Adaptive User Dynamic Interest Guidance for Generative Sequential Recommendation

Kai Zhu
zhukai-cs@whu.edu.cn
School of Computer Science
Wuhan University, Wuhan, China

Jing Li*
leejingcn@whu.edu.cn
National Engineering Research Center for Multimedia Software
Wuhan University, Wuhan, China

Jia Wu
jia.wu@mq.edu.au
School of Computing, Faculty of Science and Engineering, Macquarie University, Sydney, Australia

Yue He*
yuehe.cs@whu.edu.cn
School of Computer Science
Wuhan University, Wuhan, China

Jun Chang
chang.jun@whu.edu.cn
School of Computer Science
Wuhan University, Wuhan, China

Guohao Li
ghli156@whu.edu.cn
School of Computer Science
Wuhan University, Wuhan, China

Shuyi Zhang
2024102110015@whu.edu.cn
School of Computer Science
Wuhan University, Wuhan, China

Abstract

Recently, diffusion model-based methods have utilized user interest features as guidance conditions to achieve stable generation results in sequential recommendation tasks. However, these models struggle to capture users' dynamic interests, as the interests of different users are often inconsistent. Moreover, the fixed number of interests predefined by existing models cannot adapt to the diverse preferences of users, making it difficult to further improve recommendation performance. To address these issues, we propose a novel generative sequential recommendation framework named ADIGRec (**A**daptive User **D**ynamic **I**nterest **G**uidance for Generative Sequential **Rec**ommendation), which adaptively focuses on users' dynamic interest features. Specifically, our framework combines users' dynamic features and inherent interest features encoded from historical sequences as new guidance conditions. Furthermore, we introduce a module that injects dynamic interest features into the noise item embeddings, enabling explicit interaction with the guidance conditions during the generation phase. This approach essentially fits the noise in the target space rather than the user preference space, leading to improved recommendation diversity. Additionally, we propose a novel regularization method to mitigate the impact of user interest routing collapse on the generation results. Extensive experiments on three publicly available datasets demonstrate that our method achieves superior performance compared to established baseline methods.

CCS Concepts

• **Information systems → Recommender systems**.

Keywords

Sequential Recommendation, Diffusion Models, Multi-Interest.

ACM Reference Format:
Kai Zhu, Jing Li, Jia Wu, Yue He, Jun Chang, Guohao Li, and Shuyi Zhang. 2025. Adaptive User Dynamic Interest Guidance for Generative Sequential Recommendation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25), July 13–18, 2025, Padua, Italy.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3726302.3729888

1 Introduction

Sequential recommendation (SR) has been widely used in various domains to alleviate the problem of information overload by predicting potential user interaction items based on a user's historical sequence. Early work treated sequential recommendation as a Markov Decision Process (MDP) to model the transition patterns between items in a sequence. In terms of sequential modeling, early research used RNN or GRU framework [8, 9] to capture long-term sequence dependencies. Alternative approaches to RNNs with self-attention mechanisms have shown general superiority [29, 38]. With the proposal of the Transformer architecture, SASRec [13] was first utilized to deal with long-term dependencies in sequence recommendation tasks. Meanwhile, the GNN-based approach tries to capture higher-order dependencies between items to enhance the recommendation effect [5, 36]. Due to the different interests of users, researchers have attempted to model user multi-interest features from user history sequences for fine-grained recommendations. ComiRec [2] has proposed different routing methods based on capsule networks and self-attention to capture user multi-interest features and improve

*Corresponding authors: Jing Li and Yue He. This research is supported by the National Natural Science Foundation of China under Grant No. 62372335.

SIGIR '25, Padua, Italy

ACM ISBN 979-8-4007-1592-1/2025/07
https://doi.org/10.1145/3726302.3729888

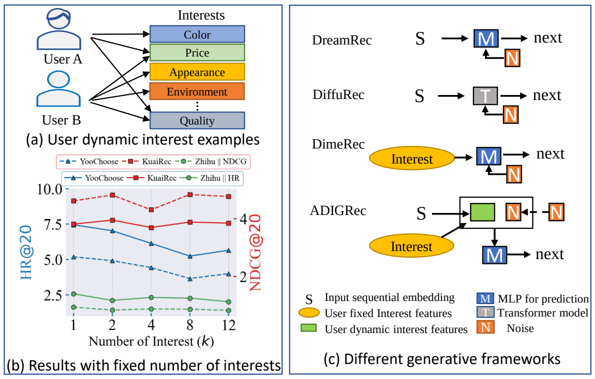


Figure 1: (a) represents the examples of user dynamic interest, which shows different users focus on different numbers of interests. (b) illustrates the different performances of DimeRec[14] with the fixed number of interests in three datasets, which means the model needs different numbers of interests to achieve better results for different datasets. (c) illustrates different generative frameworks, DreamRec[35], DiffuRec[16], DimeRec[14] and our ADIGRec.

recommendation accuracy. However, the inherent interest features lead to limited recommendation diversity.

The above paradigm is weak for exploring the probability that a user interacts with a non-interactive item. Currently, generative networks are used for recommendation, such as GAN and VAE. VAE-based methods [17, 32] and GAN-based methods [37] learn latent distributional processes to infer such probabilities, but they suffer from fitting difficulties and a posteriori collapse, respectively. Recently, researchers have attempted to employ new diffusion models to explore the embedding of target items from distributions and enhance recommendation diversity [3, 4, 16, 28, 35]. Since the guidance condition encoded directly from the user's history sequence is dynamically non-stationary, using it to guide the model to generate the final prediction results in better diversity, but it likewise introduces more uncertainty, leading to fluctuations in the recommendation effect. To mitigate this effect, DimeRec [14] captures stable user interest features as guidance conditions to achieve better recommendation results.

However, the existing approaches neglect that the number of interests followed by different users is inconsistent, and the inherent number of interest features of the model cannot be matched with different users, resulting in a limited diversity of recommendation results. Although combining the Diffusion Model can mitigate this effect, the inherent number of interest features as a guidance condition leads to a limited diversity of generated results, such as DimeRec [14]. As shown in Figure 1, setting different points of interest shows inconsistent recommendation results on different datasets, indicating that the number of users' interest features does not exactly match the model. It leads to the inherent interest features extracted by the model, making it difficult to represent the real interest preferences of different users, leading to limited results.

In addition, existing methods usually use MLP (Such as method DiffRec[28], DreamRec[35], and DimeRec[14]) or Transformer Encoder (Such as method DiffuRec[16] and CaDiRec[3]) for noise (or target item) prediction. The former has a simple structure that ensures the diversity of generated items, but lacks the interaction with the guidance conditions for display, leading to the possibility of serious bias in the generated results. The latter performs a weighting operation of noise items with user dynamic features for generating results, which essentially fits the noise in the user preference space rather than the target space, leading to limited result diversity.

To address the above problem, we propose a new generative sequence recommendation framework called **ADIGRec** (**A**daptive User **D**ynamic **I**nterest **G**uidance for Generative Sequential **Rec**ommendation), which dynamically focuses on user interest features and explicitly instructs model generation as guidance conditions. Specifically, a module **DIFGC** (Dynamic Interest Features as Guidance Condition) is proposed to integrate user dynamic interest features encoded from historical sequences with inherent interest features as explicit guidance conditions. Secondly, a module **AI^{2}M** (Adaptive Condition Insert Module) is proposed to inject this dynamic interest feature into the noise to implicitly affect the generation results of the diffusion model. Meanwhile, we propose a regularization method without artificial parameters to reduce the impact of user interest routing collapse. Ultimately, our method obtains better results on three publicly available datasets. The effectiveness of ADIGRec is evaluated through extensive experiments and comparisons with established baseline methods. Our main contributions can be summarized as follows:

- We propose a guidance condition characterization approach, which considers different users focusing on different numbers of interest points and realizes the matching demand of different users by adaptively focusing on their dynamic interest features.
- We propose ADIGRec, a framework that integrates user dynamic interest features encoded from historical sequences with intrinsic interest features as explicit guidance conditions and injects dynamic interest features into the noise to implicitly affect the generation results of the diffusion model.
- Our method obtains better results on three publicly available datasets. Among them, we propose a method to reduce the impact of user interest routing collapse without artificial parameters. Ultimately, we evaluate the effectiveness of ADIGRec through extensive experiments and comparisons with established baseline methods.

