Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.058438

ARTICLE

A Cross Attention Transformer-Mixed Feedback Video Recommendation Algorithm Based on DIEN

Jianwei Zhang[1,2,*], **Zhishang Zhao**[3], **Zengyu Cai**[3], **Yuan Feng**[4], **Liang Zhu**[3] **and Yahui Sun**[3]

[1]College of Software Engineering, Zhengzhou University of Light Industry, Zhengzhou, 450000, China

[2]Faculty of Information Engineering, Xuchang Vocational Technical College, Xuchang, 461000, China

[3]College of Computer Science and Technology, Zhengzhou University of Light Industry, Zhengzhou, 450000, China

[4]College of Electronics and Information, Zhengzhou University of Light Industry, Zhengzhou, 450000, China

*Corresponding Author: Jianwei Zhang. Email: mailzjw@163.com

Received: 12 September 2024 Accepted: 23 October 2024 Published: 03 January 2025

ABSTRACT

The rapid development of short video platforms poses new challenges for traditional recommendation systems. Recommender systems typically depend on two types of user behavior feedback to construct user interest profiles: explicit feedback (interactive behavior), which significantly influences users' short-term interests, and implicit feedback (viewing time), which substantially affects their long-term interests. However, the previous model fails to distinguish between these two feedback methods, leading it to predict only the overall preferences of users based on extensive historical behavior sequences. Consequently, it cannot differentiate between users' long-term and short-term interests, resulting in low accuracy in describing users' interest states and predicting the evolution of their interests. This paper introduces a video recommendation model called CAT-MF Rec (Cross Attention Transformer-Mixed Feedback Recommendation) designed to differentiate between explicit and implicit user feedback within the DIEN (Deep Interest Evolution Network) framework. This study emphasizes the separate learning of the two types of behavioral feedback, effectively integrating them through the cross-attention mechanism. Additionally, it leverages the long sequence dependence capabilities of Transformer technology to accurately construct user interest profiles and predict the evolution of user interests. Experimental results indicate that CAT-MF Rec significantly outperforms existing recommendation methods across various performance indicators. This advancement offers new theoretical and practical insights for the development of video recommendations, particularly in addressing complex and dynamic user behavior patterns.

KEYWORDS

Video recommendation; user interest; cross-attention; transformer

1 Introduction

In recent years, short videos have gained immense popularity, with platforms like TikTok and Kuaishou amassing large user bases and significant traffic [1]. Given the large user volume and traffic, recommendation systems (RS) have become a crucial technology in this domain [2], Recently, researchers have developed various recommender system methods focused on building user profiles based on user feedback [3] to enhance recommendation quality and achieve specific platform objectives [4].

Early recommendation models evolved from methods based on Collaborative Filtering (CF) [5] and Matrix Factorization (MF) [6] to logistic regression algorithms that integrate multiple features, eventually developing into deep learning-based methods [7], such as the AFM model [8], which was the first to introduce the attention mechanism. Recently, researchers have proposed several new recommendation system models, including the SASRec model [9], BERT4Rec model [10], and Alibaba's models, such as the Deep Interest Network (DIN) [11] and the Deep Interest Evolution Network (DIEN) [12]. These recommendation models aim to achieve the objectives of short video platforms by analyzing user behaviors, including increasing user retention, enhancing user engagement, and extending viewing time. The DIEN model, in particular, has gained significant recognition among researchers for its ability to learn and predict the evolution of user interests. For instance, Xu et al. introduced a hierarchical attention network based on the DIEN model, proposing a deep interest prediction model that utilizes hierarchical attention networks [13]. Feng et al. integrated Bi-LSTM into the DIEN model, resulting in the development of the Deep Session Interest Network [14]. Shi et al. employed neural networks for continuous modeling of interest evolution, proposing a deep time-stream framework built on the DIEN model [15].

Existing recommender systems and several models, including SASRec, BERT4Rec, DIN, and DIEN, along with their improved variants, have implemented deep learning on user behavior sequences, leading to the proposal of the concept of interest evolution. This concept has become a crucial technical foundation for short video recommendation systems. However, a common issue in these models is that they learn users' historical behavior information indiscriminately when constructing user interest profiles, neglecting the distinction between explicit and implicit feedback in reflecting user interests [16]. This undifferentiated approach may result in a misunderstanding of user interests, thereby affecting the accuracy of recommendations, this is due to explicit and implicit feedback each have distinct advantages, relying on a single type of feedback or learning both types simultaneously fails to adequately capture the full scope of users' interest states [17]. To overcome these limitations, it is essential to develop a more effective strategy for separately learning and integrating explicit and implicit feedback, thereby balancing users' long-term and short-term interests.

This paper proposes a Cross Attention Transformer-Mixed Feedback Recommendation (CAT-MFRec) model to address this issue. Unlike existing models, the CAT-MFRec model distinctly differentiates between explicit and implicit feedback during user interest modeling, utilizing Transformers to model each type of feedback separately. More importantly, the model integrates the two types of feedback through the cross-attention mechanism, maximizing the advantages of both explicit and implicit feedback, and ultimately capturing users' long-term and short-term interest states more accurately. Additionally, the model significantly improves predictions of user interest evolution while balancing short-term engagement and long-term satisfaction, thereby enhancing personalized recommendations and achieving the objectives of short video platforms.

The primary contributions of this paper are as follows:

- This paper highlights the differing performance of explicit and implicit feedback mechanisms concerning long-term and short-term interests, emphasizing the importance of separate learning. It effectively combines these mechanisms to construct user interest states.
- We propose the CAT-MFRec model, an innovative approach based on the cross-attention mechanism and Transformer. This model facilitates separate learning and effective integration

of the two types of feedback, accurately constructing users' long-term and short-term interest states and thereby enhancing predictions of their future interests.

- We validated the effectiveness of our work using a real-world dataset (KuaiRand), with results demonstrating the efficacy of our approach.

2 Related Work

2.1 Research Actuality

Video recommendation technology has emerged as a significant development area in artificial intelligence and data science in recent years. Video recommendation systems primarily analyze users' historical behaviors to determine their interest preferences, delivering personalized video content to enhance user experience and increase platform engagement [18]. However, traditional recommendation systems typically account only for users' static interests. The integration of deep learning technology into recommendation systems has led to significant breakthroughs, including the introduction of the concept of interest evolution, which signifies that user interests change dynamically over time [19].

Users have diverse interests; at any given moment, a user may possess multiple interests, a situation referred to as the "interest state." Furthermore, each interest is dynamic, undergoes its own evolutionary process, and exhibits specific causal relationships [20]. Additionally, interests can drift; at any moment, a user's interest may manifest as behavior, such as watching basketball-related videos for a time and then suddenly switching to calligraphy-related content. However, previous models are limited in their ability to predict overall user preferences based solely on a large number of historical behavior sequences [21]. These models are unable to predict the user's "next" preference. To accurately predict the user's next preference, we must thoroughly understand the user's interest state, which necessitates distinguishing between the two types of user feedback, as they have different effects on interests. We will focus on this distinction in the next section; however, previous recommendation models do not differentiate between these feedback types. These models train all of a user's historical behaviors collectively, resulting in a generalized description of the user's interest state that fails to accurately balance long-term and short-term interests [22].

Therefore, the primary challenge lies in balancing the use of two feedback methods to construct a user's interest profile while effectively managing the relationship between long-term and short-term interests.

