IUI: Intent-Enhanced User Interest Modeling for Click-Through Rate Prediction

Mao Pan*
JD.com, Inc.
Beijing, China
panmao5@jd.com

Tao Yu*
JD.com, Inc.
Beijing, China
yutao73@jd.com

Kun Zhou†
JD.com, Inc.
Beijing, China
zhoukun32@jd.com

Zheng Li
JD.com, Inc.
Beijing, China
lizheng23@jd.com

Dongyue Wang
JD.com, Inc.
Beijing, China
wangdongyue@jd.com

Zhuoye Ding
JD.com, Inc.
Beijing, China
dingzhuoye@jd.com

Xiwei Zhao
JD.com, Inc.
Beijing, China
zhaoxiwei@jd.com

Sulong Xu
JD.com, Inc.
Beijing, China
xusulong@jd.com

ABSTRACT

Click-Through Rate (CTR) prediction is becoming increasingly vital in many industrial applications, such as recommendations and online advertising. How to precisely capture users' dynamic and evolving interests from previous interactions (e.g., clicks, purchases, etc.) is a challenging task in CTR prediction. Mainstream approaches focus on disentangling user interests in a heuristic way or modeling user interests into a static representation. However, these approaches overlook the importance of users' current intent and the complex interactions between their current intent and global interests. To address these concerns, in this paper, we propose a novel intent-enhanced user interest modeling for click-through rate prediction in large-scale e-commerce recommendations, abbreviated as IUI. Methodologically, different from existing works, we consider users' recent interactions to be inspired by their implicit intent and then leverage an intent-aware network to model their current local interests in a more precise and fine-grained manner. In addition, to obtain a more stable co-dependent global and local interest representation, we employ a co-attention network capable of activating the corresponding interest in global-level interactions and capturing the dynamic interactions between global- and local-level interaction behaviors. Finally, we incorporate self-supervised learning into the model training by maximizing the mutual information between the global and local representations obtained via the above two networks to enhance the CTR prediction performance. Compared with existing methods, IUI benefits from the different granularity of user interest to generate a more accurate and comprehensive preference representation. Experimental results demonstrate that the proposed model outperforms previous state-of-the-art methods in various metrics on three real-world datasets. In addition, an online A/B test deployed on the JD recommendation platforms shows a promising improvement across multiple evaluation metrics.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; **Personalization**.

KEYWORDS

Recommender systems, Click-Through Rate Prediction, User behavior modeling

ACM Reference Format:
Mao Pan, Tao Yu, Kun Zhou, Zheng Li, Dongyue Wang, Zhuoye Ding, Xiwei Zhao, and Sulong Xu. 2023. IUI: Intent-Enhanced User Interest Modeling for Click-Through Rate Prediction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3583780.3614939

1 INTRODUCTION

Recommender systems (RSs) have been widely used to help users mitigate information overload and render valid information in various applications [9, 36]. Click-Through Rate (CTR) Prediction is the fundamental recommendation task for e-commerce and online streaming services, which aims to estimate the probability of a user clicking the candidate item [13, 24]. Consequently, modeling CTR prediction has been widely focused on by both academia and industry [42].

An effective CTR model should be capable of exploiting a user's intricate and evolving preferences, which are a variety of users' global and local interests [31, 41]. In practice, local interests are generally regarded as subsets of global interests that can be swiftly substituted by other local interests. Moreover, global interests tend to be relatively stable and more comprehensive than local interests. For instance, an Apple fan is always likely to browse Apple products (global interest), while recently he may also have shown an interest in clothes (local interest). However, how to achieve an effective CTR model remains a considerable challenge. Generally, most existing CTR prediction methods fall into two categories: conventional and deep learning-based models. The former (conventional CTR prediction) is primarily concerned with modeling feature interactions [26, 28], whereas the latter is concerned with modeling user

*Both authors contributed equally to this research.
†Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3614939

behavior [40, 43]. Typically, in feature interaction-based methods, Factorization Machines [12, 15, 26] have been widely applied in the industry. Following that, inspired by the success of deep learning in CV and NLP, deep learning-based models, such as DIN [44], DSIN [7], and DMT [11], are proposed to mine the user's interests based on their history of behavior sequences.