2 Related Work

2.1 Discriminative Sequential Recommendation

Early studies[21] focus on the orders of interacted items in modeling sequence, but ignore the intricate relationships between items. RNN-based model GRU4Rec[10], Caser[24], and Transformer-based model SASRec[13] have achieved great performance in long-term relation modeling. In addition, graph neural networks (GNNs), such as SR-GNN[30], GC-SAN[34], and MAERec[36], are introduced to capture more complex relations underlying the item transitions by constructing item sequences into different graphs. Generative models have also shown great potential in sequential recommendation. Typically, SVAE[23] and ACVAE[33] merge VAE to capture underlying dependencies for sequential recommendation, and IRGAN[27], RecGAN[1] and MFGAN[20] devise two retrieval models from generative and discriminative perspectives to consider the special demands in the information retrieval domain.

2.2 Generative Sequential Recommendation

Recently, diffusion models, which were first proposed for image synthesis tasks, have emerged as a promising paradigm in recommendation. DiffuRec[16] adds truncated Gaussian noise to the embedding of the target item in the forward process and predicts the target item by an approximator based on the Transformer structure. DiffRec[28] proposes a model for diffusion acceleration in potential space and a model with a noise addition strategy adjusted over time for different scenarios. DiffuASR[18] utilizes the diffusion model to expand the user interaction sequences, which attempts to alleviate the long tail problem. In addition, CadiRec [3] generates masked items through contextual information as an alternative to the rule-based view enhancement approach, and employs a contrastive learning strategy for model optimization. DreamRec[35] firstly proposes a "learning-to-generate" paradigm for diffusion models in sequential recommendation, which refers to generating a personalized oracle item based on user-item interactions directly, avoiding being limited by certain specific candidate items and exploring unknown data distribution without negative sampling. Further, Li et al. proposed a diffusion model DimeRec [14] with user interest features as stable guidance conditions for sequential recommendation and enabled the reconstruction task to be optimized at the same time as the recommendation task by $L2$ normalization.

3 Preliminary

This section will briefly introduce the problem definition and notations used in this article and present the diffusion model as preliminary knowledge.

3.1 Task Definition and Notations

In this paper, we use bold italicized lowercase letters (e.g., $\boldsymbol{u}$, $\boldsymbol{v}$) and bold uppercase letters (e.g, $\boldsymbol{I}$) to represent column vectors and matrices, respectively. In addition, this paper uses calligraphic fonts to represent the sets (e.g., $\mathcal{U}$, $\mathcal{V}$). In sequential recommendation, the task is to infer a user's preference and recommend the next item using a sequence of historical interactions, which can be extracted as a set arranged in time. Let $\mathcal{U}$ and $\mathcal{V}$ denote a set of users and items, respectively, and $u_i \in \mathcal{U}$ ($i = 1, 2, \ldots, |\mathcal{U}|$) or $v_j \in \mathcal{V}$ ($j = 1, 2, \ldots, |\mathcal{V}|$) represent one of them. The user's behavior sequence is usually arranged in chronological order. Therefore, this paper uses $\mathcal{S}_{u_i} = [v_1^{u_i}, v_2^{u_i}, \ldots, v_t^{u_i}, \ldots, v_{|\mathcal{S}_{u_i}|}^{u_i}]$ to represent the interaction sequence of user u_i, where b represents the item that the user interacts at the time step t and $|\mathcal{S}_{u_i}|$ represents the length of the interaction sequence. The recommendation task is to predict the next item at time step $t = |\mathcal{S}_{u_i}|$, i.e., $v_{|\mathcal{S}_{u_i}|+1}$ for the user u_i.

3.2 Diffusion Model

This section reviews the basic concepts of the diffusion model established by the groundbreaking DDPM framework Ho et al. [11] to ensure the completeness and readability of this paper. DDPM Ho et al. [11] is a representative formulation of the diffusion model, which formulates two Markov chains of forward and reverse diffusion processes, respectively, to model the underlying distribution.

Forward Diffusion Process. Let x_0 be the data point sampled from the real data distribution $q(x)$, and Gaussian noise β_t is added according to the Markov chain characteristic at time step t (t=1, t=2, . . . , t=T), where T is the total step size. This process can be formulated as:

$$\begin{aligned} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) &= \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I}), \\ q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) &= \prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}). \end{aligned} \tag{1}$$

The step sizes are controlled by a variance schedule $\{\beta_t \in (0,1)\}_{t=1}^T$. Let $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{t'=1}^{t} \alpha_{t'}$, and use the additive properties of two Gaussian distributions with different variances and the re-parameterization technique to derive $\boldsymbol{x}_t$ after t steps of noise addition as follows:

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon} \ (\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})). \tag{2}$$

Reverse Diffusion Process. The diffusion model estimates the conditional distribution between different steps of the Markov chain in the reverse process through deep networks such as U-Net Ronneberger et al. [22] and Transformer Vaswani et al. [26] (commonly used in NLP tasks), and samples according to the distribution to achieve denoising. It can be defined as:

$$\begin{aligned} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) &= \mathcal{N}(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t), \Sigma_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)), \\ p_{\boldsymbol{\theta}}(\boldsymbol{x}_{0:T}) &= p(\boldsymbol{x}_T) \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) \end{aligned} \tag{3}$$

where $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ and $\Sigma_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ are the mean and covariance of the Gaussian distribution predicted by the deep neural network with parameters $\boldsymbol{\theta}$. Since the real conditional distribution $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ is difficult to obtain,

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t(\boldsymbol{x}_t, \boldsymbol{x}_0),\ \tilde{\beta}_t \mathbf{I}) \tag{4}$$

can be derived by adding condition $\boldsymbol{x}_0$ combined with Bayesian rules, where

$$\tilde{\boldsymbol{\mu}}_t(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\boldsymbol{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_t)}{1-\bar{\alpha}_t}\boldsymbol{x}_t, \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t. \tag{5}$$

Combined with $\boldsymbol{x}_t$,

$$\boldsymbol{\mu}_{\theta}(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\right) \tag{6}$$

can be further derived to predict $\tilde{\boldsymbol{\mu}}_t$. The core of the generation task is to learn the distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}_0)$, which can be optimized by optimizing the variational boundary of negative log-likelihood:

$$L_{vlb} = \mathbb{E}_{q(\boldsymbol{x}_{0:T})}\left[\log \frac{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{0:T})}\right] \geq -\mathbb{E}_{q(\boldsymbol{x}_0)} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0), \tag{7}$$

$$L_{vlb} = \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[D_{\mathrm{KL}}\left(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, x_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\right)\right]}_{L_{t-1}} + C_1, \tag{8}$$