2.2 User Feedback

In video recommendation systems, user feedback serves as a critical foundation for optimizing recommendation algorithms. Feedback is primarily categorized into explicit and implicit types, reflecting different dimensions of user interest and differing significantly in terms of timeliness and dependence. Understanding the characteristics of these two types of feedback and utilizing them effectively is essential for optimizing recommender systems [23].

Explicit feedback refers to preferences directly expressed by users through interactive actions, such as likes, comments, favorites, and shares, which provide immediate insights. This feedback arises directly from users' active behaviors and reflects their attitudes toward the video content. For instance, a user who likes a video after viewing it provides strong positive feedback on the content. This short-term behavior allows the recommender system to rapidly capture users' short-term interests.

Implicit feedback refers to behavioral preferences that are not explicitly stated, including viewing time, frequency, browsing path, and interaction duration. Although this type of feedback does not directly indicate user preferences, it allows for inferring long-term interests due to continuous observation of user behavior, often necessitating the analysis of historical data over extended periods. While implicit signals, such as viewing duration, do not directly reflect user attitudes, they can reveal trends in user preferences. For instance, a user who watches specific content for an extended duration may demonstrate a sustained interest in that content. These patterns typically reflect long-term interests rather than short-term fluctuations [24].

In general, explicit feedback allows the recommender system to rapidly adjust recommended content to align with users' current interests, while implicit feedback reflects users' long-term interests due to its stability, thereby enhancing the representation of long-term preferences [25].

2.3 Transformer

The Transformer is a deep learning model designed for processing sequence data, utilizing a self-attention mechanism that operates independently of Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs).

The model input includes a word embedding vector, and positional information is enhanced through positional encoding. The self-attention mechanism enables each position in the input sequence to focus on other positions by calculating the relationships between the query, key, and value to assess their influence. It employs the dot product and softmax function to generate attention weights. In the multi-head attention mechanism, attention is distributed across multiple heads, with each head processing different parts of the input. Each self-attention layer is followed by a feedforward neural network, which processes the vector at each position independently. Each encoder layer comprises a multi-head self-attention layer and a feedforward network, both equipped with residual connections and layer normalization to improve training stability and efficiency. The decoder layer resembles the encoder layer but incorporates an encoder-decoder attention layer, enabling it to focus on relevant portions of the encoder's output. Finally, the decoder output is processed through a linear layer followed by a softmax layer to generate a probability distribution [26].

These designs allow the Transformer to optimize parallel processing and manage long-range dependencies, making it suitable for complex tasks such as machine translation and text generation. In this study, we replace positional encoding with precise temporal information for two primary reasons. First, positional encoding based on sine and cosine primarily offers relative positional information within the sequence. In contrast, time series data encompasses not only the order of elements but also time intervals and durations, which positional encoding cannot accurately capture. Incorporating precise time information enables the model to learn actual time points and intervals, thereby modeling temporal dependencies more effectively. Second, when processing time series data, the model must comprehend the relative positions of data points and their time intervals. By integrating precise time information, the model can more effectively capture temporal dependencies and improve its capability to handle tasks with varying time spans [27].

2.4 Cross Attention

The cross-attention mechanism is derived from classical attention mechanisms. Common attention mechanisms are typically found in self-attention mechanisms, which learn dependencies between elements within the same input dataset. In contrast, cross-attention integrates multiple inputs from diverse sources, enhancing feature representation by focusing on interactions among these inputs. This aligns closely with our requirements, as our model must handle two distinct sources of information: explicit and implicit feedback. The cross-attention mechanism effectively captures the mutual relationships and dependencies between these two types of information, facilitating better integration to predict the user's interest state. It dynamically assigns attention weights, enabling more precise learning of how different signals influence user interests.

In comparison to other methods, directly averaging or concatenating explicit and implicit feedback before sending it to the network for processing lacks dynamic weight adjustment. The cross-attention mechanism dynamically learns the importance of various feedback types in different contexts, while averaging or concatenating lacks this flexibility. Compared to graph neural networks, the cross-attention mechanism offers significant advantages in terms of parameter complexity and training efficiency when only two sources are present. Utilizing graph neural networks significantly increases model complexity, while also reducing training efficiency. When using the self-attention mechanism, dependencies cannot be captured through interactions among data from different sources [28].

In summary, the cross-attention mechanism improves the performance and adaptability of sequence-to-sequence models by allowing the decoder to dynamically focus on elements of the input sequence. It is a crucial component of contemporary deep learning models for processing complex sequence data [29].

3 Method

In this section, we present a comprehensive overview of our proposed CAT-MFRec model. We begin with a foundational overview to facilitate deeper understanding, followed by a detailed examination of the complex design and setup of specific modules.

3.1 CAT-MFRec Network Structure

The CAT-MFRec model mainly comprises three layers. From bottom to top, these layers are: 1. the behavior sequence layer; 2. the interest extraction layer; and 3. the interest evolution layer. And the final decision output section. Its structure is shown in the Fig. 1 below.

As can be seen from Fig. 1, the structure of CAT-MFRec is as follows:

1. **Embedding Layer:** This layer comprises four types of features: User Behavior, Platform Object, Context, and User Profile. In this layer. We process two types of user behavior separately: b_d represents the implicit feedback from the user's viewing time, while b_i represents the explicit feedback from the user's interactions. Both are processed separately and fed into the embedding layer. This results in two batches of embedding vectors: e_d and e_i.
2. **Interest Extractor Layer:** The Interest Extractor layer aims to mine and extract the "interest state" hidden behind the user's behavior at each moment, which is the primary focus of our improvement efforts. The cross-attention mechanism is introduced to enhance the training and learning of mixed feedback. The adopted sequence model is the Transformer model, it relies entirely on the self-attention mechanism to process the entire sequence, allowing simultaneous processing of all data points and significantly improving training speed.
3. **Interest Evolving Layer:** As shown in the figure, a weight score is calculated through attention between the hidden state h_t obtained from the Interest Extractor layer and the platform target. This weight score is then combined with h_t in the AUGRU (Attention-based Gated Recurrent Unit) to obtain the final user interest state.

4. **Subsequent Processing:** The user's interest state is obtained through the three aforementioned layers and concatenated with the Platform Object, Context features, and User Profile feature vectors. This combined input then enters the multi-layer fully connected layer to generate the final recommendation prediction.

