Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Dual-Level Attention for Feature-Aware Sequential Recommendation

[redacted] [redacted]

220242888@seu.edu.cn

Southeast University

Research Article

Keywords: Sequential recommendation, Feature-aware recommendation, Attention network, Consumer heterogeneity

Posted Date: November 25th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11851855/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Dual-Level Attention for Feature-Aware Sequential Recommendation

Yixing Wang[1*]

[1*]School of Economics and Management, Southeast University, Nanjing, China.

Corresponding author(s). E-mail(s): 220242888@seu.edu.cn;

Abstract

Sequential recommendation models mainly rely on item IDs and historical interaction sequences to predict users' next-item interaction. However, item identity alone cannot fully describe product characteristics, while users with similar interaction histories may still exhibit different preferences because of their heterogeneous characteristics. To address these limitations, this study proposes a dual-level attention framework for feature-aware sequential recommendation. The model first employs a sequence-level cross-attention fusion module, where item-ID embeddings serve as queries and projected item-feature representations serve as keys and values. This mechanism allows each item position to retrieve relevant attribute information from the historical feature sequence, thereby integrating collaborative identity signals with sequence-level product feature patterns. Then a user-conditioned self-attention module that incorporates user features into the query-key interaction process is applied, allowing user characteristics to influence the modeling of sequential dependencies. In addition, a user residual fusion layer is used to preserve stable user-level preference information. Experiments on the MovieLens-1M and KuaiRec datasets show that the proposed model achieves the highest HR@10 on both datasets and the highest NDCG@10 on MovieLens-1M. Ablation experiments further demonstrate the complementary contributions of item feature fusion and user-conditioned attention, while the attention-head analysis indicates that two attention heads provide the best performance under the current setting. The proposed framework provides a useful approach for jointly modeling consumer behavior, product features, and user heterogeneity, with potential applications in personalized recommendation and customer preference analysis.

Keywords: Sequential recommendation, Feature-aware recommendation, Attention network, Consumer heterogeneity

1 Introduction

Over the past decades, digital platforms have fundamentally reshaped consumer decision-making and service delivery. E-commerce, online travel, and social media systems continuously generate large-scale behavioral data, including click streams, browsing histories, and transaction sequences. Leveraging such data to infer user intent and predict future actions has become central to modern intelligent services, making sequential recommendation a core research problem with broad practical impact[20].

Recent surveys consistently show that Sequential recommendation(SR) models have progressed from ID-based modeling toward side-information-aware modeling, including categorical and numerical features, context signals, and multi-modal features [14]. Transformer-based architectures, represented by SASRec [9] and BERT4Rec [16], have significantly improved sequential recommendation via self-attention mechanism. Based on these studies, recent advances have extended Transformer-based SR models toward feature-aware recommendation by incorporating side information, including item features and contextual signals.

Despite substantial progress, two challenges still remain insufficiently addressed. First, in real-world scenarios, items with similar co-occurrence patterns may correspond to different utility profiles because of their differences in quality, price, promotion status, and other features [1]. On the other hand, purchase patterns can still be different for heterogeneous customers, even when they share a similar purchase history. For example, two users may have similar movie-watching histories consisting of animation and family movies. However, a younger student and an older user with family-related preferences may interpret this similar sequence differently. The former may prefer entertaining and popular animation movies, while the latter may be more interested in family-oriented or classic movies. Therefore, similar item sequences do not necessarily imply identical future preferences, and user-side heterogeneity, or user features, should be well considered when modeling sequential dependencies. Most feature-aware sequential recommendation methods focus on enriching item representations through side information, while the influence of user-side heterogeneity on dependency learning remains limited. Also, existing personalization approaches typically inject user information after sequence encoding or through separate preference modules. As a result, user features rarely participate directly in the formation of attention structures that determine how historical interactions are weighted, ignoring the fact that user features help explain why users interact with these items and how similar sequences may lead to different future preferences.

Meanwhile, Although feature-aware sequential recommendation has achieved promising results, existing feature-aware sequential recommendation models usually fuse item features through concatenation, feature aggregation, or decoupled attention mechanisms. Such approaches do enrich item representations but generally treat item identities and item features as parallel information sources, but only few studies explicitly model the cross-sequence interactions between historical item-ID representations and their corresponding projected feature representations before sequential dependency learning. Consequently, the interaction between item identity semantics and feature semantics remains insufficiently explored.

To address these limitations, we enhance sequential recommendation at the item-representation, sequence-modeling, and preference-fusion levels. **Cross-attention module for item features** selectively aligns item identities with relevant product information, producing representations that combine collaborative and content signals. **User-conditioned self-attention module** then incorporates user characteristics into dependency modeling, allowing similar behavioral sequences to be interpreted differently for heterogeneous users. Finally, **User residual fusion** directly adds the projected user-attribute representation to the conditioned sequential representation, preserving both behavior-dependent dynamic preference and profile-based stable preference. Together, these components are expected to improve the informativeness of item representations, the personalization of sequential dependencies, and the discriminative capacity of the final user representation.

Eventually, the main contributions of our work can be summarized as follows:

1. We propose a personalized feature-aware sequential recommendation framework that jointly models item identities, product attributes, explicit user attributes, and sequential interaction patterns within a unified architecture.

2. We develop a sequence-level product-feature cross-attention module, in which item-ID embeddings act as queries and projected item-feature representations act as keys and values. The module enables each item position to retrieve relevant attribute information from different positions in the historical sequence, thereby integrating collaborative identity signals with sequence-level product-feature patterns.

3. We introduce a user-conditioned self-attention mechanism to personalize the modeling of sequential dependencies according to user attributes. In addition, a residual-style late-fusion mechanism is designed to preserve the direct contribution of user attributes to candidate-item preference. Together, these mechanisms capture both the conditioning effect of user characteristics on historical behavior and their direct profile-based compatibility with candidate items.

4. We conduct extensive experiments, including comparisons with representative baselines, ablation studies, and hyperparameter analyses. The results demonstrate the effectiveness of the proposed framework and quantify the contribution of major components to next-item prediction performance.

2 Related Works

Sequential recommendation(SR) seeks to predict the next item user probably interact with, by modeling temporal dependency and preference evolution in user behavior sequences. Existing methods have progressed from traditional neural sequence encoders to Transformer-based architectures, and recent advances increasingly focus on integrating side information beyond item IDs.

