Proceedings of the 3rd International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/2025.24195

Fine-Grained Attribute Decoupling and Interest Denoising Network for Click-Through Rate Prediction

Zheng Li[1*], Kaiyao Zhu[2], Xijun Zhu[3]

[1] *College of Data Science, Qingdao University of Science and Technology, Qingdao, China*

[2] *College of Data Science, Qingdao University of Science and Technology, Qingdao, China*

[3] *College of Information Science and Technology, Qingdao University of Science and Technology, Qingdao, China*

**Corresponding Author. Email: lizheng5599@163.com*

Abstract. Click-through rate (CTR) prediction plays a critical role in personalized recommendation systems. Accurate CTR prediction not only enhances user experience and satisfaction but also brings substantial commercial value to online service platforms such as e-commerce. A key challenge in CTR prediction is the precise representation of user interest preferences. Existing methods mostly focus on item-level user interest modeling, neglecting the complex relationships between fine-grained attributes (such as category, brand, etc.), making it difficult to effectively address the noise problem caused by item attribute coupling and user interest drifting. Furthermore, identifying and distinguishing hard negative samples from noisy ones remains a critical issue that needs to be resolved. To address these challenges, we propose a fine-grained attribute decoupling and interest denoising network. This network effectively mitigates noise caused by attribute coupling and interest drifting through joint modeling of fine-grained attribute decoupling and interest denoising. Specifically, the network decouples item attributes to model user interests with a fine-grained attribute-aware interest denoising module, which handles noise caused by attribute coupling and user interest drifting. To further optimize user interest representation, we design a contrastive interest optimization module based on hard sample enhancement, ensuring a more accurate and comprehensive user interest representation. We conduct experiments on three real-world datasets and compare the proposed method with baseline approaches, validating its effectiveness.

Keywords: click-through rate prediction, interest denoising, attribute decoupling.

1. Introduction

Click-through rate (CTR) prediction plays a critical role in personalized recommendation systems. Accurate CTR prediction not only enhances user experience and satisfaction[1], but also brings significant commercial value to online advertising and e-commerce platforms. In recent years, with the development of deep learning technologies, feature interaction-based CTR prediction models[2-3] have achieved preliminary success through efficient feature combinations. However, these models struggle to effectively model users' historical behavior features and dynamic interest preferences; therefore,

DOI: 10.54254/2755-2721/2025.24195

sequence-based CTR prediction methods[4-8] have gradually become a research hotspot. By deeply analyzing user behavior sequences, these methods can more accurately capture users' behavior patterns and interest preferences.

Current sequence modeling research can be roughly divided into three categories: time window-based modeling, user behavior attribute-based modeling, and multi-behavior sequence-based modeling. Time window-based modeling includes short-term sequences[7-8], long-term sequences[5-6], and session sequences[4,9]; behavior attribute-based modeling consists of two types, which are one utilizing item attributes[10] and the other incorporating behavior details (such as dwell time, clicked images, view comments, etc.)[11]; while multi-behavior sequence-based modeling captures user behavior patterns through mixed sequences[12] or by splitting independent sequences based on behavior type[13-14]. Some studies also combine explicit and implicit feedback[15] to enhance modeling capabilities. Despite the progress made by these methods, they often only capture item-level related interests and fail to fully consider the complex interactions between fine-grained attributes of items. In fact, users' attention to different attributes of an item (such as brand, price, rating, etc.) when deciding whether to interact with the item can vary due to personal preferences and contextual changes. In this case, models struggle to accurately capture users' true interest in these fine-grained attributes. Therefore, current models need to better decouple these fine-grained attributes and effectively capture user preferences for each attribute.

User behavior sequences often contain complex noise interference, which makes it difficult to accurately represent users' true interest preferences. First, noise labels (such as misclicks[16]) in the samples constituting the user behavior sequence can lead to inaccurate mappings between behavior and interest. Second, user interests may drift over time[8], causing further noise. Among these noisy samples, some are semantically similar to positive samples but are labeled as negative samples, making it easy for models to mistakenly identify them as users' true interests[17]. In model training, sample quality is crucial for enhancing the model's ability to distinguish similar samples and avoid local optima. Therefore, if these noise issues are not properly addressed, the accuracy and generalization ability of CTR prediction will be significantly reduced. To address these noise issues, Bian et al.[18] attempt to correct noise in implicit feedback using explicit feedback, but explicit feedback data is often too sparse[19], and assuming that explicit feedback is entirely correct is unreasonable. Lin[20] and Zhao[21] reduce the impact of sample noise in implicit feedback through sample reweighting or sample selection methods, but they do not consider the impact of the temporal variation in user interests. Zhou et al.[7] focus on the problem of interest drifting, attempting to capture the temporal changes in user interests. However, they do not address the noise problem caused by fine-grained attribute coupling, and thus user interest representation may still be affected. Therefore, reasonably modeling user behavior sequences and effectively denoising them becomes a key challenge for improving CTR prediction performance[22].