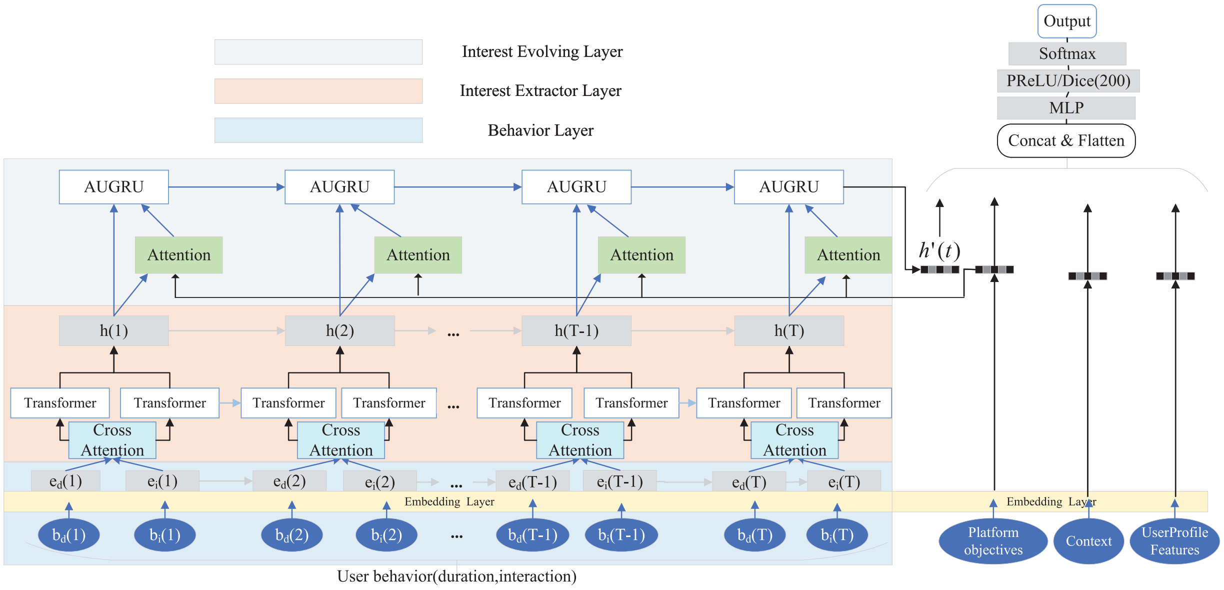

Figure 1: CAT-MFRec network structure

The loss function for the entire model is calculated as follows:

$$L = L_{target} + \alpha * L_{aux} \tag{1}$$

L_{target} represents the cross-entropy loss following the fully connected layer, and the formula is:

$$L_{\text{target}} = -\sum\nolimits_{j=1}^{N} \left[y_j \log(\hat{y}_j) + (1 - y_j)\log(1 - \hat{y}_j) \right] \tag{2}$$

Here y_j is the real situation, $\hat{y}_j$ is the probability that the model predicts the user is interested, L_{aux} is the auxiliary loss, and α is the hyperparameters used for balancing, which will be explained in the later section.

3.2 Embedding Layer

In our proposed model, in the Embedding stage, we divide user behavior features into explicit feedback $b_i = [b_i(1), b_i(2), \ldots, b_i(T-1), b_i(T)]$ brought by user interaction behavior and implicit feedback $b_d = [b_d(1), b_d(2), \ldots, b_d(T-1), b_d(T)]$ brought by user viewing time for Embedding operation, respectively, to explicit feedback word embedding vector $e_i = [e_{i1}, e_{i2}, \ldots, e_{i(T-1)}, e_{iT}]$ and implicit feedback word embedding vector $e_d = [e_{d1}, e_{d2}, \ldots, e_{d(T-1)}, e_{dT}]$. By handling explicit and implicit feedback separately, our model retains the distinct information associated with each feedback type. Subsequently, e_{ij} and e_{dj} are fed into two parallel Transformer encoders to learn patterns and dependencies specific to each feedback type.

3.3 Interest Extractor Layer

3.3.1 Transformer and Cross Attention

To accurately extract the user's interest state at each moment from their behavior, we need to implement a Transformer network capable of handling time series data. We will also incorporate the cross-attention mechanism when passing it into the self-attention mechanism of the Transformer, specifically exchanging the generated Q matrix. Its structure is shown in Fig. 2 below.

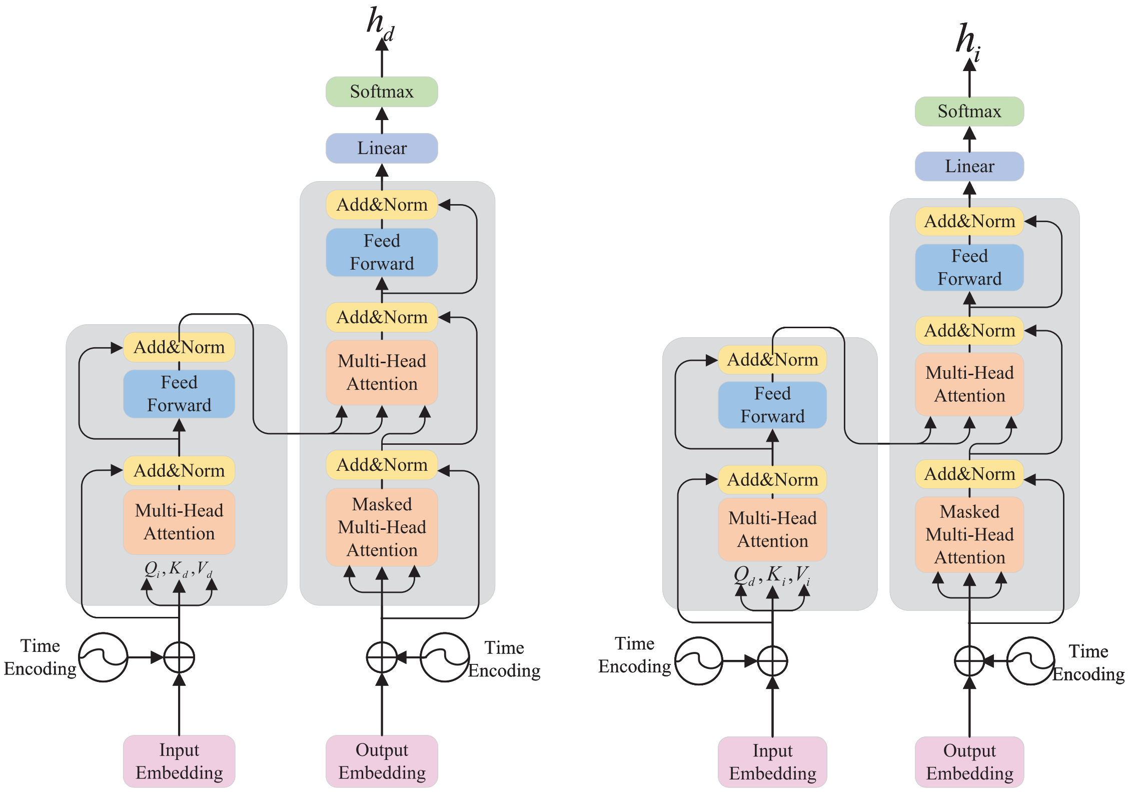

Figure 2: Transformers for time series

The input to the Transformer model consists of the embedding vector, which incorporates positional information through positional encoding. However, we will remove the Positional Encoding from the Transformer and replace it with appropriate time series encoding.

Building on the Transformer model with time series encoding, the sequential data processing methods for the explicit and implicit feedback embedding vectors are outlined below. In the encoder section of the Transformer model, accurate time information replaces the original positional encoding. First, accurate time features are extracted from the timestamps of user behavior and periodically encoded. These time features are then matched with the dimensions of e_i and e_d through linear transformation and subsequently summed.The embedding vectors Te_i and Te_d with time series information were obtained. The queries Q_i, keys K_i, and values V_i for the explicit feedback embedding vector Te_i and the queries Q_d, keys K_d, and values V_d for the implicit feedback embedding vector Te_d were derived through linear transformation.

When the explicit feedback embedding vector Te_i and the implicit feedback embedding vector Te_d are passed into the Transformer model, the query matrix Q_i representing explicit feedback is exchanged with the query matrix representing implicit feedback.This exchange allows the Transformer to process Q_d, K_i, and V_i along with Q_i, K_d, and V_d as inputs, accounting for both types of user behavior and their effects on long-term and short-term interests, as shown in Fig. 3 below.

