Information Fusion 129 (2026) 104004

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full Length Article

PIFGSR: Pluggable framework for information fusion using generative artificial intelligence (GenAI) in recommender systems

Yicheng Di [a], Hongjian Shi [b], Khushal Haider Syed [b], Ruhui Ma [b,*], Haibing Guan [b], Yuan Liu [a], Rajkumar Buyya [c]

[a] School of Artificial Intelligence and Computer Science, Jiangnan University, 1800 Lihu Avenue, 214122, Wuxi, China
[b] School of Computer Science, Shanghai Jiao Tong University, 800 Dongchuan Rd. Minhang Dist, 200240, Shanghai, China
[c] University of Melbourne, Grattan Street, 3010, Parkville, Victoria, Australia

ARTICLE INFO

Keywords:
Generative artificial intelligence
Information fusion
Diffusion model
Sequential recommender
Personalization

ABSTRACT

The application of Generative Artificial Intelligence (GenAI) in information fusion brings new breakthroughs in enhancing user experience within recommender systems, with sequential recommender being an indispensable element. Plugin-based diffusion models, a prominent GenAI approach, have proven effective in alleviating sparsity in sequential recommenders. However, by introducing disordered noise that disrupts sequential logic, these methods can distort user interest modeling and still struggle with data sparsity. To resolve these problems, we propose the novel general framework Pluggable Framework for Information Fusion Using Generative Artificial Intelligence (GenAI) in Recommender Systems (PIFGSR), which generates expanded sequences aligned with users' real interests while retaining both explicit and implicit feedback at low computational cost. Central to PIFGSR is the Potential Interest Guidance Bank (PIGB), which uses trainable parameter matrices to learn and store interest patterns from user interactions and integrate them with prior knowledge from a pre-trained diffusion model. These learned cues guide the diffusion process during inference, enabling effective information fusion and preserving users' true preference trajectories. We design the Demand Information Extraction Mechanism (DIEM), which integrates spatial row-column attention and statistical mean-variance attention to extract users' actual points of interest from local and global user interaction features. Specifically, DIEM takes a parameter matrix as input, applies convolutional projections to generate query, key, and value representations, and computes normalized correlations. Spatially, it constructs and fuses row-wise and column-wise weights to form local offset features. Statistically, it derives an attention-weighted scale from the mean and variance to perform normalization and shifting, ultimately producing a demand representation that integrates local structure with global distribution. Additionally, to ensure expanded sequences align with user interests in the original interactions, we design the Guide Matching Strategy (GMS). Compared to existing baselines, PIFGSR achieves a superior performance improvement of 9.85 %.

1. Introduction

In recent years, with the rapid development and widespread application of Generative Artificial Intelligence (GenAI), its advanced capabilities in information fusion have emerged as a significant driver for enhancing user experience in recommender systems [1]. By generating semantically coherent and contextually relevant content from heterogeneous data sources, GenAI enables the seamless integration of explicit and implicit feedback, bridging gaps caused by sparsity or modality differences. Meanwhile, as the application scope of recommender systems expands, the increasing individual differences and behavioral complexities of user groups further amplify the challenges of interest modeling [2–4]. Consequently, efficiently mining latent interest patterns from vast historical interaction data and leveraging them for precise content matching has become a critical research objective in modern recommender systems [5].

Sequential recommender systems play a key role in this task by modeling the latent temporal dependencies in user behavior sequences, thereby achieving an accurate characterization of the dynamic evolution of user interests [6–8]. Specifically, sequential recommender systems explicitly model temporal dependencies within user behavior sequences, short-term interest drift, and cross-session context, thereby enabling real-time responsiveness to user needs, improving the relevance and timeliness of recommendations, and reducing irrelevant exposure

* Corresponding author.
E-mail address: ruhuima@sjtu.edu.cn (R. Ma).

https://doi.org/10.1016/j.inffus.2025.104004
Received 25 June 2025; Received in revised form 11 November 2025; Accepted 24 November 2025
Available online 27 November 2025

and interaction overhead, which systematically enhances the user experience. By leveraging both the chronological order and contextual relevance of interactions, these systems can effectively distinguish between long-term stable preferences and short-term interest shifts, enabling more fine-grained and adaptive recommender strategies. In addition, the sequential modeling framework facilitates the timely capture of emerging interests, trend transitions, and intent changes, which is critical for applications where user needs evolve rapidly, such as news feeds, e-commerce promotions, and personalized entertainment services. Thanks to their ability to capture contextual evolution and short-term preferences, sequential recommenders significantly enhance recommender accuracy, diversity, and novelty, while also providing more generalizable support for user modeling, content matching, and long-term retention.

The plugin-based diffusion model is an important implementation of GenAI, demonstrating strong effectiveness in enhancing predictive accuracy in sequential recommender systems. In recent years, research on plugin-based diffusion models has made remarkable progress, with its core value lying in the fusion of controllable stochastic processes to enrich user interaction histories [9–11]. The existing methodological system can be systematically categorized into two major technical paradigms: (1) sequence-level diffusion methods [12]. These methods implicitly model the evolution of user behavior patterns and construct an end-to-end sequence generation framework to mitigate the challenge of data sparsity. The core idea is to treat user behavior sequences as latent trajectories in a continuous state space, leveraging the iterative denoising mechanism of the diffusion process to fuse latent state information, thereby generating statistically plausible synthetic sequences [13]. The latent trajectory in continuous state space refers to an implicit path that evolves with user behavior under continuously changing preference states, characterizing the dynamic variation of user interests over time. Specifically, this approach defines a global noise perturbation and a reverse reconstruction strategy, gradually correcting the initial noise distribution until it approximates the true user behavior pattern. (2) Item-level diffusion methods [14] represent a class of approaches that expand user interaction histories at the granularity of items. This paradigm focuses on inserting appropriately adapted pseudo-interaction items into user historical sequences to construct information-augmented data that can be directly input into standard recommender architectures. By modeling the diffusion process on the item space, these methods can capture potential but unobserved user-item relations and enrich the sequential context. Furthermore, by employing local sequence expansion strategies to fuse original and augmented sequence information, they effectively enhance the representation density for users with short sequences and provide an explicit solution for long-tail optimization [15]. Both approaches have achieved encouraging results.

However, plugin-based diffusion models still possess inherent limitations in their core design [16]. Firstly, existing methods generally adopt a global random perturbation strategy, which introduces noise indiscriminately across the entire sequence without considering its temporal or causal structure [17]. Such unordered noise disrupts the causal dependencies between items-i.e., the directional influence of earlier behaviors on subsequent ones, by altering or masking the order-sensitive contextual cues that link user actions together [18]. These dependencies are crucial for preserving the logical progression of user decision-making, where each choice is often conditioned on the accumulation of prior interactions. When this ordering is disturbed, the augmented sequence may retain superficial semantic similarity at the local level but lose alignment with the global trajectory of user interest evolution. As a result, the captured preference patterns deviate from the true behavioral distribution, leading to inaccurate modeling of user interests [19]. Secondly, to alleviate the representation distortion induced by such noise perturbations, existing approaches frequently incorporate domain-specific prior constraints to guide the generation process. However, these constraints inevitably compromise the model's generalization ability, making it less effective when adapting to diverse data distributions across different scenarios. Moreover, dynamically balancing noise intensity and semantic fidelity increases model complexity, leading to slower convergence and higher inference latency, thus limiting applicability in real-time recommender systems.

To address these issues, we propose the Pluggable Framework for Information Fusion Using Generative Artificial Intelligence (GenAI) in Recommender Systems (PIFGSR). The core of PIFGSR lies in integrating a pre-trained diffusion model with interest-driven information extracted from user interactions, so as to generate augmented sequences that align with users' actual interests while preserving both explicit and implicit feedback in the sequence. Specifically, to achieve interest-driven sequence augmentation via the diffusion model, PIFGSR mainly comprises three core modules. First, the framework leverages a pre-trained diffusion model to construct a parameter generation mechanism for the reverse process, reconstructing user behavior sequences via a noise prediction network. This design gives rise to a backbone parameter and dynamic parameters, where the former solidifies the general sequence generation capability of the pre-trained model, and the latter, through the Potential Interest Guidance Bank (PIGB), captures personalized user interests for information fusion during the diffusion process. Second, the Demand Information Extraction Mechanism (DIEM) maps the original behavior sequence into latent interest codes by analyzing local patterns and global features in the user interaction matrix from both spatial and statistical perspectives. This mechanism employs row-column attention to capture spatial correlations and combines it with mean-variance channel normalization to extract statistical patterns, forming interpretable interest representations. Finally, the Guide Matching Strategy (GMS) introduces a pre-trained sequential recommender model as a classifier in the reverse process and employs gradient guidance to ensure consistency between the augmented sequence and the user's original interests. Our contributions are summarized as follows:

- We propose PIFGSR, which generates augmented sequences that align with users' actual interests through a pre-trained diffusion model, and it can be flexibly applied to different sequential recommenders.
- We propose PIGB to fuse pre-trained diffusion priors with interaction-driven sequential signals, DIEM to extract true interests via joint local-global modeling, and GMS to align interests between expanded and original sequences.
- Comprehensive experiments on four benchmark datasets indicate that PIFGSR substantially surpasses all leading approaches. Significantly, on Yelp, PIFGSR attains a 9.85 % relative improvement in *NDCG*@5 in comparison to the top-performing methods.

2. Related work

2.1. Sequential recommender

Sequential recommender forecasts the subsequent item of interest for users by analyzing dependencies in their prior behaviours, thereby capturing the dynamic evolution of user preferences [20,21]. Early studies focus on sequential deep models to capture complex user preferences [22]. Caser [23] embeds recent interaction sequences into images in temporal and latent spaces, using convolutional filters to learn local features and capture sequential patterns while combining overall user preferences. SASRec [24] employs self-attention to balance both short- and long-term semantics, facilitating predictions based on sparse user interaction data. Recent research enhances the capability of sequential deep models to extract user interests. DuoRec [25] mitigates representation degradation by regularizing sequence representations and improving item embedding distributions from models like Transformer and BERT. AdaGCL [26] introduces two adaptive contrastive view generators to provide additional high-quality training signals for collaborative filtering, addressing data sparsity and noise. ContraRec [27] proposes a contextual contrastive task to align enhanced sequences and those with the same target item. MCLRec [28] uses a meta-learning approach to guide

model enhancer updates, capturing useful features from random data expansion. However, these approaches remain largely model-driven, being developed on static datasets, and often overlook critical issues such as data quality, intrinsic redundancy, and noise, which can hinder their ability to fully capture authentic user interests.