To address these challenges, we propose the Fine-grained Attribute Decoupling and Interest Denoising Network (FADIDN). This network effectively alleviates noise issues caused by item attribute coupling and interest drifting through joint modeling of fine-grained attribute decoupling and interest denoising. Specifically, we design an interest denoising module based on fine-grained attribute awareness. By decoupling item attributes, we personalize the modeling of user interests and handle noise issues in the fine-grained attribute dimension caused by attribute coupling and interest drifting. To further optimize user interest representation, we also design a contrastive interest optimization module based on hard sample enhancement. This module constructs high-quality positive-negative sample pairs through hard negative sample mining strategies and uses contrastive learning methods to optimize the denoised user interest representation, achieving more precise personalized modeling. In summary, our main contributions are as follows:

DOI: 10.54254/2755-2721/2025.24195

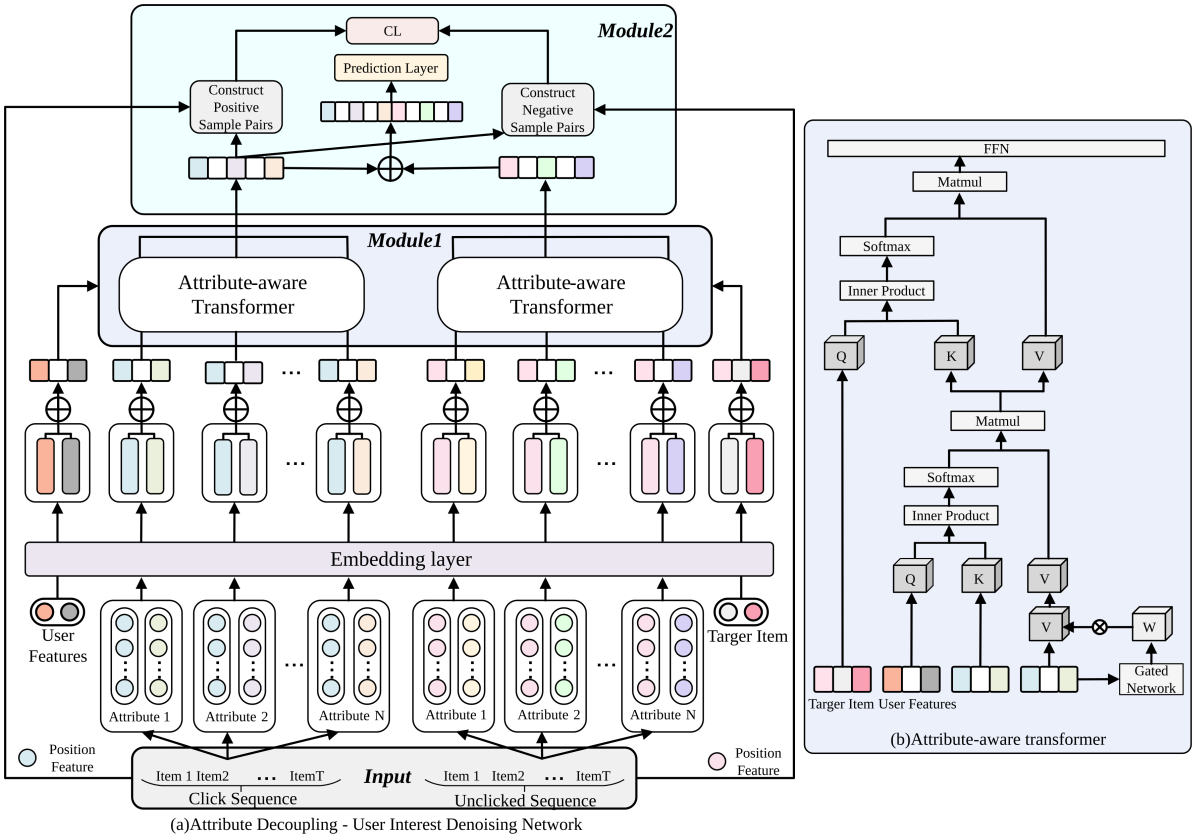
(1)We propose the Fine-grained Attribute Decoupling and Interest Denoising Network, which is the first to filter noise from user interests at the fine-grained attribute dimension. This effectively alleviates noise problems caused by attribute coupling and dynamic changes in user interests within user behavior data.

(2)We design a contrastive interest optimization strategy based on hard sample enhancement. By mining hard negative samples, we construct high-quality positive-negative sample pairs, enhancing the model's ability to distinguish similar samples, further optimizing denoised user interest representation and improving the accuracy of personalized modeling.

(3)We conduct systematic experimental evaluations on three real-world datasets. The experimental results show that FADIDN improves recommendation performance by 12.31%, 16.91%, and 9.95% compared to existing baseline models. This significant advantage demonstrates the effectiveness and practicality of the network in complex scenarios.

2. Method

In this section, we will describe FADIDN in detail. As shown in Figure 1(a), FADIDN first maps users, target items, and user behavior sequences based on item fine-grained attributes into vector representations through the feature embedding layer. Then, the fine-grained attribute-aware interest denoising module models the user's personalized interests based on the fine-grained attribute sequences of the items and addresses the noise caused by attribute coupling and user interest drifting. Next, the contrastive interest optimization module based on hard sample enhancement constructs high-quality positive and negative sample pairs using a hard negative sample mining strategy, and further optimizes the denoised user interest representations through contrastive learning. This enables the model to more accurately capture user interests and ultimately achieve click-through rate prediction.



(a)Attribute Decoupling - User Interest Denoising Network

DOI: 10.54254/2755-2721/2025.24195

Figure 1: Fine-grained Attribute Decoupling and Interest Denoising Network Framework.

2.1. Problem Definition

First, we define some symbols used in this paper. We use bold uppercase letters (e.g., $\boldsymbol{W}$) to represent matrices, bold lowercase letters (e.g., $\boldsymbol{w}$) to represent vectors, non-bold letters (e.g., w/W) to represent scalars, and Greek letters (e.g., β) to represent parameters. Suppose we have a set of users U and items I , where $u \in U$ represents any user and $i \in I$ represents any item. We comprehensively consider M attributes of user u (such as user ID, gender, age, etc.), and model two types of behavior sequences: the click sequence C and the unclick sequence V . For each item i , we model N fine-grained attributes n (e.g., item ID, category, brand, etc.). Based on these fine-grained attributes n , we construct fine-grained attribute click sequences C^n and unclick sequences V^n . The objective of FADIDN is to learn a function $\hat{y} = f\left(M_{en}\left(M_{de}\left(U, C^n, V^n, i_{tar}\right)\right)|\theta\right)$ to predict the click probability of user u on the target item $i_{tar} \in I$. Here, M_{de} represents the fine-grained attribute-aware interest denoising module, which aims to model the user's personalized interest and effectively address the noise caused by attribute coupling and interest drifting; M_{en} represents the contrastive interest optimization module based on hard sample enhancement, which constructs high-quality positive and negative sample pairs through hard negative sample mining, enhances the relationships between fine-grained attributes, and further optimizes the denoised user interest representations using contrastive learning, improving the accuracy of the user interest representation; θ represents the parameters of FADIDN; and $\hat{y}$ takes values between 0 and 1.