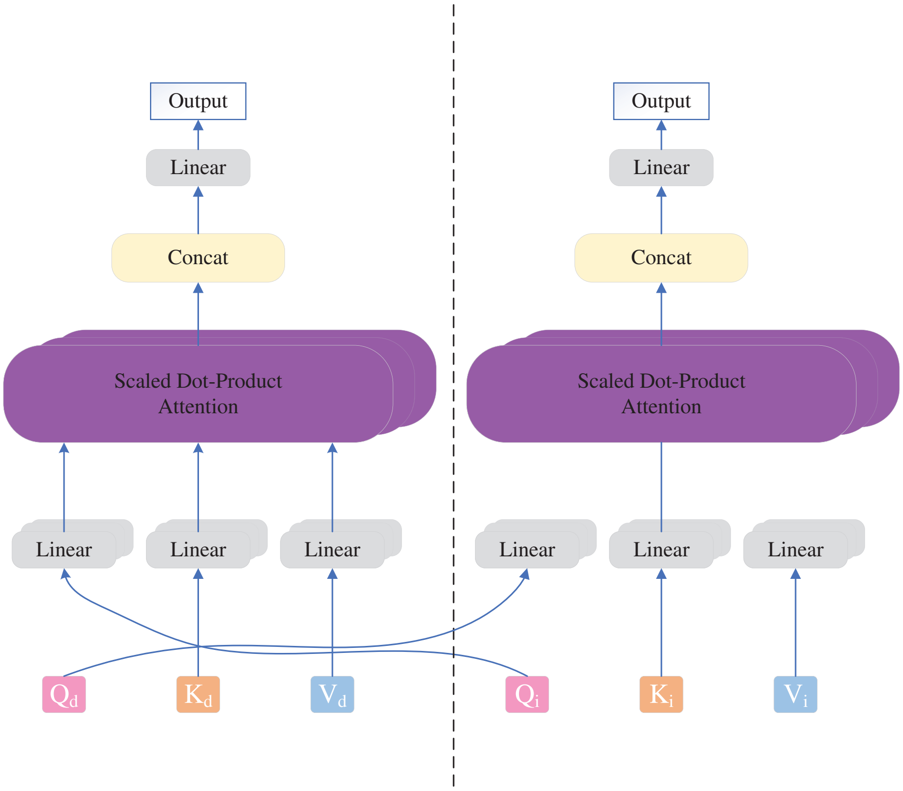

Figure 3: Cross-attention mechanism

The subsequent calculation process in the two Transformer models is the same, taking the subsequent calculation of the explicit feedback embedding vector as an example: Each head of multi-head attention is a self-attention calculation, and the attention output is calculated through three steps: Scale, Softmax, and MatMul.

Firstly, in the Scale operation, the dot product result of Q_d and K_i is scaled to prevent numerical instability caused by too large vector dimension, so that the attention score is maintained in a reasonable range. Subsequently, Softmax (normalization) is performed to obtain the attention weight, Finally, MatMul (matrix multiplication) multiplizes the attention weights with the value vector matrix V_i to obtain the attention output of each head, The formula is summarized as follows:

$$ScaledScores_i = \frac{Q_d K_i^T}{\sqrt{d_k}} \quad AttWeights_i = Softmax(ScaledScores_i) \quad AttOutput_i = AttWeights_i \cdot V_i \tag{3}$$

The above calculation process is the calculation process of multi-head attention computation:

$$Attention(Q_d, K_i, V_i) = softmax\left(\frac{Q_d K_i^T}{\sqrt{d_k}}\right) \cdot V_i \tag{4}$$

Multi-head attention then concatenates the outputs of all heads and maps them back to the original dimensions of the model through a linear layer, the output of multi-head attention is then residual connected with the original input.

In the decoder section: These embeddings help the decoder to refer to the information of different candidate videos in the prediction process. Firstly, the input is linearly transformed to generate Q, K, V vector and sent to the calculation of mask multi-head, which is similar to the multi-head attention mechanism. But after each Scale of self-attention, a masked process is added. To ensure that the model is focusing on the relevant content, we apply masks as needed to hide the parts that don't need to be focused on, After that, the query Q_2 obtained from the result of the mask multi-head attention calculation and the key K_1 and value V_1 obtained from the encoder output features are used for the multi-head cross-attention calculation of the encoder and decoder. The process of multi-head cross-attention calculation is consistent with the above multi-head attention calculation, and the overall formula is as follows:

$$CrossAttention(Q_2, K_1, V_1) = softmax\left(\frac{Q_2 K_1^T}{\sqrt{d_k}}\right) \tag{5}$$

The output features of multi-head cross-attention computation were Add&Norm and residual connected with the results of masked multi-head attention computation, and then through a feed-forward neural network and concatenated Add&Norm and residual connected with the input of feedforward neural network. Finally, the output result h_t is obtained by a liner and Softmax function.

Similarly, a subsequent computation of the implicit feedback embedding vector yields the output h_d, which is followed by a weighted average of the two probability distributions h_t and h_d with one-to-one weights to synthesize an intermediate state h_t.

3.3.2 Auxiliary Loss Design

As we said above, the overall loss of the whole network is $L = L_{target} + \alpha * L_{aux}$, this is because if the time series of users is directly added to it, new problems will also be encountered. If the behavior of users is regarded as a sequence, there is little difference between the user behavior sequence "jeans–Harun pants–wide-leg pants" and "Harun pants–jeans–wide-leg pants". That is, the sequence of user actions is actually not very sensitive to order. The problem is that hidden state h_t only captures the dependence of user behavior sequence and cannot effectively reflect user interest. If we only rely on the loss after the final fully connected layer, we can only learn the final comprehensive interest of users, while hidden state h_t cannot be guided by effective supervision signals.

Therefore, there will be an auxiliary loss, which is used to guide the learning of the intermediate states, and its structure is shown in Fig. 4 below.

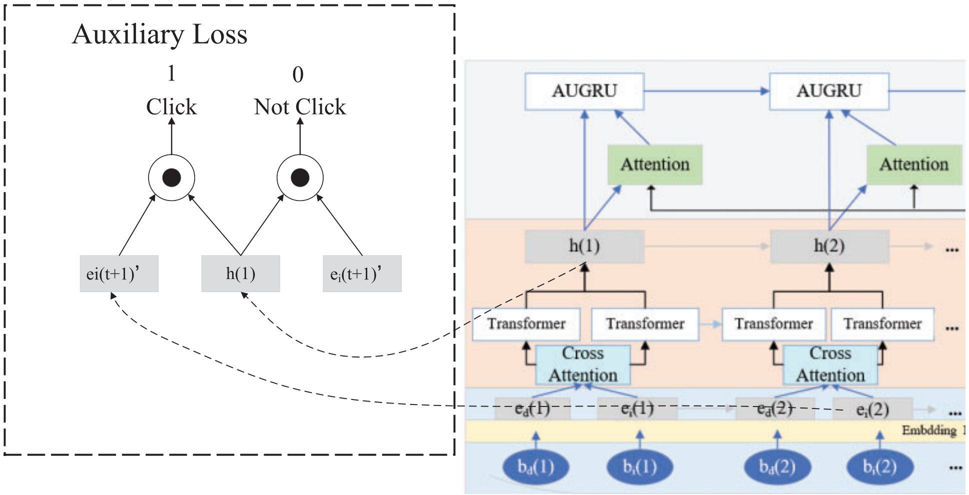

Figure 4: Auxiliary loss

As shown in the figure, this is a binary classification model used to calculate the accuracy of interest extraction. We use the user's actual behavior at the next time step $e(t+1)$ as the positive example, and the negatively sampled behavior as the negative example $e(t+1)'$. These are respectively dotted with the extracted interest h(t), then input into the designed auxiliary network to obtain the prediction results, and calculate an auxiliary loss through the log_{loss}. The formula for the auxiliary loss is:

$$L_{aux} = -\frac{1}{N}\left[\sum_{j=1}^{N}\sum_{t} log(\sigma[h_t, e_{ij}(t+1)]) + log(1-\sigma[h_t, e_{ij}(t+1)'])\right] \tag{6}$$