2.1 Transformer-based Sequential Recommendation

Sequential recommendation aims to predict the next item that a user is likely to interact with by modeling temporal dependencies in historical behavior sequences. Early neural sequential recommendation models mainly relied on recurrent neural networks (RNNs) and convolutional neural networks (CNNs). Representative studies include

GRU4Rec [6], which employs gated recurrent units to capture sequential dependencies in session-based recommendation, and DREAM [25], which utilizes recurrent architectures for next-basket prediction. Convolution-based approaches such as CASER [17] model user interaction sequences as temporal feature maps and extract local behavioral patterns through convolutional filters. Although these methods demonstrated promising performance, they often struggle to effectively capture long-range dependencies due to the inherent limitations of recurrent and convolutional structures.

The success of Transformer architectures in natural language processing has inspired a new generation of sequential recommendation models. Transformer-based methods employ self-attention mechanisms to directly model dependencies among all positions in a sequence, enabling more effective learning of long-range behavioral patterns. SASRec [9] was among the earliest Transformer-based sequential recommendation models, introducing causal self-attention to adaptively identify historical interactions relevant to next-item prediction. Subsequently, BERT4Rec [16] adopted bidirectional self-attention and masked item prediction objectives to further enhance sequence representation learning.

Building upon these foundations, recent studies have extended Transformer architectures to incorporate additional contextual signals and personalized modeling strategies. For example, PDTN [4]combines personalized feature extraction modules with dual Transformer encoders to improve user preference modeling. PAUDRec [22] integrates user preference and popularity-aware components within a Transformer framework to mitigate popularity bias. Other approaches, including MSMT-LSI [19], DTCoSR [12], and SeKeBERT4Rec [18], further enhance sequential recommendation through temporal modeling, collaborative signals, or auxiliary contextual information. Also, some personalization-oriented sequence models like HGN [13] and SSE-PT [23] emphasize that user-specific preference structure should be reflected during sequence encoding, not only in post-hoc fusion.

Despite their strong performance, most Transformer-based sequential recommendation models primarily focus on improving sequence representation learning and dependency modeling through model design. User-related contextual information is often incorporated through separate personalization modules or auxiliary representations, rather than directly participating in the formation of attention structures. Consequently, how to integrate user features into attention-based dependency learning remains an open research problem. This limitation motivates the user-conditioned attention mechanism proposed in our work.

2.2 Feature-aware Models in Sequential Recommendation

To overcome the limitations of ID-only sequence modeling, recent studies increasingly incorporate side information such as item features, contextual signals, and behavioral features into sequential recommendation models. The underlying motivation is that item IDs alone only provide limited semantic information, especially in sparse and cold-start scenarios, whereas auxiliary features can provide richer descriptions of item features and user preferences, so as to improve the effectiveness of sequential recommendation.

Early feature-aware approaches mainly focused on learning separate representations for item identities and item features. FDSA [26] introduced a feature-level self-attention architecture that models item transitions and feature transitions separately before combining them for prediction. S3-Rec [27]further strengthened feature representation learning through self-supervised objectives designed around item features and sequence semantics.

Subsequent studies explored more sophisticated feature fusion mechanisms within Transformer architectures. NOVA[10] proposed a non-invasive side-information fusion strategy that guides attention learning while minimizing disturbances to item representations. DIF-SR [24]introduced decoupled side-information attention, where different feature types are modeled through separate attention branches and later aggregated. These methods demonstrate that incorporating side information like item features directly into attention mechanisms can significantly improve recommendation performance.

More recently, CARCA [15]employed cross-attention to model interactions between historical items and candidate items, allowing contextual information and feature information to participate in the final matching process. ASIF [21] investigated the information invasion problem in side-information fusion and proposed representation alignment and homogeneous information extraction mechanisms to improve the semantic consistency between item identities and side information. FIDS [28] proposed the Feature Interaction Dual Self-attention Network to further improve feature-aware sequential recommendation. Different from FDSA, which mainly uses vanilla attention to combine heterogeneous features, FIDS emphasizes that interactions among item features should also be modeled. It first applies self-attention to learn higher-order feature interactions within each item, and then uses two independent self-attention networks to capture transition patterns from both the item sequence and the integrated feature sequence. FANGIM [8] treats item features as an additional way for user group clustering and enhance sequential recommendation through interest groups. AttrFormer [11] introduces item relations, which measures the similarity between items according to item features, and proposes relation-augmented multi-head attention to enable attention to related user sequences. Not limited to the fusion of item features, recent works like SASRec-cross [2] and DCUPSRec [7] also explore how to integrate user information into the model through simple fusion.

Although these methods have significantly improved the utilization of side information, they generally focus on feature aggregation, representation alignment, or item-to-item interaction modeling. Item identities and item features are still largely treated as parallel information sources. However, the interaction between historical item identities and historical feature patterns remains insufficiently explored. In particular, few studies explicitly model how historical item representations attend to feature information across the interaction sequence before dependency learning. To address this limitation, our work introduces a feature-aware cross-attention fusion module, where item ID embeddings act as queries and item features serve as keys and values. This design enables adaptive feature retrieval conditioned on item identity, producing feature-enhanced item representations that are subsequently used for sequential dependency modeling. The differences between our model and several representative works are shown in Table 1

Table 1 Comparison on Different modules from Related Works

Method	Item features	User context	Fusion stage	Attention Q/K Modulation
SASRec	No	No	ID-only sequence encoding	No
FDSA	Yes	No	Feature-level attention	No
S3-Rec	Yes	No	Self-supervised pre-training	No
SSE-PT	No	No	Personalized Transformer	No
NOVA	Yes	No	Side info guides attention non-invasively	Side info affects attention
DIF-SR	Yes	Partial	Decoupled side information attention layer	For decoupled side info
CARCA	Yes	No	Profile and target cross-attention and scoring	No
Our model	Yes	Explicit user features	Identity-conditioned cross attention	Q/K conditioning

To conclude, our study aims to build an attention mechanism model that further integrates multi-dimensional information such as user-related features, based on existing Transformer-based sequence recommendation models. Under practical application scenarios, it explores methods to systematically incorporate contextual information including user demographic features and interaction states, thereby addressing the existing gap in modeling user background information within current models.

3 Proposed Model

This section presents our proposed dual-level attention framework for feature-aware sequential recommendation. The model is designed to capture how sequential behavior, product-side features, and user-side features jointly influence users' future interactions. Specifically, historical item sequences reflect users' dynamic preference evolution, item features provide product-side information that affects consumer evaluation and choice, and user features represent consumer heterogeneity in preference tendencies and decision criteria. Based on this consideration, the proposed model integrates item feature fusion, user-conditioned sequential dependency learning, and user residual preference fusion into a unified architecture.

The remainder of this section is organized as follows. Section 3.1 defines the feature-aware sequential recommendation problem and introduces the main input variables. Section 3.2 describes the overall model architecture and explains how item features and user features are incorporated into the recommendation process. Section 3.3 presents the prediction layer and optimization objective for next-item recommendation.