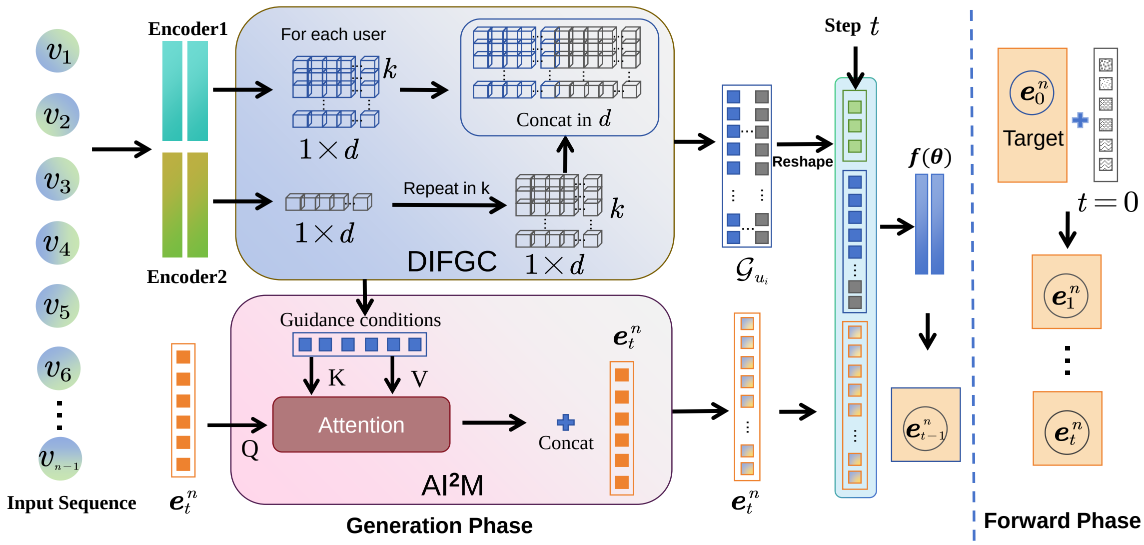


Figure 2: Illustrates the framework of ADIGRec. It contains two important modules: DIFGC (Dynamic Interest Features as Guidance Condition) captures user dynamic interest features as guidance conditions, and AI^{2}M (Adaptive Condition Insert Module) combines user interest semantics from guidance conditions as new noise to generate the next embedding in the diffusion generation phase.

where C_1 is constant. Substituted into $\tilde{\boldsymbol{\mu}}_t(\boldsymbol{x}_t, \boldsymbol{x}_0)$ and $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$, the optimization formula for omitting the coefficient term is derived:

$$L_{t-1} = \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{\epsilon}, t}\left[||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)||^2 \right], \tag{9}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$, and $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\cdot)$ can be instantiated by a deep neural network.

4 Methodology

4.1 Overview

In this section, we present the proposed method, ADIGRec, which focuses on generating user dynamic interest features as guidance and explicitly injecting this information into noisy items to enhance the reverse process for target embedding generation. As shown in Figure 2, ADIGRec consists of two key modules: DIFGC (Dynamic Interest Features as Guidance Condition) and AI^{2}M (Adaptive Condition Insert Module). Specifically, the DIFGC module aggregates the two types of features captured from user interaction sequences to generate user dynamic interest features as guidance conditions. Additionally, the AI^{2}M module injects information into noisy item embeddings via a cross-attention mechanism to strengthen the guidance during the generation phase. Unlike general residual connections, to ensure that the noisy items maintain a unique distribution while incorporating the injected information, we employ a concatenation operation to combine the injected embedding with the original embedding. Finally, in the diffusion model generation phase, the resulting noisy target embedding is concatenated with the guidance conditions and time step t, and then input into the deep neural network $f_{\boldsymbol{\theta}}$ to generate the target embedding at step $t-1$.

4.2 Dynamic Interest Features as Guidance Condition (DIFGC)

Recent studies construct guidance conditions from user interaction features (non-stationary) or user interest features (stationary) to instruct the model to generate high-quality recommendation results. However, there are inherent limitations in both kinds of guidance conditions. Non-stationary conditions are represented by user dynamic feature embeddings, which increase the diversity of the results but introduce redundant information, causing the generated results to deviate from the user's true preferences. And stationary conditions are represented by user interest feature embeddings, which ensure that the generated results do not seriously deviate from the user's true preferences, but the model usually extracts intrinsic user interests (it is not possible to match the number of interests of different users), which can lead to a reduction in the diversity of the recommendation results. Specifically, a module DIFGC is proposed to integrate user dynamic interest features encoded from historical sequences with inherent interest features as explicit guidance conditions.

This section formalizes the embedding methods for extracting user interest features and user dynamic interest features, and the module for integrating the two types of features as guidance conditions. For the user interest features, the self-attention mechanism is applied by embedding the user's historical sequence, which is consistent with the previous work [2]. In addition, the user's dynamic preference feature is extracted by two Transformer blocks.

4.2.1 Embedding Layer. Given a user sequence $\mathcal{S}_{u_i}$, we can obtain the initial embedding $\mathcal{E}_{\mathcal{S}_{u_i}} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_{|\mathcal{S}_{u_i}|}]$ through an embedding matrix of items $\mathcal{M} \in \mathbb{R}^{|\mathcal{V}| \times d}$ which can be optimized, where d represents the dimension of the embedding.

4.2.2 User Interest Features Extraction. The self-attention method based on the two-layer MLP extracts the weighted score matrix $\mathcal{W}_{u_i} \in \mathbb{R}^{|\mathcal{S}_{u_i}| \times k}$ as

$$\mathcal{W}_{u_i} = \mathbf{Softmax}\{\mathbf{MLP}_{4d \times k}[\mathbf{Tanh}(\mathbf{MLP}_{d \times 4d}[\mathcal{E}_{\mathcal{S}_{u_i}} + \mathcal{P}])]\}, \quad (10)$$

where $\mathcal{P}$ is position embedding and k is the number of interest features. The information is aggregated through a weighted score matrix to extract the user's interest features as follows:

$$\mathcal{G}^1_{u_i} \in \mathbb{R}^{k \times d} = \mathcal{W}^T_{u_i} \times \mathcal{E}_{\mathcal{S}_{u_i}}. \quad (11)$$

4.2.3 User Dynamic Interest Features Extraction. The user dynamic preference embedding is extracted through two transformer blocks, where each block consists of a self-attention layer and a feed-forward layer, and is described as follows:

$$\mathcal{G}^2_{u_i} \in \mathbb{R}^{1 \times d} = \mathcal{H}_{u_i}[-1:], \ \mathcal{H}_{u_i} = \mathbf{Trm\text{-}e}[\mathcal{E}_{\mathcal{S}_{u_i}} + \mathcal{P}]. \quad (12)$$

4.2.4 Generate Guidance Condition. Equations 11 and 12 represent the extraction methods for the user's inherent k interest features and dynamic interest features, respectively. These two types of features can be regarded as approximations of the user's long-term and short-term preferences. Their effective combination enables the construction of the user's true interaction preferences. By leveraging this property, the model can be guided to generate authentic user preference embeddings in the noise space of the target item, thereby achieving dynamic interest feature guidance. To preserve the individuality of the two features while enriching the guidance information, we concatenate them to form the final guidance condition as follows:

$$\mathcal{G}_{u_i} \in \mathbb{R}^{k \times 2d} = \mathbf{Concat}(\mathcal{G}^1_{u_i}, \mathcal{G}^2_{u_i}, dim = -1). \quad (13)$$

By concatenating along the feature dimension, the time step t is combined with the noise term in a high-dimensional space and mapped to the same space as the predicted term, enabling the generation of item embeddings that better match the user's true preferences.

4.3 Adaptive Condition Insert Module (AI^{2}M)

Existing methods based on diffusion models typically employ either MLPs (such as DiffRec[28], DreamRec[35], and DimeRec[14]) or Transformer encoders (such as DiffuRec[16] and CaDiRec[3]) for noise (or target item) prediction. The former, with its simple structure, ensures the diversity of generated items but lacks explicit interaction with the guidance conditions, which may result in significant bias in the generated results. The latter performs a weighting operation between noise items and user dynamic features to generate results, which essentially fits the noise in the user preference space rather than the target space, thereby limiting recommendation diversity. To address these limitations, we propose a module named **AI^{2}M**, which dynamically injects the guidance condition $\mathcal{G}_{u_i}$ into the noisy target embedding via a cross-attention mechanism to enhance the guidance strength. As follows:

$$\begin{aligned} \boldsymbol{x}_t^{\mathcal{G}^{u_i}} &= Softmax\left\{\frac{[(\boldsymbol{x}_t + \boldsymbol{t}_E)\boldsymbol{W}^Q](\mathcal{G}_{u_i}\boldsymbol{W}^K)^\top}{\sqrt{\boldsymbol{d}}}\right\}(\mathcal{G}_{u_i}\boldsymbol{W}^V), \\ \boldsymbol{x}_t &= \mathbf{Concate}(\boldsymbol{x}_t, \boldsymbol{x}_t^{\mathcal{G}_{u_i}}, dim = -1), \end{aligned} \quad (14)$$

where $\boldsymbol{W}^Q \in \mathbb{R}^{d \times d}$, $\boldsymbol{W}^K \in \mathbb{R}^{2d \times d}$ and $\boldsymbol{W}^V \in \mathbb{R}^{2d \times d}$ are learnable parameters and $\boldsymbol{t}_E$ is the embedding of the time step t.