Among them, $[h_t, e_{ij}(t+1)]$ represents the inner product, and σ is the Sigmoid function. From this loss design, it can be seen that the purpose is to force the interest sequence feature h_t to fit the user's behavior, so as to better capture the user's interest. The more similar h_t is to $e_{ij}(t+1)$, the larger the inner product, so $\sigma[h_t, e_{ij}(t+1)]$ is close to 1, and $log(\sigma[h_t, e_{ij}(t+1)])$ tends to 0. While $log(1-\sigma[h_t, e_{ij}(t+1)'])$ also tends to 0, so the auxiliary loss L_{aux} tends to 0. It conforms to the objective of minimizing auxiliary loss L_{aux}. Conversely, if h_t is not similar to $e_{ij}(t+1)$, $log(\sigma[h_t, e_{ij}(t+1)])$ tends to negative infinity and $log(1-\sigma[h_t, e_{ij}(t+1)'])$ also tends to negative infinity, in which case the auxiliary loss L_{aux} tends to positive infinity, in line with our min (loss) objective. And vice versa, so the final loss function is $L = L_{target} + \alpha * L_{aux}$.

The purpose of this is to extract the interest state at each time. If only the last interest state is used to supervise, all the states of the hidden layer will serve for the last state, and the extracted hidden layer state is obviously distorted.

3.4 Interest Evolving Layer

In the interest evolution layer, firstly, the attention score α_t between the interest state and the target is calculated, which represents the correlation degree between the interest sequence feature h_t and the current candidate advertisement at the current time step, and the larger the value is. It shows that the current h_t is more related to the platform goal and more concerned about the value.

Where e_t is the intermediate state to calculate the attention score α_t, q refers to the embedding vector of the current recommended target, W_1 and W_2 are the weight matrices of the projection, v is the context vector used to map to a scalar, tanh adds nonlinearity to capture complex relationships, and b is the bias term, then we normalize the attention weights, the calculation formula is summarized as follows:

$$e_t = v^T \cdot \tanh(W_1 h_t + W_2 q + b) \qquad \alpha_t = \frac{exp(e_t)}{\sum_{j=1}^{T} exp(e_j)} \tag{7}$$

After having the attention score, add it to AUGRU, that is, embed this attention operation into the AUGRU update gate, and use this layer to more pertinently simulate the interest evolution path related to the platform goal, so as to obtain the final interest, its structure is shown in Fig. 5 below.

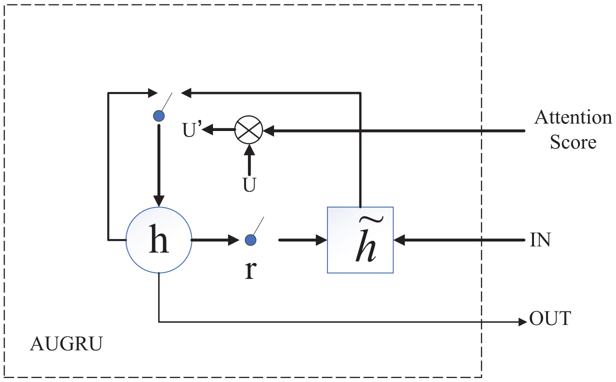

Figure 5: AUGRU structure

In AUGRU, by dynamically adjusting the update gate u_t, combined with the weight a_t of the attention mechanism, the model is allowed to selectively update the user's interest state according to the relevance of the current goal. This mechanism enables the model to give enough attention to the latest input or changes while retaining important historical information, so as to track and predict user interest changes more precisely.

In the calculation process, the vectors of the two behaviors of the user at time t are concatenated to get $x_t = Concat(b_i, b_d)$, and then the GRU update gate z_t can calculate how much historical information should be retained from the previous state h_{t-1} to the current state h_t through the current input x_t and the user interest state h_{t-1} at the previous time step, Then we add the attention score to the update gate, we get:

$$u_t' = \sigma(W_u[x_t, h_{t-1}] + b_u) \qquad \tilde{u}_t = \alpha_t * u_t' \tag{8}$$

Then we also need a candidate hidden state $\tilde{h}_t$, the key intermediate computation used to update the current hidden state of the network. This mechanism allows these networks to preserve both long-term dependence and short-term correlation of information when processing sequential data, calculated as a gated combination of the current input x_t and the previous state h_{t-1}:

$$\tilde{h}_t = \tanh(W_h[x_t, (r_t \cdot h_{t-1})] + b_h) \tag{9}$$

where r_t is the reset gate of GRU (Gated Recurrent Unit), which determines how much information should be kept from the previous state h_{t-1} when calculating the candidate state, and is calculated as follows:

$$r_t = \sigma(W_r[x_t, (r_t, h_{t-1})] + b_r) \tag{10}$$

where σ represents the Sigmoid function used to compress the linear combination of update and reset gates into the range [0, 1], and tanh represents the hyperbolic tangent activation function used to introduce nonlinearities and help the network capture complex patterns and relationships.

With the above formula, it finally follows that AUGRU state updates can be expressed as follows:

$$h'_t = (1 - \tilde{u}_t{}') * h'_t + \tilde{u}'_t * \tilde{h}'_{t-1} \tag{11}$$

The result is a vector h'_t, which represents a high-dimensional representation of the user's interest state at that particular point in time. This interest state sequence reflects how the user's interest changes over time based on their actions and interactions, and can be directly used as input to subsequent recommendation models to predict the probability of a user's click rate or other actions for a recommendation item.

3.5 Subsequent Processing

In this section, we provide a detailed exploration of the subsequent processing steps for obtaining user interest representation in the CAT-MFRec model.

Based on prior research on video recommendation, we have selected three metrics as platform goals to guide our model's learning: user time, user engagement, and user retention. The first two metrics are derived primarily from total user viewing time and the number of user interactions, while the third metric is influenced by the user's interest in the recommended content.

First, we analyze the user's interaction history with specific content on the platform over time, including viewing time and retention, to characterize user preferences. Together with contextual features (time, location, etc.) and user profile features (age, gender, interest tags, etc.), these data are encoded using one-hot encoding in the embedding layer and then concatenated.The concatenated vectors are then input into a fully connected layer for linear transformation. Following the fully connected layer, an activation function is introduced to introduce non-linear factors, enabling the model to learn and simulate more complex functions. In this model, we employ PReLU, alongside the Dice activation function.

The final feature vector is processed through the Sigmoid function of the post-classifier, converting the output to a value between 0 and 1, which indicates the accuracy of our predictions.

Its flow chart is shown in Fig. 6.