2.2. Diffusion modeling methods of GenAI in recommender

As one of the important implementation pathways of GenAI, diffusion models demonstrate strong capabilities in recommender system tasks [29], operating by progressively injecting Gaussian noise into input data through a forward process, and then learning to reverse this process to generate new samples. This mechanism has shown strong potential in addressing data sparsity issues commonly encountered in recommender systems [30]. For instance, DiffuASR [31] addresses data sparsity and long-tail user issues by generating pseudo-interaction sequences to enrich the training set. DiffRec [32] learns the generative process of user-item interactions through denoising, gradually removing noise while preserving personalized information. DreamRec [33] adds noise to target items to explore the item space distribution, better capturing true user preferences and overcoming limitations of traditional negative sampling. CDDRec [34] adopts a different strategy by decoupling optimization and generation into sequential autoregressive sub-processes, improving both sequence quality and item representation learning. Other works treat diffusion models as modular components to flexibly generate user preferences across the entire item set. For instance, SeeDRec [35] refines the diffusion objective from the item to the semantic unit level, capturing more nuanced user behavior patterns. PDRec [36] re-weights historical behaviors to emphasize high-quality interactions and reduce noise, while leveraging top-ranked unobserved items as candidate positives to introduce diverse soft signals. Despite these advancements, a common shortcoming remains: many of these methods tend to disrupt the structure of explicit and implicit feedback embedded in user sequences during the generative process. As a result, they often fail to generate expanded sequences that faithfully reflect users' genuine interests.

3. Methodology

3.1. Problem definition

The objective of sequential recommender is to suggest the subsequent item based on users' prior interaction history. The user set U is represented as $U = \{u_1, u_2, \dots, u_{|U|}\}$, and the item set I is represented as $I = \{i_1, i_2, \dots, i_{|I|}\}$. Every user $u \in U$ possesses a chronologically arranged sequence of interacted items $S_u = \{i_1^u, i_2^u, \dots, i_N^u\}$, with N representing the total number of historical interactions of the user. The sequential recommender task predicts the most likely item for interaction at step $N+1$ based on S_u, expressed as:

$$\arg\max_{i_n \in I} P(i_{N+1} = i_n | S_u), \tag{1}$$

Given the original interaction item sequence $S_{ori} = \{i_1, \dots, i_m, \dots, i_N\}$, the item-level expansion method generates a pre-sorted list of items, with expanded items inserted before i_1. Let $\hat{S}_{exp} = \{\hat{i}_{-M}, \hat{i}_{-M+1}, \dots, \hat{i}_{-1}\}$ denote the expanded sequence of S_{ori}, where M denotes the number of expanded items. Subsequently, the original sequence is converted into $\hat{S} = \{\hat{i}_{-M}, \dots, \hat{i}_{-1}, i_1, \dots, i_N\}$. PIFGSR aims to generate $\hat{S}_{exp}$ that aligns with the user's potential interests. The key notions in this paper are shown in Table 1.

3.2. System overview

Shown in Fig. 1, PIFGSR comprises training and inference: during training, PIGB learns and stores interest-driven sequential signals from interactions for fusion in the diffusion process, while DIEM extracts true interests from local and global behavioral patterns. At inference, PIGB conditions the pre-trained diffusion model and GMS steers the reverse process, so the expanded embedding sequence aligns with users' potential interests while preserving explicit and implicit feedback, yielding expanded sequences that match actual interests.

Table 1
Summary of key notations used in PIFGSR framework.

Notations	Description
U	The user set
I	The item set
S_u	Sequence of interaction items for user u
N	Historical number of user interactions
B	Interactive item sequence collection
S_{ori}	Original interactive item sequence
$\hat{S}_{exp}$	Expanded sequence
$\hat{S}$	Expanded interactive item sequence
β_t	Variance scale at time step t
i_n	Item identity
c_l	Interest representation vector
c_t	Continuous vector representation
P_l	Learnable parameter matrix
ϕ	Pre-trained sequential recommender model
M	Sequence expanded number
F	Attention matrix
T	The number of diffusion steps
λ	The guide strength

3.3. Diffusion-based sequential recommender

3.3.1. Forward process

The expanded sequence $S_{exp} = \{i_{-M}, i_{-M+1}, \dots, i_{-1}\}$ consists of a series of item identities, and we can set $x_0 = S_{exp}$ as the initial state. An embedding table is utilized to transform item identities i_n into low-dimensional dense representations $\mathbf{h}_n$. This embedding table is denoted as $\mathbf{H} \in \mathbb{R}^{|I| \times D}$, where D represents the dimensionality of the embeddings. The stacked item embeddings are represented as $\mathbf{h}_0 = \{h_{-M}, h_{-M+1}, \dots, h_{-1}\}$. We transform the one-dimensional discrete item identities into a two-dimensional continuous value matrix, which functions as the image representation matrix, compatible with the input requirements of the pre-trained diffusion model. We then iteratively incorporate Gaussian noise into the embedding sequence $\mathbf{h}_t$.The forward process gradually transforms $\mathbf{h}_0$ into standard Gaussian noise $\mathbf{h}_t \sim \mathcal{N}(0; \mathbf{I})$, expressed as:

$$\begin{aligned} q(\mathbf{h}_{1:T} \mid \mathbf{h}_0) &= \prod_{t=1}^{T} q(\mathbf{h}_t \mid \mathbf{h}_{t-1}), \\ q(\mathbf{h}_t \mid \mathbf{h}_{t-1}) &= \mathcal{N}(\mathbf{h}_t; \sqrt{1-\beta}\mathbf{h}_{t-1}, \beta_t \mathbf{I}), \end{aligned} \tag{2}$$

where $\beta_t \in (0, 1)$ denotes the variance scale at time step t, $\mathbf{I}$ is the identity matrix. Learning the variance β_t directly is difficult; thus, the reparameterization trick [37] reformulates the forward process as follows:

$$q(\mathbf{h}_t \mid \mathbf{h}_0) = \mathcal{N}(\mathbf{h}_t; \sqrt{\bar{\alpha}_t}\mathbf{h}_0, (1-\bar{\alpha}_t)\mathbf{I}), \tag{3}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. To regulate the amount of noise added in $\mathbf{h}_{1:T}$, we reparameterize $\mathbf{h}_t = \sqrt{\bar{\alpha}_t}\mathbf{h}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$, where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. We adopt a linear noise schedule for $1 - \bar{\alpha}_t$ to control the noise level in $\mathbf{h}_{1:T}$.

$$1 - \bar{\alpha}_t = z \cdot \left[\chi_{min} + \frac{t-1}{T-1}\left(\chi_{max} - \chi_{min}\right)\right], t \in \{1, \dots, T\} \tag{4}$$

where $z \in [0, 1]$ controls the noise scale, and χ_{min} and χ_{max} denote the lower and upper bounds of the added noise, respectively, both constrained to the range $[0, 1]$.

3.3.2. Reverse process

Subsequent to the forward process, the reverse process progressively reconstructs the original data $\mathbf{h}_0$ by sampling from $\mathbf{h}_t$ utilizing

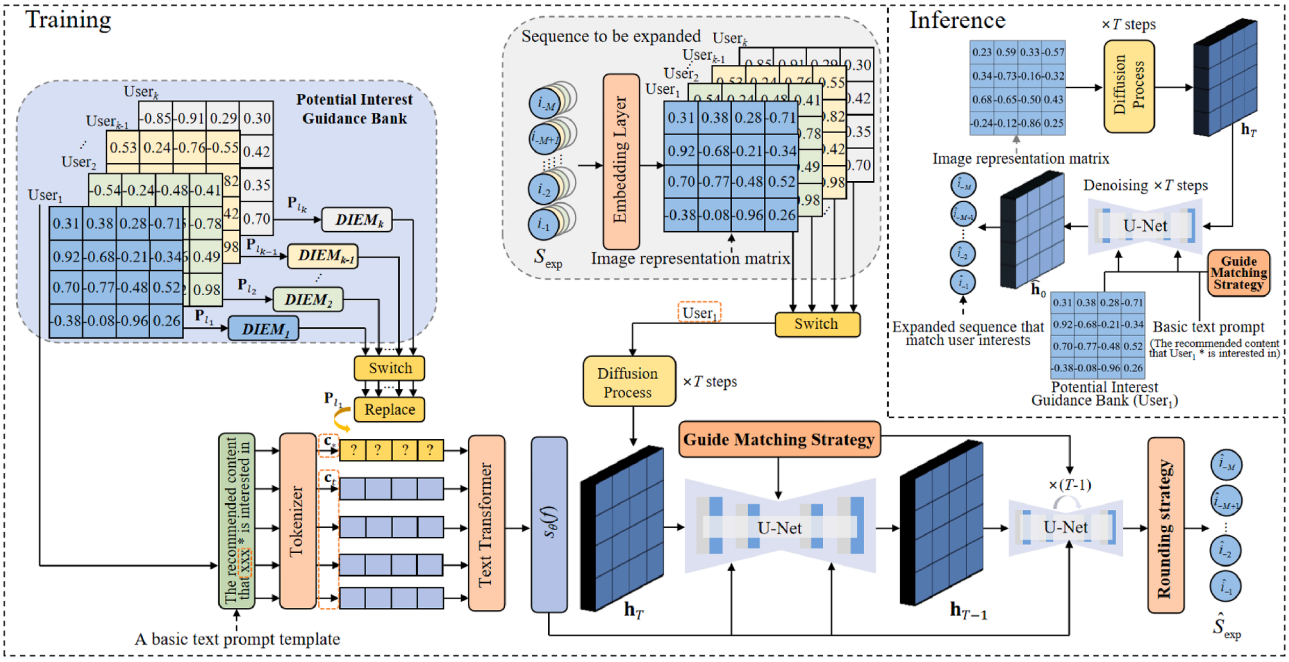

Fig. 1. The overall structure of PIFGSR consists of three main parts: PIGB, DIEM, and GMS.

the acquired diffusion model, expressed as:

$$p_\theta(\mathbf{h}_{0:T}) = p(\mathbf{h}_t)\prod_{t=1}^{T} p_\theta(\mathbf{h}_{t-1} \mid \mathbf{h}_t),$$
$$p_\theta(\mathbf{h}_{t-1} \mid \mathbf{h}_t) = \mathcal{N}(\mathbf{h}_{t-1}; \mu_\theta(\mathbf{h}_t, t), \Sigma_\theta(\mathbf{h}_t, t)), \tag{5}$$

where $\mu_\theta(\mathbf{h}_t, t)$ and $\Sigma_\theta(\mathbf{h}_t, t)$ are the mean and variance parameterized by θ. The mean of the distribution can be expressed as:

$$\mu_\theta(\mathbf{h}_t, t) = \frac{1}{\sqrt{\alpha_t}}(\mathbf{h}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}})\epsilon_\theta(\mathbf{h}_t, t), \tag{6}$$

where $\epsilon \sim N(0, \mathbf{I})$ and $\epsilon_\theta(\mathbf{h}_t, t)$ is parameterized by a neural network, such as U-Net [38], to forecast the noise ϵ that aids the transition from $\mathbf{h}_0$ to $\mathbf{h}_t$ in the forward process.