4.4 Training with diffusion model

4.4.1 Learning Phase. This subsection describes the learning phase of the diffusion model in ADIGRec, where training is achieved by adding noise to the embedding of the target item and minimizing its mean square error with respect to the predicted embedding. Consistent with DreamRec [35], the sequence recommendation task is refactored into a target-generated task as follows:

$$p_\theta(\boldsymbol{e}^n_{t-1}|\boldsymbol{e}^n_t, \mathcal{G}_{u_i}) = \mathcal{N}(\boldsymbol{e}^n_{t-1}; \boldsymbol{\mu}_\theta(\boldsymbol{e}^n_t, \mathcal{G}_{u_i}, \boldsymbol{t}_E), \Sigma_\theta(\boldsymbol{e}^n_t, \mathcal{G}_{u_i}, \boldsymbol{t}_E)), \quad (15)$$

where $\boldsymbol{\mu}_\theta$ is predicted by a trainable single-layer MLP $f_\theta(\cdot)$, $n = |\mathcal{S}_{u_i}|+1$ and $\boldsymbol{e}^n_t$ is the embedding $(\boldsymbol{x}_t)$ of the target item calculated by Equation (14). Therefore, by substituting the $p_\theta(\cdot)$ into Equation (8) and re-sampling the

$$\boldsymbol{\mu}_\theta(\boldsymbol{e}^n_t, \mathcal{G}_{u_i}, \boldsymbol{t}_E) = \sqrt{\bar{\alpha}_{t-1}} f_\theta(\boldsymbol{e}^n_t, \mathcal{G}_{u_i}, \boldsymbol{t}_E) + \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{\sqrt{1 - \bar{\alpha}_t}}\boldsymbol{\epsilon}, \quad (16)$$

the optimization goal that performs well in the text generation [15] can be further derived as:

$$L_{simple} = \mathbb{E}_{\boldsymbol{e}^n_0, \boldsymbol{\epsilon}, t}\left[\|\boldsymbol{e}^n_0 - f_\theta(\sqrt{\bar{\alpha}_t}\boldsymbol{e}^n_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}, \mathcal{G}_{u_i}, \boldsymbol{t}_E)\|^2\right]. \quad (17)$$

Algorithm 1: Learning Phase

Input: Historical sequence $\mathcal{S}_{u_i}$; Target item embedding $\boldsymbol{e}_0^{n=|\mathcal{S}_{u_i}|+1}$; Max diffusion time step T;
Output: Next target embedding.

repeat
- $\mathcal{S}_{u_i} = [v_1, v_2, \ldots, v_{|\mathcal{S}_{u_i}|}] \sim \mathcal{U}$;
- $\mathcal{E}_{\mathcal{S}_{u_i}} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_{|\mathcal{S}_{u_i}|}]$;
- Generate $\mathcal{G}_{u_i}$ through Eq 11, Eq 12 and Eq 13;
- $\boldsymbol{t}_E = Embedding(t)$; // $t \sim U(1, 2, \ldots, T)$
- Forward phase generates $\boldsymbol{e}^n_t$ via Eq 2;
- Map the embedding onto a sphere [14]: $\boldsymbol{e}^n_t \leftarrow L2(\boldsymbol{e}^n_t)$;
- Inject the guidance conditions: $L2(\boldsymbol{e}^n_t) \leftarrow \mathrm{AI}^2\mathrm{M}$;
- Reconstruction: $\hat{\boldsymbol{e}}^n_0 \leftarrow f_\theta(\boldsymbol{e}^n_t, \mathcal{G}_{u_i}, \boldsymbol{t}_E)$;
- Optimization: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \mu\nabla_\theta(\mathcal{L}_{\mathcal{G}_1} + \lambda_1 L_{simple} + \lambda_2\mathcal{L}_{ssm} + \mathcal{L}_{reg})$;

until *converged*;

4.4.2 New design strategy in learning phase. General multi-interest recommendation suffers from routing collapse, which manifests itself as a single interest expressed by a single item. Inspired by [31], we design a simple regularization loss to reduce the influence of guidance conditions. Specifically, the covariance matrix of the weighted score matrix $\mathcal{W}_{u_i} \in \mathbb{R}^{|\mathcal{S}_{u_i}| \times k}$ is computed and then the Euclidean distance of the relevant terms is minimized as follows:

$$\begin{aligned} C_{u_i} \in \mathbb{R}^{k \times k} &= \frac{1}{|\mathcal{S}_{u_i}|}(\mathcal{W}_{u_i} - \bar{\mathcal{W}}_{u_i}(dim = 0))^T(\mathcal{W}_{u_i} - \bar{\mathcal{W}}_{u_i}), \\ \mathcal{L}_{reg} &= \frac{1}{|\mathcal{U}|}\sum_{u_i \in \mathcal{U}}\left[\sum_{k=1}^{K}(\|C_{u_i}[k,:]\|_2\right]. \end{aligned} \quad (18)$$

Algorithm 2: Generation Phase

$\mathcal{S}_{u_i} = [v_1, v_2, \ldots, v_{|\mathcal{S}_{u_i}|}] \sim \mathcal{U}$;
$\mathcal{E}_{\mathcal{S}_{u_i}} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_{|\mathcal{S}_{u_i}|}]$;
Generate $\mathcal{G}_{u_i}$ through Eq 11, Eq 12 and Eq 13;
$\boldsymbol{e}_T^{n=|\mathcal{S}_{u_i}|+1} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$;
for $(t = T; t > 0; t--)$ **do**
    $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, if $t > 1$, else $\boldsymbol{\epsilon} = \boldsymbol{0}$;
    $\boldsymbol{t}_E = Embedding(t)$; // $t \sim U(1, 2, \ldots, T)$
    Map the embedding onto a sphere [14]: $\boldsymbol{e}_t^n \leftarrow L2(\boldsymbol{e}_t^n)$;
    Inject the guidance conditions: $L2(\boldsymbol{e}_t^n) \leftarrow$ AI^{2}M;
    Reconstruction: $\hat{\boldsymbol{e}}_t^n \leftarrow f_{\boldsymbol{\theta}}(\boldsymbol{e}_t^n, \mathcal{G}_{u_i}, \boldsymbol{t}_E)$;
    Generation:
    $\boldsymbol{e}_{t-1}^n = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{e}_t^n + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\boldsymbol{e}_{t-1}^n + \sqrt{\tilde{\beta}_t}\boldsymbol{\epsilon}$;
Output $\boldsymbol{e}_0^n$;

4.5 Loss of ADIGRec

The loss of model ADIGRec consists of four components, namely the interest feature supervision loss $\mathcal{L}_{\mathcal{G}^1}$ (consistent with the [14]), the reconstruction loss $\mathcal{L}_{simple}$, the retrieval loss $\mathcal{L}_{ssm}$ (softmax loss [12]) and the regularization loss L_{reg} as follows:

$$\mathcal{L}_{\mathcal{G}^1} = \frac{1}{|\mathcal{U}|}\sum_{u_i \in \mathcal{U}} -\log\left(\frac{\exp(g_{u_i} \cdot \boldsymbol{e}_0^n)}{\exp(g_{u_i} \cdot \boldsymbol{e}_0^n) + \sum_{i^- \in \mathcal{I}_{neg}} \exp(g_{u_i} \cdot \mathbf{e}_{i^-})}\right),$$

$$\mathcal{L}_{ssm} = \frac{1}{|\mathcal{U}|}\sum_{u_i \in \mathcal{U}} -\log\left(\frac{\exp(\hat{\boldsymbol{e}}_0^n \cdot \boldsymbol{e}_0^n)}{\exp(g_{u_i} \cdot \boldsymbol{e}_0^n) + \sum_{i^- \in \mathcal{I}_{neg}} \exp(g_{u_i} \cdot \mathbf{e}_{i^-})}\right), \tag{19}$$

where $\boldsymbol{g}_{u_i} = \mathcal{G}^1_{u_i}[\arg\max(\mathcal{G}_1^{u_i}\boldsymbol{e}_0^n), :]$, $\boldsymbol{e}_0^n$ is the embedding of the target item and $\hat{\boldsymbol{e}}_0^n$ is the predicted of $f(\boldsymbol{\theta})$.