2.2. Embedding Layers

For each user u in the user set U and each item i in the item set I , we construct vector representations following [18]. The representation of user u consists of multiple concatenated features: $E_u = \left[e_u^{id}, e_u^{gen.}, \cdots, e_u^{age}\right] \in \mathbb{R}^{M \times E}$, where M is the number of user attributes and E is the embedding dimension. For item i ,its representation under fine-grained attribute n is denoted as e_i^n ,and the target item's representation is $e_{i_{tar}}^n$. The click sequence of the user on fine-grained attribute n is represented as $E_c^n = \left[e_{c_1}^n, e_{c_2}^n, \ldots, e_{c_T}^n\right] \in \mathbb{R}^{T \times E}$, and the unclick sequence is $E_v^n = \left[e_{v_1}^n, e_{v_2}^n, \ldots, e_{v_T}^n\right] \in \mathbb{R}^{T \times E}$, where T is the maximum sequence length. Additionally, to model the positional information of items in the behavior sequence, we take the click sequence E_c^n as an example and describe the composition of the positional features in detail. First, we calculate the time difference between the timestamp of each item in the sequence and the timestamp of the target item [1], and then construct the corresponding position sequence $P = [p_1, p_2, \ldots, p_T]$. Next, through the embedding layer, we obtain its feature embedding $E_c^p \in \mathbb{R}^{T \times E}$. Finally, we add the positional feature embedding E_c^p and the fine-grained attribute n click sequence feature embedding E_c^n , obtaining the feature embedding of the fine-grained attribute click sequence with fused positional information $E_{c_p^n} \in \mathbb{R}^{T \times E}$. Similarly, the embedding representation of the unclick sequence for fine-grained attribute n is $E_{u_p^n} \in \mathbb{R}^{T \times E}$.

2.3. Fine-Grained Attribute-Aware Interest Denoising Module

User interests are influenced by fine-grained item attributes (e.g., category, brand, price, etc.). However, existing click-through rate prediction models[4,8] typically capture user interests at the item level and fail to fully consider the coupling relationships between item attributes and the noise issues

DOI: 10.54254/2755-2721/2025.24195

arising from this. Therefore, to accurately extract user interest preferences, we have designed a fine-grained attribute-aware interest denoising module. This module models the user's personalized interests and handles the noise caused by attribute coupling and interest drifting by using an improved Transformer model, based on the fine-grained attribute sequence of the items. Next, we take $\boldsymbol{E}_{c_p^n}$ as an example to describe the specific implementation of this module in detail.

The contribution of fine-grained item attributes varies dynamically in different scenarios. For example, a user's recent interests are reflected in specific items (e.g., item ID), while long-term interests are more evident in preferences for attributes like "category" or "brand." To address this, we introduce a gating network (i.e., a multi-layer perceptron) to dynamically weight each fine-grained attribute click sequence, capturing the user's behavior patterns on fine-grained attributes and adaptively modeling the user's long-term and short-term interests. The weight w_c^n for each fine-grained attribute click sequence can be calculated using the following gating network:

$$w_c^n = softmax(\sigma(W_l g_{l-1} + b_l))$$

Here, $g_0 = \boldsymbol{E}_{c_p^n}$, and g_{l-1} represents the output of the $l-1$ -th layer. W_l and b_l are the learnable weight matrix and bias vector of the layer, respectively, and $\sigma(\cdot)$ denotes the $LeakyReLU$ activation function. The dynamic weight w_c^n obtained through the gating network is element-wise multiplied (Hadamard product) with $\boldsymbol{E}_{c_p^n}$ to obtain the weighted fine-grained attribute click sequence. This weighted sequence is then input into the multi-head attention mechanism, where it is combined with the user profile E_u to model the personalized user interests. Each attention head $head_i$ models the user's attention patterns over the same fine-grained attribute click sequence across different subspaces:

$$head_i = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

The query vector $Q = E_u W^Q$ represents all features except for the user's historical behavior, the key vector$K = (E_{c_p^n} \odot w_c^n)W^K$, and the value vector$V = E_{c_p^n} W^V$both originate from the same fine-grained attribute sequence. The personalized interest representation o_c^n for the user in each fine-grained attribute click sequence is obtained by concatenating the outputs of each attention head as follows:

$$o_c^n = Concat(head_1, head_2, \ldots, head_h)W_F$$

Here, h represents the number of heads in the multi-head attention mechanism, and W_F is the linear transformation matrix.

User behavior sequences are often influenced by interest drifting, one significant cause of which is that users exhibit different levels of interest in fine-grained attributes of items across different time periods. This variation in interest can lead to short-term fluctuations in user click behavior. To address this, we introduce a multi-head target attention mechanism, which aligns the target item with the fine-grained attribute click sequence and combines timestamp information and relative position information from the sequence to filter out noise unrelated to the user's interests. The denoised interest representation f_c for the user in the click sequence is obtained after processing the output of the multi-head target attention through a point-wise feed-forward neural network (FFN):

$$f_c = FFN(Concat(head_1, head_2, \ldots, head_N)W_s)$$

DOI: 10.54254/2755-2721/2025.24195

Here, $head_n$ can be obtained in a similar manner to Equation (2), but in this case, $Q = e^n_{i_{tar}} W^{Q*}$ represents the feature embedding vector of the target item, $K = o^n_c W^{K*}$, and $V = o^n_c W^{V*}$, where W_s is the linear transformation matrix. Similarly, the denoised interest representation f_v for the unclick behavior sequence is obtained in the same manner.

