Engineering Applications of Artificial Intelligence 169 (2026) 114131

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Research paper

Dimos: Diffusion model with unified sequential state space for session-based recommendation

Weiyue Li [a,1], Ming Gao [a,*,1], Bowei Chen [b,1], Jingmin An [a,1], Hao Dong [a,1], Wei Jiang [c,1], Jiafu Tang [a,1]

[a] School of Management Science and Engineering, Key Laboratory of Big Data Management Optimization and Decision of Liaoning Province, Dongbei University of Finance and Economics, Dalian, 116025, China
[b] Adam Smith Business School, University of Glasgow, Glasgow, G116EY, United Kingdom
[c] Key Laboratory of Advanced Design and Intelligent Computing, Ministry of Education, School of Software Engineering, Dalian University, Dalian, 116622, China

ARTICLE INFO

Keywords:
Session-based recommendation
Diffusion model
State space model
Self-attention mechanism

ABSTRACT

Session-based recommendation aims to predict the next item based on the user–item interactions within the current session. Many existing methods adopt discriminative approaches to learn specific preference representations, while few methods introduce generative approaches to learn underlying preference distributions, failing to handle limited and noisy interactions effectively. Moreover, naive implementations of generative models face a trade-off between effectiveness and efficiency, limiting their practical utility. To address these challenges, we propose Dimos, a dual-branch framework comprising an exploring branch and an exploiting branch, which leverage diffusion models and attention networks to capture implicit and explicit preferences, respectively. At the core of Dimos is Bi-MaKAN, a novel backbone architecture featuring a pair of parameter-sharing bidirectional Mamba blocks and a Kolmogorov–Arnold network-based feature fusion layer, designed to enhance both performance and efficiency. To further improve generalization and reduce overfitting, we unify the sequential state spaces of both branches. Additionally, we introduce a linearly weighted fusion mechanism that integrates preference representations from both branches, enabling flexible adjustment of implicit and explicit preference contributions during training and inference. Extensive experiments on three real-world benchmark datasets demonstrate the superiority of Dimos, achieving up to 2.79% improvement in Recall, 3.09% in Mean Reciprocal Rank (MRR), and 3.00% in Normalized Discounted Cumulative Gain (NDCG) over state-of-the-art baselines. Efficiency evaluations show substantial gains, with reductions of 94.32% in Graphics Processing Unit (GPU) memory usage, 66.81% in training time, and 98.80% in inference time. In-depth analyses reveal a collaborative effect between the two branches during both training and inference, with dataset scale modulating their relative importance.

1. Introduction

Session-based recommendation (SBR) aims to capture the dynamic evolution of user preferences within individual sessions and represents a critical subfield of sequential recommendation research (Wang et al., 2022a; Li et al., 2025c). While session-based recommender systems (SBRSs) have shown great potential in enhancing user experiences on online platforms (Feng et al., 2019), their effectiveness is often hindered by limited contextual information and inherently noisy user interactions. One major challenge stems from strict privacy policies, which prevent access to rich user data, such as detailed profiles or long-term interaction histories, thus complicating the design of accurate and personalized SBRSs (Li et al., 2025c). Additionally, session lengths are typically short, with the median number of interactions per session being fewer than six in most widely used benchmark datasets (Li et al., 2025c). Compounding this issue, user behavior during sessions can be erratic or exploratory, leading to incidental or irrelevant interactions that introduce noise and undermine recommendation accuracy (Wang et al., 2022a).

Many SBRSs adopt discriminative approaches to address these challenges. Specifically, some works adopt sequential modeling methods to capture user behaviors, such as recurrent neural networks (RNNs) (Hidasi et al., 2016) and Transformers (Choi et al., 2024). Other approaches represent session sequences as graphs and apply graph-based

* Correspondence to: No. 217 JianShan St., Shahekou District, Dalian, PR China.
E-mail address: gm@dufe.edu.cn (M. Gao).
[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.engappai.2026.114131
Received 2 November 2025; Received in revised form 3 February 2026; Accepted 7 February 2026
Available online 12 February 2026

learning methods to model spatial dependencies among items, including graph attention networks (GATs) (Lv et al., 2025) and gated graph neural networks (GGNNs) (Lin et al., 2025). Despite their success, these discriminative methods often struggle with limited and noisy session interactions, as they rely heavily on ground-truth labels to learn explicit user preferences (Ng and Jordan, 2001; Zheng et al., 2023). This dependency limits their generalizability in real-world scenarios, where user behavior is uncertain and evolving (Li et al., 2024c), making it difficult to estimate user intent in real-time. To overcome these limitations, generative approaches have been explored, modeling preferences as probabilistic distributions rather than fixed representations. Methods based on variational autoencoders (VAEs) (Wang et al., 2022b) and adversarial learning (Chen et al., 2024) aim to capture implicit preferences by learning latent variables that reflect underlying user interests. However, such generative methods face notable challenges, including posterior collapse in VAEs (Zhao et al., 2019) and training instability in adversarial setups (Becker et al., 2022), which hinder their practical performance and adoption.

Diffusion models have recently emerged as a powerful generative framework for recommendation, offering strong probabilistic modeling capabilities and high-quality representation learning. Originally developed for tasks such as semantic segmentation (Tian et al., 2024) and image editing (Tumanyan et al., 2023), they are now gaining traction in the recommendation domain, including applications in sequential recommendation (Zhao et al., 2024; Yang et al., 2024a; Wang et al., 2024d; Li et al., 2024c; Xie et al., 2024; Gupta et al., 2024), where the goal is to capture users' evolving preferences over time. However, existing diffusion-based recommender systems often adopt Transformers (Vaswani et al., 2017) or U-Net architectures (Ronneberger et al., 2015) as the backbone for the denoising process, both of which present significant limitations. U-Net, originally designed for image tasks, struggles with modeling sequential dependencies due to its equivariance constraints (Lenc and Vedaldi, 2019) and limited receptive field, which hinders its ability to capture global user preferences (Li et al., 2024c). Transformers, while effective in sequence modeling, suffer from quadratic complexity in the self-attention mechanism, making them computationally expensive and less scalable in large-scale recommendation scenarios (Liu et al., 2024). These challenges highlight the need for alternative backbones that can better balance effectiveness and efficiency in diffusion-based SBRSs. This motivates the exploration of lightweight, sequentially expressive, and computationally scalable architectures tailored to the unique constraints of session-based recommendation.

Another key challenge in applying diffusion models to SBR lies in the underutilization of feature encoders during the forward diffusion process, which limits the model's ability to learn meaningful underlying preference distributions. While prior works have made progress by optimizing sampling strategies (Song et al., 2021; Zhang et al., 2023), designing noise schedulers (Nichol and Dhariwal, 2021; Kingma et al., 2021), and exploring alternative noise types (Bansal et al., 2023; Qi et al., 2024; Ma et al., 2025), these enhancements still struggle to achieve an effective balance between generation quality and computational efficiency (Yang et al., 2024c), leaving room for further innovation. Inspired by the success of latent diffusion models (LDMs) (Rombach et al., 2022), several recent approaches shift the diffusion process from the raw entity space to a learned latent feature space, improving inference efficiency. However, these methods often depend on large pre-trained models, which are expensive to train and fine-tune. This issue is further exacerbated in recommendation settings, where widely used ID features are far less expressive and general than natural language or visual inputs, thereby limiting the applicability and reusability of foundation models developed in other domains. To reduce deployment costs, some studies (Li et al., 2024c; Yang et al., 2023) have proposed initiating the diffusion process directly from simple initial features. However, these naive feature encodings lack essential prior knowledge, making it more difficult for the model to effectively learn user preferences, especially in short and noisy session contexts.

To address these challenges, we propose Dimos, a Diffusion Model with Unified Sequential State Space for SBR. Dimos features a dual-branch architecture that captures user preferences from both explicit and implicit perspectives. The exploiting branch employs attention networks to extract explicit preferences directly from user–item interactions, while the exploring branch leverages a latent diffusion model to uncover implicit preferences beyond observed behavior. This dual-view representation learning promotes a more comprehensive understanding of user intent. At the core of Dimos is a novel backbone module, Bi-MaKAN, designed to enhance both performance and efficiency. It consists of a pair of parameter-sharing bidirectional Mamba blocks, coupled with a Kolmogorov–Arnold network (KAN)-based feature fusion layer. Compared to standard Mamba, the bidirectional design enables the model to incorporate both past and future session interactions, improving its contextual modeling capabilities. The parameter-sharing mechanism not only reduces model complexity but also helps mitigate overfitting. The KAN-based feature fusion layer integrates features from forward and reverse session contexts, leveraging its strong capacity to model complex, nonlinear relationships in high-dimensional spaces (Hou et al., 2024). Moreover, we share the Bi-MaKAN parameters across the exploring branch's forward diffusion process and the exploiting branch's attention-based encoder. This design establishes a unified sequential state space, ensuring feature consistency and enhancing generalization across branches.

Extensive experiments conducted on three benchmark datasets demonstrate the superior performance of Dimos and confirm the effectiveness of its core components. The in-depth analyses yield several key insights. To quantify the contributions of the exploring and exploiting branches during both training and inference, two weighting mechanisms (i.e., loss weight and preference weight) are introduced to linearly combine their outputs based on relative importance. Our empirical results demonstrate a collaborative effect between the exploring and exploiting branches during both training and inference. Furthermore, we find that data scale acts as a key modulator. On smaller datasets, the exploring branch plays a more substantial role, while the influence of the exploiting branch becomes increasingly significant as the dataset grows. These findings highlight the complementary nature of the two branches and offer practical guidance for future architectural design in SBRSs. In addition, ablation studies verify that the proposed Bi-MaKAN backbone not only delivers robust feature learning but also brings significant efficiency gains. Remarkably, it reduces the number of diffusion steps from thousands to fewer than ten, without sacrificing performance. This advancement greatly enhances the practical applicability of diffusion models and paves the way for future developments in efficient and scalable recommendation systems.

Our work makes three primary contributions.

- We introduce Dimos, a dual-branch framework for session-based recommendation that combines an attention-based exploiting branch for learning explicit preferences and a diffusion-based exploring branch for modeling implicit preferences. This hybrid design enables a nuanced understanding of user intent. Empirical results show that the two branches contribute differently across training and inference phases, with the exploiting branch gaining more influence as the dataset scale increases.
- We present Bi-MaKAN, a novel state–space backbone composed of bidirectional, parameter-sharing Mamba blocks and a Kolmogorov–Arnold network-based fusion layer. This architecture enhances sequential modeling while offering robust preference representations. Importantly, Bi-MaKAN accelerates the diffusion process by reducing the number of required steps from thousands to fewer than ten, without compromising recommendation performance.

- The effectiveness and efficiency of Dimos are validated through extensive experiments on three real-world benchmark datasets, where it outperforms twenty-four competitive baselines. Detailed ablation and efficiency analyses further demonstrate the significance of each module and the model's scalability for large-scale deployment.

2. Preliminaries

Similar to sequential recommendation, SBR aims to predict the next item a user will interact with, based on their current session interactions. Formally, let the item set be denoted as $V = \{v_1, \dots, v_{|V|}\}$, where each v_i represents a unique item and $|V|$ is the total number of items. A session s_u for user u is defined as an ordered sequence $\{v_1, \dots, v_t\}$, where v_t denotes the item interacted with at the tth timestamp. Given a session s_u, the objective of a SBRS is to predict the next interaction v_{t+1} at the subsequent timestamp. The proposed model first learns a preference representation for the user based on their current session and then computes prediction scores for all candidate items in V. The top-K items with the highest scores are subsequently recommended. Next, two core components of the proposed architecture are briefly introduced: the Mamba selective state space model and the denoising diffusion probabilistic model (DDPM).

2.1. Mamba

Mamba, an optimized selective state space model (SSM), is widely used in sequence modeling tasks due to its linear scalability with sequence length and low computational cost. Instead of relying on attention mechanisms, Mamba adopts the state space model framework, encoding context through hidden states during recurrent scans. Its selection mechanism enables control over which parts of the input are integrated into the hidden states, forming the context that influences subsequent embedding updates. Formally, given the item feature sequence $E_u = \{e_1, \dots, e_t\} \in \mathbb{R}^{t\times d}$ corresponding to the session sequence $s_u = \{v_1, \dots, v_t\}$, the state equation and observation equation with zero-order hold (ZOH) discretization and selection mechanism can be written by:

$$\bar{h}_k = \bar{A}h_{k-1} + \bar{B}e_k, \tag{1}$$

$$h_k = C\bar{h}_k, \tag{2}$$

$$\bar{A} = \exp(\Delta A), \tag{3}$$

$$\bar{B} = (\Delta A)^{-1}(\exp(\Delta A) - I)\Delta B, \tag{4}$$

where $h_k \in \mathbb{R}^d$ is the kth hidden state, k is the discrete time step. Moreover, A is the state transition matrix that describes how states change over time, $B = W_B E_u$ is the input matrix that controls how inputs affect state changes, $C = W_C E_u$ denotes the output matrix that indicates how outputs are generated based on current states, and $\Delta = Softplus(BroadCast_D(W_\Delta E_u))$ is the context-aware interval for ZOH discretization. $W_B \in \mathbb{R}^{n\times d}$, $W_C \in \mathbb{R}^{n\times d}$, and $W_\Delta \in \mathbb{R}^{d\times 1}$ are the selection weights, and $\text{BroadCast}_D(\cdot)$ means to broadcast the result to all the dimensions. t, n, and d represent the input length, input feature size, and hidden channel number, respectively.

The discrete SSM, as a linear system, inherently possesses the associated property, allowing it to integrate seamlessly with convolutional computation. Specifically, it can compute the output at each time step independently, as follows:

$$H = E_u * \bar{K}, \tag{5}$$

where $\bar{K} = (C\bar{A}^0\bar{B}, \dots, C\bar{A}^k\bar{B})$ is a set of convolutional kernels, $H = \{h_0, \dots, h_k\}$ is the output hidden state sequence. Exactly, Mamba's structure combines elements of both RNNs and convolutional neural networks (CNNs), which helps it enhance efficiency during both training and inference by leveraging the strengths of sequential modeling and local pattern extraction. For simplicity, we define this process as $y = \text{Mamba}(x)$.

2.2. Diffusion model

Diffusion models are a class of probabilistic generative models that gradually corrupt data by adding noise, and then learn to invert this process to generate new samples. Compared to VAEs and GANs diffusion models have greater representation capacity and are more stable during training (Wei and Fang, 2025; Yang et al., 2024c; Lin et al., 2024). As a prominent framework within diffusion models, the DDPM has achieved notable success across several tasks, including traffic prediction (Li et al., 2024b), image reconstruction (Huberman-Spiegelglas et al., 2024), and sequential recommendation (Li et al., 2024c). Technically, DDPM adopts a forward first-order Markov chain that add Gaussian noise to the input, and a reverse first-order Markov chain that denoise and reconstruct the original input incrementally. Specifically, given the original representation $x_0 \sim q(x_0)$ and subsequent noised representations $\{x_1, \dots, x_L\}$, the forward diffusion process can be formalized using the chain rule of probability and the Markov property as follows:

$$q\left(x_{1:L}|x_0\right) = \prod_{l=1}^{L} q\left(x_l|x_{l-1}\right), \tag{6}$$

$$q\left(x_l|x_{l-1}\right) = \mathcal{N}(x_l; \sqrt{1-\beta_l}x_{l-1}, \beta_l I), \tag{7}$$

where $q(\cdot|\cdot)$ is the transition kernel, $\{\beta_1, \dots, \beta_L\}$ represents a variance schedule that controls the magnitude of noise added at each step in the Markov chain. Instead of computing the transition L times from x_0 to x_L, x_L can be obtained in a single step by marginalizing the joint distribution $q\left(x_{1:L}|x_0\right)$ as follows:

$$q\left(x_l|x_0\right) = \mathcal{N}(x_l; \sqrt{\bar{\alpha}_l}x_0, (1-\bar{\alpha}_l)I), \tag{8}$$

where $\bar{\alpha}_l = \prod_{s=0}^{l}\alpha_l$ and $\alpha_s = 1 - \beta_l$. Given x_0, we can obtain a sample of x_l by sampling a Gaussian vector $\epsilon \sim \mathcal{N}(0, I)$ and applying the transformation $x_l = \sqrt{\bar{\alpha}_l}x_0 + (1-\bar{\alpha}_l)\epsilon$.

In the reverse denoising process, DDPM-based models begin by generating an unstructured noise vector from the prior distribution, then progressively denoise it by running a learnable Markov chain in the reverse time direction. Formally,

$$p_\theta\left(x_{0:L}\right) = p(x_L)\prod_{l=1}^{L} p_\theta\left(x_{l-1}|x_l\right), \tag{9}$$

$$p_\theta\left(x_{l-1}|x_l\right) = \mathcal{N}(x_{l-1}; \mu_\theta\left(x_l, l\right), \Sigma_\theta\left(x_l, l\right)), \tag{10}$$

where $\mu_\theta\left(x_l, l\right)$ and $\Sigma_\theta\left(x_l, l\right)$ are the mean and variance learned by the denoising network with parameters θ. With the reverse Markov chain, we can generate a data sample x_0 by first sampling a noise vector $x_L \sim p(x_L)$, then iteratively sampling from the learnable transition kernel $x_{l-1} \sim p_\theta(x_{l-1}|x_l)$ until $l = 1$.

The success of the sampling process heavily relies on training the denoising network so that the learned reverse Markov chain accurately approximates the true time-reversal of the forward diffusion process. Following the foundational work (Ho et al., 2020), we can optimize the variational lower bound, simplified as the mean-squared error loss, to train the denoising network as follows:

$$\mathcal{L}_{DDPM} = \mathbb{E}_{l\in\mathcal{U}(1,L), x_0\in q(x_0), \epsilon\in\mathcal{N}(0,I)}\left[\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_l}x_0, \sqrt{1-\bar{\alpha}_l}\epsilon, l)\|^2\right], \tag{11}$$

where $\mathcal{U}(1, L)$ is a uniform distribution over the integers 1 to L. Here the denoising network ϵ_θ with parameter θ shifts to predicting the noise vector ϵ given x_l and l, which has been shown to be equivalent to predicting the mean and variance.

3. Dimos

This section outlines the architecture of the proposed Dimos framework, illustrated in Fig. 1. The framework consists of three main

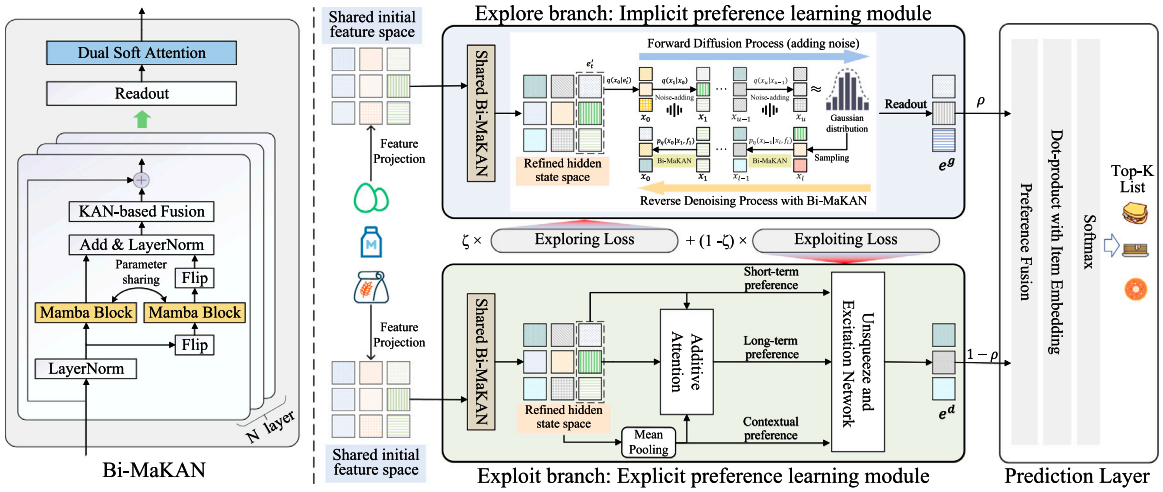


Fig. 1. Schematic overview of the proposed dual-branch framework Dimos for session-based recommendation. The top part depicts the explore branch, which employs a latent diffusion model over the Bi-MaKAN backbone to capture implicit user preferences beyond observed interactions. The bottom part illustrates the exploit branch, which adopts attention networks over the Bi-MaKAN backbone to model explicit preferences directly from user–item interactions. Both branches share the Bi-MaKAN backbone (left part), which consists of parameter-sharing bidirectional Mamba blocks and a KAN-based fusion layer, establishing a unified sequential state space for consistent feature learning.

components: the implicit preference learning module, the explicit preference learning module, and the prediction layer. The implicit preference module (explore branch) utilizes Bi-MaKAN in conjunction with a DDPM to model latent user preferences. In parallel, the explicit preference module (exploit branch) incorporates Bi-MaKAN and an Unsqueeze-and-Excitation Network (UE-Net) to capture observable user intents. To enhance model consistency and reduce overfitting, both branches operate over a shared sequential state space generated by Bi-MaKAN. The final prediction layer computes relevance scores for all candidate items and outputs the top-K recommendations.

3.1. Explore branch: Implicit preference learning module

Although LDM (Rombach et al., 2022) demonstrates that using diffusion models in the latent feature space from well-pretrained autoencoders can enhance performance and efficiency, this approach is not well-suited for addressing session-based recommendation tasks. On the one hand, the pretrain-finetune paradigm usually requires significant computational resources and time. On the other hand, item IDs, commonly used by SBRSs, vary across different scenarios, limiting the effectiveness of pre-trained SBRSs. Therefore, designing an effective feature encoder is crucial to create a robust latent feature space, enabling diffusion models to fully leverage their generative potential.

Instead of the widely adopted U-Net and Transformer, we propose a Mamba-based feature encoder due to its strong sequence modeling capacity and linear scalability with sequence length. Specifically, a pair of parameter-sharing Mamba blocks are employed to effectively capture sequential item dependencies from bidirectional session contexts. Here, parameter sharing refers to full weight sharing across the two directional blocks, meaning that the same set of trainable parameters is used for processing both the forward and backward sequences. Subsequently, we introduce the Kolmogorov–Arnold network-based feature fusion component to integrate the bidirectional sequential item dependencies. Formally,

$$\bar{E}_u = \text{Mamba}\left(E_u\right) + \text{Flip}\left(\text{Mamba}\left(\text{Flip}\left(E_u\right)_{ld}\right)\right)_{ld}, \tag{12}$$

$$E'_u = \text{KAN}(\bar{E}_u), \tag{13}$$

where $\text{Flip}(\cdot)_{ld}$ denotes flipping the input sequence along the length dimension. Here, we present two options for the fusion method $\text{KAN}(\cdot)$. The first is Fourier KAN (Xu et al., 2024). Formally,

$$\text{Fourier-KAN}(x) = \sum_{i=1}^{d}\sum_{k=1}^{g}\left(\cos\left(kx\right)a_{ik} + \sin\left(kx\right)b_{ik}\right), \tag{14}$$

where a_{ik} and b_{ik} are trainable Fourier coefficients. g refers to the grid size, which controls the number of terms (frequencies) used in the Fourier series expansion. Specifically, g determines the number of sine and cosine terms incorporated into the Fourier coefficients for each input dimension. Compared to vanilla KAN, Fourier KAN uses 1D Fourier coefficients rather than B-spline coefficients, which simplifies the optimization process and decreases the number of learnable parameters (Xu et al., 2024).