3.1 Problem Definition

Our sequential recommendation problem consists of the following elements:

$\mathcal{U} = \{1, 2, ..., U\}$ The set of users, U stands for the total number of users.

$\mathscr{I} = \{1, 2, ..., I\}$ The set of items, I stands for the total number of items.

$\mathscr{F}_u = \{F_{u_1}, F_{u_2}, ..., F_{u_N}\}$ Features of user u, N stands for the fixed number of user features.

$\mathscr{F}_i = \{F_{i_1}, F_{i2}, ..., F_{i_M}\}$ Features of item i, M stands for the fixed number of item features.

$L \in \mathbb{N}$ Maximum sequence length.

$S_u = (i_1, i_2, ..., i_N, \forall i_t \in \mathscr{I})$ Historical interaction sequence of user u with sequence length N.

Sequential recommendation can be viewed as a data-driven decision support tool for understanding consumer preference evolution and improving personalized service delivery. Consumer choices are not determined solely by past interactions; they are also shaped by product features, user features, and the fit between consumer needs and product offerings. Therefore, an effective recommendation model should not only learn behavioral patterns from historical sequences, but also explain how item-side features and user-side heterogeneity jointly affect future purchase or interaction decisions. Therefore in our model, apart from user interaction sequences S_u, we additionally incorporates user-side contextual feature vectors $\mathscr{F}_u$ and item-side feature vectors $\mathscr{F}_i$. Our model seeks to integrate sequence dynamics, user context, and item semantics to learn the prediction function:

$$P(i_{N+1}|S_u, \mathscr{F}_i, \mathscr{F}_u)$$

that estimates the probability of the next interacted item for user u.

In our model, the input sequence S_u is firstly transformed into a fixed-length sequence $s = \{s_1, s_2, s_3, ..., s_L\}$ to fit into the setting of our model. If the length of input sequence is greater than L, then the last L items will be kept. If the length of input sequence is smaller than L, then the zero padding will be applied to the sequence by filling in zero elements to the left until the length of sequence reaches L.

3.2 Model Architecture

The overall design of the proposed model is motivated by the fact that sequential recommendation is not only a problem of modeling item transition patterns, but also a problem of understanding how user preferences are shaped by both product-side features and user-side heterogeneity. Existing Transformer-based sequential recommendation models mainly focus on learning dependencies among historical item IDs. Although this design is effective for capturing behavioral patterns, it may ignore two important factors in real recommendation scenarios. First, items are not only identified by their IDs, but also characterized by explicit features such as category, price, brand, or content attributes. Second, users with similar interaction sequences may still have different preferences due to their demographic backgrounds, behavioral contexts, or long-term preference tendencies. Therefore, the proposed framework is designed to address three related questions: 1. How can item-ID representations be aligned with

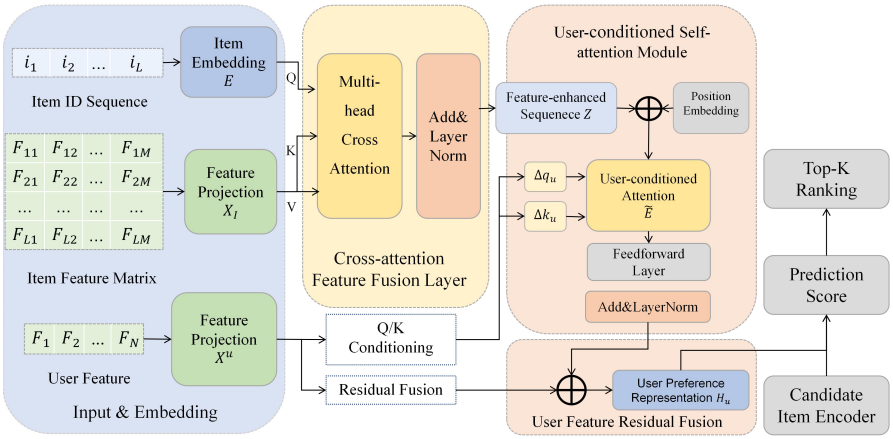

Fig. 1 Model Architecture

product-feature patterns distributed across historical interactions? 2. How can user-side information influence the learning of sequential dependencies rather than being added only after sequence encoding? 3. How can the model preserve stable user preference information while still maintaining the sequential patterns learned from historical interactions?

The overall architecture of our model is shown in Figure 1, which is centered around a Transformer encoder and features a five-layer structure: an embedding layer, a cross-attention feature fusion layer, a user-conditioned self-attention module, a user feature fusion layer, and prediction layer.

3.2.1 Embedding Layer

The embedding layer is to map discrete item IDs and other multidimensional features into a unified hidden vector space R^d, where d is the hidden dimension.

Item ID embedding layer. This layer constructs an embedding matrix $E \in \mathbb{R}^{(I+1) \times d}$ for one-hot item ID input, and the corresponding embedding vector of zero-padding value is set as a zero vector to prevent invalid information interference.

Position encoding layer. Since the attention model itself fails to recognize the temporal information of the elements within a sequence, position encoding is introduced following the pattern of previous studies. The encoding layer generates a learnable position vector $P \in \mathbb{R}^{L * d}$, which compensates for the lack of temporal information in the Transformer architecture. This position vector is then element-wise added to the item embedding vector.

Item feature projection. For the features vector $\mathscr{F}_i$ of item i, the multi-dimensional features of items are projected into hidden dimensions via a linear layer $W_I \in \mathbb{R}^{M \times d}$,

yielding item feature projection $X^I = W_I \times \mathscr{F}_i$.

User feature projection. A linear layer $W_u \in \mathbb{R}^{n \times d}$ maps a high-dimensional user static feature vector into hidden dimensions, aligning them with the dimensionality of sequence embedding and obtain user feature projection $X^u = W_u \times \mathscr{F}_u$

3.2.2 Cross-attention Feature Fusion Layer

The various features of items play a vital role in the decision making process of consumers, which means the sequence of consumer historical behaviors can also reflect how user preferences associated with item features. For the problem of sequential recommendation, leveraging the contextual information like item features can be critical to the final prediction. When it comes to the fusion of different inputs, common practice in previous studies is to concatenate those inputs into one vector, which can hardly model the differentiated preference to item features based on different consumer behaviors. Although these methods enrich item representations, they do not explicitly capture interactions between item-identity signals and product-feature patterns distributed across the historical sequence. Consumer histories may contain recurring attribute patterns that provide useful information beyond individual item identities. For example, the representation of a recently watched animation movie may be more strongly associated with the representations of earlier family-oriented movies than with those of unrelated action movies. Historical interactions contain not only item-identity information but also recurring product-feature patterns. However, simply adding or concatenating projected features with item-ID embeddings models each interaction largely in isolation. To capture the relationship between collaborative identity signals and attribute patterns distributed across the sequence, we introduce a sequence-level cross-attention fusion module (See Fig 2). To be specific, we use item ID embeddings as queries and item feature embeddings as keys and values. This design allows sequence-level cross-attention between the item-ID embedding sequence and the projected item-feature sequence. Each item-ID representation can retrieve relevant feature representations from different positions in the historical sequence, thereby integrating collaborative identity signals with sequence-level attribute patterns. Compared with static feature fusion, the cross-attention mechanism provides a more flexible and item-conditioned fusion process.