2.4. Hard Sample Enhancement-Based Contrastive Interest Optimization Module

To enhance the relationships between fine-grained attributes and further optimize the representation of user interests by mining latent interest information in the click behavior sequence, we design a contrastive learning strategy based on hard negative sample mining for the sample-enhanced interest optimization module. This module constructs high-quality positive-negative sample pairs and utilizes contrastive learning to strengthen the relationships between fine-grained attributes, thereby optimizing the denoised user interest representation to enhance its comprehensiveness.

During the positive-negative sample construction process, we use the denoised interest representation f_c output in Section 3.3 as the anchor point, and select positive and negative samples from both the click sequence and the unclick sequence. To enhance the relationships between fine-grained attributes and ensure the consistency of f_c with the selected items in terms of semantic features, the embeddings of all items are represented by the concatenated features of each attribute. Specifically, we select the top-k items from the click sequence whose semantic similarity with f_c is greater than η, and construct the positive sample set $L = \{e_1, e_2, \ldots, e_k\}$. Here, η is a hyperparameter, and the similarity between f_c and e_i is calculated using thefollowing formula:

$$\mathrm{sim}(f_c, e_i) = \frac{f_c \cdot e_i}{\|f_c\| \|e_i\|}$$

Here, $e_i \in L$. Then, we construct the positive sample pair set $z^+ = \{(f_c, e_1), (f_c, e_2), \ldots, (f_c, e_k)\}$ by pairing the item embeddings from L with f_c.

In terms of negative sample selection, unlike the previous method of random sampling from the unclick sequence[23], we adopt a strategy combining hard negative sample mining[24] and random negative sampling. Using hard negative sample mining to construct negative samples not only helps FADIDN identify those hard-to-classify negative samples, but also prevents overfitting on simple negative samples, promoting more discriminative feature learning. Specifically, we select the top-q items from the unclick sequence that are similar to f_c, forming the hard negative sample set $J = \{e_1, e_2, \ldots, e_q\}$ in a manner similar to the positive sample selection. Then, we pair the item embeddings from J with f_c, constructing the hard negative sample pair set $z^-_{hard} = \{(f_c, e_1), (f_c, e_2), \ldots, (f_c, e_q)\}$. Next, to increase the diversity of negative samples, we also employ a random negative sampling strategy, randomly selecting negative samples from the unclick sequence, and pairing their embeddings with f_c to form the random negative sample pair set z^-_{ran}. Finally, we take the union of z^-_{hard} and z^-_{ran} to obtain the final negative sample pair set z^-:

$$z^- = z^-_{hard} \cup z^-_{ran}$$

Based on the positive and negative sample pairs constructed above, we use the InfoNCE contrastive loss function L_{cl} to optimize the feature representation learning of FADIDN:

DOI: 10.54254/2755-2721/2025.24195

$$L_{cl} = -\frac{1}{N}\sum_{i=1}^{N} log\frac{exp(sim(z_i^+)/\tau)}{exp(sim(z_i^+)/\tau)+\sum_{i=1}^{K} exp(sim(z_i^-)/\tau)}$$

Here, $z_i^+ \in z^+$ and $z_i^- \in z^-$, τ is the temperature parameter, and K is the number of negative samples. In the click-through rate prediction task for attribute decoupling and interest denoising, the standard cross-entropy loss L_{ce} is used:

$$L_{ce} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i log(\hat{y}_i) + (1-y_i) log(1-\hat{y}_i)\right]$$

Here, y_i and $\hat{y}_i$ represent the true label and predicted value of the i -th sample, respectively. Finally, we jointly optimize the InfoNCE contrastive loss and the cross-entropy loss. The joint optimization objective is defined as L :

$$L = L_{ce} + \lambda L_{cl}$$

Here, λ is a hyperparameter.

3. Experiments

In this section, we validate the effectiveness of the FADIDN network through experiments on different datasets and compare it with current state-of-the-art methods. The experimental design and discussion will focus on the following four questions:

RQ1:How does FADIDN perform compared to other CTR prediction models?

RQ2:What is the impact of each module in FADIDN on the model's performance?

RQ3:How does the parameter setting of FADIDN affect the model's performance?

RQ4:Does FADIDN effectively implement denoising?

3.1. Experimental Setups

Dataset: We evaluate the proposed FADIDN on three publicly available datasets with different experimental settings. The Amazon Dataset[25] includes product reviews and related metadata from the Amazon website, covering multiple product categories. In this experiment, we selected the Beauty and Electronics categories. We treat the product ratings as indicators of user interest, where ratings of 4 and 5 are considered as user clicks, and ratings of 0-3 are considered as unclick behaviors. Another dataset used is the MovieLens Dataset[26], which includes user ratings and preferences for movies. Similarly, the user ratings for movies are treated as interest indicators, with ratings of 4 and 5 representing clicks, and ratings of 0-3 representing unclicks. The statistical information for each dataset is shown in Table 1.

Table 1 Statistical information of the dataset

Dataset	User	Goods	Categories	Samples
Beauty.	22363	12101	220	198602
Eletronics.	192403	63001	801	1689188
Movielens.	138493	27278	21	20000263

Comparison Methods: To demonstrate the effectiveness of the proposed FADIDN in sequence modeling and interest denoising, we compare FADIDN with state-of-the-art click-through rate

DOI: 10.54254/2755-2721/2025.24195

prediction models.