3.3.3. Rounding strategy

We anticipate the noise introduced at each diffusion phase utilizing Eq. (6). $\epsilon_\theta(\mathbf{h}_t, t)$ is typically implemented by U-Net. However, it is usually applied to image processing tasks, where each element in the column and row dimensions has different meanings compared to recommender tasks. Therefore, we designate the sequence dimension as the image channel to capture the sequential information of $\mathbf{h}_t$, converting the input image matrix into one-dimensional form. Each one-dimensional embedding is reshaped into a matrix of shape $(\sqrt{D}, \sqrt{D})$. Therefore, the embedding sequence is converted into a matrix with Z channels, each with a height and width equal to $\sqrt{D}$. Before downsampling, several ResNet blocks utilizing convolutional neural networks are employed on the multi-channel matrix. Subsequently, we use an AttnBlock to eliminate redundant information. During the upsampling process, the downsampled matrix is duplicated and combined to integrate additional user-interest elements. The output matrix retains the dimensions of the input, reforming the multi-channel matrix into a one-dimensional embedding vector to derive $\mathbf{h}_{t-1}$. Ultimately, the expanded embedding sequence $\hat{\mathbf{h}}_0 = \{\hat{h}_{-M}, \hat{h}_{-M+1}, \ldots, \hat{h}_{-1}\}$ can be obtained. A rounding strategy is employed to discretize the continuous value vector to yield the expanded sequence, expressed as:

$$\hat{i}_r = \max_{i_n \in I} \text{sim}(\hat{h}_r, h_n), \tag{7}$$

where r represents the position of the expanded item, where $r \in [-M, -1]$. sim$(\cdot)$ denotes the cosine similarity function, which can eliminate the impact of length differences between the expanded embeddings and the original embeddings.

3.4. Potential interest guidance bank

To acquire the expanded sequence, an intuitive method is to align the posterior distribution in the forward process with the prior distribution in the reverse phase, expressed as:

$$\mathcal{L}_{diff} = \mathbb{E}_{\mathbf{h}_0,t,\epsilon}[\| \epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{h}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t) \|_2^2], \tag{8}$$

where $\mathbf{h}_0 \sim \mathbb{E}(x)$, and $\epsilon \sim \mathcal{N}(0, 1)$. Upon convergence, the model diffuses any initial sequence into an expanded one. However, this naive generation process cannot effectively ensure that the expanded sequence is aligned with user-specific interests derived from prior interaction data. The key motivation behind our design is to strike a balance between two essential aspects: preserving the general sequence generation capability of the pre-trained diffusion model, and incorporating personalized user interests to guide the generation process. To this end, we fix the parameters of the backbone diffusion model to maintain its robust generative ability, and introduce PIGB, which consists of trainable parameter matrices tailored for user interaction sets and projected into continuous matrices. This design allows the dynamic parameters in PIGB to capture interest-driven sequential information and adjust the diffusion process, enabling the model to generate sequences that remain faithful to user preferences while preserving both explicit and implicit feedback information.

Using the diffusion model described in Section 3.3 as backbone and a CLIP text encoder for reverse denoising, we produce image matrices reflecting genuine user interests. When using the text encoder as an example, the text prompt is converted into a continuous vector representation $\mathbf{c}_t$. While $\mathbf{c}_t$ can direct the diffusion model to produce the intended image representation matrix, it fails to encapsulate the diffusion model's understanding of interest transfer. Therefore, we design several basic text prompt templates. For example, the recommender content of interest for User_1 is represented by a meaningless placeholder *. The basic text prompt templates are converted into continuous vector representations $\mathbf{c}_t$ and $\mathbf{c}_*$. Simultaneously, the learnable parameter matrix $\mathbf{P}_l$ of

PIGB is projected into the continuous interest representation vector $\mathbf{c}_l$ using DIEM, represented as $\mathbf{c}_l = DIEM(\mathbf{P}_l)$. This vector $\mathbf{c}_l$ then replaces $\mathbf{c}_*$, and both $\mathbf{c}_t$ and $\mathbf{c}_l$ are converted into a single conditional code $s_\theta(f)$. In this process, only the learnable parameter matrix $\mathbf{P}_l$ and DIEM require training, with each user having corresponding $\mathbf{P}_{(l_k)}$ and $DIEM_k$, where $DIEM_k$ captures user demand information and reveals potential interests through complex interactions. The loss function is expressed as:

$$\mathcal{L}_{diff} = \mathbb{E}_{\mathbf{h}_0,t,f,\epsilon}[\| \epsilon - \epsilon_\theta(\sqrt{\overline{\alpha}_t}\mathbf{h}_0 + \sqrt{1-\overline{\alpha}_t}\epsilon, t, DIEM(\mathbf{P}_l), \mathbf{c}_t) \|_2^2], \tag{9}$$

where $\mathbf{c}_t$ represents the text prompt. When $\mathbf{P}_l$ and DIEM are trained, PIFGSR can generate an expanded sequence that aligns with the user's actual interests while preserving both explicit and implicit user feedback information.

3.5. Demand information extraction mechanism

DIEM combines spatial and statistical perspectives to analyze variations in the parameter matrix. Using row-column attention for spatial analysis and mean-variance attention for statistical analysis, it extracts users' actual points of interest from local patterns and global features of user interactions. Row-column attention is a mechanism that models dependencies along both the row and column dimensions of the parameter matrix, aiming to capture interactions and structural patterns among features. This approach enhances accuracy in capturing user demands and uncovers potential interests in complex interactions. DIEM begins with the trainable parameter matrix $\mathbf{P}_l$, encoded into queries $\mathbf{Q}$, keys $\mathbf{K}$, and values $\mathbf{V}$, as expressed:

$$\mathbf{Q} = \mathbf{C}_\mathbf{Q} \cdot \mathbf{P}_l, \mathbf{K} = \mathbf{C}_\mathbf{K} \cdot \mathbf{P}_l, \mathbf{V} = \mathbf{C}_\mathbf{V} \cdot \mathbf{P}_l, \tag{10}$$

where $\mathbf{C}_\mathbf{Q}$, $\mathbf{C}_\mathbf{K}$, and $\mathbf{C}_\mathbf{V}$ represent trainable convolutional layers, $\cdot$ signifies element-wise multiplication. The attention matrix $\mathbf{F}$ can be expressed as:

$$\mathbf{F} = soft\max(\mathbf{Q}^\mathrm{T} \otimes \mathbf{K}), \tag{11}$$

where $\otimes$ denotes matrix multiplication, and the attention matrix $\mathbf{F}$ captures the relationships between different rows and columns. For the attention matrix $\mathbf{F}$, we construct the column-wise weight matrix $\mathbf{C}_1$ and the row-wise weight matrix $\mathbf{C}_2$. For convenience in calculations, we replicate $\mathbf{C}_1$ and $\mathbf{C}_2$ along with the columns and rows. Then, we can obtain the column and row attention matrices, expressed as:

$$\mathbf{F}_c = \mathbf{F} \cdot \mathbf{C}_1, \mathbf{F}_r = \mathbf{F} \cdot \mathbf{C}_2, \tag{12}$$

The column weight matrix $\mathbf{F}_c$ and the row weight matrix $\mathbf{F}_r$ capture the features of the matrix along the column and row directions, respectively. Next, $\mathbf{F}' = \gamma \cdot \mathbf{F}_c + (1-\gamma) \cdot \mathbf{F}_r$, where γ is the fusion coefficient. The row-column attention information is integrated into the shifted matrix $\mathbf{G}'$, enabling the extraction of local patterns, expressed as:

$$\mathbf{G}' = \mathbf{V} \cdot \mathbf{F}', \tag{13}$$

From a statistical perspective, DIEM introduces the mean and variance as normalization factors for the parameter matrix. The attention-weighted standard deviation $\mathbf{D}'$ is calculated to capture global distribution features. The attention-weighted standard deviation $\mathbf{D}'$ can be expressed as:

$$\mathbf{D}' = \sqrt{(\mathbf{V} \cdot \mathbf{V}) \otimes \mathbf{F}'^\mathrm{T} - \mathbf{G}' \cdot \mathbf{G}'}, \tag{14}$$

By combining the corresponding shifted matrix $\mathbf{G}'$ and scale $\mathbf{D}'$, $\mathbf{P}_l$ is normalized and transformed to generate a parameter matrix that incorporates both local and global information, expressed as:

$$\mathbf{c}_l = \mathbf{D}' \cdot Norm(\mathbf{P}_l) + \mathbf{G}', \tag{15}$$

This process can be represented as $\mathbf{c}_l = DIEM(\mathbf{P}_l)$.

3.6. Guide matching strategy

Effective sequence expansion must remain faithful to the user's latent interests encoded in the original interaction history; otherwise, injected items dilute supervision, introduce bias, and destabilize ranking. To enforce such fidelity, GMS injects task-aware guidance into the reverse diffusion process so that each denoising step not only reconstructs plausible sequences but also steers generation toward user-consistent preferences. This design is analogous to classifier guidance in image diffusion, where gradients from a classifier bias samples toward target classes [39]. Here, a pre-trained sequential recommender plays the role of a preference classifier: its next-item likelihood supplies gradients that nudge the diffusion trajectory toward sequences coherent with the original history S_{ori}. Operationally, given the expanded sequence S_{exp}, we treat the first item i_1 in S_{ori} as the next-item label for guidance, and define the guided noise prediction as:

$$\hat{\epsilon} = \epsilon_\theta(\mathbf{h}_t, t, DIEM(\mathbf{P}_l), \mathbf{c}_t) - \lambda \cdot \sqrt{1-\overline{\alpha}_t} \nabla_{\mathbf{h}_t} \log p_\varphi(i_1 \mid S_{\text{exp}}), \tag{16}$$

where λ represents the guidance strength, $p_\varphi(\cdot)$ is the classifier using binary cross-entropy loss.

Algorithm 1 Training and expansion process of PIFGSR.

Input: Number of diffusion steps T, sequence expanded number J, pre-trained sequential recommender model ϕ, guide strength λ, variance schedule: $\beta_{1:T}$.
Output: Expanded sequence $\hat{S}_{exp}$.