Cross-attention layer. Using item ID embedding E as Query matrix and item feature projection X^Ias Key and Value matrix, the inputs are fed into a multi-head attention module *MultiHead*(E, X^I, X^I) to enable adaptive attention of ID embedding to item features. *MultiHead*( ) can be written in the concatenation of the following attention functions:

$$Q_i, K_i, V_i = E\,W_i^Q, X^I\,W_i^K, X^I\,W_i^V, i = 1, 2, ..., h$$

Within the module, the attention weights $W \in \mathbb{R}^{B \times L \times L}$ can be represented as:

$$W_i = Softmax(\frac{Q_i K_i^T}{\sqrt{d/h}})$$

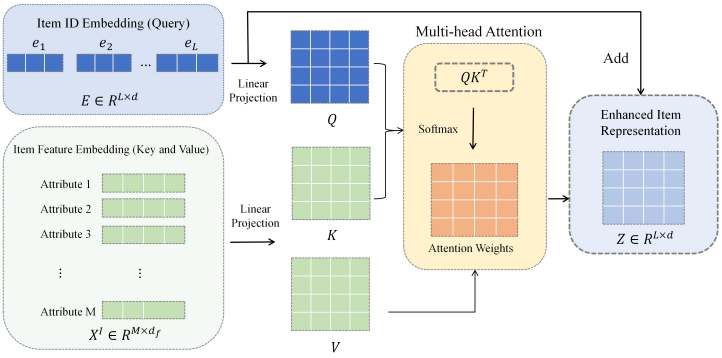

Fig. 2 Cross-attention Feature Fusion Layer

Every element $w_{i,j,k}$ in W stands for the attention weight between item ID j and item features k, which reflects how item features relate to the items within a sequence. The attention output of head i can be calculated as:

$$Attn_i = Softmax(\frac{Q_i K_i^T}{\sqrt{d/h}}) V_i$$

Where $softmax()$ stands for softmax function, h is the number of attention heads i and $\sqrt{d/h}$ is a scale factor that controls the variance of the dot-product logits and prevents the softmax distribution from being excessively concentrated. The final fused output is:

$$MultiHead(E\ ,\ X^I,\ X^I) = Concat(Attn_1, Attn_2, ..., Attn_h)$$

To avoid over-fitting, dropout rate is applied to attention layer. The final output of cross-attention layer can be represented as:

$$Z = LayerNorm(E + Dropout(MultiHead(E, X^I,\ X^I)))$$

where $LayerNorm()$ stands for layer normalization, E is Item embedding matrix. The computational complexity of the proposed cross-attention layer is $O(L^2 d)$, which is the same order as the self-attention operation used in Transformer-based recommendation models.

Compared to self-attention mechanism, which is widely used in Transformer-based sequential recommendation models, cross-attention mechanism mainly differs in its source of Query Q , Key K and Value V. To be specific, cross-attention mechanism takes one sequence as Query , and another sequence as Key and Value, while self-attention mechanism treats only one sequence as the source of both Query, Key and Value. In this way, the model is capable of analyzing the attention between two sequences. The module output Z can be interpreted as a feature-enhanced item sequence representation. Each position retains item identity semantics while injecting

feature information weighted by attention coefficients $Attn_i$. Therefore, the model does not assign a fixed global weight to each feature; instead, it learns context-dependent feature importance conditioned on sequential interactions.

3.2.3 User-conditioned Self-attention Module

To tackle the problem of personalization, our model designs a User-conditioned Self-attention module to model long-term temporal dependencies in user behavior sequences, while integrating user features into self-attention calculation. The module includes a user-conditioned self-attention layer and a point-wise feed-forward network layer.

Self-attention Block. The self-attention blocks serve as the critical module in our model for capturing long-term temporal dependencies in user behavior sequences and item associations within the sequence. Building upon the single-block self-attention and feed-forward network architecture of SASRec [9], the self-attention blocks introduces dual-mask optimization and feature-mask residual design tailored for sequence recommendation tasks. This approach can not only preserve the parallel processing advantage of self-attention but also resolves two key challenges in traditional Transformer-based sequence recommendation: future information leakage and interference from zero-filling invalid data.

The Self-attention blocks of our model include the following components:

Self-attention layer. Following the pattern of SASRec, position embedding P is added to enable the attention model to handle position information and we obtain the final input $\hat{E} = Z + P.$.

To accommodate the unidirectional temporal nature of sequence recommendations (where future user behavior is invisible to model) and the zero-padding feature in short sequences, our model employs a dual masking strategy combining causal masks with Padding masks in the self-attention layer:

$$\mathcal{M} = \mathcal{M}_{pad} \otimes \mathcal{M}_{causal}$$

Padding mask is denoted as $\mathcal{M}_{pad} \in \{0,1\}^L$, where valid positions in the sequence are assigned 1 and zero-filled positions are assigned 0. After dimension expansion, it is multiplied element-wise with the causal mask to globally mask invalid padding positions. Causal mask $\mathcal{M}_{causal} \in \{0,1\}^{L \times L}$ is a lower triangular matrix with the upper triangular elements being 0 to disable future location information and the lower triangle and diagonal being 1 to retain current and historical location information. This ensures the model learns temporal dependencies based solely on historical behavior, as well as preventing future information leakage. Through element-wise multiplication, combinatorial masking ultimately achieve attention interaction that retains only historically valid positions

Self-attention blocks capture dependencies across different positions in the sequence by using self-attention mechanism. The attention layer employs a multi-head mechanism. Input sequences are embedded into $\hat{E}$ and transformed through linear transformations into the query matrix Q, key matrix K, and value matrix V respectively. These are then split into multiple subspaces according to the number of attention heads, with independent attention weight calculations performed in each subspace. The outputs from all heads are finally concatenated and subjected to linear transformation, enabling multi-dimensional sequence dependency capture:

$$Q_i, K_i, V_i = \hat{E}\,W_i^Q, \hat{E}\,W_i^K, \hat{E}\,W_i^V, i = 1, 2, ..., h$$

$$Attn_i = Softmax(\frac{Q_i K_i^T}{\sqrt{d/h}} + \mathcal{M})\,V_i$$
$$MultiHead(Attn) = Concat(Attn_1, Attn_2, ..., Attn_h)$$

Where h is the number of attention heads, W_i^Q, W_i^K, W_i^V are the projection matrix of attention head i.