DIN[8]: DIN adaptively adjusts user behavior feature weights using an attention mechanism.

DIEN[7]: DIEN learns the temporal dependencies of the sequence with GRU to capture changes in user interest.

DSIN[4]: DSIN divides user historical behaviors into sessions and uses Transformer to learn the behavior sequences within each session.

AutoInt[27]: AutoInt utilizes a multi-head attention mechanism to construct higher-order features, enhancing the accuracy of CTR prediction.

TFNet[28]: TFNet introduces operational tensors and uses multi-layer matrices to capture interactions between features, revealing semantic space differences.

FRNET[22]: FRNET leverages an attention mechanism and multi-layer perceptrons to learn feature relationships and contextual information, enhancing data representation capabilities.

DFN[14]: DFN introduces explicit negative feedback sequences and uses a basic attention mechanism to denoise implicit negative feedback sequences.

DUMN[18]: DUMN denoises user interests by modeling four types of user behavior sequences and incorporating user profiles to mine long-term interests.

RLNF[21]: RLNF uses reinforcement learning to select effective negative samples.

AutoDenoise[20]: AutoDenoise utilizes deep reinforcement learning and a two-stage optimization strategy to automatically select noise-free data subsets.

Evaluation Metrics: We use AUC (Area Under the ROC Curve)[29] and RelaImpr[30] as performance evaluation metrics. Among them, RelaImpr measures the performance improvement of the model relative to the baseline model, typically expressed as a percentage. The calculation formula is as follows:

$$\text{RelaImpr} = \left(\frac{\text{AUC(measured model)} - 0.5}{\text{AUC(base model)} - 0.5} - 1 \right) \times 100\%$$

Parameter Settings: FADIDN is built based on the PyTorch framework, with a maximum sequence length of 50 and an embedding layer output dimension of 16. The gating network consists of two fully connected layers (256 and 16 dimensions), and both the point-wise feed-forward network and the prediction layer are also two fully connected layers (256 and 128 dimensions). The number of heads in the multi-head attention mechanism is set to 2. During the training phase, the learning rate is set to 0.01, and the Adam optimizer is used. All hyperparameters are determined through grid search, and early stopping is employed to prevent overfitting.

3.2. Comparative Experiments

Table 2 The AUC and RelaImpr performance of FADIDN compared to the baseline models on three datasets, with the best results highlighted in bold.

Datesets	Beauty.		Eletronics.		MovieLens.	
Model	AUC	RelaImpr	AUC	RelaImpr	AUC	RelaImpr
DIN	0.8127	0%	0.7962	0%	0.7926	0%
DIEN	0.8152	0.80%	0.7995	1.14%	0.7939	0.44%
DSIN	0.8176	1.57%	0.8014	1.76%	0.7968	1.44%
AutoInt	0.8213	2.75%	0.8014	1.76%	0.7992	2.26%

DOI: 10.54254/2755-2721/2025.24195

TFNet	0.8241	3.65%	0.8073	3.75%	0.8036	3.76%
FRNet	0.8314	5.98%	0.8164	6.82%	0.8053	4.34%
DFN	0.8242	3.68%	0.8081	4.02%	0.8037	3.79%
DUMN	0.8263	4.35%	0.8123	5.44%	0.8042	3.96%
RLNF+DSIN	0.8352	7.20%	0.8213	8.47%	0.8086	5.47%
RLNF+FRNet	0.8342	6.88%	0.8201	8.07%	0.8073	5.02%
AutoDenoise+DSIN	0.8402	9.79%	0.8321	12.12%	0.8125	6.80%
AutoDenoise+FRNet	0.8357	7.36%	0.8292	11.14%	0.8103	6.05%
FADIDN	0.8512	12.31%	0.8463	16.91%	0.8217	9.95%

As shown in Table 2, RLNF+DSIN and AutoDenoise+DSIN are combined models of denoising components with sequence modeling models, while RLNF+FRNet and AutoDenoise+FRNet are combined models of denoising components with non-sequence modeling models. We compared FADIDN with 12 other models on three datasets and drew the following conclusions:

(1)The average results from five random experiments indicate that FADIDN outperforms the current state-of-the-art models on all datasets, validating its effectiveness. Compared to the baseline models, FADIDN improved performance by 12.31%, 16.9%, and 9.95% on the Amazon (Beauty), Amazon (Electronics), and MovieLens datasets, respectively. Notably, on the Amazon dataset, where there are rich attribute categories, FADIDN's advantage is more pronounced, demonstrating its ability to handle complex data.

(2)The combination of sequence modeling and denoising components achieves better performance, indicating that the presence of noise indeed affects the model's ability to capture user interests. FADIDN models personalized user interests through sequence modeling while simultaneously addressing the noise issues introduced by fine-grained attribute coupling. Its performance significantly outperforms models that rely solely on sequence modeling or feature interaction representations.

3.3. Ablation Experiments

To study the effectiveness of the components in the FADIDN network, we conducted extensive ablation studies on three datasets and the results are shown in Figure 2:

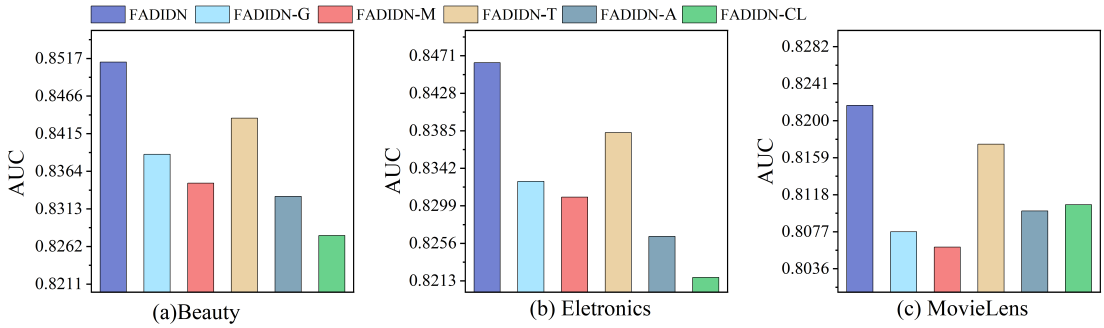


Figure 2: Ablation study results.