Training process:
Select sequences from B that satisfy $N > J$ to form B';
for S in B' **do**
  Take the first J items as S_{exp} and the rest as S_{ori};
  Train the learnable parameter matrix $\mathbf{P}_l$ to obtain $s_\theta(f)$;
  $q(\mathbf{h}_t \mid \mathbf{h}_0) = \mathcal{N}(\mathbf{h}_t; \sqrt{\overline{\alpha}_t}\mathbf{h}_0, (1-\overline{\alpha}_t)\mathbf{I})$;
  for diffusion step is T **do**
    $\hat{\epsilon} = \epsilon_\theta(\mathbf{h}_t, t, DIEM(\mathbf{P}_l), \mathbf{c}_t) - \lambda \cdot \sqrt{1-\overline{\alpha}_t} \nabla_{\mathbf{h}_t} \log p_\varphi(i_1 \mid S_{\text{exp}})$;
  end for
  $p_\theta(\mathbf{h}_{0:T}) = p(\mathbf{h}_t) \prod_{t=1}^{T} p_\theta(\mathbf{h}_{t-1} \mid \mathbf{h}_t)$;
  Calculate the loss and update the parameters through $\mathcal{L}_{diff}$;
end for
Expansion process:
for S in B **do**
  for diffusion step is T **do**
    $\hat{\epsilon} = \epsilon_\theta(\mathbf{h}_t, t, DIEM(\mathbf{P}_l), \mathbf{c}_t) - \lambda \cdot \sqrt{1-\overline{\alpha}_t} \nabla_{\mathbf{h}_t} \log p_\varphi(i_1 \mid S_{\text{exp}})$;
  end for
  Reverse denoising utilising Eq. (5) based on S and $\hat{\epsilon}$;
  Obtain the expanded sequence $\hat{S}_{exp}$;
end for
return $\hat{S}_{exp}$.

3.7. Training and expansion

The overall PIFGSR procedure is detailed in Algorithm 1. During training, a subset $B' \subset B$ is constructed by selecting sequences longer than the predefined length J. For each sequence $S \in B'$, the first J items are extracted as the expansion target S_{exp}, while the remaining items constitute S_{ori}. A learnable parameter matrix $\mathbf{P}_l$ is optimized to enhance the DIEM-guided noise prediction function ϵ_θ. The forward process applies Gaussian noise to the latent sequence representations, modeled by a variance schedule $\beta_{1:T}$, and the backward process performs iterative denoising based on the learned score function and guidance from the pre-trained recommender p_φ. The model is updated using a diffusion-based loss $\mathcal{L}_{diff}$. In the expansion phase, the learned model performs reverse denoising on the original sequences in B, where each step refines the latent state with both DIEM-based structure-aware signals and

recommendation-informed gradients. The final output is an expanded interaction set $\hat{S}_{\text{exp}}$, synthesized from the generated trajectories.

3.8. Complexity analysis

3.8.1. Time complexity

For each diffusion step t, calculating Gaussian distribution parameters has a complexity of $O(M \cdot D)$, where M is the sequence length and D is the embedding dimension. With T diffusion steps, the forward process complexity is $O(T \cdot M \cdot D)$. The reverse process, using U-Net, has a per-step complexity of $O(M \cdot D^2 \cdot L)$, where L is the network depth, leading to a total complexity of $O(T \cdot M \cdot D^2 \cdot L)$. In the DIEM module, transforming the parameter matrix $\mathbf{P}_l$ ($M \times D$) and applying row-column and mean-variance attention mechanisms result in $O(M^2 \cdot D + M \cdot D^2)$. Converting $\mathbf{P}_l$ to the interest vector $\mathbf{c}_l$ adds $O(M \cdot D)$, making the total DIEM complexity $O(M^2 \cdot D + M \cdot D^2)$. In GMS, computing the gradient for each candidate item in a set of size $|I|$ has a complexity of $O(|I| \cdot D)$, and over T steps, the total GMS complexity is $O(T \cdot |I| \cdot D)$. Thus, the overall time complexity is $O(T \cdot M \cdot D(1 + D \cdot L) + (M^2 + M \cdot D + T \cdot |I|)D)$.

3.8.2. Space complexity

In the forward process, storing the embedding sequence $\mathbf{h}_t$ requires $O(M \cdot D)$, while noise scheduling parameters β_t add a negligible $O(T)$. Thus, the forward process has a space complexity of $O(M \cdot D)$. In the reverse process, storing $\mathbf{h}_t$ and noise predictions $\epsilon_\theta(\mathbf{h}_t, t)$ requires $O(M \cdot D)$, and the U-Net model adds $O(U)$. Therefore, the reverse process has a total complexity of $O(M \cdot D + U)$. For DIEM, the parameter matrix $\mathbf{P}_l$ requires $O(M \cdot D)$, while attention mechanisms and intermediate results contribute $O(M^2 + M \cdot D)$. Thus, DIEM's complexity is $O(M^2 + M \cdot D)$. In GMS, gradient storage for $\nabla_{\mathbf{h}_t} \log p_\varphi(i_1 \mid S_{\text{exp}})$ and the classifier model contribute $O(M \cdot D)$ and $O(C)$, respectively. Therefore, GMS's complexity is $O(M \cdot D + C)$. Overall, the framework's space complexity is: $O(M \cdot D + U + M^2 + C)$.

4. Performance evaluation

In this section, we do comprehensive experiments to investigate the following research questions:

- **RQ1**: How does the performance of PIFGSR compare to established baselines?
- **RQ2**: What is the impact of the primary components of PIFGSR on its performance?
- **RQ3**: What is the impact of various hyperparameter configurations on PIFGSR?
- **RQ4**: How generalizable is PIFGSR?
- **RQ5**: Can PIFGSR mitigate the issue of data sparsity?

4.1. Experimental setup

4.1.1. Datasets

To assess the effectiveness of PIFGSR, we conduct experiments on four widely-used real-world datasets: Yelp [40], Beauty, Sport, and Toys [41]. Experiments are performed using public datasets, where reviews or ratings are regarded as implicit interactions that represent the interactions between users and items, and user sequences are constructed by ordering user interactions chronologically. To ensure data quality and meaningful evaluation, users with fewer than five interactions are filtered out. The datasets' statistical information is presented in Table 2.

Table 2
The statistics of datasets after preprocessing.

Datasets	#users	#items	#actions	avg.length	density
Yelp	30,431	20,033	1,094,826	11.5	0.04 %
Beauty	22,363	12,101	198,502	8.8	0.07 %
Sport	35,598	18,357	296,337	8.3	0.05 %
Toys	19,412	11,924	167,597	8.6	0.07 %

4.1.2. Evaluation metrics

We evaluate PIFGSR and baselines using *HR*@k and *NDCG*@k for k = 5 and 10 with leave-one-out methodology, reserving the final and penultimate items for testing and validation. P-values are used in statistical significance testing to confirm how well PIFGSR has improved performance. The null hypothesis is that the median difference between PIFGSR and the suboptimal method is negative.

4.1.3. Baselines

To illustrate the efficacy of PIFGSR framework, we compare it with three categories of baseline methods: 1) Sequential recommender methods such as DuoRec [25], AdaGCL[26], ContraRec [27], MCLRec [28], BSARec [42]; 2) Diffusion-based methods such as DiffuASR [31], DiffRec [32], DreamRec [33], CDDRec [34]; 3) Plugin-based diffusion methods such as SeeDRec [35], DR4SR [43], PDRec [36]. The detailed description of the baseline is as follows:

(i) Sequential recommender methods:

- DuoRec [25] improves the distribution of item embeddings generated by sequential deep learning models such as Transformer and BERT through regularizing sequence representations.
- AdaGCL [26] introduces two adaptive contrastive view generators to provide additional high-quality training signals for collaborative filtering.
- ContraRec [27] proposes a contextual contrastive task to encourage similar representations between augmented sequences and sequences with the same target item.
- MCLRec [28] adopts a meta-learning approach to guide the model enhancer's updates, thereby adaptively capturing useful features hidden in random data augmentations.
- BSARec [42] leverages Fourier transform to inject inductive bias of fine-grained sequential patterns and integrates low-frequency and high-frequency information to mitigate the over-smoothing issue in the self-attention mechanism.

(ii) Diffusion-based methods:

- DiffuASR [31] utilizes diffusion models to generate pseudo sequences to alleviate data sparsity and the long-tail user problem in recommender systems.
- DiffRec [32] employs diffusion models to learn the generative process of user interactions through denoising and reduces noise introduced during generation to preserve personalized user information.
- DreamRec [33] adds noise to target items to explore the latent distribution in the item space, directly reflecting users' true preferences and avoiding the limitations of negative sampling.
- CDDRec [34] divides the optimization and generation processes into more manageable autoregressive steps to produce high-quality sequence and item representations.

(iii) Plugin-based diffusion methods:

- SeeDRec [35] upgrades the diffusion objective from the item level to the semantic unit level to better guide the modeling of each user's sequential behavior.
- DDRM [44] improves the robustness of user and item embeddings by injecting controllable Gaussian noise in the forward process and iteratively eliminating the noise in the reverse multi-step denoising process.
- GlobalDiff [45] superimposes a diffusion model on top of a sequence model, uses a vectorized sample conversion mechanism to learn the global association and sequence position information of items, and combines gradual noise addition to enhance the model's personalized expression and training stability.
- DR4SR [43] employs a model-agnostic dataset regeneration framework to generate ideal training datasets with excellent cross-architecture generalization capabilities, effectively capturing the evolving preferences of users.

Table 3
The performance of different models. Optimal outcomes are indicated in **bold**, whereas suboptimal results are underlined. H represents *HR*, while N represents *NDCG*.