User-conditioned Attention. In our problem setting, every user is associated with a vector of user features and a sequence of history purchase. A natural question is whether user features determine the preference towards certain item features, and thus influence their final purchases, and whether those additional information can help achieve better sequential recommendation.

During the self-attention calculations above, sequence representation is entirely determined by similarities between historical interactions. This approach relies solely on interactions sequences themselves while overlooking the potential impact of user static features on dependency modeling. For recommendation tasks, this means that even when two users exhibit similar short-term interaction patterns, the model may still learn comparable dependency structures, failing to fully capture how user profiles modulate behavioral explanations.

To tackle the limitation, we introduce a new user-conditioned attention layer into the initial sequence encoder. The user-conditioned self-attention module is designed to incorporate user-specific context into the dependency learning process. Standard self-attention computes attention weights only based on interactions among historical items. However, users with similar behavior sequences may still have different preferences due to demographic features, behavioral states, or contextual factors. If user information is only fused after sequence encoding, it cannot directly affect how historical items are weighted during attention calculation. To address this issue, we inject user feature representations into the query and key transformations of self-attention. This allows user context to modulate the attention scores before sequence dependencies are aggregated. As a result, the model can learn personalized attention patterns, where the same historical item sequence may lead to different dependency structures for different users. Specifically, we project the user representation into the query and key spaces and use it to modulate the self-attention structure(see Fig 3).

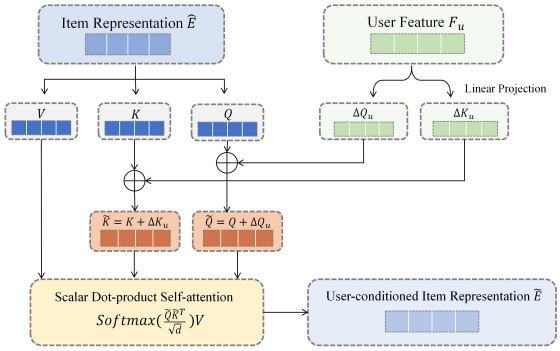

Fig. 3 User-conditioned Attention

Formally, to enable user information to directly influence attentional structures, we projects X^u into the space of Q and K, and obtain the adjustment bias:

$$\Delta q_u = W_q^u X^u, \Delta k_u = W_k^u X^u$$

Therefore, the conditioned user query and key are:

$$\tilde{Q}_i = Q_i + \Delta q_u$$

$$\tilde{K}_i = K_i + \Delta k_u$$

Which yields the final user-aware attention output:

$$Attn_i = Softmax(\frac{\tilde{Q}_i \tilde{K}_i^\top}{\sqrt{d/h}} + \mathcal{M}) V_i$$

Modifying value vectors V only affects the aggregated representations after attention computation. In contrast, conditioning query Q and key K vectors can directly influence attention score formation and dependency learning. Therefore in our model, user features are injected into the query-key interaction process rather than the value space. By replacing initial self-attention with our proposed user-conditioned attention, user information can directly influences how historical interactions are weighted and aggregated, enabling the model to capture user-specific sequential dependency patterns. Finally, by taking sequence embedding $\hat{E}_u$ as the input of self-attention layer, the outputs feature matrix are:

$$\tilde{E}_u = LayerNorm(\hat{E}_u + Dropout(MultiHead(\hat{E}_u, \hat{E}_u, \hat{E}_u)))$$

With residual connections and layer normalization similar to former steps.

Pointwise Feed-Forward Network layer. Although the output of previous self-attention layer is able to aggregate all previous items' embedding with adaptive

weights, the model fails to capture the non-linearity between the elements of input. Similar to other Transformer-based model, a feed-forward neural network consisting of two linear layers and one ReLU activation layer is applied to handle the problem:

$$E_u = FFN(\tilde{E_u}) = ReLU(W^{(2)}(W^{(1)}\tilde{E_u} + b^{(1)}) + b^{(2)})$$

We denote $H_u = F_{seq}(S_u|F_u)$ as the final output of user-conditioned self-attention module, which represents the dynamic sequential preference learned under the condition of user attributes.

3.2.4 User Residual Fusion Layer

Although user attributes have already been incorporated into the conditioned self-attention mechanism, their influence on the final prediction remains indirect. Specifically, conditioned attention mainly determines how historical interactions should be interpreted for a particular user, thereby capturing the interaction between user characteristics and sequential behavior. However, after multiple attention and feed-forward transformations, the direct preference information contained in user attributes may be weakened, and the sequence representation alone does not explicitly preserve the compatibility between the user profile and a candidate item. This limitation is particularly relevant when the observed interaction sequence is short, sparse, or insufficient to reflect the user's relatively stable preferences. Therefore, we further introduce user preference residual fusion layer to to preserve the contribution of relatively stable user characteristics and complete personalization. For each input of user features vector $\mathscr{F}_c$, a linear projection layer maps $\mathscr{F}_c \in \mathbb{R}^n$ to a hidden space consistent with sequence features:

$$X^u = W_u\mathscr{F}_u + b_u$$

Where $W_u \in \mathbb{R}^{n\times d}$ is linear projection matrix, $b_u \in \mathbb{R}^d$ is bias term and $X^u \in \mathbb{R}^d$ is projected user feature vector. The linear projection maps the encoded user feature vector into the same dimension as the sequential representation, thereby enabling element-wise residual fusion. The projected user feature representationX^u is then broadcast along the sequence dimension extended to obtain $X^u_{seq} \in \mathbb{R}^{L\times d}$. After that, by adopting residual connection method to integrate this extended matrix and the output of user-conditioned self-attention module H_u, we obtain:

$$Z_u = H_u + X^u_{seq}$$

Where Z_u is the final sequence embedding, which represents user-level preference that integrates the static information from users.

This operation can referred to as residual-style late fusion because it retains the sequence-based representation as the main pathway while introducing the projected user representation as a complementary component after sequence encoding. Unlike the typical function of residual connection which prevents gradient vanishing during the process of back propagation in deep neural network, we treat residual connection as an effective way for feature fusion. Given the candidate-item representation v_i, the

prediction score can be decomposed as:

$$\hat{y_{ui}} = Z_{ui}^{\top} v_i = H_u^{\top} v_i + {X_{seq}^u}^{\top} v_i$$

The first term captures the compatibility between the candidate item and the user's dynamic, behavior-dependent preference, whereas the second term represents the latent compatibility between the candidate item and the relatively stable preference prior inferred from user attributes.