Combining the above four losses, the optimization loss of ADIGRec can be obtained as:

$$\mathcal{L} = \mathcal{L}_{\mathcal{G}^1} + \lambda_1 L_{simple} + \lambda_2 \mathcal{L}_{ssm} + \mathcal{L}_{reg}, \tag{20}$$

where λ_1 and λ_2 are hyperparameters. The details of the learning phase of ADIGRec are shown in Algorithm 1.

4.6 Generation Phase of ADIGRec

In the generation phase, it aims to gradually generate the target from the standard Gaussian distribution from the total time step T. Substitute $f_{\boldsymbol{\theta}}$ into $\tilde{\boldsymbol{\mu}}_t(\boldsymbol{x}_t, \boldsymbol{x}_0)$ (Equation 14), and sample (t-1)th item through $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$. The specific Equation is:

$$\boldsymbol{e}_{t-1}^n = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{e}_t^n + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\boldsymbol{f}_{\boldsymbol{\theta}} + \sqrt{\tilde{\beta}_t}\boldsymbol{\epsilon}, \tag{21}$$

where $\boldsymbol{x}_t = \boldsymbol{e}_t^n$, $\boldsymbol{x}_0 = \boldsymbol{f}_\theta$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, and finally $\boldsymbol{e}_0^n$ is sampled with $\boldsymbol{\epsilon} = 0$. The details of the generation phase of ADIGRec are shown in Algorithm 2.

5 Experiments

5.1 Experimental Settings

5.1.1 Evaluation Datasets. To evaluate the performance of our ADIGRec fairly and faithfully, we adopt three real-world datasets for sequential recommendation. The statistics of datasets are shown in Table 1.

Table 1: Statistics of datasets.

Dataset	#Items	#Sequences	#Interactions	Sparsity
YooChoose	9514	128468	539436	99.96%
KuaiRec	7261	92090	737163	99.89%
Zhihu	4838	11714	77712	99.86%

- **YooChoose**: A dataset comes from RecSys Challenge 2015[1] contains click-streams of an e-commerce site in six month.
- **KuaiRec**[6]: A full-observed dataset with few missing values from a famous short-video platform, Kuaishou App.
- **Zhihu**[7]: A dataset contains complete interactions in social Q&A scenario collected from an online-knowledge-sharing community called Zhihu.

5.1.2 Evaluation Metrics. We use two widely used metrics in a recommendation system: Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG). HR@N counts the times that the ground-truth next item is in the set of the top-N items, while NDCG@N considers the actual rank of the item.

5.1.3 Baselines. We compare ADIGRec with various recommenders.

- **GRU4Rec**[10] captures the user's long-distance preference through a recurrent neural network.
- **SASRec**[13] uses a Transformer to capture users' long-term and high-order preferences in sequential recommendation.
- **DuoRec**[19] proposes a hard sample sampler and model-level dropout method to solve representation degradation.
- **ComiRec**[2] proposed capsule network and self-attention mechanism approaches for extracting user interest features, respectively.
- **DiffRec**[28] proposes a model for diffusion acceleration in potential space and a model with a noise addition strategy adjusted over time for different scenarios.
- **DiffuRec**[16] adds truncated Gaussian noise to the embedding of the target item in the forward process and predicts the target item by an approximator based on the Transformer structure.
- **DreamRec**[35] proposes a learning-to-generate paradigm, which directly generates the embedding of user pre-interactive items through a Classifier-Free Guided diffusion model.
- **DimeRec**[14] proposes a unified framework of user interest features as guidance conditions and optimizes three types of losses simultaneously.

5.1.4 The dataset detail for preprocessing. Especially, to avoid the cold-start issue brought by the inherent extreme sparsity of datasets that is not closely related to the research in this paper, we removed items with fewer than 5 interactions and sequences shorter than 3 interactions, and only the last 10 interactions were retained for each sequence [35], and the specific statistics of the three datasets are shown in Table 1. All datasets are divided into training, validation, and testing sets in a 8 : 1 : 1 ratio.

[1] https://recsys.acm.org/recsys15/challenge/

Table 2: Overall performance comparison in metrics of HR and NDCG on three datasets: YooChoose, KuaiRec, and Zhihu. The best-performing results under each metric are shown in bold, while the second-best results are highlighted with underlining. "Δ" means the increased rate of ADIGRec compared to the second-best results. "∗" indicates that the results show significance relative to the suboptimal results ($P < .05$).