Model	Yelp				Beauty				Sport				Toys			
	H@5	H@10	N@5	N@10	H@5	H@10	N@5	N@10	H@5	H@10	N@5	N@10	H@5	H@10	N@5	N@10
DuoRec	0.0155	0.0257	0.0119	0.0138	0.0368	0.0612	0.0256	0.0341	0.0227	0.0359	0.0143	0.0165	0.0432	0.0629	0.0316	0.0381
AdaGCL	0.0153	0.0248	0.0113	0.0132	0.0361	0.0594	0.0239	0.0327	0.0219	0.0325	0.0135	0.0161	0.0418	0.0610	0.0285	0.0368
ContraRec	0.0197	0.0291	0.0115	0.0162	0.0434	0.0685	0.0269	0.0339	0.0247	0.0398	0.0141	0.0192	0.0481	0.0662	0.0317	0.0384
MCLRec	0.0213	0.0302	0.0136	0.0168	0.0441	0.0675	0.0264	0.0345	0.0236	0.0391	0.0152	0.0189	0.0485	0.0678	0.0315	0.0391
BSARec	0.0219	0.0305	0.0142	0.0175	0.0446	0.0703	0.0288	0.0359	0.0243	0.0405	0.0155	0.0194	0.0497	0.0693	0.0324	0.0408
DiffuASR	0.0161	0.0271	0.0096	0.0126	0.0395	0.0601	0.0243	0.0331	0.0206	0.0338	0.0139	0.0157	0.0429	0.0625	0.0296	0.0372
DiffRec	0.0168	0.0275	0.0121	0.0140	0.0399	0.0649	0.0268	0.0335	0.0234	0.0342	0.0147	0.0172	0.0437	0.0634	0.0302	0.0385
DreamRec	0.0183	0.0284	0.0125	0.0148	0.0437	0.0692	0.0281	0.0346	0.0253	0.0366	0.0149	0.0184	0.0464	0.0638	0.0321	0.0394
CDDRec	0.0204	0.0298	0.0117	0.0166	0.0429	0.0668	0.0270	0.0336	0.0241	0.0373	0.0137	0.0176	0.0472	0.0654	0.0309	0.0376
SeeDRec	0.0219	0.0301	0.0134	0.0165	0.0452	0.0689	0.0286	0.0354	0.0254	0.0403	0.0154	0.0193	0.0495	0.0681	0.0326	0.0402
DDRM	0.0216	0.0302	0.0131	0.0160	0.0446	0.0684	0.0281	0.0349	0.0247	0.0398	0.0146	0.0188	0.0489	0.0676	0.0322	0.0401
GlobalDiff	0.0221	0.0306	0.0137	0.0169	0.0458	0.0706	0.0287	0.0355	0.0259	0.0411	0.0156	0.0195	0.0501	0.0689	0.0334	0.0412
DR4SR	0.0223	0.0304	0.0139	0.0161	0.0442	0.0695	0.0284	0.0347	0.0258	0.0410	0.0149	0.0185	0.0496	0.0687	0.0329	0.0411
PDRec	0.0225	0.0307	0.0140	0.0171	0.0460	0.0712	0.0295	0.0362	0.0261	0.0412	0.0158	0.0197	0.0505	0.0685	0.0337	0.0415
PIFGSR	**0.0246**	**0.0329**	**0.0156**	**0.0184**	**0.0494**	**0.0735**	**0.0318**	**0.0381**	**0.0274**	**0.0433**	**0.0168**	**0.0211**	**0.0521**	**0.0726**	**0.0354**	**0.0435**
Improv.	9.33%	7.16%	9.85%	5.14%	7.39%	3.23%	7.79%	5.24%	4.98%	5.09%	6.33%	7.10%	3.17%	4.76%	5.04%	4.82%
p-value	0.035	0.039	0.043	0.031	0.027	0.046	0.038	0.043	0.021	0.029	0.025	0.032	0.018	0.020	0.014	0.022

Table 4
The computational complexity of PIFGSR and plugin-based diffusion methods on four datasets.

	Model	GPU (MB)	Parameters (M)	Training Time (s)	Inference Time (s)
Yelp	SeeDRec	4245	364.92	483.07	2432.51
	DDRM	4537	386.14	521.45	2667.63
	GlobalDiff	6213	511.28	650.73	3516.58
	DR4SR	6352	528.75	682.34	3702.86
	PDRec	4709	453.61	536.17	2751.19
	PIFGSR	**3841**	**329.58**	**419.64**	**1825.42**
Beauty	SeeDRec	1051	81.79	105.16	495.91
	DDRM	1124	93.91	114.05	502.34
	GlobalDiff	1518	106.24	129.38	518.15
	DR4SR	1572	124.77	134.54	522.42
	PDRec	1197	93.58	110.32	537.56
	PIFGSR	**914**	**69.32**	**85.53**	**378.57**
Sport	SeeDRec	1486	100.17	195.55	726.50
	DDRM	1530	114.06	219.65	756.24
	GlobalDiff	2106	135.41	231.84	825.56
	DR4SR	2254	146.13	264.62	881.43
	PDRec	1635	122.06	217.16	750.32
	PIFGSR	**1239**	**85.96**	**143.82**	**537.46**
Toys	SeeDRec	1025	76.78	111.13	482.48
	DDRM	1134	90.14	128.30	506.53
	GlobalDiff	1306	95.35	137.48	542.97
	DR4SR	1378	105.65	140.16	585.14
	PDRec	1101	84.71	103.67	452.26
	PIFGSR	**849**	**61.43**	**82.94**	**342.19**

- PDRec [36] employs a reweighting mechanism on historical behaviors to identify high-quality behaviors and suppress noisy ones. It also leverages top-ranked unobserved items as potential positive samples, introducing informative and diverse soft signals.

4.1.4. Implementation details

PIFGSR is executed on a server with NVIDIA A100 GPUs, using Pytorch version 1.7.1. PIFGSR's backbone recommender network and the pre-trained sequential recommender model in GMS both use SASRec, with baseline parameters set as in the original papers. We optimize key parameters through a grid search: guidance strength λ from $\{0.1, 0.3, 0.5, 0.7, 1\}$, sequence expansion number M from $\{2, 4, 6, 8, 10\}$, noise scale z from $\{0.2, 0.4, 0.6, 0.8, 1\}$, and fusion coefficient γ from $\{0.1, 0.2, 0.4, 0.6, 0.8\}$. The embedding dimension is set to 64, batch size up to 256, Adam optimizer's learning rate is 0.001, and total diffusion steps T are 100. The dataset's maximum sequence length is 200. Each experiment is repeated 10 times, with average results presented.

4.2. Overall comparison (RQ1)

We present the performance of PIFGSR in comparison to many baseline models across four datasets in Table 3. Based on the results, we draw the following conclusions: PIFGSR routinely surpasses all baselines on most metrics and datasets. It outperforms baselines across metrics with significant p-values less than 0.05, achieving gains of up to 9.33% in *HR* and 7.16% in *NDCG*. These consistent improvements stem from a combination of complementary design choices. First, the Potential Interest Guidance Bank enables the framework to fully exploit knowledge embedded in the pre-trained diffusion model while efficiently integrating user-specific interaction information. By steering the reverse generation process with this enriched guidance, PIFGSR produces augmented sequences that reflect both general behavioral patterns and individual user preferences. Second, the Demand Information Extraction Mechanism captures fine-grained local patterns as well as holistic global structures within user interaction histories. This dual-level understanding ensures that both short-term contextual shifts and long-term preference

Table 5
The results of ablation study for PIFGSR across four datasets.

Model	Yelp				Beauty				Sport				Toys			
	H@5	H@10	N@5	N@10	H@5	H@10	N@5	N@10	H@5	H@10	N@5	N@10	H@5	H@10	N@5	N@10
w/o PIGB	0.0228	0.0312	0.0142	0.0168	0.0472	0.0719	0.0304	0.0369	0.0257	0.0415	0.0154	0.0196	0.0508	0.0692	0.0337	0.0415
w/o DIEM	0.0233	0.0318	0.0145	0.0176	0.0479	0.0722	0.0308	0.0372	0.0262	0.0420	0.0158	0.0199	0.0512	0.0702	0.0343	0.0422
w/o GMS	0.0236	0.0321	0.0151	0.0179	0.0482	0.0726	0.0311	0.0375	0.0265	0.0423	0.0160	0.0203	0.0515	0.0708	0.0345	0.0426
w/o RS	0.0242	0.0326	0.0154	0.0181	0.0489	0.0730	0.0315	0.0379	0.0268	0.0427	0.0162	0.0207	0.0518	0.0715	0.0349	0.0431
PIFGSR	**0.0246**	**0.0329**	**0.0156**	**0.0184**	**0.0494**	**0.0735**	**0.0318**	**0.0381**	**0.0274**	**0.0433**	**0.0168**	**0.0211**	**0.0521**	**0.0726**	**0.0354**	**0.0435**

trajectories are accurately modeled, overcoming the tendency of traditional approaches to focus narrowly on either local or global information. Last, the Guide Matching Strategy aligns the interests expressed in the expanded sequences with those in the original sequences, reducing interest drift and maintaining semantic fidelity.

The expanded sequences generated by diffusion models are not always beneficial for recommender systems. A comparison between DiffuASR and MCLRec reveals that expanded sequences misaligned with actual user interests can significantly limit the performance of recommender models. A potential cause for this issue is that DiffuASR frequently introduces noisy data during the reverse process, which results in the recommender model training on this flawed data. This, in turn, leads to a cumulative accumulation of errors over time, adversely affecting the overall recommender accuracy. In contrast, PDRec adopts a reweighting mechanism based on historical user behavior to effectively identify high-quality actions while suppressing the influence of noisy ones. This strategic approach allows PDRec to achieve performance that is second only to PIFGSR. This observation underscores the critical importance of guidance mechanisms in enhancing the effectiveness of expanded sequences within recommender systems.

We discover that PIFGSR performs exceptionally well with the sparse Yelp dataset compared to improvements seen in other datasets. Notably, PIFGSR also achieves strong performance on denser datasets. This success can be attributed to PIFGSR's effective utilization of the diffusion model and the seamless integration of its components, enabling the framework to generate expanded sequences that align closely with users' actual interest preferences. The experiments reveal significant limitations in both SeeDRec and PDRec, which generate expanded data for recommender models but exhibit biases that hinder their ability to accurately capture users' true interests. In particular, PDRec relies heavily on a historical behavior re-weighting mechanism to identify high-quality actions and suppress noisy behaviors, as well as on the use of top-ranked unobserved items as potential positive samples to introduce informative and diverse soft signals. However, this re-weighting mechanism may inadvertently overemphasize behaviors with higher historical weights, causing the model to neglect emerging or niche interests that are weakly represented in the interaction history (Fig. 2).

Table 4 also compares PIFGSR's computational complexity with plug-in methods, showing its superior one-epoch performance and validating its generalizability. We observe that in the sparsest dataset, Yelp, the training and inference times required by PIFGSR are relatively long. In contrast, in the denser datasets, Sport and Toys, the computational complexity of PIFGSR is significantly reduced. The reason for this difference lies in the sparsity of the Yelp dataset, which results in relatively scarce interaction information between users and items. As a result, PIFGSR requires more time to learn and extract users' potential interests. Conversely, in the Sport and Toys datasets, the interactions between users and items are more abundant. This abundance allows PIFGSR to identify and learn users' interest patterns more quickly.