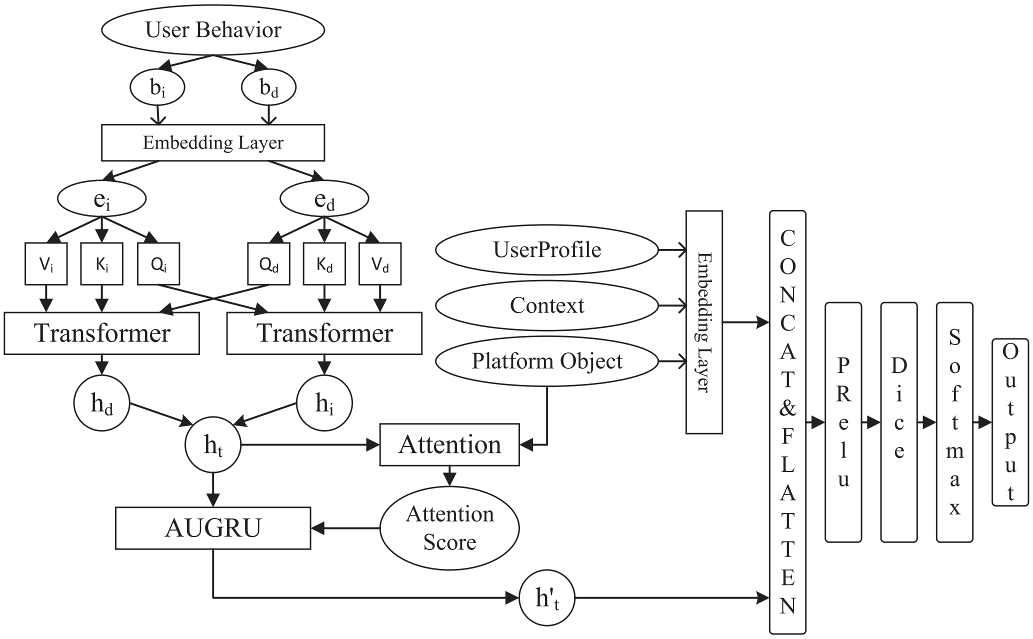

Figure 6: Flow chart

4 Experiment

We will conduct experiments to address three questions: 1. How does our improvement perform on a real dataset compared to existing recommendation methods? 2. How does the improved component impact the experiment? 3. How do the hyperparameters in the experimental setup influence the results?

4.1 Experimental Environment Setup

4.1.1 Datasets

We conduct extensive experiments using the real-world dataset KuaiRand[1], from the recommendation logs of the video-sharing mobile application Kuaishou. The dataset covers a time span from 08 April 2022, to 08 May 2022. The dataset includes explicit feedback on user interaction behaviors, such as likes and comments, as well as implicit feedback based on viewing time. This combination enables effective capture of both short-term and long-term user interests, while the dataset's large scale supports model training. The dataset is derived from an actual short video platform, providing a high-fidelity research foundation for studying user interest evolution.

We divide the dataset into three parts: the first 15 days constitute the training set, the subsequent 10 days comprise the test set, and the final 5 days serve as the validation set. To enhance explicit feedback, we combined six types of explicit feedback (is-click, is-like, is-follow, is-comment, is-forward, is-hate) into a single item labeled "clfcf." If "is-hate" equals 1, then "clfcf" is set to 0, indicating that the user dislikes the content. Conversely, if "is-hate" equals 0 and at least two of the first five items equal 1, "clfcf" is set to 1, indicating that the user likes the content. Refer to Table 1 for details on the evaluation of the preprocessed dataset.

[1] https://kuairand.com/ (accessed on 22 October 2024).

Table 1: Statistical details of the evaluation dataset

Dataset	Users	Item	Interaction	Interaction density
KuaiRand	27,284	7314	694,180	0.0805

4.1.2 Evaluation Index

To thoroughly assess the comparative methods, we must evaluate model performance regarding recommendation accuracy, user retention, engagement, and playtime using various metrics. First, to measure recommendation accuracy, we utilize the NDCG@k metric, which is widely adopted in recommendation systems to assess ranking quality and specifically evaluate the ranking results [30]. The values of NDCG typically range from 0 to 1, where values close to 1 indicate a recommendation list of very high quality, while values close to 0 indicate poor quality NDCG is calculated as follows:

$$NDCG_K = \frac{DCG_K}{IDCG_K} \tag{12}$$

To represent the three platform goals: user time, user engagement, and user retention, we formulated three metrics: *DS* (Duration Standard), *IR* (Interaction Ratio), and *PTR* (Play Time Ratio).

The user retention metric, Duration Std@k (DS@k), represents the standard deviation used to evaluate the recommendation system. This metric analyzes duration differences to assess whether the recommender system can meet users'varied time needs. A high value of DS@k indicates significant differences in the duration of recommended content, which may better satisfy diverse user needs. For the user's playback time, we use the ratio of video playback time in the user's *Topk* recommendation list to the total playback time. The baseline is set at 80%, corresponding to a value of 1. A value greater than 1 indicates that the playback time of the recommended content typically exceeds the average. This suggests that the user is interested in these recommendations, so we label this metric as Play Time Ratio@k (PTR@k). For engagement, we focus on the proportion of interactions between users and videos in the top k recommendation list. The baseline is set at 1, where a value greater than 1 indicates that the recommended content achieves higher interaction than this baseline, signaling good user acceptance. We label this measure as Interaction Ratio@k (IR@k). The formula is presented as follows:

$$\sigma_K = \sqrt{\frac{1}{K}\sum_{i=1}^{K}(d_i - \bar{d})^2} \qquad PTR@K = \frac{U_p}{0.8A_p} \qquad IR@K = \frac{U_i}{0.8A_i} \tag{13}$$

Here, σ_k denotes the standard deviation of the durations of the *Topk* recommendation items, d_i represents the duration of the i-th recommendation item, and $\bar{d}$ is the average duration of the *Topk* recommendations.

4.1.3 Implementation Details

In this invention, the dimension of the embedding layer in CAT-MFRec is set to 128, which is then increased to 256 in the linear layer of the Transformer. The number of Transformer layers is set to 2, with 8 heads, each having a dimension of 32, and the feedforward neural network dimension is 1024. The hidden layer dimension in DIEN is also set to 128, consistent with the model dimension. Adam is chosen as the optimizer for DIEN, responsible for adjusting the model's weights and parameters to minimize the loss function. Consequently, the model gradually improves its performance during training. The auxiliary loss weight decay range is: [1e-2, 1e-4, 1e-6, 0], and the search learning rate range is: [1e-1, 1e-2, 1e-3, 1e-4, 1e-5], range of micro-batch sizes: [512, 1024, 2048], to adapt to the parameters of the range [0.0–1.0, default = 0.5]. The training period consists of 10 rounds, utilizing an early stopping policy with a patience of 5. Additionally, the hyperparameter k for the top-k list metric is set to 10.

4.2 Performance Comparison

We evaluated the overall performance of the current popular recommender system models in this system, and the summary results shown in Table 2.

Table 2: Model performance comparison

Model	NDCG@10	DS@10	PTR@10	IR@10
CAT-MFRec	0.8143	77.2623	1.152	1.135
DIEN	0.7814	75.4512	1.088	1.083
DIN	0.7684	70.1514	1.069	1.058
SASRec	0.7325	66.51	1.004	1.003
BERT4Rec	0.6901	65.23	1.006	0.992
AFM	0.6425	42.33	0.978	0.966

The CAT-MFRec model demonstrates superior performance across all evaluated aspects, achieving an NDCG@10 score of 0.8143, the highest among all models. This indicates that our model excels in the relevance and ranking quality of the recommendation list. A high NDCG value indicates that the recommender system effectively ranks content most likely to interest the user at the top of the list, this performance significantly enhances user satisfaction, as users are more likely to quickly find relevant content.