Impact of the Fine-Grained Attribute-Aware Interest Denoising Module: To validate the effectiveness of the components in this module, we designed three variants: FADIDN-G, FADIDN-M, and FADIDN-T, and evaluated them on the Amazon (Beauty), Amazon (Electronics), and MovieLens datasets. The results are shown in Figure 2.

DOI: 10.54254/2755-2721/2025.24195

FADIDN-G removes the gating network, leading to a performance drop of 1.46%, 1.61%, and 1.7%, demonstrating the critical role of the gating network in capturing user behavior patterns and adaptively modeling both short-term and long-term interests.

FADIDN-M removes the multi-head target attention mechanism, resulting in a performance drop of 0.89%, 0.94%, and 0.52%, highlighting the key role of this mechanism in addressing the noise problem caused by interest drifting.

FADIDN-T removes the improved Transformer, with a performance drop of 1.93%, 1.82%, and 1.91%, underscoring its ability to capture personalized user interest preferences and handle noise effectively.

Impact of the Sample-Enhanced Interest Optimization Module: To validate the effectiveness of the components in this module, we designed two variants: FADIDN-A and FADIDN-CL, and evaluated them on the Amazon (Beauty), Amazon (Electronics), and MovieLens datasets. The results are shown in Figure 2.

FADIDN-A removes the existing positive-negative sample pair construction strategy, replacing it with positive sample pairs formed by pairing the target item with items in the click sequence, and negative sample pairs randomly sampled from the unclick sequence within the same batch. The performance drops by 2.14%, 2.35%, and 1.42%, validating the importance of constructing high-quality sample pairs for model performance.

FADIDN-CL removes the contrastive learning algorithm, resulting in a performance drop of 2.76%, 2.91%, and 1.34%, demonstrating the significant role of this algorithm in optimizing sample similarity and discriminability.

3.4. Parameter Experiments

This section investigates the impact of various hyperparameters on the performance of FADIDN. The adjustment of these hyperparameters has a direct effect on the model's prediction accuracy and generalization ability.

Network Depth: The network depth includes the number of gating network layers and the number of feed-forward neural network (FNN) layers, which are used to capture feature relationships at different levels. As shown in Table 3, the optimal choice for both the number of gating network layers and the FNN layers is 2 layers, demonstrating consistency in the design of the deep structure.

Table 3 The AUC performance of neural networks with different depths on three datasets.

Layers	Network	Beauty.	Eletronics.	Movielens.
1	Gate natework	0.8316	0.8302	0.8052
	FNN	0.8325	0.8310	0.8075
2	Gate natework	0.8391	0.8325	0.8102
	FNN	0.8412	0.8364	0.8138
3	Gate natework	0.8331	0.8314	0.8063
	FNN	0.8379	0.8330	0.8125

Number of Heads in the Multi-Head Attention Mechanism: In the fine-grained attribute-aware interest denoising module, the number of heads in the multi-head attention mechanism of the Transformer determines the model's ability to capture feature information across multiple subspaces. Increasing the number of heads allows for more comprehensive learning of sequence features, but it also increases computational resources and the risk of overfitting. Therefore, an appropriate number of

DOI: 10.54254/2755-2721/2025.24195

heads needs to be chosen for feature learning. As shown in Table 4, the optimal performance is achieved when the number of heads is set to 4.

Table 4 The AUC performance of multi-head attention with different numbers of heads on three datasets.

heads	Beauty.	Eletronics.	MovieLens.
2	0.8321	0.8316	0.8098
4	0.8423	0.8352	0.8115
8	0.8387	0.8347	0.8106
16	0.8326	0.8292	0.8083

Number of Item Attributes: Increasing the number of item attributes can provide richer information but also leads to higher computational costs and the curse of dimensionality. In e-commerce scenarios, where there are hundreds of attribute features, calculating all of them would be too costly. A reasonable number of attributes can effectively enhance the model's understanding of features, thereby improving prediction performance. As shown in Figure 3, when the number of attributes is between 0 and 6, the model's performance improves significantly. However, between 6 and 10 attributes, the performance improvement is marginal, while the storage requirements increase substantially. Therefore, we set the number of attributes to 7.

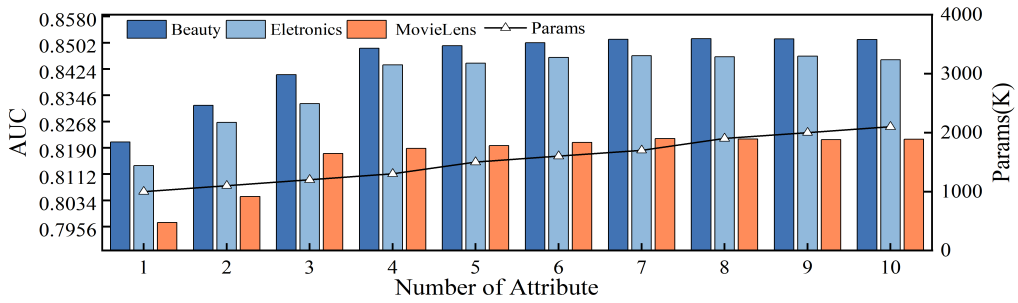


Figure 3 The number of attributes N.