4.3. Ablation study (RQ2)

To gain deeper insights into PIFGSR's design, we perform ablation studies by disabling specific components and constructing four variants: w/o RS removes the Rounding Strategy; w/o GMS disables the Guide Matching Strategy; w/o DIEM replaces DIEM with AdaAttN as described in Fig. 6. Unlike DIEM, which analyzes the user interaction matrix from both spatial and statistical perspectives to jointly capture local patterns through row-column attention and global features through mean-variance channel normalization, AdaAttN focuses on aligning feature distributions by combining high-level and low-level features to compute attention maps and applying weighted style feature statistics for point-wise adaptation; and w/o PIGB removes the Potential Interest Guidance Bank. These variants are compared against the full PIFGSR model, with results shown in Table 5, and the subsequent observations can be observed:

PIFGSR always demonstrates superior performance across all four datasets, indicating that each of the three components positively contributes to model performance. By comparing w/o GMS with PIFGSR, we observe that using GMS to ensure alignment between user interests in the expanded sequence and the original sequence leads to better performance. GMS helps maintain the coherence of user behavior, reducing noise unrelated to the user's actual interests.

PIFGSR outperforms w/o DIEM because w/o DIEM fails to extract information-centric data from both local patterns and global features in user interaction behaviors, limiting its ability to capture true user interests. Compared to PIFGSR, w/o PIGB generally results in poorer performance, emphasizing the importance of PIGB in improving the quality of expanded sequences. Disabling PIGB causes a significant performance decline.

4.4. Parameter analysis (RQ3)

4.4.1. Impact of guide strength λ

The experimental results are presented in Fig. 3. As the parameter λ increases, the performance of PIFGSR tends to decline gradually in most scenarios. This decline can be attributed to the fact that larger values of λ lead to generated items that closely resemble those found in the original interaction sequence. Such overly comparable samples limit the model's ability to incorporate a diverse range of user interests. In contrast, when λ is set to a smaller value, the user interest information embedded in the generated items aligns more effectively with the actual interests of the users, thereby enhancing the model's performance.

4.4.2. Impact of sequence expansion number M

The experimental results are presented in Fig. 4. When the parameter M is set to a large value, specifically 8 or 10, PIFGSR demonstrates a significant decline in performance. This deterioration can be attributed to the fact that an excessive number of generated items causes the model to overfit specific features of these generated samples, thereby undermining its ability to generalize effectively to the original interaction sequence. In contrast, a moderate expansion of generated items provides a sufficient number of positive samples, particularly in scenarios characterized by data sparsity. This balanced approach enhances the model's capacity to assimilate a diverse range of user preferences, ultimately leading to improved performance.

4.4.3. Impact of fusion coefficient γ

The experimental results are presented in Fig. 5. In most cases, when the parameter γ is set to a small value, specifically 0.2 or 0.4, PIFGSR

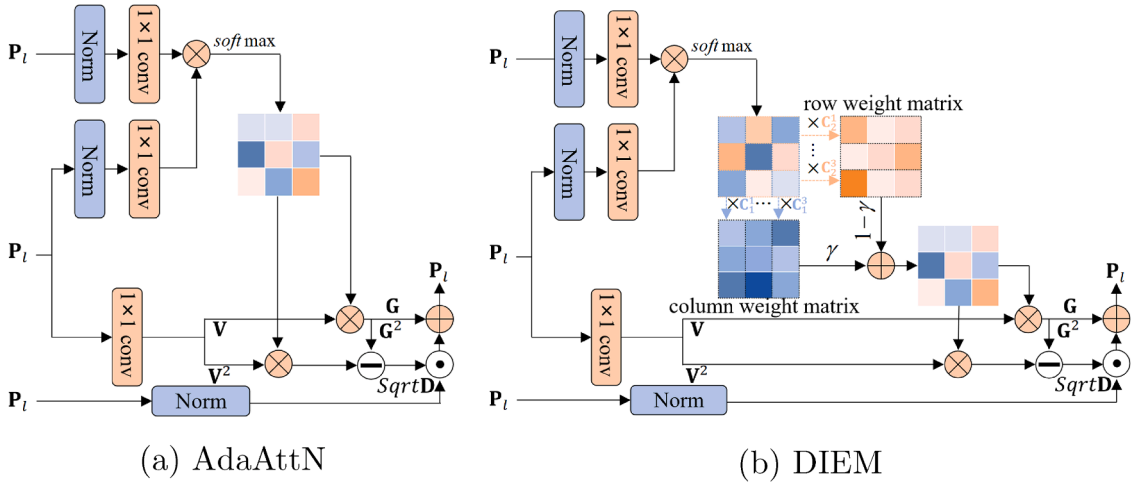

Fig. 2. Schematic diagram of the difference between AdaAttN and DIEM.

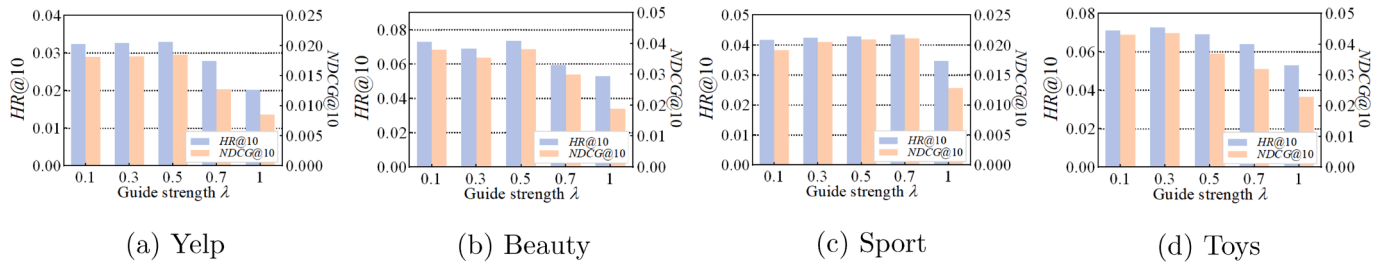

Fig. 3. Impact of guide strength λ on four datasets.

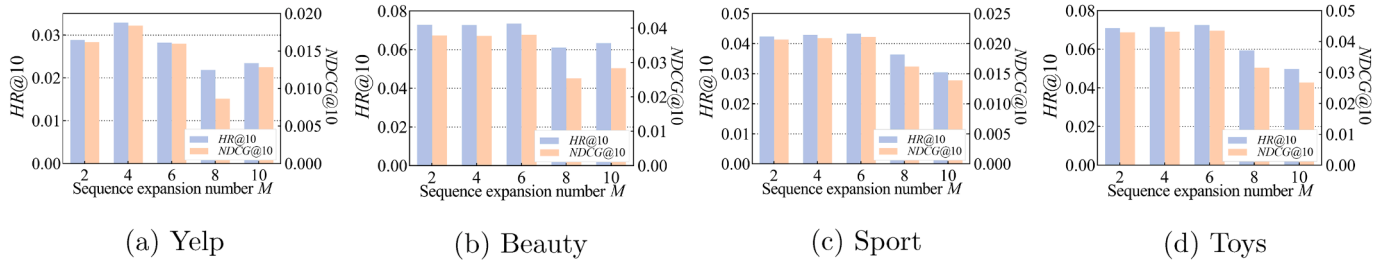

Fig. 4. Impact of sequence expansion number M on four datasets.

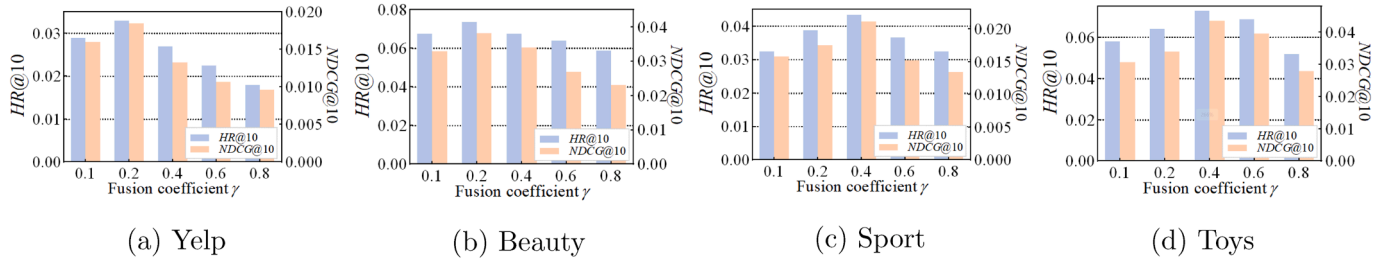

Fig. 5. Impact of fusion coefficient γ on four datasets.

demonstrates the best performance. This observation can be attributed to the functionality of the row-wise attention matrix, which effectively focuses on global information across each column, thereby capturing the overall relationships among various features. Meanwhile, the column-wise attention matrix emphasizes local differences between samples. In scenarios characterized by data sparsity and elevated noise levels, the incorporation of a greater proportion of the row-wise attention matrix plays a crucial role in effectively smoothing out anomalous data, leading to improved model performance.

4.4.4. Impact of noise scale z

The experimental results are presented in Fig. 6. In most cases, the model achieves the best results when z is set to a moderate value such as 0.4 or 0.6, indicating that an appropriate noise level can balance the trade-off between preserving sequential fidelity and exploring potential but unobserved user interests. When z is too small, the injected noise is insufficient to enrich the sparse interaction sequences, limiting the model's ability to generalize; conversely, an excessively large z may introduce disruptive perturbations that distort user preference

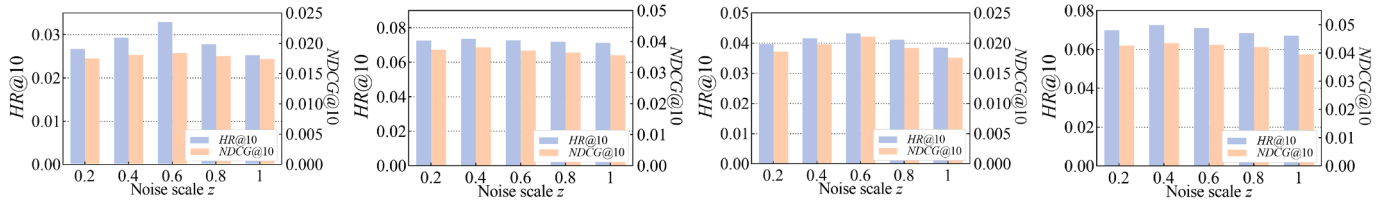

Fig. 6. Impact of noise scale z on four datasets.