Moreover, the additive pathway provides a shorter gradient route from the ranking objective to the user-feature projection layer, reducing its dependence on gradient propagation through the conditioned self-attention. During the training process of model, the gradient of loss function L can be calculated as:

$$\begin{aligned} \frac{\partial L}{\partial W} &= \frac{\partial L}{\partial Z} \cdot \frac{\partial Z}{\partial W} \\ &= \frac{\partial L}{\partial Z} \cdot \frac{\partial H}{\partial X} \cdot \frac{\partial X}{\partial W} + \frac{\partial L}{\partial Z} \cdot \frac{\partial X}{\partial W} \end{aligned}$$

The first term represents the indirect gradient propagated through the user-conditioned sequence encoder, whereas the second term is the direct gradient introduced by residual-style late fusion. Therefore, the projected user attributes can receive a direct learning signal from candidate-item ranking errors, while still influencing sequence representations indirectly through conditioned attention.

Overall, our proposed model is designed based on the joint influence of user-side features and item-side features on purchase behavior. From the item side, product features represent observable features that consumers rely on when evaluating and comparing alternatives. Purchase decisions are not determined only by item identity or historical co-occurrence patterns; they are also shaped by whether product features match users' needs and preferences. Therefore, the cross-attention feature fusion layer allows item ID embeddings to selectively attend to relevant item features, generating feature-enhanced item representations. From the user side, consumers with similar demographic or contextual features may share similar preference tendencies, while users with similar historical behavior sequences may still make different decisions due to different user profiles. Thus user features are introduced into the attention mechanism to modulate how historical interactions are weighted, and are further fused into the final representation to preserve stable user-level preference information. This architecture enables the model to jointly represent sequential behavior, product-side differences, and consumer-side heterogeneity for next-item personalized recommendation.

3.3 Prediction Layer and Optimization

The final preference representation of each user is derived from the output of the user feature fusion layer Z, where the embedding vector at the last valid position of each user sequence Z_u is extracted.

To train the model, feature-enhanced candidate item encoder is applied to both positive and negative samples, yielding the corresponding embedding Z^+ and Z^- .

Table 2 Statistics of datasets used in the paper

Datasets	User #	Item #	User features #	Item features #	Interactions #	Average Length
ML-1m	5328	3596	3	1	728550	136.74
KuaiRec	7178	10728	8	7	12530806	1746

For candidate item representation, we apply the candidate item encoder. Specifically, the candidate item ID is first mapped into an item embedding, and its corresponding item feature vector is projected into the hidden space. Then, the item-ID embedding is combined with the transformed feature representation through residual addition and layer normalization. In this way, the candidate item representation is feature-enhanced and aligned with the historical item representations. After that, by taking the dot product between user preference representation Z_u and item representation Z^+ and Z^- , we calculate each user's prediction preference score $y^+ = Z_u \bullet Z^+$ and $y^- = Z_u \bullet Z^-$ for positive and negative samples.

Our model is optimized using Bayesian Personalized Ranking (BPR) loss, which maximizes the score difference between positive and negative samples. The loss function is defined as:

$$-\frac{1}{N}\sum_{i=1}^{N}\log\big(\sigma(y_i^+ - y_i^-)\big) + \lambda\|\theta\|^2$$

Where σ is sigmoid function, λ is regularization parameter, θ are the learnable parameters of model.

To evaluate the model, the dot product score between the user's final preference representation Z and the embedding of all candidate items is calculated, and we sort the scores in descending order to generate the Top-K recommendation list. We adopt two standard metrics, HR@K and NDCG@K, for model evaluation.

4 Experiments

To evaluate the performance of our model compared to other baseline model, we used the following real-world datasets:

MovieLens 1m [5] MovieLens Dataset is a widely used benchmark dataset in sequential recommendation, which contains anonymous rating of movies made by MovieLens users. We used the version of MovieLens 1M, including approximately 1 million ratings.

KuaiRec [3] KuaiRec is a real-world dataset collected from the recommendation logs of the video-sharing mobile app Kuaishou. It contains the video interaction history from users and rich side information like video categories and user features.

The detailed information of used datasets can be found in Table 2.

For MovieLens-1m, we considered the rating of users as implicit feedback. To ensure that the dataset could indicate the preference of users, we only kept the interactions

with rating larger than or equal to 3. For KuaiRec, We treated the interactions with video watching ratio larger than 0.5 as positive feedback and filtered those smaller than 0.5.

4.1 Baseline methods

The benchmark methods we used in our experiments are summarized as follows.

SASRec [9] models the sequence by using multi-head attention mechanism to encode the consumer's historical behavior sequence into a vector representation.

SSE-PT [23] integrates user embeddings and stochastic shared embeddings into a Transformer model to achieve personalization.

S^3Rec [27] is a self-attentive neural architecture that learns the correlations among feature, item, subsequence, and sequence by utilizing the mutual information maximization (MIM) principle.

DIF-SR [24] moves different features from the input to the attention layer and decouples the attention calculation of various side information and item representation.

CARCA [15] is a context and feature-aware model that captures user profiles' dynamic nature by using a cross-attention component for scoring items.

4.2 Implementation Details

Our model is implemented in PyTorch. The optimizer is Adam optimizer, with the batch size of 256, the learning rate is 0.0001 and the dropout rate is set as 0.2. The maximum length are set as 200, 100 for datasets MovieLens and KuaiRec.Early stopping was applied based on the validation NDCG@10 for every 10 epochs. For the architecture of our model, we use 2 Transformers blocks and learnable position embedding. The number of attention heads is set as 2.

4.3 Evaluation Metrics

Two commonly used Top-N metrics, the top 10 average Hit Rate(HR@10) and the top 10 normalized discounted cumulative gain(NDCG@10) are adopted to evaluate the performance of sequence recommendation model. HR@10 counts the fraction of times that the ground-truth next item is among the top 10 items, denotes the proportion of accurately recommended items among the top 10 positions in the recommendation list and evaluates the accuracy of the recommendations by assessing how well the recommended items align with the user's interests. While NDCG@10 is a position-aware metric that evaluates the significance of the recommended items by considering their ranking in the list and assigning larger weights on higher positions. Following the evaluation process of SASRec, for each user u, we picked one positive test item which is the real item user interact with, and randomly sampled 99 negative items. Based on the rankings of these 100 items, we calculate HR@10 and NDCG@10. The results are shown in Table 3.