Joint Loss Weight $\boldsymbol{\lambda}$: In the joint optimization loss function formula, λ represents the weight of the InfoNCE contrastive loss. As shown in Figure 4(a), different datasets require different values for λ : in the Beauty dataset, the model optimizes feature representation more effectively when $\lambda = 0.21$; in the Electronics dataset, the optimal value of λ is 0.34. In high-noise scenarios, such as the MovieLens dataset, the model performs better when $\lambda = 0.52$.

InfoNCE Contrastive Loss Parameter: The temperature parameter τ controls the distribution range of the similarity between positive and negative samples in the contrastive learning loss. As shown in Figure 4(b), the optimal value of τ differs for different datasets. In the Beauty dataset, when $\tau = 0.2$, the model achieves the best performance by balancing the optimization of positive and negative sample similarities. When $\tau > 0.2$, the distinction between positive and negative samples decreases, leading to weakened model performance. Similarly, in the Electronics dataset, the optimal τ value is 0.26, and in the MovieLens dataset, the optimal τ value is 0.31. When τ exceeds these thresholds, model performance declines.

The Number of Negative Samples K: K determines the strength of optimization between positive and negative sample pairs. As shown in Figure 4(c), in the Amazon (Beauty) dataset, when K=256, the

DOI: 10.54254/2755-2721/2025.24195

number of negative samples is suitable, effectively balancing optimization performance and computational cost. In the Amazon (Electronics) dataset, the optimal value of K is 512, while in the MovieLens dataset, K=1024 is optimal. As K increases, overfitting occurs, and computational cost significantly increases, affecting model performance.

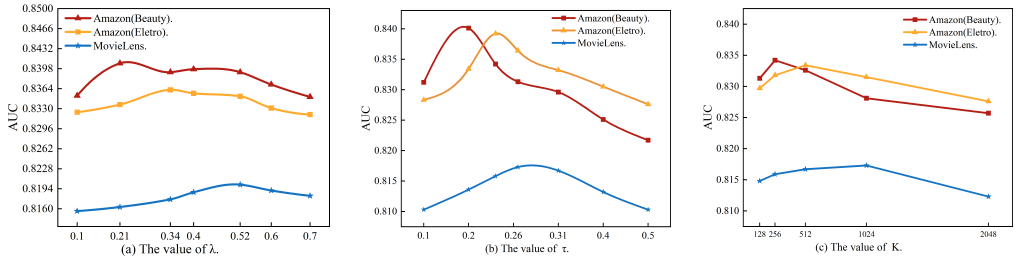


Figure 4 Analysis of the loss function hyperparameters.

3.5. Visualization Analysis

In this section, we use visualization analysis to investigate the handling capabilities of FADIDN in the following three key aspects:

User Interest Noise: In Figure 5, we present the interest representation for the same user's click and unclick behaviors. The noise is effectively filtered in both the click and unclick behavior representations, allowing the user's interest preferences to be more clearly presented.

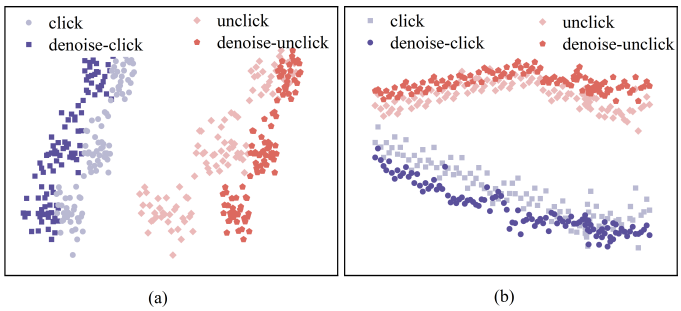
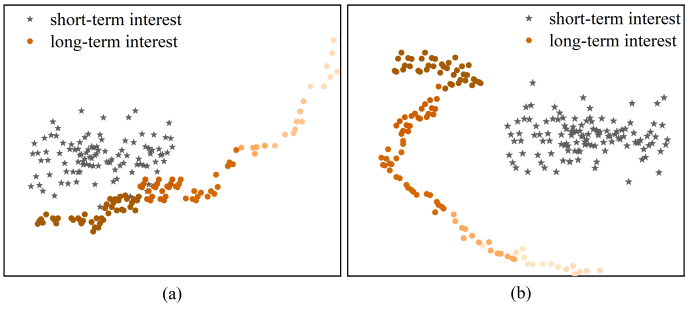


Figure 5 Visualization representation of user interest and noise.

User Long-Term and Short-Term Interests: In Figure 6, we visualize the long-term interest representation and short-term interest representation. It is clear that the model distinguishes the user's long-term and short-term interests. Compared to the distribution of the user's short-term interests, the distribution of long-term interests is more scattered.



DOI: 10.54254/2755-2721/2025.24195

Figure 6 Visualization of user long-term and short-term interests.

Personalized Weight Representations for Different Users: In Figure 7, we visualize the vector representations of several users. It is evident that, before personalization weighting, the vector representations of users are very similar. After applying personalized weighting, the vector representations of users show more distinct differences, which helps the model further understand each user's unique interests and behavior patterns.

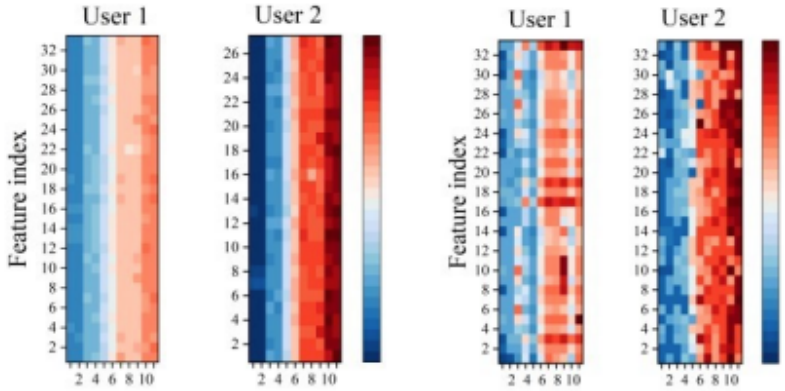


Figure 7 Visualization of personalized weight distributions across different users.