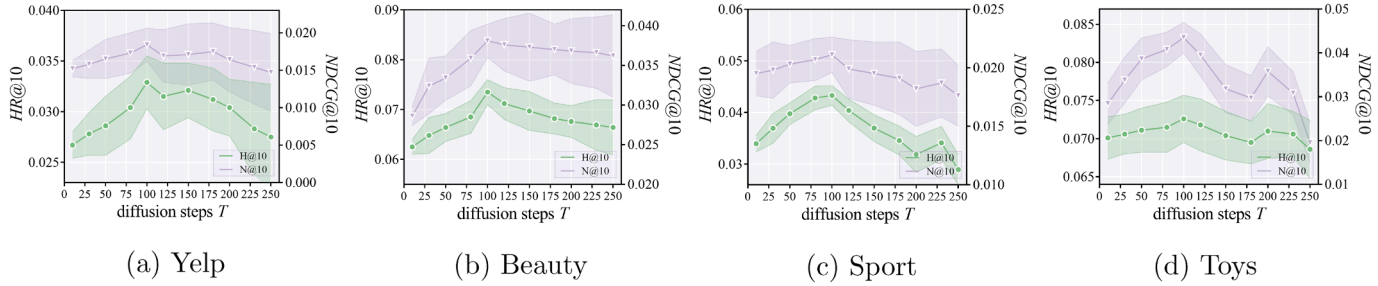

Fig. 7. Impact of different diffusion steps T on four datasets.

trajectories. Notably, on the relatively sparse Yelp dataset, a larger value of z is required to reach peak performance, suggesting that additional perturbation is beneficial in scenarios where user interaction histories are limited, as it helps uncover latent preferences.

4.4.5. Impact of diffusion steps T

To evaluate the impact of diffusion steps on model performance, we systematically train PIFGSR using a range of step counts, specifically from 10 to 250. As illustrated in Fig. 7, the results obtained from four datasets reveal a notable trend: performance tends to improve with an increasing number of diffusion steps, particularly during the initial stages of the training process. This enhancement can be attributed to the introduction of beneficial noise through additional diffusion steps, which plays a crucial role in retaining both explicit and implicit user feedback. Notably, the model achieves its peak performance at 100 diffusion steps, where it attains a local optimum. This optimal point is characterized by an effective balance between the denoising process and the preservation of essential user interest information, thereby maximizing the model's overall efficacy.

4.5. Generalizability analysis (RQ4)

To further substantiate the claim that PIFGSR functions as a model-agnostic and broadly applicable framework, we conduct experiments with GRU4Rec [46], Caser [23], and SASRec [24] as backbone models across four distinct datasets. As shown in Table 6, PIFGSR consistently yields significant performance gains across all backbones and datasets, which demonstrates that its benefits are not tied to a specific architecture. This universality stems from the design of PIFGSR: the pre-trained diffusion model provides a general backbone for sequence generation, PIGB captures personalized user interests, DIEM offers interpretable latent codes through both spatial and statistical analysis, and GMS enforces alignment with users' actual preferences. Together, these modules form a plug-and-play augmentation mechanism that can be seamlessly integrated into diverse sequential recommenders. Notably, the strongest improvements are observed on the sparsest dataset Yelp, highlighting the robustness of PIFGSR in alleviating data sparsity by generating augmented sequences that remain faithful to true user interests.

Table 6
Comparative analysis of PIFGSR against several sequential recommender models.

Model	Yelp		Beauty		Sport		Toys	
	H@10	N@10	H@10	N@10	H@10	N@10	H@10	N@10
GRU4Rec	0.0269	0.0151	0.0368	0.0234	0.0262	0.0139	0.0274	0.0143
+PIFGSR	0.0308	0.0166	0.0412	0.0256	0.0296	0.0154	0.0287	0.0159
Caser	0.0264	0.0149	0.0359	0.0245	0.0273	0.0152	0.0281	0.0152
+PIFGSR	0.0304	0.0172	0.0407	0.0276	0.0314	0.0171	0.0295	0.0167
SASRec	0.0278	0.0156	0.0631	0.0317	0.0365	0.0178	0.0668	0.0384
+PIFGSR	0.0329	0.0184	0.0735	0.0381	0.0433	0.0211	0.0726	0.0435

4.6. In-depth analysis (RQ5)

4.6.1. Different sequence lengths analysis

We present the recommender performance at *HR*@10 for different user groups based on varying historical interaction counts, as illustrated in Fig. 8. In the Yelp dataset, users are systematically categorized into five distinct groups according to their interaction counts N: short ($3 \leq N \leq 5$), mid-short ($5 \leq N \leq 6$), medium ($6 \leq N \leq 7$), mid-long ($7 < N \leq 10$), and long ($N > 10$). This categorization allows for a nuanced analysis of the model's performance across different user engagement levels.

We compare PIFGSR with DR4SR and PDRec and observe consistent gains across all cohorts grouped by history length, demonstrating strong robustness in sequential recommendation. The advantage is most evident for users with short or moderate histories, where sparsity is acute, indicating that sequence expansion effectively enriches limited interaction signals. As sequences grow longer, PIFGSR maintains steady gains, showing that the expansion strategy not only mitigates sparsity but also improves representation quality for users with richer histories. Results further show that SASRec performance rises with added expansion, confirming that sequence level augmentation strengthens the model's ability to capture both short-term and long-term dependencies under diverse data conditions.

4.6.2. Further analysis of sparsity improvement

In the Yelp dataset, we define the top 30 % of items by interaction frequency as head items and the remainder as tail items to assess expansion quality and sparsity mitigation. As shown in Fig. 9, compared with PDRec and DR4SR, all methods perform better on head items because

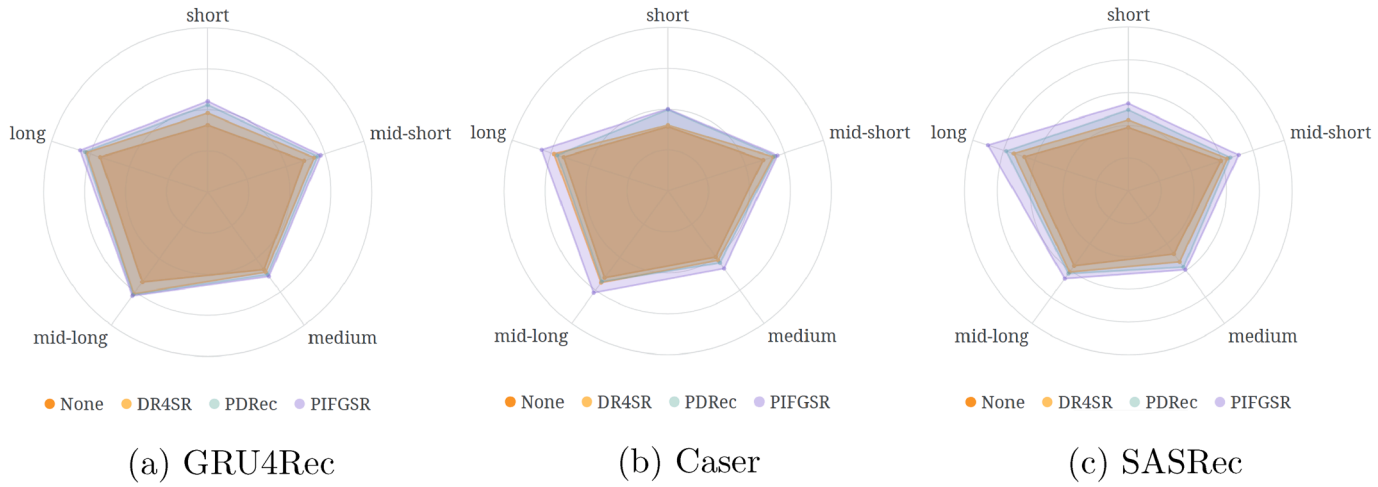

Fig. 8. Performance after grouping users on the Yelp dataset.

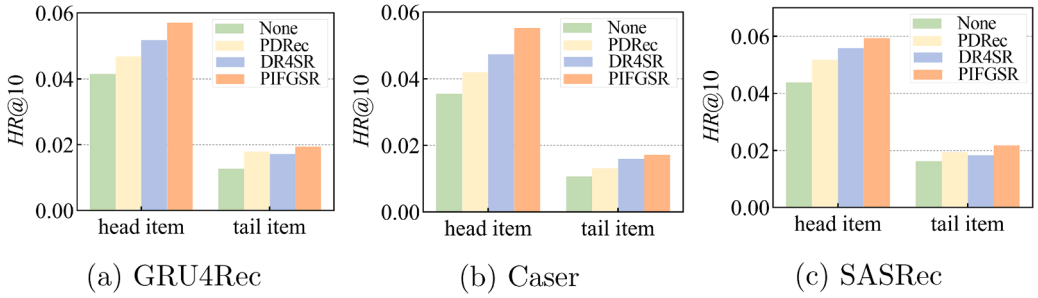

Fig. 9. Performance after dividing user interactions into head and tail items on the Yelp dataset.

they benefit from denser interaction signals. Notably, PIFGSR attains the best results on both head and tail, indicating user-aligned expansions that effectively alleviate sparsity. Its robust performance across item categories underscores its value for recommender systems, especially when interaction data is limited.

5. Conclusion

This paper presents PIFGSR, a pluggable framework that overcomes key limitations of diffusion-based sequential recommenders by tightly coupling three purpose-built modules. PIGB learns and caches potential-interest guidance from user interactions and fuses it with priors from a pre-trained diffusion model, thereby conditioning generation to reduce noise and redundancy while preserving preference trajectories. DIEM extracts users' true points of interest by jointly modeling local structure and global distribution, improving the fidelity of sequence representations delivered to the generator. GMS injects task-aware guidance during the reverse process to align expanded sequences with interests encoded in the original history, preventing drift and maintaining consistency. Extensive experiments on four real-world datasets show that PIFGSR improves recommendation accuracy by 9.85 %, with particularly strong gains for users with short histories, and does so with low computational cost, underscoring its practicality. Future work will focus on improving PIFGSR's adaptability and timeliness in multi-domain scenarios by advancing cross-domain joint modeling for stronger generalization, designing online incremental expansion for evolving user behaviors, and incorporating fine-grained interest modeling with multimodal fusion to enhance accuracy and personalization.

CRediT authorship contribution statement

Yicheng Di: Writing – original draft, Validation, Methodology; **Hongjian Shi:** Investigation, Conceptualization; **Khushal Haider Syed:** Writing – original draft, Resources; **Ruhui Ma:** Supervision, Funding acquisition; **Haibing Guan:** Validation, Funding acquisition; **Yuan Liu:** Formal analysis, Conceptualization; **Rajkumar Buyya:** Writing – original draft, Validation.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was funded by the National Natural Science Foundation of China under grant number (No. 62472200, No.61972182). This work was supported in part by the Shanghai Key Laboratory of Scalable Computing and Systems. This research was funded by project supported by Shanghai Municipal Science and Technology Major Project.