Dataset	Method	HR@5(%)	NDCG@5(%)	HR@10(%)	NDCG@10(%)	HR@20(%)	NDCG@20(%)	HR@50(%)	NDCG@50(%)
YooChoose	SASRec	2.32±0.05	1.17±0.05	2.78±0.06	1.39±0.04	3.90±0.06	1.56±0.11	5.29±0.07	1.99±0.05
	GRU4Rec	2.64±0.07	1.12±0.04	2.98±0.03	1.52±0.05	4.12±0.05	1.76±0.08	6.27±0.10	2.12±0.04
	DuoRec	2.67±0.03	1.21±0.08	3.12±0.07	1.47±0.05	4.54±0.06	1.71±0.04	6.71±0.04	2.23±0.06
	ComiRec	2.81±0.07	1.18±0.04	3.21±0.07	1.34±0.03	5.25±0.10	1.82±0.06	8.54±0.09	2.32±0.08
	DiffRec	2.89±0.05	1.34±0.03	3.60±0.05	1.57±0.06	4.30±0.03	1.92±0.07	6.61±0.05	1.87±0.06
	DiffuRec	3.28±0.10	1.55±0.03	3.66±0.06	1.87±0.08	4.93±0.05	2.21±0.06	6.81±0.15	2.57±0.08
	DreamRec	3.42±0.02	1.54±0.06	4.05±0.06	1.74±0.05	4.76±0.08	1.93±0.07	8.35±0.03	2.61±0.08
	DimeRec	2.44±0.11	1.25±0.06	4.28±0.04	1.84±0.03	6.14±0.21	2.31±0.10	10.11±0.14	3.15±0.08
	ADIGRec	**3.68±0.04***	**1.86±0.02***	**5.54±0.05***	**2.46±0.04***	**7.91±0.07***	**3.05±0.06***	**12.62±0.12***	**3.98±0.08***
	Δ	+7.60%	+20.78%	+29.44%	+33.70%	+28.83%	+32.03%	+24.83%	+26.35%
KuaiRec	SASRec	3.57±0.04	1.26±0.06	3.47±0.08	1.61±0.04	3.86±0.04	1.97±0.06	7.42±0.12	3.55±0.09
	GRU4Rec	3.21±0.06	1.19±0.08	3.23±0.09	1.56±0.06	3.69±0.08	1.82±0.09	6.98±0.14	3.28±0.05
	DuoRec	4.17±0.06	2.24±0.07	4.32±0.09	2.35±0.06	4.54±0.10	2.52±0.08	8.12±0.12	3.72±0.05
	ComiRec	4.01±0.06	3.28±0.05	5.24±0.05	3.42±0.07	6.84±0.09	4.27±0.10	9.48±0.16	4.62±0.08
	DiffRec	3.34±0.06	1.21±0.04	3.40±0.08	1.60±0.07	3.80±0.07	2.17±0.08	7.12±0.12	3.62±0.09
	DiffuRec	2.71±0.04	1.72±0.05	2.95±0.04	1.80±0.06	4.32±0.11	2.18±0.18	7.92±0.07	2.91±0.11
	DreamRec	4.80±0.05	4.28±0.03	5.09±0.06	4.37±0.05	5.31±0.04	4.41±0.02	5.81±0.06	4.53±0.04
	DimeRec	4.27±0.05	3.48±0.06	5.43±0.06	3.85±0.07	7.26±0.10	4.31±0.08	11.14±0.12	5.07±0.11
	ADIGRec	**5.30±0.04***	**4.40±0.04***	**6.55±0.05***	**4.80±0.07***	**8.59±0.06***	**5.31±0.08***	**12.72±0.11***	**6.13±0.08***
	Δ	+10.42%	+2.80%	+20.63%	+9.84%	+18.32%	+20.41%	+14.18%	+20.91%
Zhihu	SASRec	0.41±0.03	0.21±0.02	0.90±0.03	0.45±0.03	1.75±0.03	0.66±0.04	4.12±0.08	1.12±0.06
	GRU4Rec	0.44±0.04	0.24±0.03	1.01±0.03	0.48±0.02	1.81±0.06	0.72±0.03	4.35±0.09	1.24±0.03
	DuoRec	0.46±0.02	0.28±0.03	1.16±0.04	0.54±0.03	2.06±0.03	0.74±0.04	4.86±0.11	1.31±0.09
	ComiRec	0.45±0.04	0.24±0.06	1.05±0.04	0.43±0.06	1.99±0.06	0.67±0.09	4.89±0.06	1.24±0.05
	DiffRec	0.55±0.06	0.32±0.03	1.08±0.06	0.47±0.05	1.84±0.09	0.68±0.04	4.61±0.10	1.30±0.11
	DiffuRec	0.56±0.03	0.33±0.03	1.21±0.04	0.52±0.05	1.82±0.06	0.71±0.04	4.75±0.11	1.33±0.08
	DreamRec	0.58±0.04	0.34±0.02	1.29±0.04	0.56±0.05	2.11±0.06	0.76±0.04	3.66±0.12	1.07±0.06
	DimeRec	0.51±0.06	0.26±0.03	1.13±0.07	0.46±0.04	2.32±0.09	0.76±0.08	5.32±0.06	1.34±0.09
	ADIGRec	**0.78±0.03***	**0.38±0.04***	**1.61±0.05***	**0.64±0.04***	**3.00±0.11***	**0.99±0.05***	**6.52±0.08***	**1.68±0.07***
	Δ	+34.48%	+11.76%	+24.81%	+14.29%	+42.18%	+30.26%	+22.56%	+25.37%

5.1.5 Training Protocol. We implement all models with Python 3.10 and PyTorch 2.1.1 on Nvidia GeForce RTX 4090. For all models, the dimension of item embedding is set as 64, the batch size is set to 256, and the weight of L2 regularization, if required, is tuned in the range of $\{1e-3, 1e-4, 1e-5, 1e-6, 1e-7\}$. For diffusion mechanism-based methods, the sum of diffusion steps T is searched from the set $\{50, 100, 200, 500, 1000, 2000\}$. For negative samples, we uniformly sampled 10 items that users had never interacted with, consistent with previous work. The parameters λ_1 and λ_2 are adjusted from $\{0.01, 0.1, 1.0, 10.0\}$. In this paper, we set $\lambda_1 = 0.1$ and $\lambda_2 = 1$ for the YooChoose and KuaiRec datasets, and $\lambda_1 = 0.1$ and $\lambda_2 = 0.01$ for the Zhihu dataset. The number of interests k for all models that need to extract user interest features is set to 4. And all models are accelerated by the AdamW optimizer during training, and the learning rate is set to 0.005.

5.2 Recommendation Performance

In this section, we compare ADIGRec with baseline models based on different techniques, including conventional methods, multi-interest recommendation methods, and state-of-the-art approaches incorporating diffusion models. The results are presented in Table 2. Overall, ADIGRec consistently outperforms all baselines across all metrics on the three datasets, demonstrating the superiority of our method, which introduces contrastive learning to enhance diffusion models for sequential recommendation tasks.

As shown in Table 2, diffusion model-based methods generally outperform conventional recommendation methods. This may be attributed to the ability of diffusion models to capture the underlying data distribution and reconstruct target embeddings, resulting in more diverse recommendations. However, it is observed that the multi-interest recommendation method ComiRec achieves better results than general diffusion model-based methods on several metrics. This suggests that user interest features capture stable user preferences, leading to more consistent recommendation results compared to dynamic user features. ADIGRec significantly outperforms all baselines on all datasets for HR@5, HR@10, HR@20, HR@50, NDCG@5, NDCG@10, NDCG@20, and NDCG@50. For example, in terms of retrieving 20 items, compared to the second-best method, ADIGRec achieves improvements in HR@20 of 28.83%, 18.32%, and 42.18%, and improvements in NDCG@20 of 32.03%,

Table 3: Ablation study on different components. The metrics use HR@20(%) and NDCG@20 (%).

Guidance	AI^2M	$\mathcal{L}_{reg}$	YooChoose		KuaiRec		Zhihu	
			HR	NDCG	HR	NDCG	HR	NDCG
$\mathcal{G}^2_{u_i}$	x	x	4.76	1.93	5.31	4.41	2.11	0.76
$\mathcal{G}^1_{u_i}$	x	x	6.14	2.31	7.38	4.40	2.32	0.76
$\mathcal{G}_{u_i}$	x	x	7.23	2.69	7.64	4.65	2.53	0.84
$\mathcal{G}_{u_i}$	✓	x	7.58	2.79	8.24	5.06	2.77	0.91
$\mathcal{G}^1_{u_i}$	x	✓	7.16	2.64	7.49	4.53	2.46	0.82
$\mathcal{G}_{u_i}$	x	✓	7.41	2.73	7.82	4.84	2.65	0.87
$\mathcal{G}_{u_i}$	✓	✓	**7.91**	**3.05**	**8.59**	**5.31**	**3.00**	**0.99**

20.41%, and 30.26% on the YooChoose, KuaiRec, and Zhihu datasets, respectively. Compared with DimeRec, ADIGRec outperforms it across all evaluation metrics. Specifically, our proposed DIFGC module extracts both inherent user interest features and dynamic user features, effectively constructing dynamic interest features as guidance conditions. This expands the representation of interest features and endows them with dynamic characteristics. As a guidance condition in the reverse stage of the model, it facilitates the generation of more diverse results. Additionally, the AI^{2}M module injects the guidance conditions into the noise terms, increasing their interaction and enhancing the guidance strength. In summary, ADIGRec achieves significant advantages over general diffusion model-based methods across three different datasets.