For the other three metrics, CAT-MFRec achieves a score of 77.2623 on the DS@10 metric, this metric reflects the diversity of recommended content durations. The high score of CAT-MFRec indicates its ability to provide users with a wide range of content options to accommodate various viewing preferences. This approach increases user engagement over time, by offering diverse content, the platform can keep users engaged for longer periods, which is essential for enhancing overall user retention.

The score for PTR@10 is 1.152, this performance surpasses that of other models, highlighting CAT-MFRec's ability to stimulate user interaction. Higher interaction rates are typically linked to better content quality and user satisfaction, suggesting that CAT-MFRec excels in personalized content matching. Frequent interaction with recommended content can foster greater user loyalty, which is essential to building a long-term relationship between users and the platform.

CAT-MFRec also achieves the highest performance on the CR@10 metric. This indicates that CAT-MFRec effectively understands and meets users' deeper interests, which is crucial for enhancing user engagement and platform retention. As users invest more time on the platform, they become more integrated into the ecosystem, reducing the likelihood of churn. Furthermore, extended viewing times can lead to increased monetization opportunities, including higher ad impressions and improved subscription retention.

4.3 Ablation Experiment

In order to study the impact of each component, we made the following design, introducing some variants as follows: G-C: Replace Transformer with GRU and keep Cross-Attention; T-FC: Keep Transformer, but disable Cross-Attention; G-FC: Replace Transformer with GRU and disable Cross-Attention; RD: Removes the viewing duration from the input; RC: Removes explicit feedback from the input. The results are shown in Table 3 below.

Table 3: Results of ablation experiments

Model	NDCG@10	DS@10	PTR@10	IR@10	Epoch time
CAT-MFRec	0.8143	77.2623	1.152	1.135	289 s
G-C	0.8052	76.442	1.137	1.125	412 s
T-FC	0.7851	75.453	0.928	0.964	355 s
G-FC	0.7725	74.2623	1.088	1.083	435 s
RD	0.4676	43.228	–	0.686	281 s
RC	0.4651	42.768	0.712	–	271 s

The results indicate that the cross-attention mechanism and the Transformer module are crucial to the CAT-MFRec model, as they effectively integrate user behavior data and dynamically update user interest status. The performance differences observed when these components were removed or modified are substantial. For instance, removing the cross-attention mechanism resulted in a marked decrease in both recommendation relevance and overall ranking quality, underscoring its essential role in capturing complex user behavior interactions. Likewise, the absence of the Transformer module significantly affected the model's ability to track changes in user interest over time, leading to decreased prediction accuracy.

CAT-MFRec exhibits the highest training efficiency, the performance of T-FC remains superior to that of G-FC even when the cross-attention mechanism is disabled. primarily due to the parallel training of the two types of feedback using the Transformer. Additionally, the experiment confirms the significance of mixed feedback in enhancing the model's understanding of user preferences. Excluding mixed feedback substantially diminished the model's ability to deliver accurate recommendations. These performance differences underscore the critical contributions of each component to the overall model effectiveness.

In conclusion, these experiments not only validate the rationale behind the original model design but also highlight the specific contributions of individual components, providing valuable insights for future optimization and evolution. Future research could focus on refining the cross-attention mechanism to more effectively capture user behavior patterns or exploring alternative architectures.

4.4 Hyperparameter Experiments

Hyperparameter tuning is essential in deep learning, as selecting the right hyperparameters improves prediction accuracy, and generalization while preventing overfitting. In this study, we performed a systematic hyperparameter optimization experiment to investigate the specific impact of various hyperparameter configurations on model performance and to identify the optimal parameter combination.

To achieve this goal, we evaluated various parameters, including the dimensions of the embedding vector d, learning rates η, and the performance of the multi-head attention mechanism in the Transformer h, we also assessed the size of the hidden layer d_{hidden}, the default values for the parameters in the experiment are: $d = 128$, $\eta = 0.1$, $h = 8$, $d_{hidden} = 128$. In this study, we focused solely on evaluating the final NDCG metric. The hyperparameter settings and results are presented in the following Table 4.

Table 4: Parameter settings and results

Parameter	Settings				
d	32 0.8025	64 0.8089	128 0.8131	256 0.8032	512 0.7994
η	0.1 0.8122	0.2 0.8101	0.3 0.8004	0.4 0.7974	0.5 0.7841
h	8 0.8133	12 0.8024	16 0.8011	32 0.8007	64 0.8008
d_{hidden}	32 0.8022	64 0.8076	128 0.8129	256 0.8066	512 0.8011

The model performs best when the embedding vector dimension is 128, However, increasing the embedding dimension further results in decreased performance. This suggests that a larger embedding vector can increase model complexity and degrade performance. The model performance shows little difference between learning rates of 0.1 and 0.4; however, there is an overall downward trend, particularly at a learning rate of 0.5. This indicates that a higher learning rate may cause the model to update too rapidly, hindering convergence. The model achieves optimal performance with 8 heads, but performance slightly declines when the number of heads increases to 12 or more. This suggests that an excessive number of heads increases model complexity without yielding significant performance gains. The model performs optimally with a hidden layer size of 128; however, further increases in hidden layer size lead to a slight decline in performance. This may be attributed to the reduced generalization ability resulting from excessively large hidden layers.

The results of the hyperparameter experiments indicate that the optimal hyperparameter combination for the CAT-MFRec model includes an embedding vector dimension of 128, a learning rate of 0.1, 8 Transformer heads, and a hidden layer size of 128. These hyperparameters provide an optimal balance between model complexity and performance, enhancing the model's generalization ability. As the experimental parameters increase, the model's performance does not significantly improve and may even decline in some cases, this may result from increased complexity and computational demands associated with larger parameter values, shows that the model is a model that pays more attention to balance.

5 Conclusions

Addressing the limitations of existing recommender systems in handling user feedback, this paper proposes a hybrid model that combines both explicit and implicit user feedback. The CAT-MFRec (Cross Attention Transformer-Deep Interest Evolution Network Recommendation) model aims to enhance user experience on short video platforms. Short video platforms like TikTok and Kuaishou boast a large user base, making an effective recommendation system essential for enhancing user retention, improving engagement, and increasing watch time. This model integrates explicit and implicit user feedback to more accurately capture user interest drift and dynamic changes. Utilizing the cross-attention mechanism and Transformer architecture, CAT-MFRec effectively addresses the long-distance dependence problem in sequence data, optimizing the accuracy of feature extraction and interest prediction. Experimental results indicate that, compared to traditional recommendation methods and other deep learning models, the proposed model exhibits significant advantages in key performance indicators, including user engagement, retention rate, and viewing time in short video recommendation scenarios.

Acknowledgement: I would like to express my heartfelt gratitude to all those who contributed to this paper. Their dedication and insights were crucial in shaping the outcomes of this work.

Funding Statement: This research is partially supported by National Natural Science Foundation of China (62072416), Key Research and Development Special Project of Henan Province (221111210500), Key Technologies R&D Program of Henan Province (232102211053, 242102211071).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Jianwei Zhang, Zhishang Zhao, draft manuscript preparation: Zhishang Zhao, Zengyu Cai, data collection: Yahui Sun, analysis and interpretation of results: Yuan Feng, Liang Zhu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data and materials utilized in this review originate from publicly available databases and previously published studies, with proper citations included throughout the text. References to these sources can be found in the bibliography.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

[1] Y. H. Wang, T. J. Gu, and S. Y. Wang, "Causes and characteristics of short video platform internet community taking the tiktok short video application as an example," in *2019 IEEE Int. Conf. Consum. Elect.-Taiwan (ICCE-TW)*, Yilan, Taiwan, 2020. doi: 10.1109/ICCE-TW46550.2019.8992021.