4. Conclusion

FADIDN, through its fine-grained attribute-decoupled user interest modeling approach, enhances the model's ability to capture user interest in the fine-grained attributes of items, overcoming the limitation of existing click-through rate prediction methods that focus only on item-level user interests. Moreover, FADIDN effectively addresses the noise problem caused by attribute coupling and user interest drifting, improving click prediction performance through both denoising and enhancement. Future research could further explore how to optimize the model on larger-scale datasets, improving its computational efficiency and generalization ability.

References

[1]Chen, Qiwei, et al. "Behavior sequence transformer for e-commerce recommendation in alibaba." Proceedings of the 1st international workshop on deep learning practice for high-dimensional sparse data. 2019.

[2]Guo, Huifeng, et al. "DeepFM: a factorization-machine based neural network for CTR prediction." arXiv preprint arXiv:1703.04247 (2017).

[3]Wang, Ruoxi, et al. "Deep & cross network for ad click predictions." Proceedings of the ADKDD'17. 2017. 1-7.

[4]Feng, Yufei, et al. "Deep session interest network for click-through rate prediction." arXiv preprint arXiv:1905.06482 (2019).

[5]Pi, Qi, et al. "Practice on long sequential user behavior modeling for click-through rate prediction." Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019.

[6]Pi, Qi, et al. "Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction." Proceedings of the 29th ACM International Conference on Information & Knowledge Management. 2020.

DOI: 10.54254/2755-2721/2025.24195

[7]Zhou, Guorui, et al. "Deep interest evolution network for click-through rate prediction." Proceedings of the AAAI conference on artificial intelligence. Vol. 33. No. 01. 2019.

[8]Zhou, Guorui, et al. "Deep interest network for click-through rate prediction." Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. 2018.

[9]Lei, Chenyi, Shouling Ji, and Zhao Li. "Tissa: A time slice self-attention approach for modeling sequential user behaviors." The World Wide Web Conference. 2019.

[10]Xu, Weinan, et al. "Deep interest with hierarchical attention network for click-through rate prediction." Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval. 2020.

[11]Gu, Yulong, et al. "Hierarchical user profiling for e-commerce recommender systems." Proceedings of the 13th international conference on web search and data mining. 2020.

[12]Wu, Chuhan, et al. "Feedrec: News feed recommendation with various user feedbacks." Proceedings of the ACM Web Conference 2022. 2022.

[13]Gu, Yulong, et al. "Deep multifaceted transformers for multi-objective ranking in large-scale e-commerce recommender systems." Proceedings of the 29th ACM international conference on information & knowledge management. 2020.

[14]Yang, Yatao, et al. "Finn: Feedback interactive neural network for intent recommendation." Proceedings of the Web Conference 2021. 2021.

[15]Xie, Ruobing, et al. "Deep feedback network for recommendation." Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence. 2021.

[16]Lv, Fuyu, et al. "Xdm: Improving sequential deep matching with unclicked user behaviors for recommender system." International Conference on Database Systems for Advanced Applications. Cham: Springer International Publishing, 2022.

[17]Wang, Wenjie, et al. "Denoising implicit feedback for recommendation." Proceedings of the 14th ACM international conference on web search and data mining. 2021.

[18]Bian, Zhi, et al. "Denoising user-aware memory network for recommendation." Proceedings of the 15th ACM conference on recommender systems. 2021.

[19]Wang, Zongwei, et al. "Efficient bi-level optimization for recommendation denoising." Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining. 2023.

[20]Lin, Weilin, et al. "Autodenoise: Automatic data instance denoising for recommendations." Proceedings of the ACM Web Conference 2023. 2023.

[21]Zhao, Pu, et al. "RLNF: reinforcement learning based noise filtering for click-through rate prediction." Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2021.

[22]Han, Ruidong, et al. "Enhancing CTR Prediction through Sequential Recommendation Pre-training: Introducing the SRP4CTR Framework." Proceedings of the 33rd ACM International Conference on Information and Knowledge Management. 2024.

[23]Lv, Fuyu, et al. "Unclicked User Behaviors Enhanced SequentialRecommendation." arXiv preprint arXiv:2010.12837 (2020).

[24]Yang, Haoran, et al. "Generating counterfactual hard negative samples for graph contrastive learning." Proceedings of the ACM web conference 2023. 2023.

[25]Ren, Kan, et al. "Learning multi-touch conversion attribution with dual-attention mechanisms for online advertising." Proceedings of the 27th acm international conference on information and knowledge management. 2018.

[26]Harper, F. Maxwell, and Joseph A. Konstan. "The movielens datasets: History and context." Acm transactions on interactive intelligent systems (tiis) 5.4 (2015): 1-19.

DOI: 10.54254/2755-2721/2025.24195

[27]Song, Weiping, et al. "Autoint: Automatic feature interaction learning via self-attentive neural networks." Proceedings of the 28th ACM international conference on information and knowledge management. 2019.

[28]Wu, Shu, et al. "Tfnet: Multi-semantic feature interaction for ctr prediction." Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval. 2020.

[29]Fawcett, Tom. "An introduction to ROC analysis." Pattern recognition letters 27.8 (2006): 861-874.

[30]Yan, Ling, et al. "Coupled group lasso for web-scale ctr prediction in display advertising." International conference on machine learning. PMLR, 2014.