The second option is group-rational KAN (GR-KAN). Formally,

$$\text{GR-KAN}(x) = \left[\sum_{i=1}^{d}\omega_{i,1}F_{\lfloor 1/d_g\rfloor}(x_i) \quad \dots \quad \sum_{i=1}^{d}\omega_{i,d}F_{\lfloor 1/d_g\rfloor}(x_i)\right], \tag{15}$$

where

$$F(x_i) = \frac{a_0 + a_1x_i + \cdots + a_mx_i^m}{1 + |b_1x_i + \cdots + b_nx_i^n|},$$

and i is the index of the input dimension, g denotes the number of groups. Each group contains $d_g = d_{in}/g$ dimensions, with the group index determined by $\lfloor i/d_g \rfloor$. $F(\cdot)$ is the input-wise rational function, specifically implemented using the Safe Padé Activation Unit (Molina et al., 2020) with the coefficients a_m and b_n to ensure training stability. Compared to vanilla KAN, GR-KAN replace B-spline functions with rational functions as the base functions to enhance the model's expressiveness, stability, and computational efficiency (Yang and Wang, 2024; Zhang et al., 2025). For simplicity, we omit layer normalization and dropout operations in the equations above. For brevity, we define this process as $E'_u = \text{Bi-MaKAN}(E_u;\xi)$, where ξ are the parameters of the Bi-MaKAN. Our Bi-MaKAN incorporates sequential dependencies into item representations, allowing it to refine the initial hidden feature space and capture contextual relationships more effectively.

Subsequently, by treating the user's historical behavior sequence as an information diffusion process (Niu et al., 2024), the DDPM is adopted to learn the underlying distribution for capturing the evolution of user preferences: $e^g_u = \text{DDPM}(E'_u)$.

In the forward diffusion process, we incrementally add Gaussian noise to the sequential dependency-aware hidden state of the target item e'_i, until the complete transformation into a thoroughly Gaussian noise x_u after T diffusion steps. Then, the current noised hidden state x_u is adopted to modify the hidden state of each historical item in s_u, denoted as $Z_{x_u} = \{z_1, \dots, z_l\}$. Furthermore, the denoising network is used to refine the reconstructed the hidden state of the target item $\hat{e}'_i$

from Z_{x_u}, bringing it closer to the original hidden state e'_i. Formally, the forward diffusion process can be formulated as follows:

$$x_u = q(x_u|x_0, s_u), \tag{16}$$

$$\hat{x}_0 = \text{Bi-MaKAN}(Z_{x_u}; \eta), \tag{17}$$

$$z_i = e'_i + \lambda_i \odot (x_u + f_u), \tag{18}$$

where $\odot$ denotes the Hadamard product. $f_{(\cdot)}$ is the sinusoidal position encoding (Ho et al., 2020) that represents different diffusion steps, allowing the model to recognize the current noise level. λ_i is sampled from a Gaussian distribution $\lambda_i \sim \mathcal{N}(\delta, \delta)$, where δ is a hyperparameter which defines both the mean and variance. λ_i modulates the amount of noise injected, introducing uncertainty into the modeling of user interest evolution.

In the reverse denoising process, we aim to recover the sequential dependency-aware hidden state of the target item x_0 iteratively from a pure Gaussian noise x_l. We firstly sample the noised target item representation x_l from a standard Gaussian distribution $\mathcal{N}(0, I)$. Subsequently, similar to the diffusion process, x_l is adopted to adjust the hidden state of each historical item in s_u, denoted as Z_{x_l}. Furthermore, the denoised network is used to estimating the original hidden state $\hat{x}_0$. After that, x_{l-1} is estimated by Eq. (10). We repeat the above process until we arrive at x_0. Formally,

$$\hat{x}_0 = \text{Bi-MaKAN}(Z_{x_l}; \eta), \tag{19}$$

$$z_i = e'_i + \lambda_i \odot (x_l + f_l), \tag{20}$$

$$x_{l-1} = p(x_{l-1}|\hat{x}_0, x_l), \tag{21}$$

where λ_i allows the importance of each latent aspect of a historical item to be iteratively adjusted in a user-aware manner during the reverse denoising process. Moreover, the reparameterization trick is adopted to facilitate better optimization by connecting model parameters with noise variables. Formally,

$$x_{l-1} = \tilde{\mu}_l(x_l, \hat{x}_0) + \tilde{\beta}_l \epsilon', \tag{22}$$

$$\tilde{\mu}_l(\mathbf{x}_l, \hat{x}_0) = \frac{\sqrt{\overline{\alpha}_{l-1}}\beta_l}{1-\overline{\alpha}_l}\hat{x}_0 + \frac{\sqrt{\alpha_l}(1-\overline{\alpha}_{l-1})}{1-\overline{\alpha}_l}\mathbf{x}_l, \tag{23}$$

$$\tilde{\beta}_l = \frac{1-\overline{\alpha}_{l-1}}{1-\overline{\alpha}_l}\beta_l, \tag{24}$$

where ϵ' is the vector sampled from Gaussian distribution $\mathcal{N}(0, I)$. So far, we obtain the refined representation of user preference $e^g_u = \hat{x}_0$, which considers the sequential interactions and uncertain behaviors.

3.2. Exploit branch: Explicit preference learning module

This module adopts Bi-MaKAN and UE-Net to capture the explicit preference e^d_u. To ensure the consistency of the sequential dependency-aware hidden state space, we share parameters between the Bi-MaKAN in this module and the Bi-MaKAN used before the forward diffusion process: $E'_u = \text{Bi-MaKAN}(E_u; \xi)$. This parameter sharing establishes the unified sequential state space, a core design principle of Dimos.

Formally, the unified sequential state space $\mathcal{F}_\xi$ is the feature space induced by the shared Bi-MaKAN with parameter ξ. Formally, for the input item feature sequence E_u, there exists a unique corresponding item state sequence E'_u. The unification is ensured by the parameter sharing of the Bi-MaKAN across the explore and exploit branches. Consequently, for the same input E_u, both branches compute the identical state sequence E'_u.

The unified sequential state space guarantees that the implicit preference distribution explored by the diffusion process and the explicit intents modeled by the attention mechanism are semantically aligned in the same feature space, enabling their effective fusion. Furthermore, the unified sequential state space acts as a strong regularizer, allowing the model to learn robust, general-purpose sequential features from both generative and discriminative signals simultaneously, which improves parameter efficiency and reduces the risk of overfitting in either branch.

Subsequently, we identify various user intents and adaptively fuse them to generate the robust preference representation. Specifically, in line with related works (Li et al., 2023a,b), the hidden state of the last item is considered as the short-term intent, i.e. $c_{st} = e'_t$. Compared to directly capturing long-term intent based on short-term intent, we additionally introduce contextual intent to alleviate the impact of the potential noised short-term intent caused by unexpected clicks or interest drift. Formally, we adopt the additive attention network as follows:

$$c_{lt} = \sum_{v_i \in s_u} \phi_i e'_i, \tag{25}$$

$$\phi_i = Softmax\left(W_1^T \sigma\left(\left[W_2 e'_i \| W_3 c_{con} \| W_4 c_{st}\right]\right)\right), \tag{26}$$

$$c_{con} = \frac{1}{t}\sum_{v_i \in s_u} e'_i, \tag{27}$$

where σ denotes the sigmoid activation function, $W_1 \in \mathbb{R}^{3d}$, and $W_2, W_3, W_4 \in \mathbb{R}^{d\times d}$ are the learnable parameters. Furthermore, the UE-Net is adopted to fuse various preference representations as follows:

$$C_{intent} = Stack\left(W_5 c_{st}, W_6 c_{con}, W_7 c_{lt}\right) \tag{28}$$

$$\gamma = Softmax\left(C_{intent}\right) \tag{29}$$

$$e^d_u = Squeeze - sum\left(\gamma C_{intent}\right), \tag{30}$$

where $Stack(\cdot)$ denotes the concatenation in the additional preference dimension, i.e. unsqueeze operation. γ is the adaptive preference weight for each preference representation, i.e. excitation operation. $Squeeze - sum(\cdot)$ denotes a dimension reduction operation based on sum fusion. $W_5, W_6, W_7 \in \mathbb{R}^{d\times d}$ are the learnable parameters. Instead of a shallow feed-forward network, our UE-Net offers greater flexibility for preference fusion by assigning dynamic attention weights across additional preference dimensions.

3.3. Prediction layer and optimization

The prediction layer first calculates the recommendation score for each candidate item, then selects the top-K items to form the recommendation list. For clarity in understanding the significance of explicit and implicit preferences in recommendations, a weighted linear method is adopted to fuse both types of preferences: $e_u = \rho \cdot e^g_u + (1-\rho) \cdot e^d_u$, where $\rho \in [0, 1]$ is the predefined preference weight. Subsequently, the recommended score is computed by inner product: $\text{score}_i = e_u \cdot e_i$. Furthermore, the optimization objective is defined as the cross-entropy of the ground-truth and the prediction scores. Similarly, we predefined the loss weight $\zeta \in [0, 1]$ to indicate the different importance of both modules during model training. Formally,

$$\hat{y}^g_i = Softmax(e^g_u \cdot e_i), \qquad \hat{y}^d_i = Softmax(e^d_u \cdot e_i), \tag{31}$$

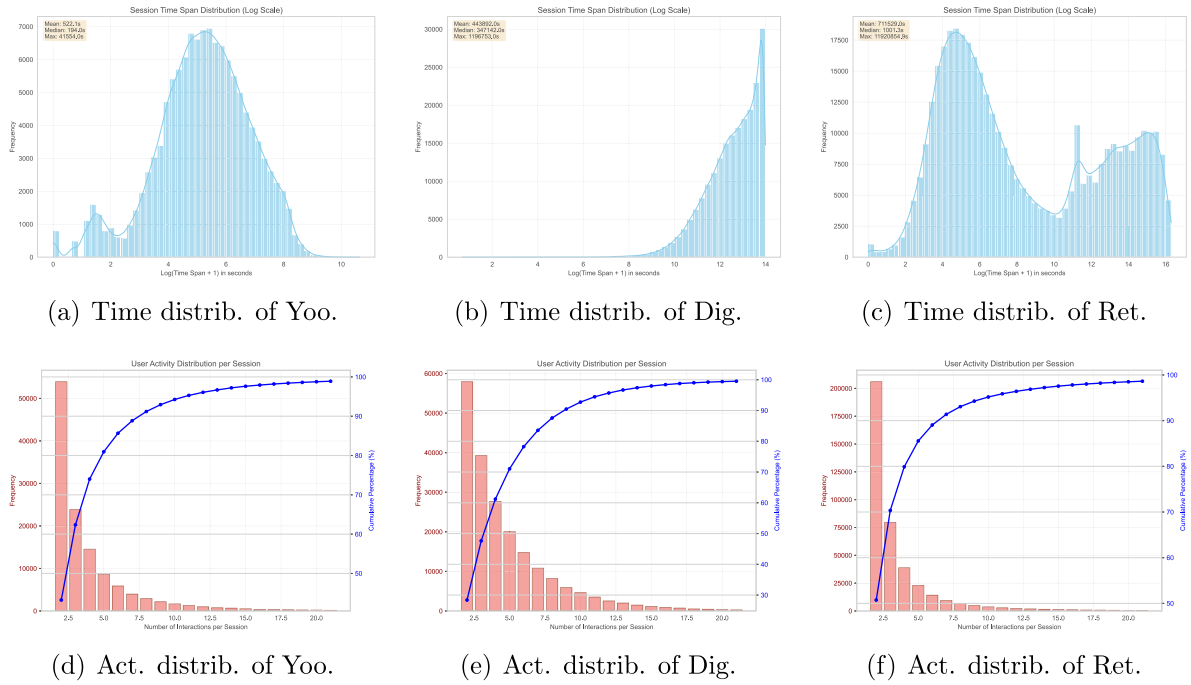
$$\mathcal{L}_g = -\sum_{i=1}^{|V|} y_i \log(\hat{y}^g_i), \qquad \mathcal{L}_d = -\sum_{i=1}^{|V|} y_i \log(\hat{y}^d_i), \tag{32}$$

$$\mathcal{L}_{total} = \zeta \cdot \mathcal{L}_g + (1-\zeta) \cdot \mathcal{L}_d, \tag{33}$$

where y_i is the one-hot encoding vector of the ground truth.

3.4. Time complexity analysis

The time complexity of the proposed Dimos framework stems from four key components. First, the time complexity of the Bi-MaKAN (Fourier KAN version) shared between implicit preference learning module and implicit preference learning module is $O(2td^2(1+g))$, where t is the session length, d is the hidden dimension, and g is the grid size for the Fourier transform operation. Second, the time complexity of the DDPM is $O(2tTd^2(1+g))$, where T is the number of diffusion steps. Third, the time complexity of the long-term intent computation and



(a) Time distrib. of Yoo. (b) Time distrib. of Dig. (c) Time distrib. of Ret.

(d) Act. distrib. of Yoo. (e) Act. distrib. of Dig. (f) Act. distrib. of Ret.

Fig. 2. Session time span distribution (Time distrib.) and user activity distribution (Act. distrib.) of Yoochoose 1/64 (Yoo.), Diginetica (Dig.) and Retailrocket (Ret.).

the UE-Net $O(4td^2+8td)$. Finally, the time complexity of the prediction layer is $O(|V|d)$, where $|V|$ is the number of candidate items. Therefore, the overall time complexity of Dimos is $(6+2g+2T+2gT)td^2+(8t+|V|)d$. With usually a grid size of 1, the overall time complexity depends on the diffusion steps, session length, and hidden dimensions.

4. Experiments

This section conducts comprehensive experiments to evaluate the effectiveness and efficiency of our proposed model. They are designed to address the following key research questions: (RQ1) How does Dimos perform in SBR tasks compared to twenty-four competitive baseline models across three real-world datasets? (RQ2) What impact do the main components of Dimos have on recommendation performance? (RQ3) What specific roles do the exploit and explore branches play during the training and inference stages? (RQ4) How does Dimos perform under varying hyperparameter settings? (RQ5) How well does Dimos generalize across user groups with different levels of activity? (RQ6) What is the impact of varying session lengths on the effectiveness of Dimos? (RQ7) How does the computational efficiency of Dimos compare with that of representative baseline models? (RQ8) Finally, how can we interpret both explicit and implicit user preferences captured by Dimos? The implementation code of our proposed model is publicly available for reference at: https://anonymous.4open.science/r/Dimos-1374.

4.1. Experimental settings

We begin by introducing the datasets used in the experiments, along with the preprocessing methods applied to prepare the data. Next, we outline the evaluation metrics employed to assess the performance of all models. Finally, we describe the baseline models used for comparison and provide details on the implementation settings.

4.1.1. Datasets and preprocessing

The overall performance of our Dimos and twenty-four competitive baseline models are evaluated on three public real-world datasets, including Yoochoose, Diginetica, and Retailrocket. Specifically, Yoochoose, released for the RecSys Challenge 2015, contains anonymized e-commerce clickstream data collected over six months.[2] Due to its large scale, we extracted the most recent 1/64 of the dataset based on the timestamp, consistent with Pan et al. (2020), Qiu et al. (2022), Qiu et al. (2020), and Wu et al. (2019). Diginetica, released for the CIKM Cup 2016, comprises anonymized user clickstream data collected from an online retail platform.[3] Retailrocket, sourced from an e-commerce platform over 4.5 months, includes three anonymized behavioral event types: view, add-to-cart, and transaction.[4] For our experiments, we exclusively utilized view interactions from Retailrocket.

To gain deeper insights into the characteristics of the datasets, we analyze the session time span distributions and user activity distribution across the three benchmark datasets. Fig. 2(a), (b), and (c) present the session time span distributions for the three benchmark datasets, revealing distinct temporal engagement patterns across different e-commerce platforms. The Yoochoose 1/64 dataset exhibits relatively brief user sessions, indicating predominantly short, goal-oriented browsing behavior. In contrast, both Diginetica and Retailrocket demonstrate significantly longer session durations. Fig. 2(d) (e), and (f) illustrate the user activity distributions across the three datasets. All datasets exhibit similar long-tailed patterns, i.e., the majority of sessions consist of very few interactions. In summary, the observed patterns further validate the selection of these three datasets for comprehensive evaluation, as they collectively represent diverse real-world scenarios in terms of user engagement intensity and temporal dynamics.

Following Hou et al. (2022), Pan et al. (2022b), and Qiu et al. (2022), the sessions longer than 1 and the items appearing more than 4 times are reserved in all the datasets. Table 1 shows the statistics for the three datasets. For fair comparison, the data augment method (Tan

[2] https://recsys.acm.org/recsys15/challenge
[3] http://cikm2016.cs.iupui.edu/cikm-cup
[4] https://www.kaggle.com/retailrocket/ecommerce-dataset

Table 1
Summary of the used datasets.

Dataset	# Sessions	# Items	# Interactions	Avg. session length	Avg. action per item	Sparsity
Yoochoose 1/64	124 724	17 606	528 858	4.24	30.04	99.9759%
Diginetica	204 062	42 172	989 204	4.85	23.46	99.9885%
Retailrocket	328 904	58 000	1 413 976	4.30	24.38	99.9926%

et al., 2016) is adopted which generates the sessions and corresponding labels by splitting the input session. For example, for an input session $S = \{v_1, \ldots, v_{|S|}\}$, the generated sessions and the corresponding labels are $(\{v_1\}, v_2), (\{v_1, v_2\}, v_3), \ldots, (\{v_1, \ldots, v_{|S|-1}\}, v_{|S|})$. Moreover, leave-one-out strategy is adopted to split datasets. Specifically, we preserve the last and the second last interactions in each session as the testing and validation data, while the rest is taken as the training data.

4.1.2. Evaluation metrics, baseline models and their implementation

We adopt three evaluation metrics for recommendation performance: *Recall@K*, *Mean Reciprocal Rank* (MRR@K), and *Normalized Discounted Cumulative Gain* (NDCG@K). Specifically, Recall@K measures the proportion of test cases in which the correct item appears within the top-K recommended list. MRR@K is a ranking-based metric that computes the average reciprocal rank of the first relevant item across all test cases. NDCG@K, another ranking metric, considers both the relevance and the position of recommended items, assigning higher scores to correctly ranked items that appear earlier in the list. To enable a more comprehensive comparison, we evaluate each metric at $K = 5, 10, 15$, and 20. To mitigate potential biases, the rankings of all candidate items are considered during evaluation.

Regarding efficiency, we employ five metrics: GPU memory consumption, training time, inference time, the number of parameters, and floating point operations (FLOPs). GPU memory consumption denotes the memory consumed during model training and inference. Training time corresponds to the computational cost per epoch during the training phase, while inference time corresponds to the cost per epoch during inference. The number of parameters indicates the total amount of learnable weights in the model, reflecting its scale, capacity, and storage requirements. FLOPs measure the total computational workload required for a single forward pass, commonly used to estimate the model's theoretical computational complexity and speed.

Table 2 summarizes the baseline models, which can be broadly categorized into six groups: (1) non-neural methods, (2) traditional neural methods, (3) GNN-based methods, (4) Mamba-based methods, (5) traditional generative methods, and (6) diffusion-based methods. Our proposed Dimos and all the baseline models are implemented based on the popular recommendation framework RecBole (Zhao et al., 2021) for easy development and reproduction. Following Peintner et al. (2023), Wu et al. (2019), the embedding dimension and batch size is set to 100. The initial learning rate is set to 0.001 and will decay by 10% after each 3 epochs. The Adam optimizer is adopted to train parameters. The max length of session is set to 20 for all the baseline models and our Dimos. The SSM state expansion factor, the local convolution width, and the block expansion factor is respectively set to 50, 4, and 2 for all models that include Mamba blocks. To alleviate over-fitting problem, the dropout strategy with 20% ratio has been applied to our model. The stacking layer number of Bi-MaKAN is searched among {1, 2, 3, 4}. The diffusion step is searched among {2, 5, 10, 20, 40, 80, 160, 320, 640}. The noise strength δ ranges from 1e-6 to 1e-2, with a step size of 10. The noise scheduler is selected from {sqrt, cosine, truncate cosine, truncate linear }. The preference weight and loss weight range from 0 to 1, with a step size of 0.1. The grid-size of Fourier KAN is set as 1. The group number of GR-KAN is set as 10 and its rational layer is initialized to behave like an identity function. For all other parameters, the baseline models follow the optimal configurations reported in their respective references. To ensure robustness and reduce variance, all models were evaluated over 5 independent runs with different random seeds. The best results of all the models are recorded. Additionally, the statistically significant results ($p < 0.05$) are confirmed by a paired t-test against the best baseline model on each dataset, ensuring that the observed improvement is not attributable to random chance.

4.2. Overall performance

To address RQ1, we conduct an empirical evaluation of overall performance on the session-based recommendation task across three real-world e-commerce datasets: Yoochoose 1/64, Diginetica, and Retailrocket. To enhance readability and support consistent cross-dataset comparisons, we report the average performance of our proposed *Dimos* model and all baseline models in terms of Recall, MRR, and NDCG across varying lengths of the top-K recommendation list, as shown in Table 3. Additional dataset-specific results and implementation details are provided in Appendix.

On average, our proposed *Dimos* model outperforms all baseline methods across all metrics and datasets, demonstrating its strong generalization ability. As shown in the last row of Table 3, Dimos achieves improvements over the best-performing baseline ranging from 1.43% to 2.79% in terms of Recall@K ($K = 5, 10, 15, 20$). Similarly, the improvements in MRR@K and NDCG@K fall within 2.87% to 3.09% and 2.49% to 3.00%, respectively. These results suggest that Dimos is particularly effective at ranking the target item higher in the recommendation list, rather than merely including it. Several design choices contribute to the superior performance of Dimos.

First, Dimos incorporates explicit and implicit preference learning modules to balance preference exploitation and exploration. The exploitation branch learns specific preference representations, while the exploration branch captures underlying preference distributions. Serving as the backbone, Bi-MaKAN exhibits strong context modeling capability for both branches. Specifically, Bi-MaKAN transforms the latent item embedding space into a sequentially aware hidden state space, effectively capturing meaningful user behavior patterns. In addition, the proposed UE-Net adaptively integrates multiple user intents into a unified preference representation, further enhancing the model's capacity for preference learning.