[2] H. Ko, S. Lee, Y. Park, and A. Choi, "A survey of recommendation systems: Recommendation models, techniques, and application fields," *Electronics*, vol. 11, no. 1, 2022, Art. no. 141. doi: 10.3390/electronics11010141.

[3] E. Raad, R. Chbeir, and A. Dipanda, "User profile matching in social networks," in *2010 13th Int. Conf. Netw.-Based Inf. Syst.*, 2010, pp. 297–304.

[4] L. Yuan, H. Xia, and Q. Ye, "The effect of advertising strategies on a short video platform: Evidence from tiktok," *Ind. Manag. Data Syst.*, vol. 122, no. 8, pp. 1956–1974, 2022. doi: 10.1108/IMDS-12-2021-0754.

[5] X. Su and T. M. Khoshgoftaar, “A survey of collaborative filtering techniques,” *Adv. Artif. Intell.*, vol. 2009, no. 1, 2009, Art. no. 421425. doi: 10.1155/2009/421425.

[6] F. Zhang, “A matrix decomposition and its applications,” *Linear Multilinear A.*, vol. 63, no. 10, pp. 2033–2042, 2015. doi: 10.1080/03081087.2014.933219.

[7] M. Naumov *et al.*, “Deep learning recommendation model for personalization and recommendation systems,” 2019, *arXiv:1906.00091*.

[8] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T. Chua, “Attentional factorization machines: Learning the weight of feature interactions via attention networks,” 2017, *arXiv:1708.04617*.

[9] W. -C. Kang and J. McAuley, “Self-attentive sequential recommendation,” in *2018 IEEE Int. Conf. Data Min. (ICDM)*, Singapore, 2018, pp. 197–206.

[10] F. Sun *et al.*, “BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer,” in *Proc. 28th ACM Int. Conf. Inform. Know. Manag., CIKM ’19*, New York, NY, USA, Association for Computing Machinery, 2019, pp. 1441–1450. doi: 10.1145/3357384.3357895.

[11] G. Zhou *et al.*, “Deep interest network for click-through rate prediction,” in *Proc. 24th ACM SIGKDD Int. Conf. Know. Disc. & Data Min., KDD ’18*, New York, NY, USA, Association for Computing Machinery, 2018, pp. 1059–1068. doi: 10.1145/3219819.3219823.

[12] G. Zhou *et al.*, “Deep interest evolution network for click-through rate prediction,” *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, pp. 5941–5948, Jul. 2019. doi: 10.1609/aaai.v33i01.33015941.

[13] W. Xu, H. He, M. Tan, Y. Li, J. Lang and D. Guo, “Deep interest with hierarchical attention network for click-through rate prediction,” in *Proc. 43rd Int. ACM SIGIR*, 2020, pp. 1905–1908.

[14] Y. Feng *et al.*, “Deep session interest network for click-through rate prediction,” 2019, *arXiv:1905.06482*.

[15] S. ShuTing *et al.*, “Deep time-stream framework for click-through rate prediction by tracking interest evolution,” *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 4, pp. 5726–5733, Apr. 2020. doi: 10.1609/aaai.v34i04.6028.

[16] G. Jawaheer, P. Weller, and P. Kostkova, “Modeling user preferences in recommender systems: A classification framework for explicit and implicit user feedback,” *ACM TIIS*, vol. 4, no. 2, pp. 1–26, Jun. 2014. doi: 10.1145/2512208.

[17] R. G. Sakthivelan, P. Rjendran, and M. Thangavel, “Retraction note: A video analysis on user feedback based recommendation using A-FP hybrid algorithm,” *Multimed. Tools Appl.*, vol. 82, no. 10, 2022, Art. no. 15923. doi: 10.1007/s11042-022-13866-0.

[18] L. Tian *et al.*, “Understanding user behavior at scale in a mobile video chat application,” in *Proc. 2013 ACM Int. Joint Conf. Pervasive Ubiq. Comput., UbiComp ’13*, New York, NY, USA, Association for Computing Machinery, 2013, pp. 647–656. doi: 10.1145/2493432.2493488.

[19] T. Jia, D. Wang, and B. K. Szymanski, “Quantifying patterns of research-interest evolution,” *Nat. Hum. Behav.*, vol. 1, no. 4, 2017, Art. no. 0078. doi: 10.1038/s41562-017-0078.

[20] C. -M. Au Yeung, N. Gibbins, and N. Shadbolt, “Multiple interests of users in collaborative tagging systems,” in *Weav. Serv. People World Wide Web*, 2009, pp. 255–274.

[21] S. Zhang *et al.*, “Causerec: Counterfactual user sequence synthesis for sequential recommendation,” in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inform. Retr., SIGIR ’21*, New York, NY, USA, Association for Computing Machinery, 2021, pp. 367–377.

[22] B. Liu, D. Li, J. Wang, Z. Wang, B. Li and C. Zeng, “Integrating user short-term intentions and long-term preferences in heterogeneous hypergraph networks for sequential recommendation,” *Inform. Process. Manag.*, vol. 61, no. 3, 2024, Art. no. 103680. doi: 10.1016/j.ipm.2024.103680.

[23] K. Sun, T. Qian, T. Chen, Y. Liang, Q. V. H. Nguyen and H. Yin, “Where to go next: Modeling long- and short-term user preferences for point-of-interest recommendation,” *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 1, pp. 214–221, Apr. 2020. doi: 10.1609/aaai.v34i01.5353.

[24] J. Shao, “Effectiveness of explicit and implicit corrective feedback in a video-based scmc environment,” *Int. J. Linguist. Trans. Stud.*, vol. 3, no. 3, pp. 15–28, Aug. 2022. doi: 10.36892/ijlts.v3i3.249.

[25] D. Li *et al.*, "CARM: Confidence-aware recommender model via review representation learning and historical rating behavior in the online platforms," *Neurocomputing*, vol. 455, pp. 283–296, 2021. doi: 10.1016/j.neucom.2021.03.122.

[26] F. Daneshfar, "Enhancing low-resource sentiment analysis: A transfer learning approach," *Passer J. Basic and Appl. Sci.*, vol. 6, no. 2, pp. 265–274, 2024. doi: 10.24271/psr.2024.440793.1484.

[27] N. Ranjbar, S. Momtazi, and M. Homayoonpour, "Explaining recommendation system using counterfactual textual explanations," *Mach. Learn.*, vol. 113, no. 4, pp. 1989–2012, 2024. doi: 10.1007/s10994-023-06390-1.

[28] A. Vaswani *et al.*, "Attention is all you need," 2017, *arXiv:1706.03762*.

[29] Y. Liu, G. Sun, Y. Qiu, L. Zhang, and L. V. Gool, "Transformer in convolutional neural networks," *Comput. Sci.-Comput. Vis. Pattern Recognit.*, vol. 3, 2021. doi: 10.48550/arXiv.2106.03180.

[30] A. Rashed, S. Elsayed, and L. Schmidt-Thieme, "Context and attribute-aware sequential recommendation via cross-attention," in *Proc. 16th ACM CRS, RecSys '22*, 2022, pp. 71–80.