Table 3 Overall performance on ML-1m and KuaiRec datasets

Model	Metric	MovieLens	KuaiRec
SASRec	HR@10	0.8361	0.8622
	NDCG@10	0.5955	0.5334
SSE-PT	HR@10	0.8140	0.8685
	NDCG@10	0.5764	0.5332
S^3Rec	HR@10	0.8157	0.8903
	NDCG@10	0.5710	0.5701
DIF-SR	HR@10	0.8263	0.9006
	NDCG@10	0.6081	0.6150
CARCA	HR@10	0.5313	0.6973
	NDCG@10	0.2956	0.3810
Our model	HR@10	0.8894	0.9114
	NDCG@10	0.6774	0.6128

4.4 Performance Comparison

In this section, we present the experimental results and conduct several analyses to explain the reasons.

4.4.1 Overall Performance

Table 3 reports the overall performance of different sequential recommendation models on the MovieLens and KuaiRec datasets. Overall, the proposed model achieves strong and competitive performance across both datasets. On MovieLens, our model obtains the best results on both evaluation metrics, with an HR@10 of 0.8894 and an NDCG@10 of 0.6774. Compared with SASRec, which mainly models item sequences through self-attention, our model improves HR@10 from 0.8361 to 0.8894 and NDCG@10 from 0.5955 to 0.6774. This indicates that incorporating item features and user-related information can improve both recommendation accuracy and ranking quality. On KuaiRec, our model achieves the highest HR@10 of 0.9114, which outperforms SASRec, SSE-PT, S3Rec, and DIF-SR. This result suggests that the proposed framework is effective in retrieving the relevant item into the top-10 recommendation list. In terms of NDCG@10, our model obtains 0.6128, which is slightly lower than DIF-SR. However, it should be noticed that the model structure of DIF-SR is more complicated than our model, which resulted in significantly longer training time under similar training settings. This indicates that although DIF-SR may achieve better performance, our model still has an advantage in training efficiency.

The overall performance can be explained by the design of the proposed dual-level attention framework. The cross-attention feature fusion layer allows item representations to retrieve relevant projected feature representations from the historical feature sequence, while the user-conditioned self-attention module enables user features to influence the learning of sequential dependencies. Compared with methods that only model item sequences or use side information in a relatively independent way, the

Table 4 Performance on different maximum sequence length on ML-1m dataset

Maximum Sequence Length	100	150	200
HR@10	0.8795	0.8829	0.8894
NDCG@10	0.6576	0.6563	0.6774

proposed model jointly captures sequential behavior, item-side features, and user-side heterogeneity.

4.4.2 Hyperparameter Sensitivity

We evaluate the effect of different maximum sequence lengths. As shown in Table4, the model achieves the best performance when the maximum sequence length is set to 200, with an HR@10 of 0.8894 and an NDCG@10 of 0.6774. When the maximum sequence length increases from 100 to 200, both HR@10 and NDCG@10 generally improve. This indicates that longer historical behavior sequences provide more useful preference information for next-item prediction. In sequential recommendation, user preferences are often reflected through long-term interaction patterns, and a longer sequence allows the model to capture richer behavioral dependencies. However, the improvement from 100 to 150 is relatively limited, and NDCG@10 slightly decreases when the sequence length changes from 100 to 150. This suggests that simply increasing sequence length does not always lead to consistent ranking improvement. Longer sequences may introduce noisy or weakly related historical interactions. Therefore, the choice of maximum sequence length should balance useful long-term preference signals and potential behavioral noise.

Finally, we analyze the influence of the number of attention heads. As shown in Table 5, the model achieves the best performance when the number of attention heads is set to 2, with an HR@10 of 0.8894 and an NDCG@10 of 0.6774. When only one attention head is used, the model still obtains competitive results, but its performance is lower than the two-head setting. This suggests that a single attention head may not be sufficient to capture diverse sequential dependency patterns and feature-related preference signals. However, increasing the number of attention heads from 2 to 4 does not further improve performance. Although the HR@10 remains close, NDCG@10 decreases noticeably. One possible reason could be that too many attention heads split the hidden representation into smaller subspaces, which may weaken the representation capacity of each head and introduce unnecessary noise. Therefore, a moderate number of attention heads can provide a better balance between model expressiveness and ranking stability.

4.4.3 Ablation Study

Furthermore, we examine the contribution of different modules. As shown in Table 6, Removing the cross-attention feature fusion module achieves an HR@10 of 0.8635 and an NDCG@10 of 0.6332, while removing the user-conditioned self-attention module achieves an HR@10 of 0.8690 and an NDCG@10 of 0.6407. Both settings outperform

Table 5 Performance on different attention heads

Number of Attention Heads	HR@10	NDCG@10
h=1	0.8847	0.6631
h=2	0.8894	0.6774
h=4	0.8883	0.6604

Table 6 Performance on different model components on ML-1m dataset

	HR@10	NDCG@10
w/o Cross-attention feature fusion	0.8635	0.6332
w/o User-conditioned self-attention	0.8690	0.6407
Full model	0.8894	0.6774

several baseline methods reported in the overall performance comparison, indicating that item-side feature fusion and user-side personalized attention are both useful for improving recommendation performance. When all modules are included, the model achieves the best performance, with HR@10 of 0.8894 and NDCG@10 of 0.6774. Compared with using only cross-attention feature fusion, the full model improves HR@10 by 0.0204 and NDCG@10 by 0.0367. Compared with using only user-conditioned self-attention, the full model improves HR@10 by 0.0259 and NDCG@10 by 0.0442. These results show that the two modules are complementary rather than redundant. The cross-attention feature fusion layer enhances item representations by incorporating feature information, while the user-conditioned self-attention module personalizes the learning of sequential dependencies according to user features.

5 Conclusion

In this study, we propose a dual-level attention framework for feature-aware sequential recommendation. The framework is motivated by the observation that next-item decisions depend not only on item-transition patterns but also on product-side information and user-side heterogeneity. To jointly model these factors, the proposed model integrates product-feature information and user attributes at different stages of the sequential recommendation process. First, we introduce a sequence-level cross-attention fusion module in which item-ID embeddings act as queries and projected item-feature representations act as keys and values. Instead of assigning explicit importance weights to individual feature dimensions, this module models the relevance between item identities and the overall feature representations at different historical positions. The resulting feature-enhanced sequence preserves collaborative identity signals while incorporating product-feature patterns distributed across historical interactions. Positional embeddings and the subsequent Transformer encoder then enable the model to learn temporal dependencies among these feature-enhanced item representations. Second, we introduce a user-conditioned self-attention mechanism

to incorporate user attributes into sequential dependency learning. This mechanism allows the interpretation of historical interactions to vary according to user-specific information, helping distinguish users who exhibit similar interaction histories but possess different profiles. In addition, residual-style late fusion directly combines the projected user representation with the encoded sequential representation before candidate scoring. This additive pathway preserves a direct profile-based contribution to candidate-item matching and provides a shorter gradient path from the ranking objective to the user-feature projection layer.