Second, Mamba-based baseline models show competitive performance. In particular, RecMamba consistently outperforms RNN-based, CNN-based, and Transformer-based baselines, achieving the best results across most metrics. This highlights the potential of Mamba-style structures in session-based recommendation. Compared to CNN-based models (Yuan et al., 2019), Mamba benefits from a global receptive field that enables it to capture the evolution of user preferences throughout the entire session. In contrast to RNN-based methods (Hidasi et al., 2016), Mamba employs a selective mechanism that determines which interactions to retain in memory, enhancing its ability to model long-range dependencies. Compared to Transformer-based approaches (Kang and McAuley, 2018), Mamba trains in parallel with superlinear time complexity and switches to a recurrent mode with linear complexity during inference, avoiding the quadratic overhead of self-attention mechanisms.

Third, among generative models, diffusion-based methods (Li et al., 2024c; Liu et al., 2025a) outperform traditional VAE-based (Wang et al., 2022b), adversarial learning-based (Chen et al., 2024), and hybrid models (Xie et al., 2021), showcasing their ability to learn compact, high-quality representations in a probabilistic generative manner. VAE-based models suffer from limited representation capacity and posterior collapse, while adversarial models face training instability,

Table 2
Summary of the baseline models.

Category	Method	Description
Non-neural methods	Pop	Pop recommends the most universally popular items to all users, irrespective of individual preferences. Despite its simplicity and efficiency, POP often serves as a fundamental baseline model.
	Item-KNN (Sarwar et al., 2001)	Item-KNN is a collaborative filtering method that predicts a user's preference for a target item by aggregating their interactions with items most similar to it.
	CORE-ave (Hou et al., 2022)	CORE-ave is a simple yet effective framework for session-based recommendation that maintains a consistent representation space throughout encoding and decoding, addressing the issue of inconsistent predictions.
Traditional neural methods	GRU4Rec (Hidasi et al., 2016)	GRU4Rec stacks multiple gated recurrent unit (GRU) layers to encode the session sequence into a final state. It also applies the ranking loss to train the model.
	NextItNet (Yuan et al., 2019)	NextItNet is the CNN-based method that adopts dilated convolutions to increase receptive fields instead of suboptimal pooling operation.
	SASRec (Kang and McAuley, 2018)	SASRec introduces causal self-attention network to capture sequential item transition patterns. It also employs positional encoding and layer normalization to preserve sequence order and stabilize training,
	MSDCCL (Zhu et al., 2024)	MSDCCL employs soft and hard denoising strategies to mitigate noise while preserving informative interactions, coupled with cross-signal contrastive learning to enhance robustness by contrasting valid signals against noisy counterparts.
GNN-based methods	SRGNN (Wu et al., 2019)	SRGNN transforms session sequences into session graphs and applies graph gated neural network to capture pair-wise item transition relations.
	CMGNN (Wang et al., 2023a)	CMGNN is a novel contrastive multi-level graph neural network that captures complex and high-order item transition information.
	GSAU (Cao et al., 2025)	GSAU designs a novel objective function to enforce alignment and uniformity between graph learning module and sequence learning module.
Mamba-based methods	Mamba4Rec (Liu et al., 2024)	Mamba4Rec enhances the basic Mamba block with Transformer techniques for efficient sequential recommendation.
	RecMamba (Yang et al., 2024b)	RecMamba adopts the basic Mamba block to capture long-term user preference.
	EchoMamba4Rec (Wang et al., 2024b)	EchoMamba4Rec enhances Mamba-based models with Fourier transform layer, GLU, and bi-directional mechanism.
	MLSA4Rec (Su and Huang, 2024)	MLSA4Rec combines the basic Mamba block with low-rank decomposition self-attention to leverage complementary advantages.
	SIGMA (Liu et al., 2025b)	SIGMA proposes a partially flipped Mamba with a dense selective gate and a feature extract GRU, addressing challenges in context modeling and short sequence modeling.
	SS4Rec (Xiao et al., 2025)	SS4Rec leverages SSMs to capture the continuous-time dynamics of user interests, addressing limitations of discrete-time methods in modeling irregularly spaced interactions.
Traditional generative methods	ACVAE (Xie et al., 2021)	ACVAE incorporates adversarial learning under the AVB framework, contrastive learning for user personalization, and a convolutional layer to enhance short-term sequential relationships.
	ContrastVAE (Wang et al., 2022b)	ContrastVAE is a two-branched VAE framework guided by ContrastELBO and employing model and variational augmentation.
	SparseEnNet (Chen et al., 2024)	SparseEnNet is an adversarial method that explores the hidden space to generate more robust enhanced items in sequence recommendation.
Diffusion-based methods	DiffuRec (Li et al., 2024c)	DiffuRec fuses the target item embedding into the diffusion process to generate historical interaction representations.
	DiffRec (Wang et al., 2023b)	DiffRec corrupts user interaction histories by injecting scheduled Gaussian noise in the forward process, then iteratively recovers the original interactions using a parameterized neural network.
	L-DiffRec (Wang et al., 2023b)	L-DiffRec compresses high-dimensional user–item interactions into a latent space via item clustering and variational encoding, then performs the diffusion process in this compressed space before decoding back to the original interaction dimension for ranking.
	CaDiRec (Cui et al., 2024)	CaDiRec uses conditional generation for augmented views and employs both preceding and succeeding items for contrastive learning.
	PreferDiff (Liu et al., 2025a)	PreferDiff introduces a personalized ranking loss to enhance ranking accuracy and speed up convergence by focusing on hard negatives in diffusion-based recommenders.

mode collapse, and convergence issues. Also, their relatively weak feature extraction capabilities limit them to capture complex behavioral patterns.

Lastly, consistent with recent findings (Ma et al., 2024a; Qu and Nobuhara, 2025; Niu et al., 2025; Li et al., 2025b; Benigni et al., 2025), we observe that not all diffusion-based models achieve competitive performance. Specifically, models such as DiffRec and L-DiffRec (Wang et al., 2023b) employ MLP-based denoising networks, which lack sequential inductive bias and thus hinder effective distribution learning in the diffusion process. Furthermore, CaDiRec (Cui et al., 2024) struggles on datasets with short interaction sequences (Liu et al., 2021), where contrastive learning based on data-level augmentation (Dang et al., 2024) is less effective. Based on these insights, our Dimos model omits contrastive learning and instead adopts Bi-MaKAN as the denoising network for the diffusion process.

4.3. Ablation studies

To address RQ2, we conduct four groups of ablation experiments designed to evaluate the effectiveness of the overall framework, the structure of Bi-MaKAN, and the functionality of Bi-MaKAN when serving as the forward feature encoder and the denoising network. Additionally, we examine the performance of alternative neural architectures when used as both the feature encoder and the denoising network. Furthermore, we investigate the effectiveness of our simple linear preference fusion method. Similarly, to ensure robustness and reduce variance,

Table 3
Average performance of the examined models across the three datasets.

Models	Category	@5			@10			@15			@20		
		Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG
Pop	Non-neural methods	2.00	1.05	1.28	3.21	1.22	1.67	4.09	1.29	1.91	5.17	1.35	2.16
Item-KNN		18.57	11.41	13.09	25.70	12.36	15.39	29.99	12.70	16.53	32.97	12.87	17.23
CORE-ave		43.35	28.45	32.16	53.45	29.81	35.44	59.04	30.25	36.92	62.68	30.46	37.78
GRU4Rec	Traditional neural methods	42.38	28.22	31.75	52.33	29.56	34.97	57.76	29.99	36.41	61.42	30.19	37.27
NextItNet		35.18	22.78	25.86	44.77	24.06	28.96	50.21	24.49	30.40	53.92	24.69	31.27
SASRec		43.01	28.99	32.48	53.27	30.37	35.80	58.95	30.82	37.30	62.76	31.03	38.21
MSDCCL		35.61	22.95	26.09	45.41	24.26	29.26	51.11	24.71	30.77	55.07	24.93	31.71
SRGNN	GNN-based methods	42.89	28.42	32.02	52.97	29.77	35.29	58.41	30.20	36.73	62.10	30.41	37.60
CMGNN		44.18	29.18	32.91	54.35	30.54	36.21	59.76	30.97	37.64	63.42	31.17	38.50
GSAU		41.25	27.51	30.93	51.37	28.87	34.21	57.01	29.31	35.70	60.88	29.53	36.61
Mamba4Rec	Mamba-based methods	43.56	29.01	32.63	53.49	30.35	35.85	58.91	30.77	37.29	62.51	30.98	38.14
RecMamba		44.59	29.92	33.57	54.40	31.24	36.75	59.72	31.66	38.17	63.23	31.86	38.99
EchoMamba4Rec		43.48	29.56	33.03	52.76	30.81	36.04	57.75	31.21	37.36	61.16	31.39	38.17
MLSA4Rec		43.35	29.31	32.80	52.84	30.58	35.88	58.00	30.99	37.25	61.44	31.18	38.06
SIGMA		42.93	28.67	32.22	52.80	29.99	35.41	58.27	30.42	36.86	61.91	30.63	37.72
SS4Rec		44.17	29.61	33.24	54.02	30.93	36.43	59.39	31.36	37.85	62.99	31.56	38.70
ACVAE	Traditional generative methods	8.59	3.66	5.07	10.95	4.54	6.12	12.51	5.06	6.66	13.66	5.41	7.05
ContrastVAE		1.25	0.74	0.87	1.96	0.82	1.09	2.59	0.87	1.25	3.06	0.90	1.36
SparseEnNet		13.44	7.62	9.06	18.95	8.35	10.84	22.72	8.65	11.83	25.59	8.81	12.51
DiffuRec	Diffusion-based methods	40.15	27.76	30.85	48.57	28.89	33.58	53.33	29.27	34.84	56.54	29.45	35.60
DiffRec		13.18	10.00	10.79	15.68	10.33	11.60	17.14	10.45	11.98	18.19	10.51	12.23
L-DiffRec		12.71	8.83	9.79	16.43	9.32	10.98	19.09	9.53	11.69	21.06	9.64	12.15
CaDiRec		12.81	7.25	8.63	18.32	7.98	10.40	22.11	8.28	11.41	24.84	8.44	12.05
PreferDiff		30.20	25.71	26.84	32.60	26.04	27.62	33.88	26.14	27.95	34.75	26.19	28.16
Dimos (Ours)	Hybrid method	**45.83***	**30.85***	**34.58***	**55.62***	**32.16***	**37.75***	**60.87***	**32.58***	**39.14***	**64.33***	**32.77***	**39.96***
Improvement		2.79%	3.09%	3.00%	2.24%	2.96%	2.72%	1.86%	2.90%	2.56%	1.43%	2.87%	2.49%

For each column, the best performance and the second best performance methods are denoted in **bold** and underlined fonts respectively.

* Indicates that the improvement over the strongest baseline is statistically significant ($p < 0.05$) based on a paired t-test.

Table 4
Effects of the overall framework.

Dataset	Model	Explore branch	Exploit branch	Unified state space	@5			@10			@15			@20		
					Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG
Yoochoose 1/64	w/o explore branch	✗	✔	✗	44.08	27.39	31.54	55.56	28.94	35.26	60.97	29.36	36.70	64.36	29.55	37.50
	w/o exploit branch	✔	✗	✗	47.46	29.50	33.97	59.53	31.13	37.89	65.36	31.60	39.44	68.68	31.78	40.22
	w/o unified state space	✔	✔	✗	48.08	29.76	34.31	59.90	31.35	38.15	65.62	31.80	39.67	68.95	31.99	40.46
	Dimos (Ours)	✔	✔	✔	**48.37***	**30.07***	**34.62***	**60.34***	**31.69***	**38.51***	**65.92***	**32.13***	**39.99***	**69.34***	**32.32***	**40.80***
Diginetica	w/o explore branch	✗	✔	✗	27.33	15.97	18.79	37.69	17.35	22.13	44.15	17.86	23.84	48.85	18.13	24.95
	w/o exploit branch	✔	✗	✗	31.22	18.52	21.67	42.47	20.02	25.30	49.32	20.56	27.11	54.26	20.84	28.28
	w/o unified state space	✔	✔	✗	29.51	16.73	19.89	41.40	18.31	23.73	48.94	18.90	25.72	54.22	19.20	26.97
	Dimos (Ours)	✔	✔	✔	**31.42***	**18.64***	**21.81***	**42.52***	**20.12***	**25.39***	**49.51***	**20.67***	**27.24***	**54.39***	**20.94***	**28.39***
Retailrocket	w/o explore branch	✗	✔	✗	55.09	41.77	45.12	60.70	42.53	46.94	63.50	42.75	47.68	65.41	42.85	48.13
	w/o exploit branch	✔	✗	✗	56.77	42.86	46.35	63.37	43.75	48.49	66.68	44.01	49.37	68.84	44.14	49.88
	w/o unified state space	✔	✔	✗	55.58	41.36	44.92	62.57	42.30	47.19	66.23	42.59	48.16	68.60	42.72	48.72
	Dimos (Ours)	✔	✔	✔	**57.70***	**43.83***	**47.31***	**63.99***	**44.68***	**49.36***	**67.17***	**44.93***	**50.20***	**69.25***	**45.05***	**50.69***

For each column, the best performance and the second best performance methods are denoted in **bold** and underlined fonts respectively.

* Indicates that the improvement over the strongest baseline is statistically significant ($p < 0.05$) based on a paired t-test.

all models were evaluated over 5 independent runs with different random seeds. The best results of all the models are recorded. Statistical significance ($p < 0.05$) is verified for every ablation study by comparing against the best ablation model on each dataset.

4.3.1. *Effects of the overall framework*

The first group of ablation study investigates the impact of key components in our dual-branch framework. Specifically, we compare three variants: (1) w/o explore branch, which removes the implicit preference learning module; (2) w/o exploit branch, which removes the explicit preference learning module; and (3) w/o unified state space, which uses separate Bi-MaKAN components to independently model the sequential state spaces in the two branches.

The results are presented in Table 4. We observe that the performance of all variants drops when any component of the full Dimos framework is removed. The complete model consistently achieves the best results, demonstrating the effectiveness of its unified architecture. The performance gains can be attributed to the complementary roles of the two branches. The *explore branch* leverages a diffusion-based generative process to capture implicit user preferences by learning the underlying distribution of user behavior. In contrast, the *exploit branch* focuses on modeling explicit preferences by identifying and integrating diverse user intents. Moreover, the dual-branch design is a structured approach to leverage the collaborative strengths of the generative and discriminative paradigms. Specifically, the explore branch provides a robust, distributional prior of user preferences, helping to regularize and generalize the discriminative branch, especially under sparsity. Conversely, the exploit branch provides strong, instance-specific supervisory signals, anchoring the generative process to the observed data and preventing it from diverging into implausible regions. Furthermore, the shared sequential state space plays a crucial role in maintaining feature consistency across both branches, further contributing to performance improvements.

Interestingly, the variant without the exploit branch slightly outperforms the variant without the explore branch. This indicates that the generative component has a relatively stronger influence on performance. Unlike fixed preference representations, the generative approach provides a probabilistic view of user behavior, which enhances robustness and helps mitigate the effects of exposure bias.

Table 5
Effects of the Bi-MaKAN's structure.

Dataset	Model	Shared Parameter	Fusion Method	Bidirectional Mechanism	@5			@10			@15			@20		
					Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG
Yoochoose 1/64	w/o Bidirectional Mechanism	✗	None	✗	4.03	1.98	2.49	6.55	2.32	3.31	8.49	2.47	3.82	10.46	2.58	4.28
	w/o Parameter Sharing	✗	Fourier KAN	✔	47.83	29.72	34.23	59.83	31.34	38.12	65.39	31.78	39.60	68.88	31.97	40.42
	MLP fusion	✔	Linear	✔	47.84	29.53	34.08	59.97	31.16	38.02	65.64	31.61	39.52	68.99	31.80	40.32
	GR-KAN fusion	✔	GR-KAN	✔	48.27	29.85	34.43	**60.35***	31.48	38.36	**66.05***	31.93	39.87	**69.34**	32.12	40.64
	Fourier KAN fusion	✔	Fourier KAN	✔	**48.37***	**30.07***	**34.62***	60.34	**31.69***	**38.51***	65.92	**32.13***	**39.99***	**69.34**	**32.32***	**40.80***
Diginetica	w/o Bidirectional Mechanism	✗	None	✗	29.75	17.56	20.58	40.74	19.02	24.13	47.56	19.56	25.93	52.45	19.84	27.09
	w/o Parameter Sharing	✗	Fourier KAN	✔	31.19	18.51	21.65	42.19	19.98	25.21	49.09	20.52	27.04	53.99	20.80	28.19
	MLP fusion	✔	Linear	✔	31.20	18.62	21.74	42.36	20.11	25.34	49.29	20.65	27.18	54.33	20.94	28.37
	GR-KAN fusion	✔	GR-KAN	✔	**31.59***	**18.74***	**21.93***	**42.79***	**20.24***	**25.55***	**49.72***	**20.78***	**27.38***	**54.64***	**21.06***	**28.54***
	Fourier KAN fusion	✔	Fourier KAN	✔	31.42	18.64	21.81	42.52	20.12	25.39	49.51	20.67	27.24	54.39	20.94	28.39
Retailrocket	w/o Bidirectional Mechanism	✗	None	✗	56.76	43.29	46.68	63.02	44.14	48.71	66.10	44.39	49.53	68.22	44.51	50.03
	w/o Parameter Sharing	✗	Fourier KAN	✔	55.66	41.27	44.88	62.71	42.22	47.17	66.32	42.51	48.12	68.76	42.64	48.70
	MLP fusion	✔	Linear	✔	57.63	43.55	47.08	63.85	44.39	49.10	67.01	44.64	49.94	69.05	44.75	50.42
	GR-KAN fusion	✔	GR-KAN	✔	**57.95***	43.68	47.26	**64.27***	44.53	49.32	**67.40***	44.78	50.14	**69.48***	44.90	50.64
	Fourier KAN fusion	✔	Fourier KAN	✔	57.70	**43.83***	**47.31***	63.99	**44.68***	**49.36***	67.17	**44.93***	**50.20***	69.25	**45.05***	**50.69***

For each column, the best performance and the second best performance methods are denoted in **bold** and underlined fonts respectively.

* Indicates that the improvement over the strongest baseline is statistically significant ($p < 0.05$) based on a paired t-test.

4.3.2. Effects of the Bi-MaKAN's structure

The second group of ablation study aims to demonstrate the effectiveness of the Bi-MaKAN's structure. Specifically, we produce five variants for comparison, including: (1) w/o Bidirectional Mechanism, which adopts vanilla Mamba block for feature learning; (2) w/o Parameter Sharing, which removes the parameter sharing between the forward Mamba block and the reverse Mamba block; (3) MLP fusion, which adopts the naive linear projection to fuse the bidirectional contextual features; (4) GR-KAN fusion, which adopts the GR-KAN to fuse the bidirectional contextual features; and (5) Fourier KAN fusion, which adopts the Fourier KAN to fuse the bidirectional contextual features.

As shown in Table 5, removing parameter sharing and bidirectional mechanism leads to sub-optimal performance, demonstrating the effectiveness of both approaches. Specifically, bidirectional mechanism expands the receptive field to facilitate sequence modeling capacity, especially for sparser datasets. Additionally, parameter sharing reduces model complexity and over-fitting risk. Notably, w/o Bidirectional Mechanism exhibited a pronounced performance collapse on the Yoochoose 1/64 dataset, demonstrating the instability of a vanilla Mamba backbone in Dimos with limited data. This insight motivated the investigation into the contributions of the denoising network and the forward feature encoder via subsequent ablation studies.

In terms of three feature fusion methods, MLP-based approach fails to achieve promising performance, suggesting the superiority of KAN-based fusion methods. Specifically, KAN adopts the same fully connected architecture as MLPs but differ by placing learnable activation functions on the edges rather than applying fixed activation functions at the nodes, as in standard MLPs (Liu et al., 2025c). Therefore, KAN exhibits less bias toward low-frequency components compared to MLPs, which are prone to spectral bias and tend to fit low-frequency features first (Wang et al., 2025). However, the recursive computations in vanilla KAN significantly slows down performance. Moreover, vanilla KAN requires unique parameters and base functions for each input–output pair. It leads to exponential growth in parameters and computation overhead as the network scales (Yang and Wang, 2024).

By adopting 1D Fourier coefficients instead of B-spline coefficients, Fourier KAN offers easier optimization due to the denser nature of Fourier coefficients, which operate on a global scale, in contrast to the local nature of splines. Moreover, the introduced Fourier coefficients benefits from periodicity, making the functions more numerically bounded and helping to avoid issues related to going out of the grid. In terms of GR-KAN, it adopts rational functions instead of B-spline functions to enhance efficiency. Additionally, by sharing function coefficients and base functions across groups of edges, it significantly reduces computational complexity. Furthermore, its carefully designed weight initialization strategy maintains consistent activation variance across layers, improving training stability and performance.

4.3.3. Effects of the bi-makan as the denoising network

The third group of ablation study is designed to demonstrate the effectiveness of our Bi-MaKAN as the denoising network. Specifically, we fix the forward feature encoder as Bi-MaKAN with the Fourier KAN fusion method and compare the performance of various methods as the denoising network, including Transformer, GRU, Fourier KAN, MLP, SU-Net (Liu et al., 2023), Mamba, and our Bi-MaKAN. Notably, these variants are employed in traditional LDM framework (Rombach et al., 2022), equivalently removing the implicit preference learning module. The results are shown in Table 6.

We have the following observations. First, with different neural networks serving as the denoising network, all variants achieve satisfactory performance, demonstrating the generalizability of our Dimos framework with respect to the choice of the denoising network. Second, the variant with the Bi-MaKAN-based denoising network achieves the best performance on the Retailrocket dataset, while the variant with the vanilla Mamba-based denoising network outperforms other methods on the Yoochoose 1/64 and Diginetica datasets. This highlights the effectiveness of Mamba-based methods within the Dimos framework. Third, these variants fail to consistently achieve the best performance on all the three datasets. One reason is that these generative methods focus on modeling implicit preferences, while neglecting to capture the explicit preferences, leading to biased preference learning.

4.3.4. Effects of the Bi-MaKAN as the forward feature encoder

The fourth group of ablation study is designed to demonstrate the effectiveness of our Bi-MaKAN as the forward feature encoder. Similarly, we fix the denoising network as Bi-MaKAN with the Fourier KAN fusion method and compare the performance of various neural networks used as the forward feature encoder. Notably, following LDM (Rombach et al., 2022), we further conduct experiments without using any forward feature encoder as the baseline, denoted as "None".

From the results as shown in Table 7, we observe that the variant without forward feature encoder achieves competitive performance on the Yoochoose 1/64 and Diginetica datasets, demonstrating the effectiveness of our Bi-MaKAN as the denoising network. Furthermore, the variant with Bi-MaKAN-based forward feature network outperforms other variants on the Retailrocket dataset, suggesting its potential to handle the large scaled datasets. Lastly, our Dimos framework outperforms all the variants, indicating that simultaneously learning both explicit and implicit preferences facilitates more accurate recommendations.

4.3.5. Evaluation of alternative networks as the backbone

In the fifth group of ablation study, we aim to further investigate the adaptability between the proposed Bi-MaKAN module and the overall Dimos framework. Specifically, we simultaneously alter the forward feature encoder and the denoising network to other neural networks as mentioned in Section 4.3.3. The results are shown in Table 8.

