933*, 2022.

[Guo *et al.*, 2024] Zhiqiang Guo, Guohui Li, Jianjun Li, Chaoyang Wang, and Si Shi. Dualvae: Dual disentangled variational autoencoder for recommendation. In *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*, pages 571–579. SIAM, 2024.

[Harper and Konstan, 2015] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

[Harvey *et al.*, 2022] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible diffusion modeling of long videos. *Advances in Neural Information Processing Systems*, 35:27953–27965, 2022.

[Hidasi *et al.*, 2015] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

[Ho *et al.*, 2020] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[Ho *et al.*, 2022] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.

[Huang *et al.*, 2024] Hongtao Huang, Chengkai Huang, Xiaojun Chang, Wen Hu, and Lina Yao. Dual conditional diffusion models for sequential recommendation. *arXiv preprint arXiv:2410.21967*, 2024.

[Jiangzhou *et al.*, 2024] Deng Jiangzhou, Wang Songli, Ye Jianmei, Ji Lianghao, and Wang Yong. Dgrm: Diffusion-gan recommendation model to alleviate the mode collapse problem in sparse environments. *Pattern Recognition*, page 110692, 2024.

[Kang and McAuley, 2018] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE, 2018.

[Li *et al.*, 2023] Zihao Li, Aixin Sun, and Chenliang Li. Diffurec: A diffusion model for sequential recommendation. *ACM Transactions on Information Systems*, 42(3):1–28, 2023.

[Li *et al.*, 2024a] Nuo Li, Bin Guo, Yan Liu, Yasan Ding, Lina Yao, Xiaopeng Fan, and Zhiwen Yu. Hierarchical constrained variational autoencoder for interaction-sparse recommendations. *Information Processing & Management*, 61(3):103641, 2024.

[Li *et al.*, 2024b] 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. *arXiv preprint arXiv:2408.12153*, 2024.

[Lipman *et al.*, 2022] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[Liu *et al.*, 2022] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[Liu *et al.*, 2023] Sijia Liu, Jiahao Liu, Hansu Gu, Dongsheng Li, Tun Lu, Peng Zhang, and Ning Gu. Autoseqrec: Autoencoder for efficient sequential recommendation. In *Proceedings of the 32nd ACM International Conference*

on Information and Knowledge Management, pages 1493–1502, 2023.

[Nakkiran *et al.*, 2024] Preetum Nakkiran, Arwen Bradley, Hattie Zhou, and Madhu Advani. Step-by-step diffusion: An elementary tutorial. *arXiv preprint arXiv:2406.08929*, 2024.

[Ni *et al.*, 2019] Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 188–197, 2019.

[Nichol and Dhariwal, 2021] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021.

[Sedhain *et al.*, 2015] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th international conference on World Wide Web*, pages 111–112, 2015.

[Song *et al.*, 2020] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[Sun *et al.*, 2019] 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 *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450, 2019.

[Tang and Wang, 2018] Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 565–573, 2018.

[Tang *et al.*, 2024] Min Tang, Lixin Zou, Shujie Cui, Shiuanni Liang, and Zhe Jin. Unbiased recommendation through invariant representation learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 280–296. Springer, 2024.

[Tang *et al.*, 2025] Min Tang, Shujie Cui, Zhe Jin, Shiuanni Liang, Chenliang Li, and Lixin Zou. Sequential recommendation by reprogramming pretrained transformer. *Information Processing & Management*, 62(1):103938, 2025.

[Vaswani, 2017] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[Wang *et al.*, 2023] Wenjie Wang, Yiyan Xu, Fuli Feng, Xinyu Lin, Xiangnan He, and Tat-Seng Chua. Diffusion recommender model. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 832–841, 2023.

[Wang *et al.*, 2024] Yu Wang, Zhiwei Liu, Liangwei Yang, and Philip S Yu. Conditional denoising diffusion for sequential recommendation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 156–169. Springer, 2024.

[Wu *et al.*, 2017] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. Recurrent recommender networks. In *Proceedings of the tenth ACM international conference on web search and data mining*, pages 495–503, 2017.

[Wu *et al.*, 2019] Qiong Wu, Yong Liu, Chunyan Miao, Binqiang Zhao, Yin Zhao, and Lu Guan. Pd-gan: Adversarial learning for personalized diversity-promoting recommendation. In *IJCAI*, volume 19, pages 3870–3876, 2019.

[Wu *et al.*, 2023] Tong Wu, Zhihao Fan, Xiao Liu, Hai-Tao Zheng, Yeyun Gong, Jian Jiao, Juntao Li, Jian Guo, Nan Duan, Weizhu Chen, et al. Ar-diffusion: Auto-regressive diffusion model for text generation. *Advances in Neural Information Processing Systems*, 36:39957–39974, 2023.

[Xu *et al.*, 2019] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Jiajie Xu, Victor S Sheng S. Sheng, Zhiming Cui, Xiaofang Zhou, and Hui Xiong. Recurrent convolutional neural network for sequential recommendation. In *The world wide web conference*, pages 3398–3404, 2019.

[Yang *et al.*, 2024] Zhengyi Yang, Jiancan Wu, Zhicai Wang, Xiang Wang, Yancheng Yuan, and Xiangnan He. Generate what you prefer: Reshaping sequential recommendation via guided diffusion. *Advances in Neural Information Processing Systems*, 36, 2024.

[Ye *et al.*, 2023] Yaowen Ye, Lianghao Xia, and Chao Huang. Graph masked autoencoder for sequential recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 321–330, 2023.

[Yu *et al.*, 2024] Pinrui Yu, Dan Luo, Timothy Rupprecht, Lei Lu, Zhenglun Kong, Pu Zhao, Yanyu Li, Octavia Camps, Xue Lin, and Yanzhi Wang. Fastervd: on acceleration of video diffusion models. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 8838–8842, 2024.

[Yu *et al.*, 2025] Qing Yu, Lixin Zou, Xiangyang Luo, Xiangyu Zhao, and Chenliang Li. Uniform graph pretraining and prompting for transferable recommendation. *ACM Transactions on Information Systems*, 2025.

[Zou *et al.*, 2019] Lixin Zou, Long Xia, Zhuoye Ding, Jiaxing Song, Weidong Liu, and Dawei Yin. Reinforcement learning to optimize long-term user engagement in recommender systems. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2810–2818, 2019.

[Zou *et al.*, 2020] Lixin Zou, Long Xia, Yulong Gu, Xiangyu Zhao, Weidong Liu, Jimmy Xiangji Huang, and Dawei Yin. Neural interactive collaborative filtering. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 749–758, 2020.