References

[1] Y. Di, H. Shi, R. Ma, H. Gao, Y. Liu, W. Wang, FedRL: a reinforcement learning federated recommender system for efficient communication using reinforcement selector and hypernet generator, ACM Trans. Recommender Syst. 4 (1) (2025) 1–31.

[2] J. Chen, J. Wang, F. Zhang, Y. Cao, K. Wei, L. Huang, H. Zhu, F. Kou, H. Huang, Dual-view collaboration fusion on diffusion learning for bundle recommendation, Inf. Fusion 4 (1) (2025) 31.
[3] Z. Jiang, Z. Cai, H. Zhao, B. Zhang, T. Wu, Y. Qiang, R. Ma, H. Guan, R. Buyya, HeShare: energy-aware and efficient multi-task GPU sharing in heterogeneous GPU-based computing systems, IEEE Trans. Comput. (5555) 1 (1) 1–12.
[4] L. Jin, Z. Cai, H. Wang, Z. Zhang, R. Ma, H. Guan, Y. Liu, B. Rajkumar, Ephemera: accelerating I/O-intensive serverless workloads with a harvested in-memory file system, ACM Trans. Archit. Code Optim. 22 (3) (2025)1544–3566.
[5] Y. Di, X. Wang, H. Shi, C. Fan, R. Zhou, R. Ma, Y. Liu, Personalized consumer federated recommender system using fine-grained transformation and hybrid information sharing, IEEE Trans. Consum. Electron. 71 (2) (2025)7254–7268.
[6] Y. Feng, Y.-a. Geng, Y. Zhu, Z. Han, X. Yu, K. Xue, H. Luo, M. Sun, G. Zhang, M. Song, PM-MOE: mixture of experts on private model parameters for personalized federated learning, in: Proceedings of the ACM on Web Conference 2025, 2025, pp. 134–146.
[7] L. Lin, R. Ma, Z. Wang, Z. Cai, H. Xu, B. Zhang, R. Ma, R. Buyya, HWDSQP: a historical weighted and dynamic scheduling quantum protocol to enhance communication reliability, IEEE J. Sel. Areas Commun.43 (8) (2025)2810–2824.
[8] J. Liu, Z. Cai, Y. Liu, H. Li, Z. Zhang, R. Ma, R. Buyya, SMore: enhancing GPU utilization in deep learning clusters by serverless-based co-location scheduling, IEEE Trans. Parallel Distrib. Syst. 36 (5) (2025)903–917.
[9] Z. Xing, Q. Feng, H. Chen, Q. Dai, H. Hu, H. Xu, Z. Wu, Y.-G. Jiang, A survey on video diffusion models, ACM Comput. Surv. 57 (2) (2024) 1–42.
[10] A. Ma, J. Feng, K. Cao, J. Wang, Y. Wang, Q. Zhang, Z. Zhang, Lay2Story: extending diffusion transformers for layout-togglable story generation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2025, pp. 16102–16111.
[11] Z. Zhang, Q. Zhang, G. Li, J. Luan, M. Yang, Y. Wang, L. Zhao, DyArtbank: diverse artistic style transfer via pre-trained stable diffusion and dynamic style prompt artbank, Knowl. Based Syst. 310 (2025) 112959.
[12] L. Gong, S. Guo, Y. Lin, Y. Liu, E. Zheng, Y. Shuang, Y. Lin, J. Hu, H. Wan, STCDM: spatio-temporal contrastive diffusion model for check-in sequence generation, IEEE Trans. Knowl. Data Eng.37 (4) (2025)2141–2154. .
[13] W. Liao, Y. Zhu, Y. Li, Q. Zhang, Z. Ou, X. Li, RevGNN: negative sampling enhanced contrastive graph learning for academic reviewer recommendation, ACM Trans. Inf. Syst. 43 (1) (2024) 1–26.
[14] L. Chen, W. Yuan, T. Chen, G. Ye, N.Q.V. Hung, H. Yin, Adversarial item promotion on visually-aware recommender systems by guided diffusion, ACM Trans. Inf. Syst. 42 (6) (2024) 1–26.
[15] Z. Chen, Y. Zheng, J.C. Gee, TransMatch: a transformer-based multilevel dual-stream feature matching network for unsupervised deformable image registration, IEEE Trans. Med. Imaging 43 (1) (2023) 15–27.
[16] S. Zhao, D. Chen, Y.-C. Chen, J. Bao, S. Hao, L. Yuan, K.-Y.K. Wong, Uni-ControlNet: all-in-one control to text-to-image diffusion models, Adv. Neural Inf. Process. Syst. 36 (2024), pp. 11127–11150.
[17] D. Zhang, S. Zheng, Y. Zhu, H. Yuan, J. Gong, J. Tang, MCAP: low-pass GNNs with matrix completion for academic recommendations, ACM Trans. Inf. Syst. 43 (2) (2025) 1–29.
[18] N.S. Hada, S. Maloth, C. Jatoth, U. Fiore, S. Sharma, S. Chatharasupalli, R. Buyya, A novel recommendation system for vaccines using hybrid machine learning model, in: Machine Intelligence Techniques for Data Analysis and Signal Processing: Proceedings of the 4th International Conference MISP 2022, Volume 1, Springer, 2023, pp. 433–442.
[19] S. Bera, T. Dey, A. Mukherjee, R. Buyya, E-CropReco: a dew-edge-based multiparametric crop recommendation framework for internet of agricultural things, J. Supercomput. 79 (11) (2023) 11965–11999.
[20] T. Peng, H. Yuan, Y. Zhang, Y. Li, P. Dai, Q. Wang, S. Wang, W. Wu, TagRec: temporal-aware graph contrastive learning with theoretical augmentation for sequential recommendation, IEEE Trans. Knowl. Data Eng. 37 (5) (2025) 3015–3029.
[21] H. Shi, Y. Di, X. Ruan, M. Liao, Q. Zhang, R. Ma, H. Guan, Efficient federated recommender system with adaptive model pruning and momentum-based batch adjustment, ACM Trans. Recommender Syst. 3 (3) (2025) 1–23.
[22] W. Li, J. Cao, J. Wu, C. Huang, R. Buyya, A collaborative filtering recommendation method based on discrete quantum-inspired shuffled frog leaping algorithms in social networks, Future Gener. Comput. Syst. 88 (2018) 262–270.
[23] J. Tang, K. Wang, Personalized top-n sequential recommendation via convolutional sequence embedding, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, 2018, pp. 565–573.
[24] W.-C. Kang, J. McAuley, Self-attentive sequential recommendation, in: 2018 IEEE International Conference on Data Mining (ICDM), 2018, pp. 197–206.
[25] R. Qiu, Z. Huang, H. Yin, Z. Wang, Contrastive learning for representation degeneration problem in sequential recommendation, in: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, 2022, pp. 813–823.
[26] Y. Jiang, C. Huang, L. Huang, Adaptive graph contrastive learning for recommendation, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 4252–4261.
[27] C. Wang, W. Ma, C. Chen, M. Zhang, Y. Liu, S. Ma, Sequential recommendation with multiple contrast signals, ACM Trans. Inf. Syst. 41 (1) (2023) 1–27.
[28] X. Qin, H. Yuan, P. Zhao, J. Fang, F. Zhuang, G. Liu, Y. Liu, V. Sheng, Meta-optimized contrastive learning for sequential recommendation, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2023, pp. 89–98.
[29] Y. Di, H. Shi, X. Wang, R. Ma, Y. Liu, Federated recommender system based on diffusion augmentation and guided denoising, ACM Trans. Inf. Syst. 43 (2) (2025) 1–36.
[30] X. He, W. Fan, R. Wang, Y. Wang, Y. Wang, S. Pan, X. Wang, Balancing user preferences by social networks: a condition-guided social recommendation model for mitigating popularity bias, Neural Netw. 187 (2025) 107317.
[31] Q. Liu, F. Yan, X. Zhao, Z. Du, H. Guo, R. Tang, F. Tian, Diffusion augmentation for sequential recommendation, in: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, 2023, pp. 1576–1586.
[32] W. Wang, Y. Xu, F. Feng, X. Lin, X. He, T.-S. Chua, Diffusion recommender model, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2023, pp. 832–841.
[33] Z. Yang, J. Wu, Z. Wang, X. Wang, Y. Yuan, X. He, Generate what you prefer: reshaping sequential recommendation via guided diffusion, in: Advances in Neural Information Processing Systems, 36, 2023, pp. 24247–24261.
[34] Y. Wang, Z. Liu, L. Yang, P.S. Yu, Conditional denoising diffusion for sequential recommendation, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2024, pp. 156–169.
[35] H. Ma, R. Xie, L. Meng, Y. Yang, X. Sun, Z. Kang, SeeDRec: sememe-based diffusion for sequential recommendation, in: Proceedings of IJCAI, 2024, pp. 1–9.
[36] H. Ma, R. Xie, L. Meng, X. Chen, X. Zhang, L. Lin, Z. Kang, Plug-in diffusion model for sequential recommendation, in: Proceedings of the AAAI Conference on Artificial Intelligence, 38, 2024, pp. 8886–8894.
[37] D.P. Kingma, Auto-encoding variational Bayes, arXiv:1312.6114 (2013).
[38] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, Adv. Neural Inf. Process. Syst. 33 (2020) 6840–6851.
[39] L. Zhang, A. Rao, M. Agrawala, Adding conditional control to text-to-image diffusion models, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 3836–3847.
[40] Z. Cheng, X. Chang, L. Zhu, R.C. Kanjirathinkal, M. Kankanhalli, MMALFM: explainable recommendation by leveraging reviews and images, ACM Trans. Inf. Syst. 37 (2) (2019) 1–28.
[41] J. McAuley, C. Targett, Q. Shi, A. Van Den Hengel, Image-based recommendations on styles and substitutes, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2015, pp. 43–52.
[42] Y. Shin, J. Choi, H. Wi, N. Park, An attentive inductive bias for sequential recommendation beyond the self-attention, in: Proceedings of the AAAI Conference on Artificial Intelligence, 38, 2024, pp. 8984–8992.
[43] M. Yin, H. Wang, W. Guo, Y. Liu, S. Zhang, S. Zhao, D. Lian, E. Chen, Dataset regeneration for sequential recommendation, in: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024, pp. 3954–3965.
[44] J. Zhao, W. Wenjie, Y. Xu, T. Sun, F. Feng, T.-S. Chua, Denoising diffusion recommender model, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2024, pp. 1370–1379.
[45] M. Luo, Y. Li, C. Lin, Enhancing sequential recommendation with global diffusion, in: Proceedings of the AAAI Conference on Artificial Intelligence, 39, 2025, pp. 12309–12318.
[46] B. Hidasi, Session-based recommendations with recurrent neural networks, arXiv:1511.06939 (2015).