Table 6
Effects of the Bi-MaKAN as the denoising network. For the experimental results on each dataset, the **bold**-faced number is the best score and the underlined number is the second best score.

Dataset	Denoising Network	@5			@10			@15			@20		
		Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG
Yoochoose 1/64	Transformer	47.69	29.46	34.00	59.68	31.08	37.89	65.48	31.54	39.43	68.77	31.72	40.21
	GRU	47.46	29.46	33.94	59.53	31.09	37.85	65.03	31.52	39.31	68.38	31.71	40.11
	Fourier KAN	46.34	28.79	33.15	58.16	30.38	36.99	63.67	30.82	38.45	67.11	31.01	39.27
	MLP	45.87	28.13	32.54	57.86	29.74	36.43	63.65	30.20	37.96	67.06	30.39	38.77
	SU-Net	48.01	29.61	34.18	60.06	31.24	38.10	65.68	31.68	39.59	69.15	31.88	40.41
	Mamba	47.76	29.78	34.25	59.55	31.37	38.08	65.09	31.81	39.55	68.51	32.00	40.36
	Bi-MaKAN	47.46	29.50	33.97	59.53	31.13	37.89	65.36	31.60	39.44	68.68	31.78	40.22
	Dimos (Ours)	**48.37***	**30.07***	**34.62***	**60.34***	**31.69***	**38.51***	**65.92***	**32.13***	**39.99***	**69.34***	**32.32***	**40.80***
Diginetica	Transformer	29.54	17.46	20.45	40.66	18.94	24.04	47.73	19.50	25.91	52.70	19.78	27.09
	GRU	30.29	17.96	21.01	41.27	19.42	24.56	47.91	19.94	26.32	52.83	20.22	27.48
	Fourier KAN	26.67	15.25	18.08	26.67	16.72	21.64	44.68	17.27	23.49	49.74	17.56	24.68
	MLP	27.02	15.56	18.39	38.06	17.03	21.96	45.18	17.59	23.84	50.40	17.88	25.07
	SU-Net	29.87	16.90	20.11	41.54	18.45	23.88	48.98	19.04	25.84	54.14	19.33	27.07
	Mamba	30.75	18.26	21.36	41.86	19.74	24.94	48.69	20.28	26.75	53.58	20.55	27.91
	Bi-MaKAN	30.17	17.77	20.84	41.44	19.27	24.48	48.42	19.82	26.33	53.42	20.10	27.51
	Dimos (Ours)	**31.42***	**18.64***	**21.81***	**42.52***	**20.12***	**25.39***	**49.51***	**20.67***	**27.24***	**54.39***	**20.94***	**28.39***
Retailrocket	Transformer	56.72	43.10	46.52	63.32	43.99	48.66	66.70	44.26	49.55	68.93	44.38	50.08
	GRU	56.71	43.25	46.63	62.76	44.07	48.60	65.67	44.30	49.37	67.56	44.41	49.82
	Fourier KAN	55.84	42.35	45.74	62.43	43.24	47.88	65.78	43.51	48.76	68.01	43.63	49.29
	MLP	52.16	39.09	42.36	59.19	40.04	44.64	62.89	40.33	45.62	65.36	40.47	46.21
	SU-Net	55.62	41.16	44.78	62.63	42.10	47.06	66.16	42.38	48.00	68.59	42.52	48.57
	Mamba	57.33	43.21	46.75	63.63	44.06	48.80	66.70	44.30	49.62	68.78	44.42	50.11
	Bi-MaKAN	57.52	43.47	47.00	63.80	44.32	49.04	67.02	44.57	49.89	69.15	44.69	50.39
	Dimos (Ours)	**57.70***	**43.83***	**47.31***	**63.99***	**44.68***	**49.36***	**67.17***	**44.93***	**50.20***	**69.25***	**45.05***	**50.69***

For each column, the best performance and the second best performance methods are denoted in **bold** and underlined fonts respectively.
* Indicates that the improvement over the strongest baseline is statistically significant ($p < 0.05$) based on a paired t-test.

We have the following observations. First, the variants with Fourier KAN-based and MLP-based backbones are not as competitive. It is attributed that these non-sequential models struggle to capture dynamic behavioral patterns, hindering representative underlying distribution learning. Surprisingly, the variant with vanilla Mamba-based backbone underperforms other variants due to severe over-fitting, revealing its potential limitation when applied to small-scale datasets. Moreover, the variant with widely adopted Transformer-based backbone consistently fail to achieve promising performance, while the variant with SU-Net-based backbone gains the best performance on the Yoochoose 1/64 and Diginetica datasets. However, both of them fail to achieve the improved performance in larger dataset, Yelp. These observations motivate the design of Bi-MaKAN, which demonstrates competitive performance across all three datasets. Furthermore, our Dimos consistently outperforms all the variants, highlighting the effectiveness of integrating explicit and implicit preference learning for session-based recommendation.

4.3.6. Effects of the preference fusion method

In the sixth group of ablation study, we aim to investigate the effectiveness of the simple linear preference fusion method. Specifically, we compared our Dimos with two variants adopted MLP and Hadamard product as preference fusion method, respectively. The results are shown in Table 9.

We have the following observations. First, the MLP-based fusion performs remarkably poorly across all datasets. This severe performance degradation suggests that a deep, non-linear transformation of the preference vectors may inadvertently destroy or obfuscate the distinct, complementary information encoded in each branch, likely leading to optimization difficulties and loss of critical signals. Second, the Hadamard product serves as a much stronger baseline, achieving the second-best performance. However, its consistent suboptimal performance compared to our linear method suggests that forcing a purely multiplicative fusion couples the two signals or amplify noise, rather than optimally balancing their contributions. Lastly, our simple linear fusion strategy consistently and significantly outperforms both non-linear alternatives on every dataset and across all metrics. Its success demonstrates that preserving the original structure of the learned preference representations and allowing for an additive combination is more effective than enforcing complex and non-linear interactions for our model.

4.4. Effects of explicit and implicit preference learning modules in training and inference

To address RQ3, we vary the loss weight ζ and the preference weight ρ from 0 to 1 in increments of 0.1, respectively. Notably, when setting $\zeta = 0$ and $\rho = 0$, only the implicit preference learning module works during training and inference. Moreover, when setting $\zeta = 1$ and $\rho = 1$, only the implicit preference learning module works during training and inference.

From the results as shown in Fig. 3, we observe that setting ζ as 0 and 1 fails to achieve promising performance, suggesting the complementary strengths of both preference learning modules during training. Specifically, the two modules focus on capturing explicit and implicit user preferences by adopting attention network and DDPM, respectively. This dual-view preference learning facilitates more robust preference learning. The sharp performance degradation at extreme ζ values stems from insufficient gradient signals from one module, leading to suboptimal feature adaptation in the hidden space. Moreover, the loss weight ζ effectively balances the contribution of each module's gradients during training. When ζ varies within the range of 0.5 to 0.7, it achieves a favorable trade-off that prevents either module from overwhelming the other. We further find that the optimal loss weight is

Table 7
Effects of the Bi-MaKAN as the forward feature encoder. For the experimental results on each dataset, the **bold**-faced number is the best score and the underlined number is the second best score.

Dataset	Forward Encoder	@5			@10			@15			@20		
		Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG
Yoochoose 1/64	None	48.11	29.85	34.39	60.03	29.85	38.26	65.60	31.90	39.74	69.09	32.10	40.57
	Transformer	47.25	29.13	33.64	59.34	30.77	37.56	65.25	31.23	39.13	68.75	31.43	39.96
	GRU	47.26	29.32	33.78	59.10	30.92	37.63	64.79	31.37	39.14	68.07	31.56	39.92
	Fourier KAN	46.95	28.95	33.43	58.82	30.55	37.28	64.54	31.00	38.80	68.06	31.20	39.63
	MLP	47.21	28.90	33.45	59.43	30.54	37.42	65.22	31.00	38.95	68.59	31.19	39.75
	SU-Net	47.88	29.50	34.07	59.87	31.12	37.96	65.61	31.57	39.49	69.04	31.77	40.30
	Mamba	47.75	29.69	34.18	59.71	31.31	38.07	65.36	31.75	39.57	68.73	31.94	40.36
	Bi-MaKAN	47.46	29.50	33.97	59.53	31.13	37.89	65.36	31.60	39.44	68.68	31.78	40.22
	Dimos (Ours)	**48.37***	**30.07***	**34.62***	**60.34***	**31.69***	**38.51***	**65.92***	**32.13***	**39.99***	**69.34***	**32.32***	**40.80***
Diginetica	None	31.22	18.52	21.67	42.47	20.02	25.30	49.32	20.56	27.11	54.26	20.84	28.28
	Transformer	29.80	17.67	20.67	40.84	19.14	24.24	47.82	19.69	26.09	52.84	19.97	27.27
	GRU	29.65	17.51	20.52	40.76	18.99	24.10	47.52	19.52	25.89	52.35	19.79	27.04
	Fourier KAN	25.06	14.27	16.94	35.77	15.69	20.39	42.64	16.23	22.21	47.63	16.51	23.39
	MLP	27.78	16.06	18.96	38.98	17.54	22.57	46.17	18.11	24.47	51.30	18.40	25.69
	SU-Net	29.56	16.81	19.97	41.04	18.34	23.67	48.29	18.90	25.59	53.43	19.19	26.80
	Mamba	31.13	18.49	21.62	42.00	19.94	25.13	48.72	20.47	26.91	53.55	20.74	28.05
	Bi-MaKAN	30.17	17.77	20.84	41.44	19.27	24.48	48.42	19.82	26.33	53.42	20.10	27.51
	Dimos (Ours)	**31.42***	**18.64***	**21.81***	**42.52***	**20.12***	**25.39***	**49.51***	**20.67***	**27.24***	**54.39***	**20.94***	**28.39***
Retailrocket	None	56.77	42.86	46.35	63.37	43.75	48.49	66.68	44.01	49.37	68.84	44.14	49.88
	Transformer	56.56	42.90	46.32	63.03	43.77	48.43	66.30	44.03	49.29	68.44	44.15	49.80
	GRU	55.96	42.71	46.04	62.21	43.56	48.07	65.26	43.80	48.88	67.39	43.92	49.38
	Fourier KAN	56.07	42.73	46.08	62.65	43.62	48.21	65.96	43.88	49.09	68.21	44.01	49.62
	MLP	55.20	41.83	45.18	61.97	42.74	47.38	65.46	43.02	48.30	67.82	43.15	48.86
	SU-Net	57.00	42.99	46.51	63.13	43.82	48.50	66.25	44.07	49.33	68.22	44.18	49.80
	Mamba	57.21	43.48	46.93	63.39	44.31	48.94	66.46	44.56	49.75	68.56	44.68	50.25
	Bi-MaKAN	57.52	43.47	47.00	63.80	44.32	49.04	67.02	44.57	49.89	69.15	44.69	50.39
	Dimos (Ours)	**57.70***	**43.83***	**47.31***	**63.99***	**44.68***	**49.36***	**67.17***	**44.93***	**50.20***	**69.25***	**45.05***	**50.69***

For each column, the best performance and the second best performance methods are denoted in **bold** and underlined fonts respectively.
* Indicates that the improvement over the strongest baseline is statistically significant ($p < 0.05$) based on a paired t-test.

Table 8
Evaluation of alternative networks as the backbone. For the experimental results on each dataset, the **bold**-faced number is the best score and the underlined number is the second best score.

Dataset	Backbone	@5			@10			@15			@20		
		Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG
Yoochoose 1/64	Transformer	46.76	28.84	33.29	58.68	30.44	37.16	64.27	30.88	38.65	67.82	31.08	39.48
	GRU	47.23	29.26	33.73	59.18	30.87	37.61	64.86	31.32	39.11	68.22	31.51	39.91
	Fourier KAN	4.34	2.23	2.75	7.08	2.60	3.64	8.74	2.72	4.07	10.85	2.84	4.57
	MLP	4.28	2.20	2.71	7.08	2.57	3.62	8.88	2.71	4.09	10.71	2.81	4.52
	SU-Net	48.13	29.48	34.12	60.11	31.09	38.01	65.70	31.54	39.49	69.05	31.73	40.28
	Mamba	4.03	1.98	2.49	6.55	2.32	3.31	8.49	2.47	3.82	10.46	2.58	4.28
	Bi-MaKAN	47.46	29.50	33.97	59.53	31.13	37.89	65.36	31.60	39.44	68.68	31.78	40.22
	Dimos (Ours)	**48.37***	**30.07***	**34.62***	**60.34***	**31.69***	**38.51***	**65.92***	**32.13***	**39.99***	**69.34***	**32.32***	**40.80***
Diginetica	Transformer	28.34	16.66	19.55	39.08	18.09	23.02	46.03	18.64	24.86	50.98	18.91	26.03
	GRU	30.31	17.94	21.00	41.19	19.39	24.52	47.93	19.92	26.30	52.68	20.18	27.42
	Fourier KAN	0.33	0.16	0.20	0.64	0.20	0.30	0.83	0.22	0.35	1.00	0.23	0.39
	MLP	0.35	0.16	0.20	0.60	0.19	0.28	0.85	0.21	0.35	1.05	0.22	0.40
	SU-Net	31.39	18.50	21.70	42.10	19.93	25.16	48.76	20.46	26.92	53.53	20.72	28.05
	Mamba	29.75	17.56	20.58	40.74	19.02	24.13	47.56	19.56	25.93	52.45	19.84	27.09
	Bi-MaKAN	30.17	17.77	20.84	41.44	19.27	24.48	48.42	19.82	26.33	53.42	20.10	27.51
	Dimos (Ours)	**31.42***	**18.64***	**21.81***	**42.52***	**20.12***	**25.39***	**49.51***	**20.67***	**27.24***	**54.39***	**20.94***	**28.39***
Retailrocket	Transformer	54.97	41.73	45.05	61.58	42.62	47.19	64.96	42.89	48.09	67.20	43.01	48.62
	GRU	56.52	42.94	46.35	62.75	43.78	48.37	65.82	44.02	49.19	67.89	44.14	49.68
	Fourier KAN	1.47	1.11	1.20	1.86	1.16	1.33	2.16	1.19	1.41	2.41	1.20	1.47
	MLP	1.44	1.07	1.16	1.89	1.13	1.31	2.16	1.15	1.38	2.41	1.16	1.44
	SU-Net	56.41	42.59	46.06	62.47	43.41	48.04	65.59	43.66	48.86	67.65	43.78	49.35
	Mamba	56.76	43.29	46.68	63.02	44.14	48.71	66.10	44.39	49.53	68.22	44.51	50.03
	Bi-MaKAN	57.52	43.47	47.00	63.80	44.32	49.04	67.02	44.57	49.89	69.15	44.69	50.39
	Dimos (Ours)	**57.70***	**43.83***	**47.31***	**63.99***	**44.68***	**49.36***	**67.17***	**44.93***	**50.20***	**69.25***	**45.05***	**50.69***

For each column, the best performance and the second best performance methods are denoted in **bold** and underlined fonts respectively.
* Indicates that the improvement over the strongest baseline is statistically significant ($p < 0.05$) based on a paired t-test.

Table 9
Effects of the simple linear preference fusion method. For the experimental results on each dataset, the **bold**-faced number is the best score and the underlined number is the second best score.

Dataset	Model	@5			@10			@15			@20		
		Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG
Yoochoose 1/64	MLP-fusion	1.25	0.72	0.85	2.51	0.88	1.25	3.77	0.98	1.58	4.95	1.04	1.86
	Hadamard-fusion	46.58	28.61	33.08	58.23	30.19	36.87	63.90	30.64	38.37	67.48	30.84	39.22
	Dimos (Ours)	**48.37***	**30.07***	**34.62***	**60.34***	**31.69***	**38.51***	**65.92***	**32.13***	**39.99***	**69.34***	**32.32***	**40.80***
Diginetica	MLP-fusion	0.03	0.01	0.02	0.10	0.02	0.04	0.21	0.03	0.07	0.29	0.03	0.09
	Hadamard-fusion	29.16	17.28	20.22	39.63	18.67	23.60	46.25	19.19	25.35	51.00	19.46	26.48
	Dimos (Ours)	**31.42***	**18.64***	**21.81***	**42.52***	**20.12***	**25.39***	**49.51***	**20.67***	**27.24***	**54.39***	**20.94***	**28.39***
Retailrocket	MLP-fusion	0.01	0.01	0.01	0.08	0.01	0.03	0.12	0.02	0.04	0.18	0.02	0.05
	Hadamard-fusion	56.40	42.69	46.13	62.42	43.50	48.09	65.43	43.74	48.89	67.44	43.85	49.36
	Dimos (Ours)	**57.70***	**43.83***	**47.31***	**63.99***	**44.68***	**49.36***	**67.17***	**44.93***	**50.20***	**69.25***	**45.05***	**50.69***

For each column, the best performance and the second best performance methods are denoted in **bold** and underlined fonts respectively.
* Indicates that the improvement over the strongest baseline is statistically significant ($p < 0.05$) based on a paired t-test.

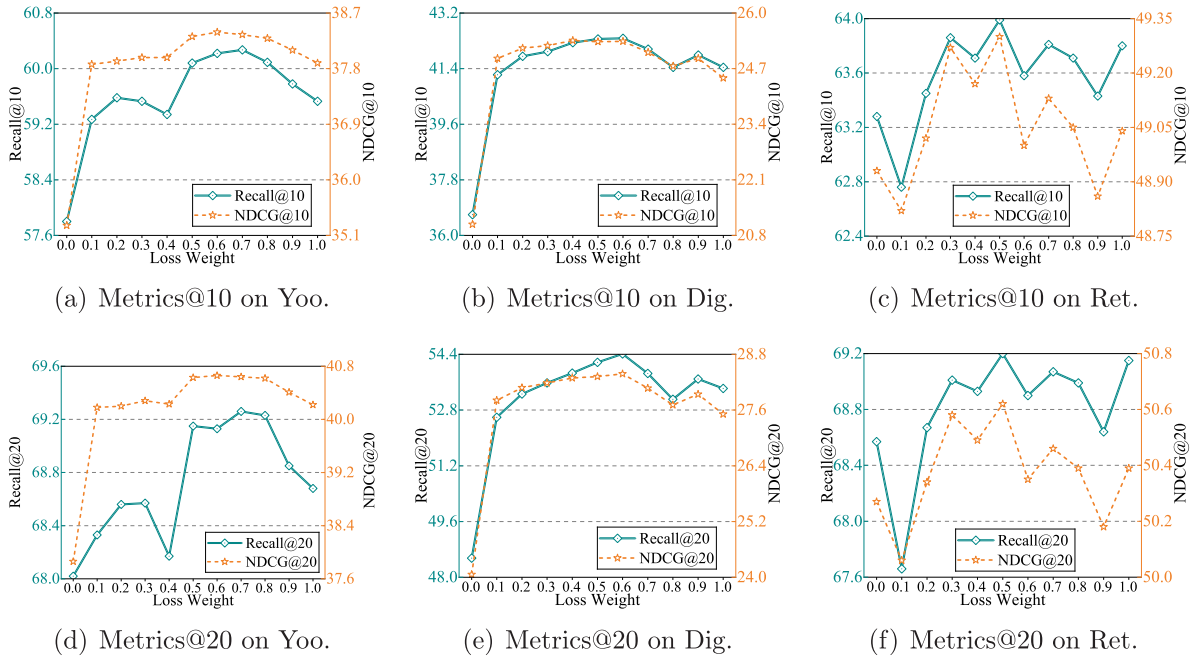


Fig. 3. Performance of Dimos with varying loss weight on Yoochoose 1/64 (Yoo.), Diginetica (Dig.) and Retailrocket (Ret.).

0.7 on Yoochoose 1/64, 0.6 on Diginetica, and 0.5 on Retailrocket. In terms of the preference weight, as shown in Fig. 4, we observe that our Dimos fails to achieve competitive performance with $\rho = 0$, it achieves the promising performance with $\rho = 1$ across all the three datasets. Specifically, the optimal preference weight is 1 on Yoochoose 1/64, 0.9 on Diginetica, and 0.8 on Retailrocket.

Analysis of the loss weight and the preference weight reveals the collaborative effect between exploit branch and explore branch. Specifically, we assume that the explore branch expands coverage of implicit preferences, while the exploit branch captures explicit preferences to sharpen decision boundaries for critical recommendations (Choi et al., 2023; Lobashev et al., 2025). Moreover, the dataset scale modulates the collaboration, aligning with the insights on discriminative and generative learning paradigms (Ng and Jordan, 2001; Zheng et al., 2023). The exploit branch, as a discriminative model, directly learns the decision boundary for next-item prediction from the data. Its performance is highly dependent on data volume: more data provides a richer and more diverse set of user–item interactions, allowing the model to overcome sparsity, refine its attention mechanisms, and capture more robust explicit patterns. In contrast, the generative explore branch excels at capturing the underlying data distribution, making it particularly valuable when observable signals are sparse.

4.5. Hyperparameter sensitivity

To address RQ4, we investigate the impact of the key hyperparameters on performance, including the Bi-MaKAN stacking layers and the noise strength. Specifically, the Bi-MaKAN stacking layers and the noise strength are adjusted within the range of {1, 2, 3, 4} and {1e-2, 1e-3, 1e-4, 1e-5, 1e-6}. Furthermore, we investigate the impact of different noise schedules, including the sqrt noise scheduler, the cosine noise scheduler, the truncated cosine noise scheduler, and the truncated linear noise scheduler.

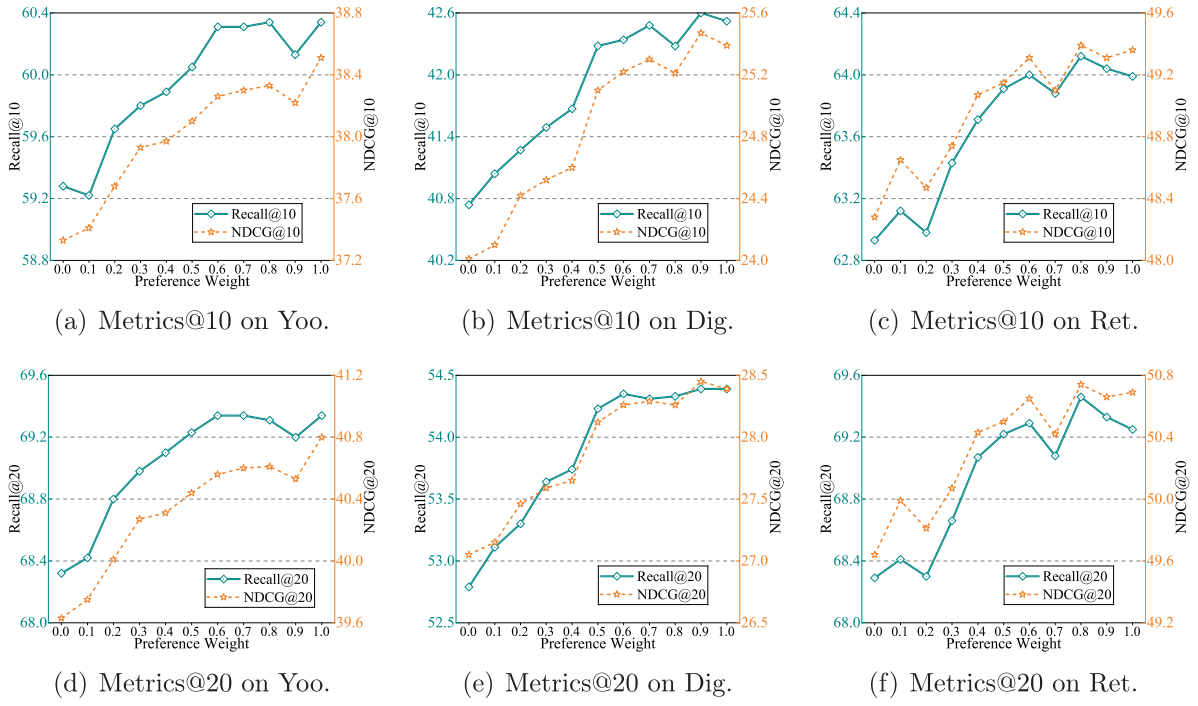


Fig. 4. Performance of Dimos with varying preference weight on Yoochoose 1/64 (Yoo.), Diginetica (Dig.) and Retailrocket (Ret.).