Experimental results demonstrate the effectiveness of our proposed framework. The proposed model achieves the best HR@10 on both datasets and obtains the best NDCG@10 on MovieLens. These results indicate that jointly modeling item-side features and user-side features can improve the ability of sequential recommendation models to retrieve relevant items.

From a management perspective, the findings suggest that personalized recommendation can benefit from jointly considering product differentiation and consumer heterogeneity. Product attributes provide observable information associated with consumers' historical choices, while user attributes help characterize relatively stable differences in preference tendencies. By combining these signals with sequential behavior, the proposed framework may support more differentiated product ranking, personalized marketing, and customer preference analysis. Nevertheless, the present findings are based on offline recommendation experiments and should not be interpreted as causal evidence of consumers' decision mechanisms.

Several directions remain for future research. First, more fine-grained user behavior signals and contextual information can be incorporated to better capture dynamic consumer preferences and improve prediction performance. Second, more advanced feature interaction mechanisms can be explored to improve ranking performance, especially on datasets with dense interactions. Third, future studies can evaluate the proposed framework in real business scenarios to further examine its managerial value and practical applicability.

Statements and Declarations

Funding. The author received no funding for this work.

Competing interests. The author declares no competing interests.

Author contributions. Yixing Wang conceived the study, developed the methodology and model, implemented the experiments, analyzed the results, and wrote and revised the manuscript.

Data availability. The datasets used in this study are all publicly available. The MovieLens-1M dataset is available from GroupLens at https://grouplens.org/datasets/movielens/1m/, and the KuaiRec dataset is available at https://github.com/chongminggao/KuaiRec.

References

[1] Boka TF, Niu Z, Neupane RB (2024) A survey of sequential recommendation systems: Techniques, evaluation, and future directions. Information Systems 125:102427

[2] Fischer E, Dallmann A, Hotho A (2023) Personalization through user attributes for transformer-based sequential. In: Recommender Systems in Fashion and Retail: Proceedings of the Fourth Workshop at the Recommender Systems Conference (2022), Springer Nature, p 25

[3] Gao C, Li S, Lei W, et al (2022) Kuairec: A fully-observed dataset and insights for evaluating recommender systems. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management. Association for Computing Machinery, New York, NY, USA, CIKM '22, p 540–550, https://doi.org/10.1145/3511808.3557220, URL https://doi.org/10.1145/3511808.3557220

[4] Ge M, Wang C, Qin X, et al (2025) Personalized dual transformer network for sequential recommendation. Neurocomputing 622:129244

[5] Harper FM, Konstan JA (2015) The movielens datasets: History and context. ACM Trans Interact Intell Syst 5(4). https://doi.org/10.1145/2827872, URL https://doi.org/10.1145/2827872

[6] Hidasi B, Karatzoglou A, Baltrunas L, et al (2015) Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:151106939

[7] Hu J, Zhao F, Chen R (2023) Diverse characteristics of users for personalized sequential recommendation. https://doi.org/10.21203/rs.3.rs-3355174/v1, URL https://doi.org/10.21203/rs.3.rs-3355174/v1, preprint

[8] Jiang W, Fan Y, Du Y, et al (2025) Feature-level attention network with group-aware interest modeling for sequential recommendation. Neurocomputing 639:130302

[9] Kang WC, McAuley J (2018) Self-attentive sequential recommendation. In: 2018 IEEE international conference on data mining (ICDM), IEEE, pp 197–206

[10] Liu C, Li X, Cai G, et al (2021) Noninvasive self-attention for side information fusion in sequential recommendation. In: Proceedings of the AAAI conference on artificial intelligence, pp 4249–4256

[11] Liu G, Yang F, Jiao Y, et al (2025) Learning attribute as explicit relation for sequential recommendation. In: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1, pp 800–811

[12] Liu L (2024) Dynamic time-aware collaborative sequential recommendation with attention-based network. Knowledge and Information Systems 66(3):1639–1655

[13] Ma C, Kang P, Liu X (2019) Hierarchical gating networks for sequential recommendation. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pp 825–833

[14] Pan LW, Pan WK, Wei MY, et al (2026) A survey on sequential recommendation. Frontiers of Computer Science 20(3):2003606

[15] Rashed A, Elsayed S, Schmidt-Thieme L (2022) Context and attribute-aware sequential recommendation via cross-attention. In: Proceedings of the 16th ACM conference on recommender systems, pp 71–80

[16] Sun F, Liu J, Wu J, et al (2019) Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In: Proceedings of the 28th ACM international conference on information and knowledge management, pp 1441–1450

[17] Tang J, Wang K (2018) Personalized top-n sequential recommendation via convolutional sequence embedding. In: Proceedings of the eleventh ACM international conference on web search and data mining, pp 565–573

[18] Veeramani H, Thapa S, Naseem U (2023) Temporally dynamic session-keyword aware sequential recommendation system. In: 2023 IEEE International Conference on Data Mining Workshops (ICDMW), IEEE, pp 157–164

[19] Wang H, Wang Y, Liu Y (2024) A sequential recommendation model for balancing long-and short-term benefits. International Journal of Computational Intelligence Systems 17(1):75

[20] Wang S, Hu L, Wang Y, et al (2019) Sequential recommender systems: challenges, progress and prospects. arXiv preprint arXiv:200104830

[21] Wang S, Shen B, Min X, et al (2024) Aligned side information fusion method for sequential recommendation. In: Companion Proceedings of the ACM Web Conference 2024, pp 112–120

[22] Wu J, Yang B, Mao R, et al (2024) Popularity-aware sequential recommendation with user desire. Expert Systems with Applications 237:121429

[23] Wu L, Li S, Hsieh CJ, et al (2020) Sse-pt: Sequential recommendation via personalized transformer. In: Proceedings of the 14th ACM conference on recommender systems, pp 328–337

[24] Xie Y, Zhou P, Kim S (2022) Decoupled side information fusion for sequential recommendation. In: Proceedings of the 45th international ACM SIGIR conference

on research and development in information retrieval, pp 1611–1621

[25] Yu F, Liu Q, Wu S, et al (2016) A dynamic recurrent model for next basket recommendation. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, pp 729–732

[26] Zhang T, Zhao P, Liu Y, et al (2019) Feature-level deeper self-attention network for sequential recommendation. In: IJCAI, pp 4320–4326

[27] Zhou K, Wang H, Zhao WX, et al (2020) S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In: Proceedings of the 29th ACM international conference on information & knowledge management, pp 1893–1902

[28] Zhu Y, Yao S, Sun X (2024) Feature interaction dual self-attention network for sequential recommendation. Frontiers in Neurorobotics 18:1456192