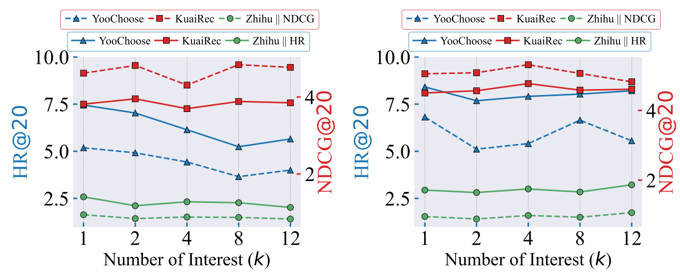
5.3 Ablation Study

5.3.1 ***Ablation on different components.*** To evaluate the impact of different components on recommendation performance, we conducted ablation experiments to assess the combined effects of various guidance inputs, the presence or absence of AI^2M, and the use of $\mathcal{L}_{reg}$ during training. As shown in Table 3, across all datasets, $\mathcal{G}^1 u_i$ outperforms $\mathcal{G}^2_{u_i}$, indicating that capturing user interest as a guidance condition generates results that are closer to the user's true preferences. When considering only different guidance conditions, using our $\mathcal{G}_{u_i}$ as the guidance condition improves HR@20 by 15.08% and NDCG@20 by 14.13% on the YooChoose dataset, HR@20 by 3.4% and NDCG@20 by 5.38% on the KuaiRec dataset, and HR@20 by 8.30% and NDCG@20 by 9.52% on the Zhihu dataset, compared to methods using $\mathcal{G}^1_{u_i}$. This demonstrates that leveraging user dynamic interest features as guidance conditions can more precisely capture users' preferences and enhance the quality of the generated results. When incorporating AI^2M, our method achieves an average improvement of 7.18% in both metrics across the three datasets. This improvement is attributed to the adaptive injection of user dynamic interest features into the noise item embeddings, enabling explicit interaction with the guidance conditions during the generation phase. Finally, when $\mathcal{L}_{reg}$ is applied, performance is significantly improved compared to the non-regularized setting under the same guidance condition and component, as it reduces the impact of user interest routing collapse on the generated results.

5.3.2 ***Ablation on dynamic interest guidance conditions with different operations.*** To highlight the advantages of the method of integrating user dynamic features and user interest features in the DIFGC module, we compare it with three operations,

Table 4: Ablation study of dynamic interest guidance conditions with different feature operations. The metrics use HR@20(%) and NDCG@20 (%).

Operation	YooChoose		KuaiRec		Zhihu	
	HR	NDCG	HR	NDCG	HR	NDCG
Add	2.99	0.98	4.58	3.87	1.91	0.66
Multiple	5.29	2.17	5.46	4.11	2.14	0.73
Concat in k	6.92	2.45	7.47	4.59	2.57	0.88
DIFGC	**7.91**	**3.05**	**8.59**	**5.31**	**3.00**	**0.99**



(a) DimeRec Results (b) ADIGRec Results

Figure 3: Illustrates the result with different interest numbers for three datasets.

namely addition, multiplication and splicing, according to the dimension of k (extended to $k \times d$), and the results are shown in Table 4. From the results, it is clear that our method achieves the best results on three different datasets. The results of addition and multiplication operations are significantly lower than the other methods. The reason may be that direct addition or multiplication of different user features destroys their uniqueness, leading to feature mixing and loss of their respective representativeness. The effect of generating spliced features with size k as the oriented condition is still weaker than our method. It may be since the feature space of the signal is not extended, leading to insufficient model exploration. Unlike our method, which splices according to the d dimension, the feature space of the signal is expanded without destroying the individuality of the two features themselves. During the generation phase, a broader space can be explored, allowing the model to generate diverse results.

5.4 Further Visualization and Analysis

5.4.1 ***The impact on different interest number of*** k. To illustrate the effect of different numbers of interests on recommendation performance, we evaluate our method and DimeRec using various values of k, set to [1, 2, 4, 8, 12]. As shown in Fig. 3, ADIGRec consistently outperforms DimeRec[14] across all datasets for different values of k. Additionally, the recommendation performance of both methods varies under different k settings across the three datasets. To ensure a fair comparison, all multi-interest recommendation methods in this paper are set to $k = 4$, consistent with the setting used for DimeRec. Notice that the variability of recommendation results of ADIGRec under different k settings is smaller than that of

Table 5: The effect of $\mathcal{L}_{reg}$ on the results of DimeRec and ADIGRec for different k. The metrics use HR@20(%) and NDCG@20 (%).

k	methods	$\mathcal{L}_{reg}$	YooChoose		KuaiRec		Zhihu	
			HR	NDCG	HR	NDCG	HR	NDCG
$k=2$	DimeRec	x	7.03	2.55	7.78	4.82	2.11	0.84
		✓	7.28↑3.56%	2.60↑1.96%	8.06↑3.60%	4.87↑1.04%	2.23↑5.69%	0.85↑1.19%
	ADIGRec	x	7.53	2.70	7.88	4.50	2.64	0.81
		✓	7.68↑1.99%	2.89↑7.04%	8.21↑4.18%	5.08↑12.89%	2.81↑6.44%	0.89↑9.88%
$k=4$	DimeRec	x	6.14	2.31	7.26	4.31	2.32	0.88
		✓	7.16↑16.61%	2.64↑14.29%	7.49↑3.17%	4.53↑5.10%	2.61↑12.50%	0.92↑4.55%
	ADIGRec	x	7.58	2.79	8.24	5.06	2.77	0.91
		✓	7.91↑4.35%	3.05↑9.32%	8.59↑4.25%	5.31↑4.94%	3.00↑8.30%	0.99↑8.79%
$k=8$	DimeRec	x	5.24	1.93	7.64	4.84	2.27	0.87
		✓	6.11↑16.60%	2.39↑23.83%	7.82↑2.36%	4.93↑1.86%	2.49↑9.69%	0.90↑3.45%
	ADIGRec	x	6.37	2.40	7.49	4.46	2.55	0.91
		✓	8.03↑26.06%	3.72↑55.00%	8.24↑10.01%	5.06↑13.45%	2.84↑11.37%	0.94↑3.30%
$k=12$	DimeRec	x	5.56	2.10	7.57	4.77	2.02	0.83
		✓	6.35↑14.21%	2.31↑10.00%	7.26↓4.10%	4.31↓9.64%	2.36↑16.84%	0.89↑7.23%
	ADIGRec	x	5.28	2.02	7.61	4.66	2.88	0.95
		✓	8.21↑55.49%	3.13↑54.95.45%	8.29↑8.94%	4.82↑3.43%	3.22↑11.81%	1.07↑12.63%

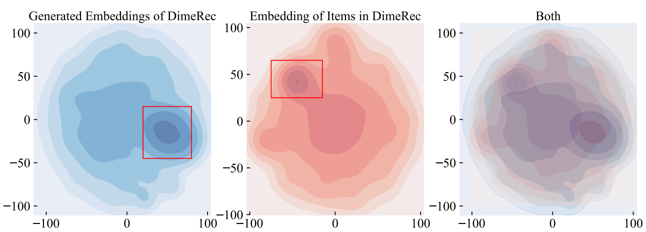


Figure 4: Visualization of embeddings with DimeRec.

DimeRec under the evaluation metrics HR@20 (the maximum difference is 8.68% and 29.66% in the YooChoose dataset, respectively) and NDCG@20 (17.85% and 27.99%). This suggests that combining users' inherent interest features with users' dynamic interest features can capture users' dynamic interests accurately, and enables the model to explore the space of users' long-term and short-term features under different k to improve the generation.

5.4.2 ***Effect of L_{reg} on results under different k.*** This paper further investigates the impact of the proposed regularization term $\mathcal{L}_{reg}$ on other multi-interest SR models. As shown in Table 5, across all datasets, ADIGRec consistently outperforms its variant without the regularization term under different k settings. For DimeRec[14], slightly lower recommendation performance is observed compared to its variant without the regularization term on the KuaiRec dataset with $k = 12$, while improvements are demonstrated across the remaining datasets and k settings. This may be attributed to the difficulty of optimizing the regularization term when the number of interests k is large, which may require additional modules or parameter fine-tuning to achieve better performance. Overall, in multi-interest SR models, $\mathcal{L}_{reg}$ effectively alleviates the issue of interest routing collapse without introducing additional parameters, thereby enhancing the model's performance.