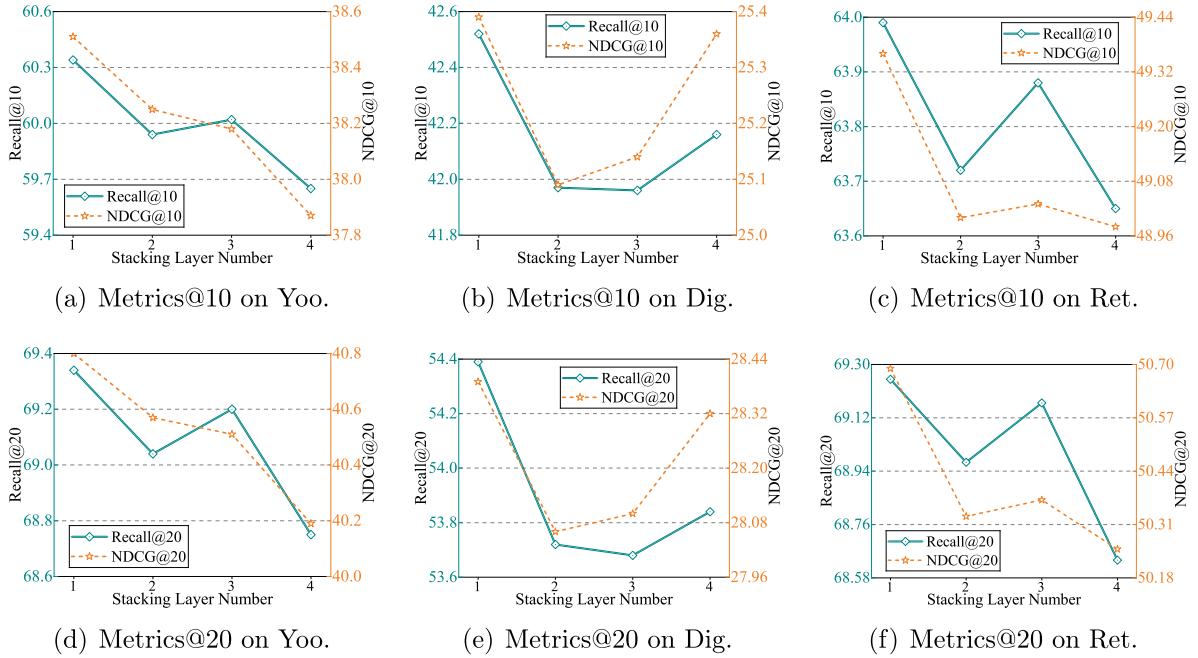


Fig. 5. Performance of Dimos with varying Bi-MaKAN layer number on Yoochoose 1/64 (Yoo.), Diginetica (Dig.) and Retailrocket (Ret.).

As shown in Fig. 5, we observe that just one layer can achieve the best performance on the three datasets, demonstrating the strong representation learning capacity of our Bi-MaKAN. Specifically, the bidirectional Mamba blocks facilitate more comprehensive context modeling, while the parameter sharing ensures consistent feature learning. Additionally, the KAN-based feature fusion method improves the learning process for high-frequency features compared to traditional MLP. Furthermore, stacking more layers can lead to a decrease in performance due to overfitting, while the introduction of excessive parameters results in lower training efficiency.

Regarding the noise strength parameter δ, Fig. 6 demonstrates that our proposed Dimos achieves strong performance with a small δ, i.e., 1e-5. However, performance degrades significantly when δ increases to 1e-1. Recall that δ controls the mean and variance of the sampling distribution for λ, which regulates the discriminative power of item representations. As δ grows, λ tends to take larger values.

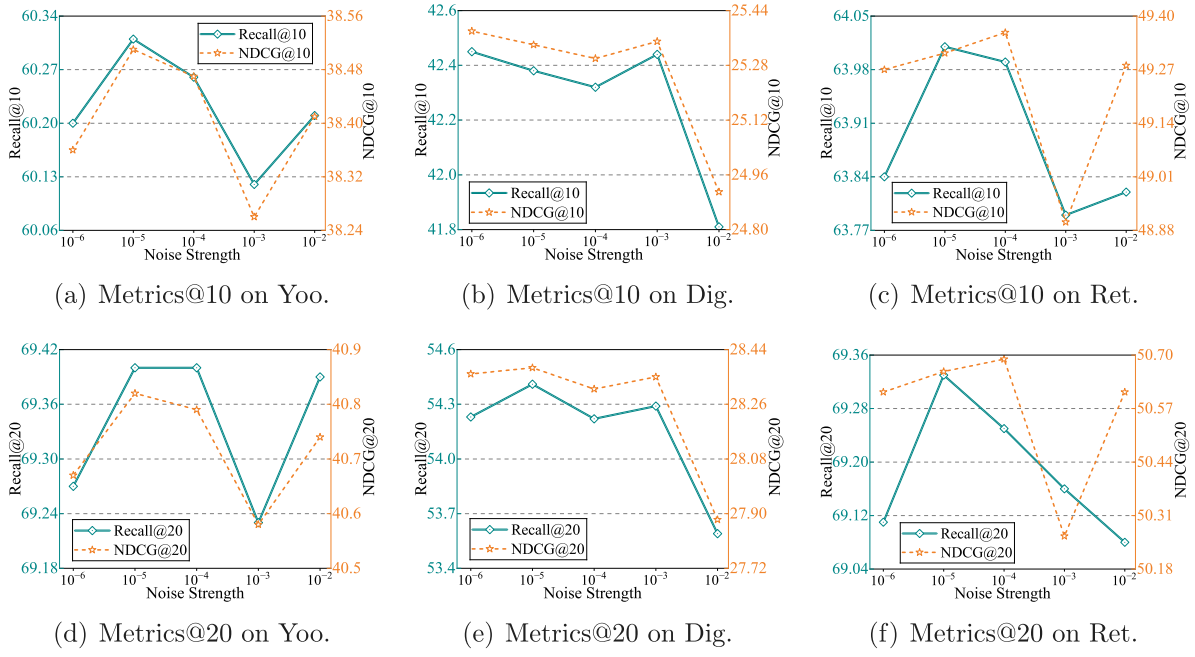


Fig. 6. Performance of Dimos with varying noise strength on Yoochoose 1/64 (Yoo.), Diginetica (Dig.) and Retailrocket (Ret.).

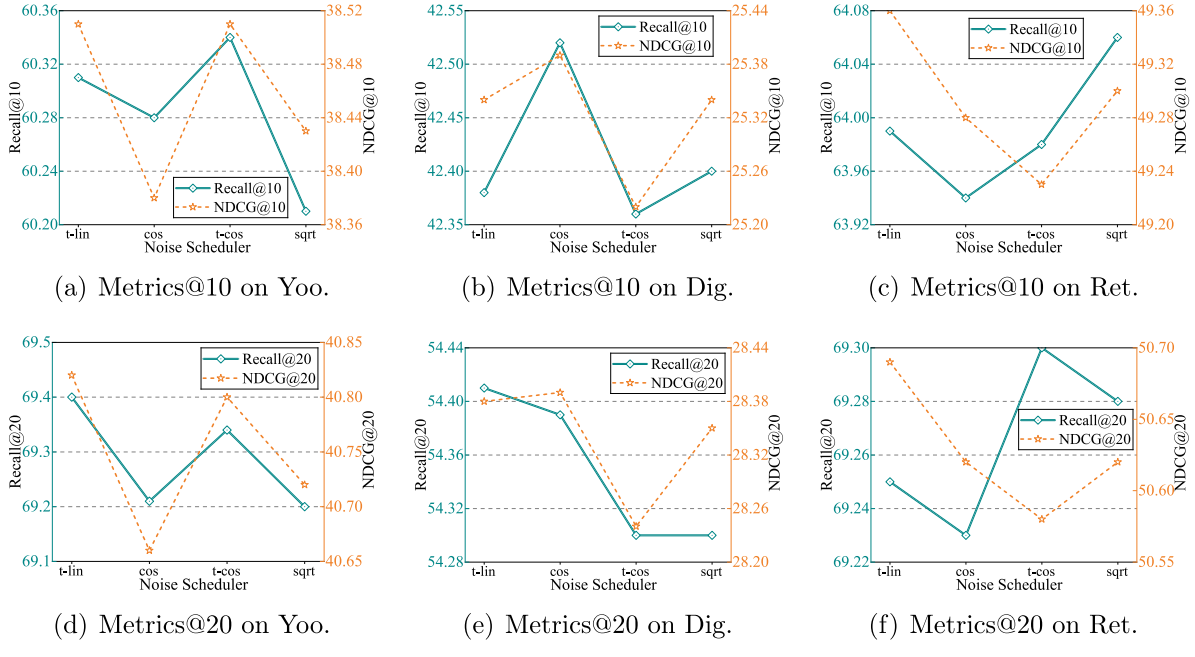


Fig. 7. Performance of Dimos with varying noise scheduler on Yoochoose 1/64 (Yoo.), Diginetica (Dig.) and Retailrocket (Ret.).

Unfortunately, an excessively large λ introduces substantial noise into the historical interaction sequences, corrupting the original interactions and impairing the capacity to precisely infer user preferences.

Regarding the noise scheduler, as shown in Fig. 7, the performance differences between various noise schedulers are slight, which aligns with recent empirical findings (Niu et al., 2025; Du et al., 2023). Specifically, the truncated linear noise scheduler performs well on the Yoochoose 1/64 and RetailRocket datasets, while the cosine linear noise scheduler outperforms other noise schedulers on the Diginetica dataset.

4.6. User group study

The user group study addresses RQ5, demonstrating that Dimos can deliver effective recommendations for users with varying levels of activity. Specifically, we categorize the users in the Diginetica dataset into three groups based on their activity levels: cold-start users (session length of 5 or fewer), common users (session length of 5 to 15), and active users (session length greater than 15). The dataset statistics are presented in Table 10, which indicate that 71.03% of the users are classified as cold-start users, 26.94% as common users, and 2.03% as

Table 10
Statistics of three user groups derived from Diginetica.

Dataset	# Sessions	# Items	# Interactions	Avg.session length	Avg. action per item	Sparsity
Cold-start group	144 939	40 782	444 678	3.07	10.90	99.99%
Common group	54 984	41 011	462 360	8.41	11.27	99.98%
Active group	4141	23 071	82 166	19.85	3.56	99.91%

Table 11
Model performance on the three user groups.

Model	Cold-start group						Common group						Active group					
	@10			@20			@10			@20			@10			@20		
	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG
CMGNN	38.10	18.93	23.47	47.17	19.56	25.76	35.62	15.61	20.30	47.31	16.41	23.25	28.55	13.12	16.76	36.50	13.67	18.77
GSAU	34.08	15.27	19.68	45.09	16.03	22.46	35.03	15.48	20.05	47.37	16.33	23.17	24.52	9.96	13.35	34.18	10.62	15.78
SS4Rec	35.32	17.24	21.50	45.10	17.92	23.97	34.80	15.89	20.32	46.09	16.67	23.17	25.22	11.91	15.04	32.10	12.38	16.77
DiffuRec	32.61	16.68	20.44	40.46	17.23	22.43	34.29	15.65	20.02	45.14	16.40	22.76	2.44	2.33	2.36	2.56	2.34	2.39
DiffRec	6.14	4.34	4.77	6.71	4.38	4.92	14.17	6.53	8.32	18.87	6.85	9.50	21.64	10.55	13.15	29.30	11.07	15.08
PreferDiff	4.48	2.78	3.18	5.32	2.84	3.39	0.46	0.14	0.21	0.80	0.16	0.30	0.46	0.15	0.22	0.77	0.17	0.29
Bi-MaKAN as denoiser	0.74	0.26	0.37	1.33	0.30	0.52	0.50	0.16	0.24	0.90	0.19	0.34	25.92	12.66	15.78	32.61	13.12	17.46
Dual Bi-MaKAN	36.65	18.54	22.82	45.14	19.13	24.97	36.64	16.54	21.25	48.28	17.35	24.19	24.18	11.57	14.55	31.30	12.07	16.36
Dimos (Ours)	**39.30**	**20.20**	**24.49**	**48.20**	**20.75**	**26.49**	**37.85**	**17.32**	**22.14**	**49.16**	**18.11**	**25.00**	**29.46**	**14.12**	**17.50**	**36.98**	**14.63**	**19.38**

active users. This distribution reveals that most real-world sessions are short, making it challenging to capture user behavioral patterns due to limited contextual information. Subsequently, we explore the performance of our Dimos and six representative baseline models, including three explicit preference modeling methods, i.e., CM-GNN (Wang et al., 2023a), GSAU (Cao et al., 2025), and SS4Rec (Xiao et al., 2025), and three implicit preference modeling methods, i.e., DiffuRec (Li et al., 2024c), DiffRec (Wang et al., 2023b), and PreferDiff (Liu et al., 2025a). To further valid the effectiveness of our framework, we evaluate two variants: (1) Bi-MaKAN as denoiser, and (2) Dual Bi-MaKAN. Specifically, the Bi-MaKAN as denoiser only replaces the denoising network of DiffuRec with Bi-MaKAN, while the Dual Bi-MaKAN excluding the implicit preference learning module from Dimos.

Our experimental results across three user groups reveal several findings regarding recommendation performance under different activity levels. As shown in Table 11, most baseline models achieve the best performance on the cold-start group, while performs poorly on the active group. One reason is that the long session length in the active group makes it particularly challenging to capture genuine user preferences.

Among baseline models, three explicit preference modeling methods consistently outperform three implicit preference modeling methods across the three user groups. Moreover, two exceptional cases warrant special discussion. First, DiffRec shows an inverse performance trend, where it performs well on the active group and fails to achieve promising performance on the cold-start group. Second, DiffuRec demonstrates reasonable performance on cold-start and common groups, while collapsing dramatically in the active group. These observations suggest the potential limitations of the MLP-based and Transformer-based denoising network. Furthermore, the failure of these generative methods highlights the effectiveness of the forward feature encoder, which is overlooked by most existing diffusion-based SBRSs, in capturing accurate user preferences. Notably, Dual Bi-MaKAN demonstrates its superiority over the three generative baseline models, as its Bi-MaKAN-based forward feature encoder effectively facilitates the learning of the representative underlying distribution.

Although Dual Bi-MaKAN outperforms most baseline model in most cases, it fails to achieve the best performance on the cold-start and active groups. Our proposed Dimos demonstrates superior performance across all user groups, achieving the best results in every metric. The consistent superiority of Dimos highlights the effectiveness of combining explicit preference modeling with implicit preference modeling.

4.7. Impact of session length

To address RQ6, we conduct two sets of experiments aimed at evaluating how session length influences the performance of Dimos. We first evaluate Dimos and six representative baseline models on the KuaiRand-Pure dataset, which features a significantly longer average session length (44.51) compared to the three datasets in Section 4.2, i.e., Yoochoose 1/64 (4.24), Diginetica (4.85), and Retailrocket (4.30). KuaiRand-Pure is an emerging recommendation dataset collected from the recommendation logs of the video-sharing mobile app Kuaishou, providing a distinct domain.[5] The overall performance comparison is summarized in Table 12. In most cases, our Dimos achieves the best performance compared to the baseline models, indicating its effectiveness of preference learning from long sessions. Notably, on the two ranking metrics MRR and NDCG, Dimos shows its consistent advantage over all baselines at all recommendation list lengths. This suggests that Dimos not only retrieves relevant items but also ranks them more accurately at the top of the list.

To further investigate how the maximum session length affects performance, we tune the maximum session length among {5, 10, 20, 50, 100 }. The results are presented in Fig. 8. We observe that performance generally improves as the maximum session length increases from 5 to 20, with the best overall results achieved at 20. However, when the maximum session length is extended further to 50 or 100, performance slightly declines. One reason is that setting too large maximum session length may introduce noise and irrelevant early interactions, diminishing the focus on recent relevant behaviors. In summary, our Dimos benefits from appropriately longer session contexts, highlighting its parameter efficiency and robustness across a reasonable range of session lengths.

4.8. Model efficiency

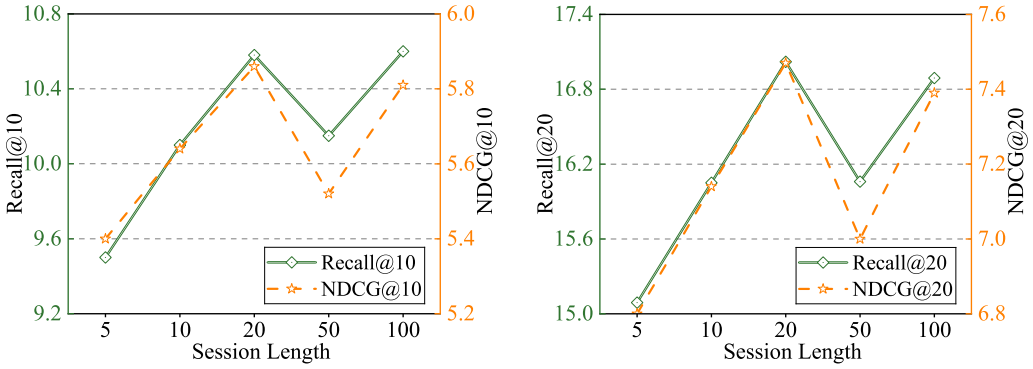
To address RQ7, we first evaluate the overall efficiency of our Dimos and six representative baseline models on three datasets, including three non-diffusion methods, i.e., CM-GNN (Wang et al., 2023a), GSAU (Cao et al., 2025), and SS4Rec (Xiao et al., 2025), and three diffusion-based methods, i.e., DiffuRec (Li et al., 2024c), DiffRec (Wang et al., 2023b), and PreferDiff (Liu et al., 2025a). Additionally, we investigate the impact of diffusion steps to demonstrate why our model is so efficient.

[5] https://kuairand.com/

Table 12
The performance of the examined models on the KuaiRand-Pure dataset.

Models	@5			@10			@15			@20		
	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG
CMGNN	5.70	3.31	3.90	9.40	3.80	5.08	12.67	4.05	5.95	15.71	4.22	6.67
GSAU	5.79	3.29	3.90	9.42	3.76	5.06	12.56	4.01	5.89	15.37	4.17	6.56
SS4Rec	6.36	3.38	4.11	**10.63**	3.93	5.48	13.88	4.19	6.33	**17.05**	4.37	7.08
DiffuRec	5.60	3.06	3.68	9.20	3.53	4.83	12.22	3.76	5.63	14.89	3.91	6.26
DiffRec	5.94	2.96	3.69	10.02	3.49	4.99	13.19	3.74	5.83	15.84	3.89	6.46
PreferDiff	2.73	1.40	1.73	3.56	1.52	2.00	4.67	1.60	2.29	5.41	1.65	2.47
Dimos (Ours)	**6.62***	**3.92***	**4.59***	10.58	**4.44***	**5.86***	**14.01***	**4.71***	**6.76***	17.02	**4.88***	**7.47***

For each column, the best performance and the second best performance methods are denoted in **bold** and underlined fonts respectively.
* Indicates that the improvement over the strongest baseline is statistically significant ($p < 0.05$) based on a paired t-test.



(a) Recall@10 and NDCG@10 (b) Recall@20 and NDCG@20

Fig. 8. Performance of Dimos with varying maximum session length on KuaiRand-Pure.

4.8.1. Overall efficiency

All experiments were conducted on a server equipped with an Intel(R) Xeon(R) Gold 6426Y CPU and a single NVIDIA GeForce RTX 4090 GPU. To ensure a fair and consistent comparison of training and inference efficiency, all models were trained and evaluated under an identical software environment: Python 3.8.10, PyTorch 2.0.0, and CUDA 11.8. The embedding dimension and batch size are consistently set to 100 throughout the training process. For each diffusion-based model, we adopted the optimal number of diffusion steps as reported in their respective original works. Specifically, the diffusion steps were set to 32 for DiffuRec, 5 for DiffRec, and 20 for PreferDiff. For our proposed Dimos, following the results of parameter sensitivity analysis, we set the diffusion steps to 2 on the Yoochoose 1/64 dataset, 5 on the Diginetica dataset, and 10 on the Retailrocket dataset. For each epoch, we evaluate the computational costs, including GPU memory usage, training duration, inference time, floating point operations (FLOPs), and the number of model parameters, as summarized in Tables 13 and 14.

Our observations are as follows. First, models based on implicit preference learning generally exhibit greater efficiency in terms of GPU memory utilization and training time compared to explicit preference learning models. Notably, Dimos requires only 7.3% of the GPU memory and 33.7% of the training time of CMGNN on average across all datasets. However, this training efficiency often comes at the expense of longer inference durations—for instance, DiffRec requires over ten times the inference time of the second-slowest model. Second, Dimos demonstrates a balanced efficiency in both spatial and temporal dimensions. It maintains GPU memory usage and training time comparable to diffusion-based models, while achieving inference speeds competitive with non-diffusion models. This dual advantage stems from the use of Bi-MaKAN as the backbone, which replaces computationally expensive self-attention with lightweight Mamba blocks to avoid quadratic time complexity. Additionally, a parameter-sharing mechanism significantly reduces memory requirements by eliminating redundant parameter storage. Third, Dimos exhibits strong scalability as dataset size increases. In contrast to DiffRec, whose inference time escalates rapidly on larger datasets, Dimos maintains a manageable computational overhead. This scalability makes Dimos particularly suitable for real-world deployment. While PreferDiff is the most efficient in terms of runtime, it consistently underperforms in recommendation effectiveness, highlighting the trade-off between efficiency and accuracy.

For FLOPs and the number of model parameters, our Dimos model requires only 0.45M FLOPs, which is dramatically lower than diffusion-based methods like DiffuRec (156.16M FLOPs) and DiffRec (53.02M FLOPs). Meanwhile, the model size of Dimos (2.27M) is comparable to most baseline models, suggesting a favorable space complexity.

4.8.2. Impact of diffusion steps

To validate that the high efficiency of Dimos primarily stems from its Bi-MaKAN-based denoising network, we evaluate four representative methods across varying numbers of diffusion steps: (1) Dimos, (2) Dual Bi-MaKAN, which employs Bi-MaKAN as both the forward feature encoder and the denoising network, (3) Bi-MaKAN as denoiser only, which removes the forward encoder, and (4) DiffuRec, the leading generative baseline.

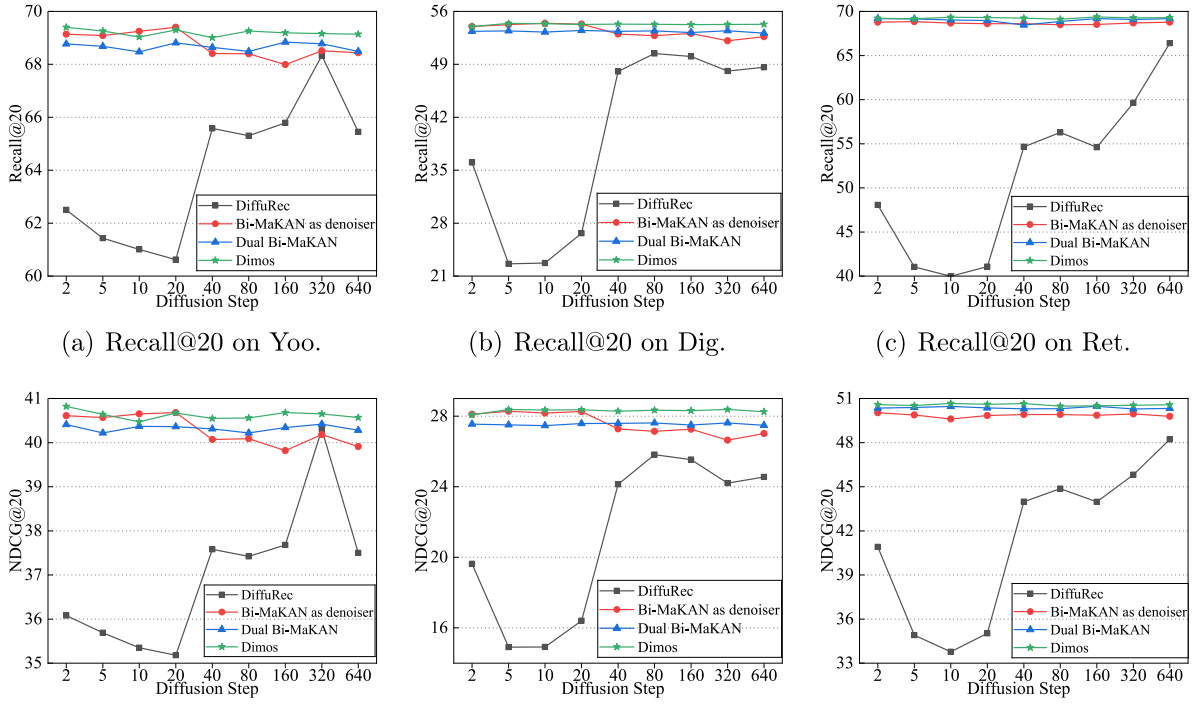
The results are illustrated in Fig. 9 and we draw the following observations. First, Dimos and its Bi-MaKAN-based variants consistently outperform DiffuRec even with a small number of diffusion steps, demonstrating the effectiveness of Bi-MaKAN as a backbone for diffusion-based SBRSs. Second, the superior efficiency of Dimos originates from its Bi-MaKAN-based denoising network. Specifically, by replacing the Transformer-based denoising network in DiffuRec with Bi-MaKAN, Dimos achieves higher performance with significantly fewer diffusion steps. Prior studies have highlighted that the number of diffusion steps substantially affects the performance of diffusion models (Ulhaq et al., 2022; Lin et al., 2024; Yang et al., 2024c; Li et al., 2025b). On average, Dimos reaches its peak performance with 53.33 times fewer diffusion steps than DiffuRec. Specifically, Dimos achieves

Table 13
Efficiency performance on the three datasets.

Dataset	Method	GPU memory (MB)	Training time (s)	Inference time (s)
Yoochoose 1/64	CMGNN	13 437	506.70	19.05
	GSAU	1231	206.27	1.80
	SS4Rec	4127	192.97	6.90
	DiffuRec	769	159.00	38.10
	DiffRec	1045	17.19	425.67
	PreferDiff	747	229.50	5.49
	Dimos (Ours)	763	170.72	5.10
Diginetica	CMGNN	13 455	961.08	46.42
	GSAU	1719	404.25	4.03
	SS4Rec	4163	357.01	16.50
	DiffuRec	1017	299.65	89.20
	DiffRec	1399	36.98	863.91
	PreferDiff	975	425.64	12.96
	Dimos (Ours)	1023	319.03	18.24
Retailrocket	CMGNN	12 469	1487.75	45.87
	GSAU	2401	600.03	4.63
	SS4Rec	4173	556.22	15.08
	DiffuRec	1181	456.73	81.79
	DiffRec	24 085	53.54	904.32
	PreferDiff	1125	650.56	12.79
	Dimos (Ours)	1191	505.67	30.71

Table 14
Floating-point operations and model size of the methods on the Yoochoose 1/64 dataset.

Method	CMGNN	GSAU	SS4Rec	DiffuRec	DiffRec	PreferDiff	Dual Bi-MaKAN	Dimos (Ours)
FLOPs (M)	3.81	4.84	0.02	156.16	53.02	5.24	0.19	0.45
# model parameters (M)	1.90	14.48	1.98	2.14	10.62	1.96	2.20	2.27



(a) Recall@20 on Yoo. (b) Recall@20 on Dig. (c) Recall@20 on Ret.

(d) NDCG@20 on Yoo. (e) NDCG@20 on Dig. (f) NDCG@20 on Ret.

Fig. 9. Performance of Dimos, Dimos's two variants, and DiffuRec with varying diffusion steps on Yoochoose 1/64 (Yoo.), Diginetica (Dig.) and Retailrocket (Ret.).

optimal results at 2 steps on Yoochoose 1/64, 5 steps on Diginetica, and 10 steps on Retailrocket, while DiffuRec requires 160, 80, and over 640 steps on the respective datasets—and still fails to outperform Dimos. Third, while Dual Bi-MaKAN incorporates Bi-MaKAN for both encoding and denoising to enhance distribution learning, it suffers from reduced performance stability and even underperforms on Diginetica. Dimos further improves upon Dual Bi-MaKAN by introducing an implicit preference learning module, which enhances both overall effectiveness and robustness across datasets.

This observed convergence acceleration can be attributed to the theoretically grounded design of the Bi-MaKAN backbone. The efficiency-accuracy trade-off in diffusion models is fundamentally tied to the representational capacity of the denoising network. Our Bi-MaKAN addresses the limitations of common backbones in sequence modeling: it provides linear-complexity, long-range dependency modeling via Mamba blocks, full bidirectional context via shared-parameter bidirectional processing, and expressive feature fusion via a lightweight KAN. Consequently, a single denoising step performed by this more powerful network yields a more accurate estimate of the clean data distribution than a step performed by a weaker network (e.g., an MLP or a Transformer under constrained computational budgets). This higher per-step fidelity directly translates to the empirical observation that our model requires drastically fewer steps (2–10) to reach peak performance, whereas methods with less effective backbones (e.g., DiffuRec) need orders of magnitude more steps to compensate, yet still may not achieve the same final quality.

4.9. Visualized case study

To address RQ8, we conduct two sets of visualization experiments aimed at demonstrating the functions of explicit and implicit preferences, as well as their impact on recommendation lists. First, we visualize the representations of explicit and implicit preferences learned by the exploit and explore branches, respectively. For each dataset, we randomly select a batch of sessions and apply singular value decomposition (SVD) to project the high-dimensional representations of explicit and implicit preferences into a two-dimensional feature space.

The resulting visualizations are shown in Fig. 10, from which we draw the following observations and insights. First, by comparing subfigures (a) and (b), (d) and (e), and (g) and (h), we observe that the distributions of implicit preferences are more concentrated, whereas those of explicit preferences are more dispersed. This suggests that users' implicit behaviors tend to be more homogeneous and exhibit shared patterns across sessions, while their explicit feedback reveals greater diversity and personalization. Second, the lower-left and upper-right regions of subfigures (c), (f), and (i) show large areas with light coloration, indicating substantial discrepancies between the preference representations learned by the exploit and explore branches. This finding supports the idea that each branch captures distinct aspects of user preferences: the explore branch uncovers broad, underlying behavioral patterns, while the exploit branch focuses on identifying session-specific or personalized preferences. Additionally, the upper-left and lower-right regions of the same subfigures also display considerable light-colored areas, suggesting notable variation within both implicit and explicit preferences. This intra-preference variability reflects the complexity and multi-faceted nature of user motivations, even within the same type of preference.

In the second case study, we compare the recommendation results from Dimos and its two variants, i.e., w/o explore branch and w/o exploit branch. We randomly select a session (ID: 104669) from the Yoochoose 1/64 dataset. For this session, we generate the top-5 recommended items along with their corresponding prediction probabilities for the three models. The results are shown in Fig. 11.

We observe that three items (IDs: 16034, 16064, and 16071) consistently appear in the top-5 recommendations across all three model configurations. This convergence indicates a strong consensus signal, suggesting these items are highly relevant to the user's immediate session context and are reliably captured by both explicit and implicit modeling approaches. Furthermore, we observe that w/o explore branch uniquely recommends items 16054 and 14723. This implies these items exhibit a strong, direct correlation with the observed sequence of user actions, reflecting short-term and session-specific patterns. Additionally, w/o exploit branch uniquely recommends items 16056 and 16028. This highlights the branch's capacity for exploratory discovery, capturing implicit preferences beyond the historical interactions. Finally, our Dimos synthesizes these perspectives into a more comprehensive recommendation list. It retains the three consensus items while introducing two unique items (IDs: 16035 and 15953). The two unique items likely represent a synergistic balance where Dimos resolves inconsistencies between the immediate session context and latent preferences. Consequently, the final ranking reflects a calibrated trade-off: it maintains grounding in the observed context while promoting a balanced set of candidates that bridges short-term relevance and broader user interest.

In summary, these visualizations highlight the complementary strengths of the exploit and explore branches in modeling user preferences. Their combined use offers a more comprehensive understanding of user behavior, thereby enhancing the effectiveness of session-based recommendation systems.

5. Related work

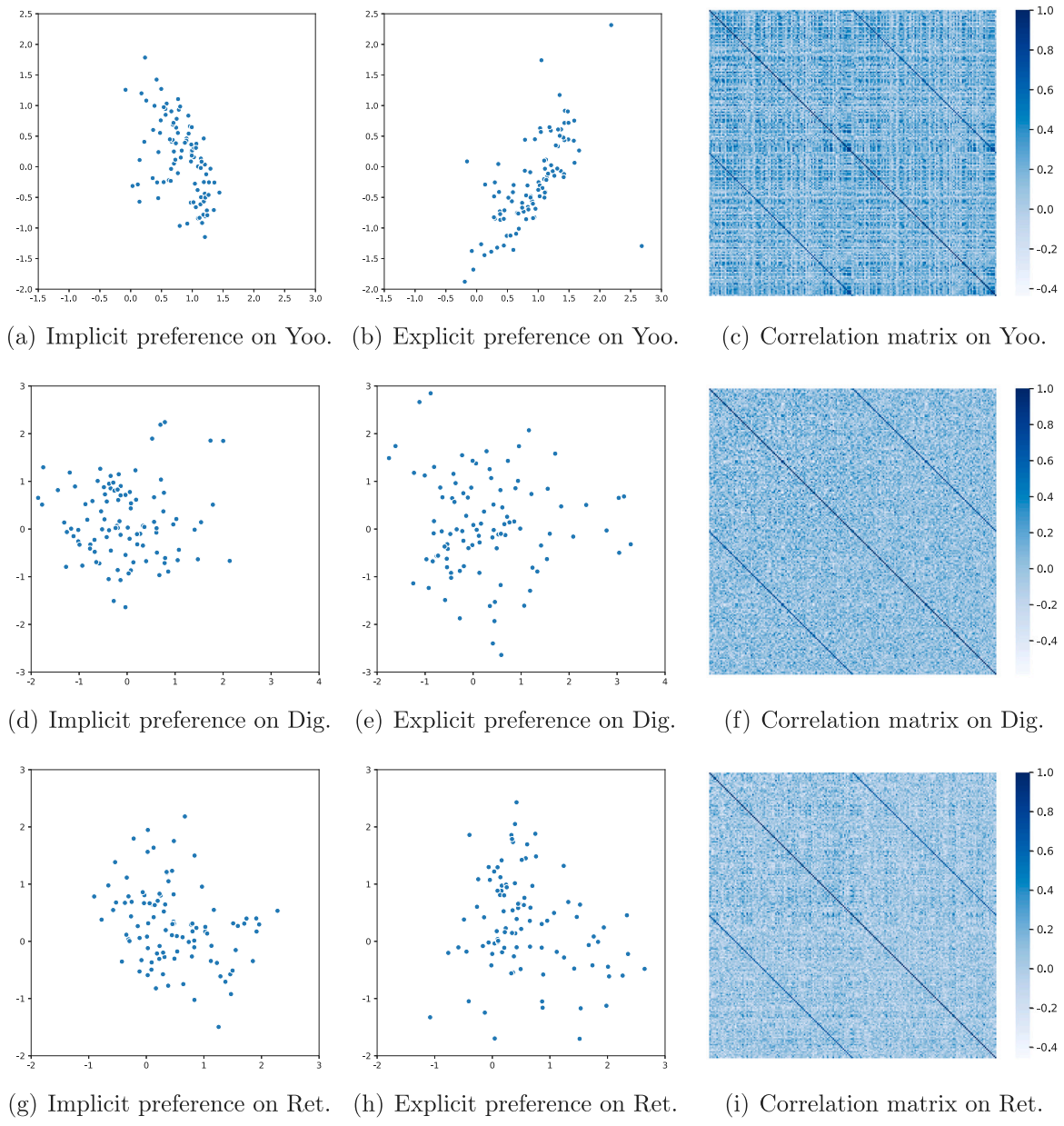
This section reviews key related works to contextualize the placement and contribution of Dimos within the broader literature. It begins with an overview of recent research trends in discriminative and generative SBRSs, followed by a discussion on the emerging use of Mamba and diffusion models in the recommender systems domain.

5.1. Session-based recommender systems

SBR has emerged as a prominent research direction, aiming to capture user preferences by modeling the sequential dependencies and temporal patterns within item interaction sequences. Unlike traditional sequential recommendation methods that rely on complete historical data to build long-term user profiles, SBR emphasizes the current session context, making it particularly suitable for scenarios involving new users and real-time dynamic recommendations. However, session data typically presents unique challenges. The length of sessions—measured by the number of interactions, is generally short, with the median session length falling below six items in most widely used public datasets. Moreover, despite the chronological ordering of items within sessions, clear sequential behavioral patterns are often lacking. To address this, some studies have transformed the inherently sequential session data into various forms of session graphs to better support preference learning. Following dominant paradigms in the field, existing SBRS approaches can be broadly categorized into two groups, as summarized in Table 15: discriminative methods and generative methods. To offer a more refined taxonomy, this classification is further organized along two key dimensions: (1) the data structure used for session representation, and (2) the specific learning strategy adopted for modeling user preferences.

5.1.1. Explicit preference modeling methods

As the widely adopted paradigm, discriminative SBRSs aim to learn the representations of user preferences from interactions. Similar to most sequential recommender systems, sequence learning-based SBRSs adopt sequential neural networks as backbone to model behavioral patterns from session sequences. These methods assume that the sequential order of user–item interactions reflect the user preferences, while the recent interactions are more significant than the older interactions. Specifically, as one of the pioneering works, GRU4Rec (Hidasi et al.,



(a) Implicit preference on Yoo. (b) Explicit preference on Yoo. (c) Correlation matrix on Yoo.

(d) Implicit preference on Dig. (e) Explicit preference on Dig. (f) Correlation matrix on Dig.

(g) Implicit preference on Ret. (h) Explicit preference on Ret. (i) Correlation matrix on Ret.

Fig. 10. Distribution visualizations of explicit preference and implicit preference representations, along with the visualized corresponding preference correlation matrices, on Yoochoose 1/64 (Yoo.), Diginetica (Dig.) and Retailrocket (Ret.).

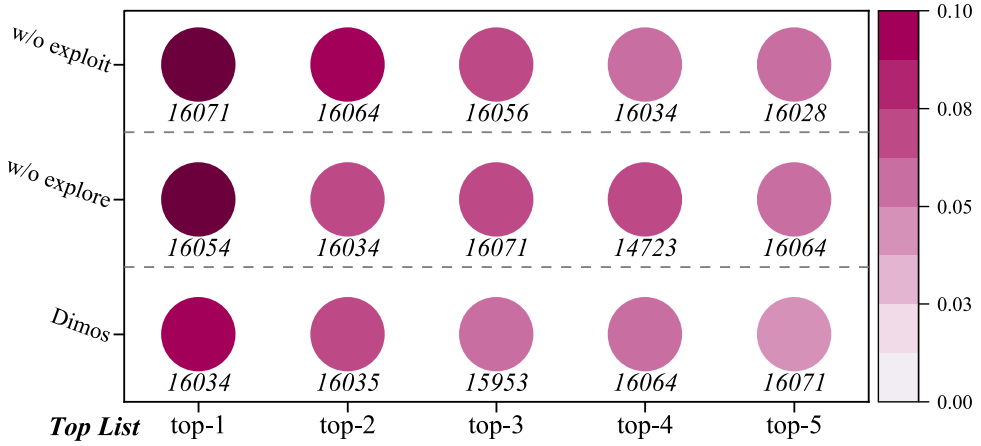


Fig. 11. Impact of the explicit and implicit preferences on the recommendation lists. Darker colors indicate higher normalized recommendation scores.

Table 15
Overview of representative SBRSs. We summarize them along three dimensions: paradigm, data structure, main approaches.

Paradigm	Model	Data structure	Main approaches
Discriminative method	GRU4Rec (Hidasi et al., 2016)	Session sequence	GRU
	NARM (Li et al., 2017)	Session sequence	GRU, Attention
	STAMP (Liu et al., 2018)	Session sequence	MLP, Attention
	NextItNet (Yuan et al., 2019)	Session sequence	Dilated CNN
	PARSRec (Gholami et al., 2022)	Session sequence	Attention-fused RNN
	LSIDN (Zhang et al., 2024a)	Session sequence	GRU, Attention
	MiaSRec (Choi et al., 2024)	Session sequence	Sparse attention, Attention
	SRGNN (Wu et al., 2019)	Vanilla session graph, Session sequence	GGNN, Attention
	LESSR (Chen and Wong, 2020)	Edge-order preserving multigraph, Shortcut graph, Session sequence	GGNN, GAT, Attention
	SGNN-HN (Pan et al., 2020)	Session star graph, Session sequence	GGNN, Attention
	GCE-GNN (Wang et al., 2020)	Vanilla session graph, Global session graph, Session sequence	GAT, Attention
	FGNN (Qiu et al., 2020)	Broadly connected session	Weighted GAT, Set2Set (Vinyals et al., 2016)
	CGL (Pan et al., 2022a)	Vanilla session graph, Global session graph, Session sequence	GGNN, GAT, Attention
	CMGNN (Wang et al., 2023a)	Vanilla session graph, Global session graph, Hypergraph, Session sequence	GAT, Attention, HCNN
	PosRec (Qiu et al., 2022)	Vanilla session graph, Session sequence	Position-aware GGNN, Attention
	GCARM (Pan et al., 2022b)	Vanilla session graph, Global session graph, Session sequence	Graph co-attention network
	H3GNN (Yin et al., 2024)	Hierarchical hypergraph, Global session graph	HCNN, GCN, Attention
	M^3T (Zhuo et al., 2024)	Forward session graph, Reverse session graph, Session sequence	GAT, GRU, Attention
	Wang et al. (2024c)	Item Knowledge Graph, Session sequence	Heterogeneous graphformer, Attention
	RNMSR (Wang et al., 2024e)	Similarity-based item-pairwise session graph	MLP-based GNN, MLP
	MHCL (Guo et al., 2025)	Behavior-based global hypergraph, Local session heterogeneous hypergraph	HCNN, Attention
	SOFA (Li et al., 2025a)	Session sequence	Temporal convolutional networks
Generative method	CVRM (Wang et al., 2018)	Session sequence	Variational recurrent model
	VASER (Zhou et al., 2019)	Session sequence	Normalizing flows, GRU, Attention
	DCFGAN (Zhao et al., 2022)	Session sequence	GAN, GRU
	PO4ISR (Sun et al., 2024)	Session sequence	GPT-3.5-turbo, Prompt engineering
Mixed method	VASER-DA (Zhong et al., 2020)	Session sequence	Normalizing flows, Variational attention, Deterministic attention, GRU
	VASER-VA (Zhong et al., 2020)		
	MMSBR (Zhang et al., 2024b)	Session sequence	DALL· E, Wasserstein self-attention, Bert, Hierarchical pivot transformer, GoogLeNet
	LLM-BRec (Jalan et al., 2024)	Heterogeneous graph	MiniLM-L6-v2, Prompt engineering, Bert, GAT
	LLMGR (Guo et al., 2024)	Vanilla session graph	LLaMA2-7B, GGNN, Attention, Prompt engineering
	ALKDRec (Du et al., 2025)	Session sequence	GPT-4-turbo, Knowledge distillation
	LLM4SBR (Qiao et al., 2025)	Vanilla session graph	Qwen-7B-Chat, GGNN, Attention, Prompt engineering
	Dimos (Ours)	Session sequence	DDPM, Bi-MaKAN, Attention

CNN denotes convolutional neural network. GRU denotes gate recurrent unit. LSTM denotes long short-term memory network. GCN denotes graph convolutional network. GAT denotes graph attention network. GGNN denotes gated graph neural network. HCNN denotes hypergraph CNN. GAN denotes generative adversarial network.

2016) constructs session sequences from recent interactions and leverages GRU to capture evolving user preferences. Moreover, by treating each batch of session sequences as an image, NextItNet (Yuan et al., 2019) combines masked filters with 1D dilated CNNs to expand receptive fields, facilitating long-term user preference learning. Recently, MiaSRec (Choi et al., 2024) introduces frequency encoding to reflect repeat patterns and adopts sparse attention network to select essential user intents.