5.4.3 ***Visualization of embeddings using t-SNE.*** To visually compare the recommendation performance of DimeRec and ADIGRec, this section utilizes t-SNE [25] and Gaussian-kde to visualize the item embeddings of the two models on the YooChoose dataset. The results are shown in Figure 4 and Figure 5. In the figures, the blue regions represent the embeddings of the generated results for both models, while the red regions represent the embeddings of all items, with the overlap indicating their recommendation performance. As shown in Figure 4, the darker parts of the blue and red regions are concentrated in the middle of the overlap, indicating that the generated results of DimeRec align with most of the users' preferences. However, the regions highlighted by the red boxes in the figure show no overlap, suggesting that the inherent user interest features limit the exploration space of the model, preventing it from generating certain items that may be of interest to users. In contrast, as shown in Figure 5, the overlap between the darker parts of the blue and red regions is more pronounced, and the distance between the item embeddings in the regions marked by the red boxes is significantly reduced. This indicates that ADIGRec more effectively captures users' dynamic interest features, thereby generating embeddings that better align with users' true preferences.

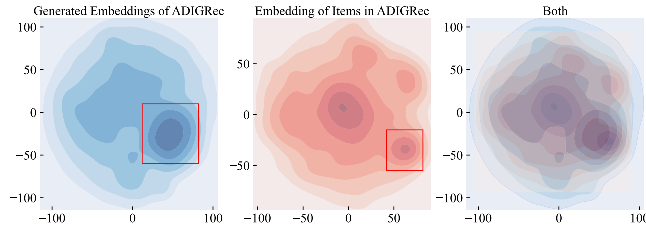


Figure 5: Visualization of Embeddings with ADIGRec.

5.4.4 ***Complexity analysis.*** The main parameters of ADIGRec consist of the query weights for all items, resulting in a complexity of $O(Nd)$, where N is the total number of items and d is the embedding dimension. The overall complexity of ADIGRec is primarily determined by the attention mechanism for guidance condition extraction $O(n^2d)$, the feed-forward neural network $O(nd^2)$, and the cross-attention in the signal injection module $O(kd)$, where n denotes the sequence length. Compared with SASRec[13] and DimeRec[14], the first two terms of the complexity remain the same, while k is smaller than both n and d, ensuring that the overall complexity of ADIGRec is comparable to these models. In terms of inference efficiency, ADIGRec and DimeRec both achieve millisecond-level latency in offline testing.

6 Conclusion

In this paper, we propose a novel generative sequential recommendation framework, ADIGRec, which considers the dynamic interests of users, acknowledging that different users may follow inconsistent interests, and adaptively leverages these interests to further improve recommendation performance. ADIGRec combines user dynamic features and inherent interest features encoded from historical sequences to provide effective guidance conditions. Furthermore, it introduces a module that injects user dynamic interest features into the noise item embeddings, enabling explicit interaction with the guidance conditions during the generation phase and thereby enhancing recommendation diversity. In addition, a new regularization strategy is proposed to mitigate the impact of user interest routing collapse on the generation results. ADIGRec achieves superior results on three publicly available datasets, and extensive experiments demonstrate the effectiveness of our method.

References

[1] Homanga Bharadhwaj, Homin Park, and Brian Y Lim. 2018. RecGAN: recurrent generative adversarial networks for recommendation systems. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 372–376.

[2] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable multi-interest framework for recommendation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 2942–2951.

[3] Ziqiang Cui, Haolun Wu, Bowei He, Ji Cheng, and Chen Ma. 2024. Context Matters: Enhancing Sequential Recommendation with Context-aware Diffusion-based Contrastive Learning. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 404–414.

[4] Hanwen Du, Huanhuan Yuan, Zhen Huang, Pengpeng Zhao, and Xiaofang Zhou. 2023. Sequential recommendation with diffusion models. *arXiv preprint arXiv:2304.04541* (2023).

[5] Ziwei Fan, Zhiwei Liu, Jiawei Zhang, Yun Xiong, Lei Zheng, and Philip S Yu. 2021. Continuous-time sequential recommendation with temporal graph collaborative transformer. In *Proceedings of the 30th ACM international conference on information & knowledge management*. 433–442.

[6] Chongming Gao, Shijun Li, Wenqiang Lei, Jiawei Chen, Biao Li, Peng Jiang, Xiangnan He, Jiaxin Mao, and Tat-Seng Chua. 2022. KuaiRec: A fully-observed dataset and insights for evaluating recommender systems. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 540–550.

[7] Bin Hao, Min Zhang, Weizhi Ma, Shaoyun Shi, Xinxing Yu, Houzhi Shan, Yiqun Liu, and Shaoping Ma. 2021. A Large-Scale Rich Context Query and Recommendation Dataset in Online Knowledge-Sharing. *arXiv preprint arXiv:2106.06467* (2021).

[8] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 843–852.

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

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

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

[12] Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2014. On using very large target vocabulary for neural machine translation. *arXiv preprint arXiv:1412.2007* (2014).

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

[14] Wuchao Li, Rui Huang, Haijun Zhao, Chi Liu, Kai Zheng, Qi Liu, Na Mou, Guorui Zhou, Defu Lian, Yang Song, et al. 2025. DimeRec: A Unified Framework for Enhanced Sequential Recommendation via Generative Diffusion Models. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*. 726–734.

[15] Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. 2022. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems* 35 (2022), 4328–4343.

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

[17] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*. 689–698.

[18] Qidong Liu, Fan Yan, Xiangyu Zhao, Zhaocheng Du, Huifeng Guo, Ruiming Tang, and Feng Tian. 2023. Diffusion Augmentation for Sequential Recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 1576–1586.

[19] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2022. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*. 813–823.

[20] Ruiyang Ren, Zhaoyang Liu, Yaliang Li, Wayne Xin Zhao, Hui Wang, Bolin Ding, and Ji-Rong Wen. 2020. Sequential recommendation with self-attentive multi-adversarial network. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. 89–98.

[21] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. 811–820.

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 234–241.

[23] Noveen Sachdeva, Giuseppe Manco, Ettore Ritacco, and Vikram Pudi. 2019. Sequential variational autoencoders for collaborative filtering. In *Proceedings of the twelfth ACM international conference on web search and data mining*. 600–608.

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

[25] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[27] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 515–524.

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

[29] Liwei Wu, Shuqing Li, Cho-Jui Hsieh, and James Sharpnack. 2020. SSE-PT: Sequential recommendation via personalized transformer. In *Proceedings of the 14th ACM conference on recommender systems*. 328–337.

[30] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 346–353.

[31] Yueqi Xie, Jingqi Gao, Peilin Zhou, Qichen Ye, Yining Hua, Jae Boum Kim, Fangzhao Wu, and Sunghun Kim. 2023. Rethinking multi-interest learning for candidate matching in recommender systems. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 283–293.

[32] Zhe Xie, Chengxuan Liu, Yichi Zhang, Hongtao Lu, Dong Wang, and Yue Ding. 2021. Adversarial and contrastive variational autoencoder for sequential recommendation. In *Proceedings of the web conference 2021*. 449–459.

[33] Zhe Xie, Chengxuan Liu, Yichi Zhang, Hongtao Lu, Dong Wang, and Yue Ding. 2021. Adversarial and contrastive variational autoencoder for sequential recommendation. In *Proceedings of the Web Conference 2021*. 449–459.

[34] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. 2019. Graph contextualized self-attention network for session-based recommendation.. In *IJCAI*, Vol. 19. 3940–3946.

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

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

[37] Xianwen Yu, Xiaoning Zhang, Yang Cao, and Min Xia. 2019. VAEGAN: A Collaborative Filtering Framework based on Adversarial Variational Autoencoders.. In *IJCAI*. 4206–4212.

[38] Fajie Yuan, Xiangnan He, Haochuan Jiang, Guibing Guo, Jian Xiong, Zhezhao Xu, and Yilin Xiong. 2020. Future data helps training: Modeling future contexts for session-based recommendation. In *Proceedings of the web conference 2020*. 303–313.