Although items within sessions are organized chronologically, clear order patterns are often absent (Wang et al., 2022a; Li et al., 2025c). For instance, shopping sessions are sometimes unordered, as users may add a basket of items without following a specific sequence, e.g., {bread, milk, eggs}. In such unordered sessions, the dependencies among items are based on co-occurrence rather than sequential order, making traditional sequence models unsuitable for capturing the hidden relationships. Consequently, some SBRSs first transform session sequences into various session graphs to represent complex

contextual relationships among items, then adopt random walk and graph neural networks to learn item features. Specifically, SRGNN (Wu et al., 2019) transforms session sequences into vanilla session graphs and adopts GGNN to learn item features. Moreover, M^3T (Zhuo et al., 2024) integrates the sequence-view information and the graph-view information of items in a session, highlighting the ambiguity between cross-view features. Additionally, RNMSR (Wang et al., 2024e) constructs the similarity-based item-pairwise session graph to capture the dependencies within the session. Recently, MHCL (Guo et al., 2025) leverages graph learning on the session heterogeneous hypergraph and the multi-behavior line graph to capture user preferences. Furthermore, SOFA (Li et al., 2025a) designs a session-oriented fairness-aware algorithm to achieve global-oriented fairness by maximizing session-oriented fairness while maintaining high session utilities.

5.1.2. Implicit preference modeling methods

As another paradigm, generative SBRSs adopt variational autoencoders, normalizing flows, adversarial learning, and LLMs to learn the underlying distribution for preference modeling. Although relatively few works have been proposed in this direction, they have demonstrated promising performance. Specifically, CVRM (Wang et al., 2018) employs the stochastic latent variable to capture the knowledge of frequent click patterns and impose variability for the sequential behavior modeling. Moreover, VASER (Zhou et al., 2019) integrates normalizing flows and variational inference for enhanced probabilistic modeling. Additionally, DCFGAN (Zhao et al., 2022) integrates reinforcement learning to leverage immediate user feedback and employs adversarial training combined with enhanced negative sampling to improve recommendation performance. Furthermore, PO4ISR (Sun et al., 2024) discovers varying numbers of semantic intents hidden in different sessions for more accurate and comprehensible recommendations through iterative prompt optimization.

5.1.3. Combining explicit with implicit preference modeling

Despite some progress achieved by existing discriminative and generative methods, most of them still fall short of delivering significantly improved performance. Specifically, discriminative methods are often hindered by data sparsity and exposure bias, while generative methods suffer from limited representation capacity and training instability (Lin et al., 2024). Consequently, some works integrate both paradigms to learn preference representations. Specifically, VASER-VA (Zhong et al., 2020) introduces soft attention as auxiliary latent features to enhance the effectiveness of variational inference. Moreover, SessionRec (Huang et al., 2025) addresses the fundamental misalignment between conventional next-item prediction paradigm and real-world recommendation scenarios. Additionally, MMSBR (Zhang et al., 2024b) models multimodal information including descriptive information, i.e., images and text, and numerical information, i.e. price, to characterize user preferences. Moreover, ALKDRec (Du et al., 2025) is an active LLM-based knowledge distillation Recommendation method for a sustainable and effective solution to SBR. Furthermore, LLM4SBR (Qiao et al., 2025) integrates semantic and behavioral signals from multiple views. We can find that most existing mixed methods strive to unlock the power of large pre-trained models for preference learning. While prompt engineering offers benefits in efficiency and usability, it also presents several limitations. Specifically, its effectiveness is highly contingent on the quality of manually crafted prompts, which frequently necessitate extensive trial and error (Sahoo et al., 2024). Moreover, compared to full training, prompt engineering affords less flexibility, restricting the model's adaptability (Chen et al., 2023).

5.2. Sequential recommendation with Mamba

As a promising sequential neural network, Mamba shows superior performance in different areas (Wang et al., 2024a), such as language modeling and image restoration. Mamba have recently emerged as a powerful backbone for sequential recommender systems. Specifically, Mamba4Rec (Liu et al., 2024) introduces a vanilla Mamba block to replace the self-attention component of the standard transformer encoder, whereas RecMamba (Yang et al., 2024b) substitutes the entire transformer encoder with the vanilla Mamba block. Moreover, EchoMamba4Rec (Wang et al., 2024b) introduces a bidirectional Mamba module that integrates both forward and reverse Mamba components, enabling the model to utilize information from past and future interactions. Additionally, SSD4Rec (Qu et al., 2024) marks the variable- and long-length item sequences with sequence registers and processes the item representations with bidirectional structured state space duality blocks. Furthermore, SIGMA (Liu et al., 2025b) introduces a bidirectional, partially flipped Mamba that incorporates a well-designed dense selective gate to assign weights to each direction, thereby addressing challenges in context modeling. Recently, SS4Rec (Xiao et al., 2025) integrates a time-aware SSM to manage irregular time intervals and a relation-aware SSM to capture contextual dependencies. While Mamba has shown promise in discriminative settings for sequential recommendation, its capabilities in generative paradigms for SBR are yet to be systematically investigated.

5.3. Sequential recommendation with diffusion models

Recently, diffusion models have emerged as the state-of-the-art in generative modeling paradigms, demonstrating promising performance across various domains such as computer vision (Fuest et al., 2024), natural language processing (Yang et al., 2024c), and recommender systems (Lin et al., 2024). Compared to VAEs and GANs, the denoising process in diffusion models enhances their ability to capture multi-grained feature representations and to generate high-quality samples (Lin et al., 2024). Particularly, Compared to traditional approaches, diffusion-based recommender systems effectively address challenges related to insufficient collaborative signals, weak latent representations, and noisy data (Lin et al., 2024).

Some diffusion-based works focus on designing effective denoising networks to improve recommendation performance. Specifically, DiffuRec (Li et al., 2024c) and DiffRec (Du et al., 2023) introduce Transformer encoder as denoising network, while T-DiffRec (Zhao et al., 2024) and MISD (Li et al., 2024a) adopt MLP-based denoising network. Moreover, DiffuASR (Liu et al., 2023) treats the sequence dimension as the image channel to adapt the U-Net-based denoising network, allowing it to preserve sequential information while effectively predicting the added noise. Other works adopt diffusion models to generate high-quality data for sequential recommendation. Specifically, CaDiRec (Cui et al., 2024) employs a context-aware diffusion model to generate alternative items for the given positions within a sequence. Additionally, Diff4Rec (Wu et al., 2023) employs a curriculum-scheduled diffusion augmentation method to generate user–item interactive data. Recent works focus on applying diffusion models within various latent spaces to reduce computational resource requirements while maintaining their quality and flexibility. Specifically, DiffRIS (Niu et al., 2024) and IDSRec (Niu et al., 2025) incorporate implicit feature extraction into the diffusion process to resist noisy interactions. Moreover, DiQDiff (Mao et al., 2025) quantizes sequences into semantic vectors based on a codebook, extracting robust guidance to understand user interests. Furthermore, SeeDRec (Ma et al., 2024b) enhances the diffusion objective and maintains low computational costs by elevating it from the item level to the sememe level. Despite some success, existing diffusion-based sequential recommender systems still rely on sub-optimal forward feature encoder and denoising network, hindering the ability to achieve full potential in performance.

6. Discussion and conclusion

This study explores the application of diffusion models in the session-based recommendation task, serving as a response to the future research directions outlined in DiffuRec (Li et al., 2024c). In this paper, we propose a novel framework Dimos, which integrate explicit and implicit preference learning to improve recommendation performance. Moreover, we tailor Dimos's backbone as Bi-MaKAN, which adopts a pair of bidirectional Mamba blocks with shared parameters to increase the receptive field for better item feature learning, while alleviating over-fitting. Furthermore, we introduce the KAN-based method to fuse the bidirectional features effectively and efficiently. Extensive experiments conducted on three real-world datasets demonstrate that Dimos can achieve the state-of-the-art performance. Three groups of ablation studies validate the effectiveness of the overall framework, the structure of Bi-MaKAN, and the preference fusion method, respectively. The subsequent three groups of ablation studies confirm the suitability of Bi-MaKAN within Dimos. Particularly, based on the results of efficiency experiments, we empirically find that adopting Bi-MaKAN can significantly reduce the number of diffusion steps, thereby improving the efficiency of Dimos.

We acknowledge several threats to the validity of our findings. First, while the present work focuses on session-based recommendation, it is worth noting that recent research has expanded into lifelong sequential recommendation, which deals with extremely long and evolving user histories (Yang et al., 2024b). The architectural design of our model, particularly its efficient sequential modeling capability, makes it a promising candidate for this challenging setting. To realize this potential for lifelong sequences, specific modifications would be required to address its distinct characteristics. Second, this study focuses on sequential IDs, integrating rich side information (e.g., multi-modal features, social context) could alter the learning dynamics between branches.

The current work opens several avenues for future research. First, exploring a more formal theoretical foundation for the proposed model constitutes a promising research direction, such as analyzing its convergence properties or representational capacity from the perspectives of dynamical systems or information theory. Second, while the current study employs a static, tunable weight for preference fusion to ensure interpretability, developing dynamic and context-aware fusion methods, such as those based on gating networks or user profile conditioning, presents a promising direction to further enhance the model's adaptability. Finally, to rigorously validate the practical efficacy and business impact of Dimos, performing rigorous A/B testing in real-world, large-scale recommendation platforms is a crucial next step. This will allow us to assess its performance under dynamic, production-environment conditions, including user engagement metrics and long-term satisfaction, beyond offline accuracy.

CRediT authorship contribution statement

Weiyue Li: Writing – original draft, Validation, Software, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Ming Gao:** Writing – review & editing, Supervision, Resources. **Bowei Chen:** Writing – review & editing. **Jingmin An:** Writing – review & editing. **Hao Dong:** Writing – review & editing, Visualization. **Wei Jiang:** Writing – review & editing. **Jiafu Tang:** Writing – review & editing.

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.

Acknowledgments

This research was supported and funded by the National Natural Science Foundation of China (No. 72293563, 72501053); the Basic Scientific Research Project of Liaoning Provincial Department of Education, China (No. JYTZD2023050); the Natural Science Foundation of Liaoning Province, China (No. 2024-MS-175); the Liaoning Province Key Research and Development Project, China (No. 2024JH2/102400020); Joint Plan of Liaoning Province Science and Technology Plan, China (No. 2024-MSLH-009); and the Dalian Scientific and Technological Talents Innovation Support Plan, China (No. 2022RG17). The authors would also like to thank Hongyu Wang of the Institute of Computing Technology, Chinese Academy of Sciences, for helpful discussions related to this work.

Appendix. Overall performance compared with baseline models

The performance of Dimos is compared with that of competing baseline models on the Yoochoose 1/64, Diginetica, and Retailrocket datasets in Tables A.16, A.17, and A.18, respectively.

Table A.16
Recommendation performance of examined models on the Yoochoose 1/64 dataset.

Models	Category	@5			@10			@15			@20		
		Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG
Pop	Non-neural methods	5.04	2.69	3.25	7.95	3.09	4.21	10.07	3.26	4.77	12.69	3.40	5.38
Item-KNN		26.70	16.83	19.17	35.41	18.01	22.01	39.96	18.37	23.22	42.87	18.53	23.91
CORE-ave		44.79	26.60	31.12	57.05	28.25	35.10	63.19	28.74	36.73	66.80	28.95	37.59
GRU4Rec	Traditional neural methods	45.72	27.93	32.36	57.87	29.57	36.30	63.69	30.03	37.84	67.22	30.23	38.68
NextItNet		40.07	23.62	27.70	52.13	25.23	31.61	58.21	25.71	33.22	62.04	25.93	34.12
SASRec		47.09	28.96	33.46	59.44	30.62	37.47	65.43	31.10	39.06	69.03	31.30	39.92
MSDCCL		44.13	26.36	30.77	56.51	28.02	34.79	62.71	28.52	36.43	66.48	28.73	37.32
SRGNN	GNN-based methods	46.18	28.13	32.61	58.43	29.78	36.59	64.26	30.24	38.14	67.79	30.44	38.97
CMGNN		47.73	29.34	33.91	60.01	30.99	37.90	65.70	31.44	39.40	69.22	31.64	40.24
GSAU		46.23	28.50	32.91	58.17	30.11	36.79	63.97	30.57	38.32	67.71	30.78	39.21
Mamba4Rec	Mamba-based methods	46.57	28.31	32.85	58.79	29.96	36.82	64.70	30.43	38.39	68.23	30.63	39.22
RecMamba		47.04	28.89	33.40	58.91	30.49	37.26	64.61	30.94	38.77	68.01	31.13	39.57
EchoMamba4Rec		45.44	28.25	32.52	56.99	29.82	36.28	62.40	30.25	37.72	65.91	30.44	38.55
MLSA4Rec		45.72	28.63	32.88	57.25	30.18	36.62	62.86	30.63	38.11	66.25	30.82	38.91
SIGMA		46.48	28.35	32.86	58.69	30.00	36.82	64.58	30.46	38.39	68.15	30.67	39.23
SS4Rec		47.18	28.80	33.37	59.30	30.44	37.31	65.23	30.91	38.88	68.80	31.11	39.72

(*continued on next page*)

Table A.16 (*continued*).

Models	Category	@5			@10			@15			@20		
		Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG
ACVAE	Traditional generative methods	14.36	3.12	4.19	15.27	3.67	6.00	15.58	3.98	7.07	15.74	4.20	7.94
ContrastVAE		3.26	1.97	2.29	5.08	2.19	2.86	6.73	2.31	3.29	7.86	2.38	3.55
SparseEnNet		22.83	13.60	15.89	29.97	14.55	18.20	34.25	14.89	19.33	37.24	15.06	20.03
DiffuRec	Diffusion-based methods	43.38	26.68	30.83	55.02	28.25	34.61	60.94	28.72	36.18	64.55	28.92	37.03
DiffRec		30.52	24.13	25.72	35.23	24.76	27.24	37.80	24.96	27.92	39.66	25.06	28.36
L-DiffRec		19.13	13.07	14.56	24.74	13.81	16.37	28.79	14.13	17.44	31.56	14.29	18.09
CaDiRec		22.12	13.10	15.33	29.54	14.09	17.73	34.07	14.45	18.94	37.13	14.62	19.66
PreferDiff		34.57	26.16	28.26	39.01	26.77	29.71	41.17	26.94	30.28	42.46	27.02	30.59
Dimos (Ours)	Hybrid method	**48.37***	**30.07***	**34.62***	**60.34***	**31.69***	**38.51***	**65.92***	**32.13***	**39.99***	**69.34***	**32.32***	**40.80***

For each column, the best performance and the second best performance methods are denoted in **bold** and underlined fonts respectively.
* Indicates that the improvement over the strongest baseline is statistically significant ($p < 0.05$) based on a paired t-test.

Table A.17
Recommendation performance of examined models on the Diginetica dataset.

Models	Category	@5			@10			@15			@20		
		Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG
Pop	Non-neural methods	0.52	0.23	0.30	0.95	0.29	0.44	1.29	0.32	0.53	1.59	0.33	0.60
Item-KNN		19.27	11.05	13.01	28.34	12.25	15.94	34.48	12.73	17.56	38.99	12.98	18.63
CORE-ave		30.65	18.27	21.34	41.39	19.70	24.81	48.21	20.23	26.61	53.02	20.51	27.75
GRU4Rec	Traditional neural methods	26.98	15.57	18.39	37.98	17.03	21.94	45.00	17.58	23.80	50.14	17.87	25.01
NextItNet		20.11	11.10	13.33	29.74	12.38	16.43	36.16	12.89	18.13	40.92	13.15	19.25
SASRec		28.76	16.91	19.85	39.93	18.40	23.45	47.04	18.96	25.33	52.23	19.25	26.56
MSDCCL		19.53	10.67	12.86	29.31	11.96	16.01	36.07	12.49	17.79	41.35	12.79	19.04
SRGNN	GNN-based methods	27.85	16.23	19.10	38.91	17.70	22.67	45.93	18.25	24.53	51.10	18.54	25.75
CMGNN		29.34	17.06	20.10	40.63	18.56	23.75	47.65	19.12	25.60	52.73	19.40	26.80
GSAU		27.55	16.08	18.92	38.48	17.53	22.45	45.62	18.09	24.34	50.79	18.39	25.56
Mamba4Rec	Mamba-based methods	28.31	16.46	19.39	39.43	17.94	22.98	46.36	18.48	24.81	51.44	18.77	26.01
RecMamba		30.07	17.71	20.77	41.17	19.18	24.35	48.19	19.73	26.21	53.12	20.01	27.37
EchoMamba4Rec		29.49	17.67	20.60	40.03	19.07	24.00	46.62	19.59	25.74	51.38	19.85	26.87
MLSA4Rec		28.62	16.86	19.77	39.37	18.29	23.24	46.11	18.82	25.03	50.94	19.09	26.17
SIGMA		27.05	15.88	18.64	37.84	17.31	22.12	44.92	17.87	23.99	50.03	18.15	25.20
SS4Rec		29.29	17.60	20.50	40.05	19.03	23.97	46.81	19.56	25.76	51.84	19.84	26.94
ACVAE	Traditional generative methods	8.02	5.42	7.35	12.68	7.01	8.33	16.02	7.95	8.74	18.58	8.60	8.97
ContrastVAE		0.34	0.17	0.21	0.54	0.19	0.27	0.74	0.21	0.33	0.93	0.22	0.37
SparseEnNet		11.22	5.78	7.12	17.80	6.65	9.24	22.94	7.05	10.60	27.11	7.29	11.58
DiffuRec	Diffusion-based methods	28.17	16.60	19.46	38.78	18.01	22.89	45.57	18.54	24.68	50.49	18.82	25.85
DiffRec		6.15	3.77	4.36	8.44	4.07	5.09	9.90	4.19	5.48	10.97	4.25	5.73
L-DiffRec		17.13	12.10	13.35	22.04	12.75	14.92	25.48	13.02	15.83	28.10	13.16	16.45
CaDiRec		11.35	5.90	7.24	18.16	6.80	9.43	23.37	7.20	10.81	27.36	7.43	11.75
PreferDiff		20.19	15.84	16.93	22.49	16.15	17.68	23.85	16.26	18.04	24.95	16.32	18.30
Dimos (Ours)	Hybrid method	**31.42***	**18.64***	**21.81***	**42.52***	**20.12***	**25.39***	**49.51***	**20.67***	**27.24***	**54.39***	**20.94***	**28.39***

For each column, the best performance and the second best performance methods are denoted in **bold** and underlined fonts respectively.
* Indicates that the improvement over the strongest baseline is statistically significant ($p < 0.05$) based on a paired t-test.

Table A.18
Recommendation performance of examined models on the Retailrocket dataset.

Models	Category	@5			@10			@15			@20		
		Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG
Pop	Non-neural methods	0.44	0.23	0.28	0.72	0.27	0.37	0.90	0.29	0.42	1.24	0.31	0.50
Item-KNN		9.75	6.36	7.08	13.35	6.83	8.22	15.53	7.00	8.80	17.05	7.09	9.15
CORE-ave		54.60	40.49	44.03	61.92	41.48	46.40	65.73	41.78	47.41	68.22	41.92	48.00
GRU4Rec	Traditional neural methods	54.44	41.17	44.50	61.14	42.08	46.67	64.58	42.35	47.59	66.89	42.48	48.13
NextItNet		45.36	33.61	36.54	52.45	34.56	38.84	56.27	34.86	39.85	58.79	35.00	40.45
SASRec		53.18	41.11	44.12	60.45	42.08	46.48	64.37	42.40	47.52	67.02	42.54	48.15
MSDCCL		43.18	31.82	34.65	50.41	32.79	36.99	54.55	33.11	38.09	57.38	33.27	38.76
SRGNN	GNN-based methods	54.65	40.90	44.35	61.56	41.84	46.60	65.05	42.11	47.52	67.40	42.25	48.07
CMGNN		55.46	41.14	44.73	62.40	42.07	46.98	65.92	42.35	47.91	68.30	42.48	48.47
GSAU		49.97	37.95	40.95	57.47	38.96	43.38	61.45	39.27	44.43	64.14	39.43	45.07

(*continued on next page*)

Table A.18 (*continued*).

Models	Category	@5			@10			@15			@20		
		Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG
Mamba4Rec		55.80	42.27	45.66	62.26	43.14	47.76	65.66	43.41	48.66	67.87	43.54	49.18
RecMamba		56.65	43.17	46.55	63.11	44.05	48.65	66.37	44.31	49.52	68.55	44.43	50.03
EchoMamba4Rec	Mamba-based	55.51	42.77	45.97	61.26	43.54	47.83	64.24	43.78	48.63	66.19	43.89	49.09
MLSA4Rec	methods	55.70	42.43	45.76	61.91	43.27	47.78	65.04	43.52	48.61	67.14	43.63	49.10
SIGMA		55.25	41.77	45.15	61.87	42.66	47.30	65.31	42.93	48.21	67.55	43.06	48.74
SS4Rec		56.03	42.43	45.84	62.72	43.33	48.01	66.12	43.60	48.91	68.34	43.72	49.44
ACVAE	Traditional	3.38	2.43	3.67	4.89	2.95	4.03	5.92	3.25	4.17	6.67	3.44	4.25
ContrastVAE	generative	0.15	0.08	0.10	0.26	0.09	0.13	0.31	0.09	0.14	0.40	0.10	0.16
SparseEnNet	methods	6.27	3.47	4.16	9.08	3.85	5.07	10.98	4.00	5.57	12.42	4.08	5.91
DiffuRec		48.89	40.01	42.26	51.92	40.42	43.24	53.48	40.54	43.66	54.59	40.61	43.92
DiffRec	Diffusion-	2.86	2.10	2.29	3.37	2.17	2.46	3.71	2.20	2.54	3.95	2.21	2.60
L-DiffRec	based	1.86	1.31	1.45	2.50	1.40	1.66	3.00	1.44	1.79	3.52	1.47	1.91
CaDiRec	methods	4.97	2.76	3.31	7.25	3.06	4.04	8.89	3.19	4.47	10.03	3.26	4.74
PreferDiff		35.85	35.14	35.32	36.30	35.20	35.46	36.61	35.22	35.54	36.85	35.24	35.60
Dimos (Ours)	Hybrid method	**57.70***	**43.83***	**47.31***	**63.99***	**44.68***	**49.36***	**67.17***	**44.93***	**50.20***	**69.25***	**45.05***	**50.69***

For each column, the best performance and the second best performance methods are denoted in **bold** and underlined fonts respectively.

* Indicates that the improvement over the strongest baseline is statistically significant ($p < 0.05$) based on a paired t-test.

Data availability

Data will be made available on request.

References

Bansal, A., Borgnia, E., Chu, H., Li, J., Kazemi, H., Huang, F., Goldblum, M., Geiping, J., Goldstein, T., 2023. Cold diffusion: Inverting arbitrary image transforms without noise. In: Conference on Neural Information Processing Systems.

Becker, E., Pandit, P., Rangan, S., Fletcher, A.K., 2022. Instability and local minima in GAN training with kernel discriminators. In: Conference on Neural Information Processing Systems.

Benigni, M., Dacrema, M.F., Jannach, D., 2025. Diffusion recommender models and the illusion of progress: A concerning study of reproducibility and a conceptual mismatch. CoRR, abs/2505.09364.

Cao, Y., Yang, L., Liu, Z., Liu, Y., Wang, C., Liang, Y., Peng, H., Yu, P.S., 2025. Graph-sequential alignment and uniformity: Toward enhanced recommendation systems. In: International World Wide Web Conference. pp. 888–892.

Chen, T., Wong, R.C., 2020. Handling information loss of graph neural networks for session-based recommendation. In: ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 1172–1180.

Chen, B., Zhang, Z., Langrené, N., Zhu, S., 2023. Unleashing the potential of prompt engineering in large language models: a comprehensive review. CoRR, abs/2310.14735.

Chen, J., Zou, G., Zhou, P., Wu, Y., Chen, Z., Su, H., Wang, H., Gong, Z., 2024. Sparse enhanced network: An adversarial generation method for robust augmentation in sequential recommendation. In: AAAI Conference on Artificial Intelligence. pp. 8283–8291.

Choi, J., Hong, S., Park, N., Cho, S., 2023. Blurring-sharpening process models for collaborative filtering. In: International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1096–1106.

Choi, M., Kim, H., Cho, H., Lee, J., 2024. Multi-intent-aware session-based recommendation. In: International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 2532–2536.

Cui, Z., Wu, H., He, B., Cheng, J., Ma, C., 2024. Context matters: Enhancing sequential recommendation with context-aware diffusion-based contrastive learning. In: ACM International Conference on Information and Knowledge Management. pp. 404–414.

Dang, Y., Yang, E., Guo, G., Jiang, L., Wang, X., Xu, X., Sun, Q., Liu, H., 2024. TiCoSeRec: Augmenting data to uniform sequences by time intervals for effective recommendation. IEEE Trans. Knowl. Data Eng. 36 (6), 2686–2700.

Du, Y., Sun, Z., Wang, Z., Chua, H., Zhang, J., Ong, Y., 2025. Active large language model-based knowledge distillation for session-based recommendation. In: AAAI Conference on Artificial Intelligence. pp. 11607–11615.

Du, H., Yuan, H., Huang, Z., Zhao, P., Zhou, X., 2023. Sequential recommendation with diffusion models. CoRR, abs/2304.04541.

Feng, Y., Lv, F., Shen, W., Wang, M., Sun, F., Zhu, Y., Yang, K., 2019. Deep session interest network for click-through rate prediction. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019. pp. 2301–2307.

Fuest, M., Ma, P., Gui, M., Fischer, J.S., Hu, V.T., Ommer, B., 2024. Diffusion models and representation learning: A survey. CoRR, abs/2407.00783.

Gholami, E., Motamedi, M., Aravindakshan, A., 2022. Parsrec: Explainable personalized attention-fused recurrent sequential recommendation using session partial actions. In: ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 454–464.

Guo, N., Cheng, H., Liang, Q., Chen, L., Han, B., 2024. Integrating large language models with graphical session-based recommendation. CoRR, abs/2402.16539.

Guo, L., Zhou, S., Tang, H., Zheng, X., Luo, Y., 2025. Multi-behavior hypergraph contrastive learning for session-based recommendation. IEEE Trans. Knowl. Data Eng. 37 (3), 1325–1338.

Gupta, M., Gupta, P., Vig, L., 2024. Guided diffusion-based counterfactual augmentation for robust session-based recommendation. CoRR, abs/2410.21892.

Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D., 2016. Session-based recommendations with recurrent neural networks. In: International Conference on Learning Representations. pp. 1–10.

Ho, J., Jain, A., Abbeel, P., 2020. Denoising diffusion probabilistic models. In: Conference on Neural Information Processing Systems.

Hou, Y., Hu, B., Zhang, Z., Zhao, W.X., 2022. CORE: simple and effective session-based recommendation within consistent representation space. In: International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1796–1801.

Hou, Y., zhang, D., Wu, J., Feng, X., 2024. A comprehensive survey on Kolmogorov arnold networks (KAN). CoRR, abs/2407.11075.

Huang, L., Guo, H., Peng, L., Zhang, L., Wang, X., Wang, D., Wang, S., Wang, J., Wang, L., Chen, S., 2025. SessionRec: Next session prediction paradigm for generative sequential recommendation. CoRR, abs/2502.10157.

Huberman-Spiegelglas, I., Kulikov, V., Michaeli, T., 2024. An edit friendly DDPM noise space: Inversion and manipulations. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12469–12478.

Jalan, R., Prakash, T., Pedanekar, N., 2024. LLM-BRec: Personalizing session-based social recommendation with LLM-BERT fusion framework. In: The Second Workshop on Generative Information Retrieval.

Kang, W., McAuley, J.J., 2018. Self-attentive sequential recommendation. In: IEEE International Conference on Data Mining. pp. 197–206.

Kingma, D., Salimans, T., Poole, B., Ho, J., 2021. Variational diffusion models. In: Conference on Neural Information Processing Systems, vol. 34, pp. 21696–21707.

Lenc, K., Vedaldi, A., 2019. Understanding image representations by measuring their equivariance and equivalence. Int. J. Comput. Vis. 127 (5), 456–476.

Li, A., Cheng, Z., Liu, F., Gao, Z., Guan, W., Peng, Y., 2023a. Disentangled graph neural networks for session-based recommendation. IEEE Trans. Knowl. Data Eng. 35 (8), 7870–7882.

Li, J., Deng, K., Li, J., Ren, Y., 2025a. Session-oriented fairness-aware recommendation via dual temporal convolutional networks. IEEE Trans. Knowl. Data Eng. 37 (2), 923–935.

Li, W., Huang, R., Zhao, H., Liu, C., Zheng, K., Liu, Q., Mou, N., Zhou, G., Lian, D., Song, Y., Bao, W., Yu, E., Ou, W., 2025b. DimeRec: A unified framework for enhanced sequential recommendation via generative diffusion models. In: ACM International Conference on Web Search and Data Mining. pp. 726–734.

Li, Q., Ma, H., Jin, W., Ji, Y., Li, Z., 2024a. Multi-interest network with simple diffusion for multi-behavior sequential recommendation. In: SIAM International Conference on Data Mining. pp. 734–742.

Li, J., Ren, P., Chen, Z., Ren, Z., Lian, T., Ma, J., 2017. Neural attentive session-based recommendation. In: ACM on Conference on Information and Knowledge Management. pp. 1419–1428.

Li, Z., Sun, A., Li, C., 2024c. DiffuRec: A diffusion model for sequential recommendation. ACM Trans. Inf. Syst. 42 (3), 66:1–66:28.
Li, Z., Wang, X., Yang, C., Yao, L., McAuley, J.J., Xu, G., 2023b. Exploiting explicit and implicit item relationships for session-based recommendation. In: ACM International Conference on Web Search and Data Mining. pp. 553–561.
Li, Z., Yang, C., Chen, Y., Wang, X., Chen, H., Xu, G., Yao, L., Sheng, M., 2025c. Graph and sequential neural networks in session-based recommendation: A survey. ACM Comput. Surv. 57 (2), 40:1–40:37.
Li, Y., Yu, Z., He, G., Shen, Y., Li, K., Sun, X., Lin, S., 2024b. SPD-DDPM: denoising diffusion probabilistic models in the symmetric positive definite space. In: AAAI Conference on Artificial Intelligence. pp. 13709–13717.
Lin, R., Liu, C., Zhong, H., Yuan, C., Chen, G., Jiang, Y., Tang, Y., 2025. Motif and supernode-enhanced gated graph neural networks for session-based recommendation. Neural Netw. 187, 107406.
Lin, J., Liu, J., Zhu, J., Xi, Y., Liu, C., Zhang, Y., Yu, Y., Zhang, W., 2024. A survey on diffusion models for recommender systems. CoRR, abs/2409.05033.
Liu, Z., Fan, Z., Wang, Y., Yu, P.S., 2021. Augmenting sequential recommendation with pseudo-prior items via reversely pre-training transformer. In: International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1608–1612.
Liu, C., Lin, J., Wang, J., Liu, H., Caverlee, J., 2024. Mamba4Rec: Towards efficient sequential recommendation with selective state space models. CoRR, abs/2403.03900.
Liu, Z., Liu, Q., Wang, Y., Wang, W., Jia, P., Wang, M., Liu, Z., Chang, Y., Zhao, X., 2025b. SIGMA: selective gated mamba for sequential recommendation. In: AAAI Conference on Artificial Intelligence. pp. 12264–12272.
Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljacic, M., Hou, T.Y., Tegmark, M., 2025c. KAN: Kolmogorov-arnold networks. In: International Conference on Learning Representations.
Liu, Q., Yan, F., Zhao, X., Du, Z., Guo, H., Tang, R., Tian, F., 2023. Diffusion augmentation for sequential recommendation. In: ACM International Conference on Information and Knowledge Management. pp. 1576–1586.
Liu, Q., Zeng, Y., Mokhosi, R., Zhang, H., 2018. STAMP: short-term attention/memory priority model for session-based recommendation. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1831–1839.
Liu, S., Zhang, A., Hu, G., Qian, H., Chua, T., 2025a. Preference diffusion for recommendation. In: International Conference on Learning Representations.
Lobashev, A., Guskov, D., Larchenko, M.A., Tamm, M.V., 2025. Hessian geometry of latent space in generative models. CoRR, abs/2506.10632.
Lv, M., Liu, X., Xu, Y., 2025. Dynamic multi-interest graph neural network for session-based recommendation. In: AAAI Conference on Artificial Intelligence. pp. 12328–12336.
Ma, N., Tong, S., Jia, H., Hu, H., Su, Y., Zhang, M., Yang, X., Li, Y., Jaakkola, T.S., Jia, X., Xie, S., 2025. Inference-time scaling for diffusion models beyond scaling denoising steps. CoRR, abs/2501.09732.
Ma, H., Xie, R., Meng, L., Chen, X., Zhang, X., Lin, L., Kang, Z., 2024a. Plug-in diffusion model for sequential recommendation. In: AAAI Conference on Artificial Intelligence. pp. 8886–8894.
Ma, H., Xie, R., Meng, L., Yang, Y., Sun, X., Kang, Z., 2024b. SeeDRec: Sememe-based diffusion for sequential recommendation. In: International Joint Conference on Artificial Intelligence. pp. 2270–2278.
Mao, W., Liu, S., Liu, H., Liu, H., Li, X., Hu, L., 2025. Distinguished quantized guidance for diffusion-based sequence recommendation. In: International World Wide Web Conference. pp. 425–435.
Molina, A., Schramowski, P., Kersting, K., 2020. Padé activation units: End-to-end learning of flexible activation functions in deep networks. In: International Conference on Learning Representations.
Ng, A.Y., Jordan, M.I., 2001. On discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes. In: Conference on Neural Information Processing Systems. pp. 841–848.
Nichol, A.Q., Dhariwal, P., 2021. Improved denoising diffusion probabilistic models. In: International Conference on Machine Learning, vol. 139, pp. 8162–8171.
Niu, Y., Xing, X., Jia, Z., Liu, R., Xin, M., 2025. Implicit local-global feature extraction for diffusion sequence recommendation. Eng. Appl. Artif. Intell. 139, 109471.
Niu, Y., Xing, X., Jia, Z., Liu, R., Xin, M., Cui, J., 2024. Diffusion recommendation with implicit sequence influence. In: International World Wide Web Conference. pp. 1719–1725.
Pan, Z., Cai, F., Chen, W., Chen, H., 2022b. Graph co-attentive session-based recommendation. ACM Trans. Inf. Syst. 40 (4), 67:1–67:31.
Pan, Z., Cai, F., Chen, W., Chen, C., Chen, H., 2022a. Collaborative graph learning for session-based recommendation. ACM Trans. Inf. Syst. 40 (4), 72:1–72:26.
Pan, Z., Cai, F., Chen, W., Chen, H., de Rijke, M., 2020. Star graph neural networks for session-based recommendation. In: ACM International Conference on Information and Knowledge Management. pp. 1195–1204.
Peintner, A., Mohammadi, A.R., Zangerle, E., 2023. SPARE: shortest path global item relations for efficient session-based recommendation. In: ACM Conference on Recommender Systems. pp. 58–69.
Qi, Z., Bai, L., Xiong, H., Xie, Z., 2024. Not all noises are created equally:Diffusion noise selection and optimization. CoRR, abs/2407.14041.
Qiao, S., Zhou, W., Wen, J., Gao, C., Luo, Q., Chen, P., Li, Y., 2025. Multi-view intent learning and alignment with large langue models for session-based recommendation. ACM Trans. Inf. Syst..
Qiu, R., Huang, Z., Chen, T., Yin, H., 2022. Exploiting positional information for session-based recommendation. ACM Trans. Inf. Syst. 40 (2), 35:1–35:24.
Qiu, R., Huang, Z., Li, J., Yin, H., 2020. Exploiting cross-session information for session-based recommendation with graph neural networks. ACM Trans. Inf. Syst. 38 (3), 22:1–22:23.
Qu, Y., Nobuhara, H., 2025. Intent-aware diffusion with contrastive learning for sequential recommendation. CoRR, abs/2504.16077.
Qu, H., Zhang, Y., Ning, L., Fan, W., Li, Q., 2024. SSD4rec: A structured state space duality model for efficient sequential recommendation. CoRR, abs/2409.01192.
Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B., 2022. High-resolution image synthesis with latent diffusion models. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10674–10685.
Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, vol. 9351, pp. 234–241.
Sahoo, P., Singh, A.K., Saha, S., Jain, V., Mondal, S., Chadha, A., 2024. A systematic survey of prompt engineering in large language models: Techniques and applications. CoRR, abs/2402.07927.
Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J., 2001. Item-based collaborative filtering recommendation algorithms. In: International World Wide Web Conference. pp. 285–295.
Song, J., Meng, C., Ermon, S., 2021. Denoising diffusion implicit models. In: International Conference on Learning Representations.
Su, J., Huang, Z., 2024. MLSA4Rec: Mamba combined with low-rank decomposed self-attention for sequential recommendation. CoRR, abs/2407.13135.
Sun, Z., Liu, H., Qu, X., Feng, K., Wang, Y., Ong, Y.S., 2024. Large language models for intent-driven session recommendations. In: International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 324–334.
Tan, Y.K., Xu, X., Liu, Y., 2016. Improved recurrent neural networks for session-based recommendations. In: Workshop on Deep Learning for Recommender Systems. pp. 17–22.
Tian, C., Tao, C., Dai, J., Li, H., Li, Z., Lu, L., Wang, X., Li, H., Huang, G., Zhu, X., 2024. ADDP: learning general representations for image recognition and generation with alternating denoising diffusion process. In: International Conference on Learning Representations.
Tumanyan, N., Geyer, M., Bagon, S., Dekel, T., 2023. Plug-and-play diffusion features for text-driven image-to-image translation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1921–1930.
Ulhaq, A., Akhtar, N., Pogrebna, G., 2022. Efficient diffusion models for vision: A survey. CoRR, abs/2210.09292.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. In: Conference on Neural Information Processing Systems. pp. 5998–6008.
Vinyals, O., Bengio, S., Kudlur, M., 2016. Order matters: Sequence to sequence for sets. In: International Conference on Learning Representations.
Wang, S., Cao, L., Wang, Y., Sheng, Q.Z., Orgun, M.A., Lian, D., 2022a. A survey on session-based recommender systems. ACM Comput. Surv. 54 (7), 154:1–154:38.
Wang, Z., Chen, C., Zhang, K., Lei, Y., Li, W., 2018. Variational recurrent model for session-based recommendation. In: ACM International Conference on Information and Knowledge Management. pp. 1839–1842.
Wang, F., Gao, X., Chen, Z., Lyu, L., 2023a. Contrastive multi-level graph neural networks for session-based recommendation. IEEE Trans. Multimed. 25, 9278–9289.
Wang, Y., He, X., Zhu, S., 2024b. EchoMamba4Rec: Harmonizing bidirectional state space models with spectral filtering for advanced sequential recommendation. CoRR, abs/2406.02638.
Wang, Y., Javari, A., Balaji, J., Shalaby, W., Derr, T., Cui, X., 2024c. Knowledge graph-based session recommendation with session-adaptive propagation. In: International World Wide Web Conference. pp. 264–273.
Wang, Y., Liu, Z., Wang, Y., Zhao, X., Chen, B., Guo, H., Tang, R., 2024d. Diff-MSR: A diffusion model enhanced paradigm for cold-start multi-scenario recommendation. In: ACM International Conference on Web Search and Data Mining. pp. 779–787.
Wang, Y., Siegel, J.W., Liu, Z., Hou, T.Y., 2025. On the expressiveness and spectral bias of KANs. In: International Conference on Learning Representations.
Wang, X., Wang, S., Ding, Y., Li, Y., Wu, W., Rong, Y., Kong, W., Huang, J., Li, S., Yang, H., Wang, Z., Jiang, B., Li, C., Wang, Y., Tian, Y., Tang, J., 2024a. State space model for new-generation network alternative to transformers: A survey. CoRR, abs/2404.09516.
Wang, Z., Wei, W., Cong, G., Li, X., Mao, X., Qiu, M., 2020. Global context enhanced graph neural networks for session-based recommendation. In: International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 169–178.
Wang, Z., Wei, W., Feng, S., Mao, X., Qiu, M., Chen, D., Fang, R., 2024e. Exploiting group-level behavior pattern for session-based recommendation. IEEE Trans. Knowl. Data Eng. 36 (1), 152–166.
Wang, W., Xu, Y., Feng, F., Lin, X., He, X., Chua, T., 2023b. Diffusion recommender model. In: International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 832–841.

Wang, Y., Zhang, H., Liu, Z., Yang, L., Yu, P.S., 2022b. ContrastVAE: Contrastive variational AutoEncoder for sequential recommendation. In: ACM International Conference on Information and Knowledge Management. pp. 2056–2066.

Wei, T., Fang, Y., 2025. Diffusion models in recommendation systems: A survey. CoRR, abs/2501.10548.

Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., Tan, T., 2019. Session-based recommendation with graph neural networks. In: AAAI Conference on Artificial Intelligence. pp. 346–353.

Wu, Z., Wang, X., Chen, H., Li, K., Han, Y., Sun, L., Zhu, W., 2023. Diff4Rec: Sequential recommendation with curriculum-scheduled diffusion augmentation. In: ACM International Conference on Multimedia. pp. 9329–9335.

Xiao, W., Wang, H., Zhou, Q., Wang, Q., 2025. SS4rec: Continuous-time sequential recommendation with state space models. CoRR, abs/2502.08132.

Xie, Z., Liu, C., Zhang, Y., Lu, H., Wang, D., Ding, Y., 2021. Adversarial and contrastive variational autoencoder for sequential recommendation. In: International World Wide Web Conference. pp. 449–459.

Xie, W., Zhou, R., Wang, H., Shen, T., Chen, E., 2024. Bridging user dynamics: Transforming sequential recommendations with Schrödinger bridge and diffusion models. In: ACM International Conference on Information and Knowledge Management. pp. 2618–2628.

Xu, J., Chen, Z., Li, J., Yang, S., Wang, W., Hu, X., Ngai, E.C.H., 2024. Fourierkan-GCF: Fourier Kolmogorov-arnold network - an effective and efficient feature transformation for graph collaborative filtering. CoRR, abs/2406.01034.

Yang, J., Li, Y., Zhao, J., Wang, H., Ma, M., Ma, J., Ren, Z., Zhang, M., Xin, X., Chen, Z., Ren, P., 2024b. Uncovering selective state space model's capabilities in lifelong sequential recommendation. CoRR, abs/2403.16371.

Yang, X., Wang, X., 2024. Kolmogorov-arnold transformer. CoRR, abs/2409.10594.

Yang, Z., Wu, J., Wang, Z., Wang, X., Yuan, Y., He, X., 2023. Generate what you prefer: Reshaping sequential recommendation via guided diffusion. In: Conference on Neural Information Processing Systems.

Yang, H., Yuan, J., Yang, S., Xu, L., Yuan, S., Zeng, Y., 2024a. A new creative generation pipeline for click-through rate with stable diffusion model. In: International World Wide Web Conference. pp. 180–189.

Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., Zhang, W., Cui, B., Yang, M., 2024c. Diffusion models: A comprehensive survey of methods and applications. ACM Comput. Surv. 56 (4), 105:1–105:39.

Yin, Z., Han, K., Wang, P., Zhu, X., 2024. H3GNN: hybrid hierarchical HyperGraph neural network for personalized session-based recommendation. ACM Trans. Inf. Syst. 42 (3), 63:1–63:30.

Yuan, F., Karatzoglou, A., Arapakis, I., Jose, J.M., He, X., 2019. A simple convolutional generative network for next item recommendation. In: ACM International Conference on Web Search and Data Mining. pp. 582–590.

Zhang, J., Fan, Y., Cai, K., Wang, K., 2025. Kolmogorov-arnold Fourier networks. CoRR, abs/2502.06018.

Zhang, X., Li, B., Jin, B., 2024a. Denoising long- and short-term interests for sequential recommendation. In: SIAM International Conference on Data Mining. pp. 544–552.

Zhang, Q., Tao, M., Chen, Y., 2023. gDDIM: Generalized denoising diffusion implicit models. In: International Conference on Learning Representations.

Zhang, X., Xu, B., Ma, F., Li, C., Yang, L., Lin, H., 2024b. Beyond co-occurrence: Multi-modal session-based recommendation. IEEE Trans. Knowl. Data Eng. 36 (4), 1450–1462.

Zhao, J., Li, H., Qu, L., Zhang, Q., Sun, Q., Huo, H., Gong, M., 2022. DCFGAN: an adversarial deep reinforcement learning framework with improved negative sampling for session-based recommender systems. Inform. Sci. 596, 222–235.

Zhao, W.X., Mu, S., Hou, Y., Lin, Z., Chen, Y., Pan, X., Li, K., Lu, Y., Wang, H., Tian, C., Min, Y., Feng, Z., Fan, X., Chen, X., Wang, P., Ji, W., Li, Y., Wang, X., Wen, J., 2021. RecBole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In: ACM International Conference on Information and Knowledge Management. pp. 4653–4664.

Zhao, S., Song, J., Ermon, S., 2019. Infovae: Balancing learning and inference in variational autoencoders. In: AAAI Conference on Artificial Intelligence. pp. 5885–5892.

Zhao, J., Wang, W., Xu, Y., Sun, T., Feng, F., Chua, T., 2024. Denoising diffusion recommender model. In: International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1370–1379.

Zheng, C., Wu, G., Bao, F., Cao, Y., Li, C., Zhu, J., 2023. Revisiting discriminative vs. Generative classifiers: Theory and implications. In: International Conference on Machine Learning, vol. 202, pp. 42420–42477.

Zhong, T., Wen, Z., Zhou, F., Trajcevski, G., Zhang, K., 2020. Session-based recommendation via flow-based deep generative networks and Bayesian inference. Neurocomputing 391, 129–141.

Zhou, F., Wen, Z., Zhang, K., Trajcevski, G., Zhong, T., 2019. Variational session-based recommendation using normalizing flows. In: International World Wide Web Conference. pp. 3476–3475.

Zhu, X., Li, L., Liu, W., Luo, X., 2024. Multi-level sequence denoising with cross-signal contrastive learning for sequential recommendation. Neural Netw. 179, 106480.

Zhuo, X., Qian, S., Hu, J., Dai, F., Lin, K., Wu, G., 2024. Multi-hop multi-view memory transformer for session-based recommendation. ACM Trans. Inf. Syst. 42 (6), 144:1–144:28.