Despite their effectiveness, existing approaches still suffer from the following challenges: Firstly, mainstream CTR prediction methods mainly focus on modeling more comprehensive users' interests from a macro perspective while ignoring the importance of users' current intent. As the ancients say, "After even just three days' absence, a scholar must be regarded with new eyes." Moreover, according to psychological theory [1, 2], user actions are driven by a set of intents. Furthermore, the lifetime of different intents may be quite different, i.e., some of them may last for a long time with lots of interaction behaviors, while others may not. As illustrated in Figure 1, without considering the user's current intent, the iPhone in the collection of candidate items will be ranked in front of trousers because of the large number of electronic products, even if the user's current intent is clothes and accessories. Consequently, we leverage the user's local interaction behaviors inspired by the user's current implicit intent to model the user's current local interests in a more precise and fine-grained manner, where the user's local interaction behaviors are the most recent T interactions and the global interactions are the user history interaction sequences including the local interactions. Secondly, most existing CTR approaches either model the local and global interactions as a whole, which may make local interests overwhelmed by global interests, or concatenate/weighted-pooling the users' global and local interests mechanically without considering the internal relationship between the user's global and local interests. Typically, when users' local interactions are sparse or even nonexistent, global interactions will serve as a supplement to obtain more stable and fine-grained local interests. Besides, local interactions have also been impacted by global interactions in a relatively recent period. For example, if a user clicks a pair of shoes in the current local interactions, the global interactions related to clothes and accessories should be activated, and then the ranking order of the candidate items should be determined.

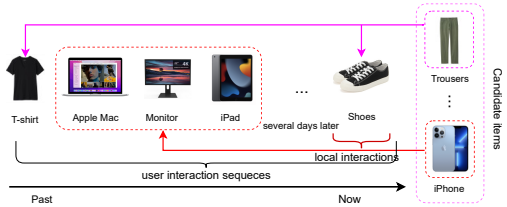

Figure 1: An example of user interaction records on a popular e-commerce platform illustrates the necessity of modeling the implicit intent. Without considering the user's current intent (clothes & accessories) as reflected in local interactions, the iPhone would be ranked ahead of trousers.

To overcome the above issues, we propose a novel CTR prediction model named "Intent-Enhanced User Interest Modeling for Click-Through Rate Prediction (IUI)". In IUI, to capture the user's current local interests, we first regard the user's local interactions as implicit intent and then employ an **I**ntent-**A**ware **N**etwork (IAN) to obtain a stable local user interest representation as well as alleviate the impact of the number and distribution of the user's interaction items. After that, we use a **C**o-**A**ttention **N**etwork (CAN) to obtain a more stable co-dependent global and local interest representation capable of activating the corresponding interest in global interactions and capturing the dynamic interactions between global- and local-level interaction behaviors. In addition, we leverage the aggregation layer to aggregate the user's interest representation across different aspects. Finally, we integrate self-supervised learning into the training of the network as an auxiliary task to improve the CTR prediction performance.

In summary, the main contributions of this work are as follows:

- We propose a novel intent-aware approach named IUI for CTR prediction. To the best of our knowledge, this is the first work to utilize the user's local interactions as current implicit intent to model their local interests in a fine-grained manner.
- We design a co-attention network to capture the dynamic interactions between the user's global- and local-level interaction behavior and obtain the co-dependent representation of global and local interests.
- Extensive experiments conducted on three real-world datasets demonstrate that IUI outperforms the state-of-the-art methods. It is noteworthy that IUI has been deployed in the JD online recommender system and obtained significant improvement in various metrics.

2 RELATED WORK

2.1 Click-Through Rate Prediction

The prediction of CTR plays an important role in many fields, ranging from advertising [43], web search [6], and recommendation [40], which has been extensively studied for many years. Generally, existing CTR prediction methods can be summarized into two categories: conventional and deep learning-based models.

Conventional CTR mainly focuses on feature interaction modeling. The ground-breaking research projects proposed here are primarily based on Logistic Regression (LR) [28], collaborative filtering (CF) [32], Bayesian model [10], and other techniques. Owing to its simplicity, explainability, and effectiveness, LR and its variants [8] have been widely used in the industry on the eve of deep learning. However, these models can not deal with complex feature interactions. To alleviate the above drawbacks, Factorization Machines-based models [15, 21, 26, 38] are proposed, such as FM [26] and FFM [15]. Specifically, FFM injects field-aware into FM to improve the ability of feature interaction. Despite its effectiveness, the performance of the conventional CTR prediction model deteriorated as the number of data and feature dimensions increased. Despite effectiveness, with the increasing number of data and the feature dimension, the performance of the conventional CTR prediction model encouraged bottleneck. Later, with the rapid development of deep learning, various neural networks are applied to CTR prediction. To get rid of the exhausting feature engineering, Wide&Deep [5] proposed by Google creatively replaces the manual feature transformation with neural networks. Compared with LR,

Wide&Deep has strong memorization and generalization ability, which pioneered the combination of different models. Hereafter, a lot of research work has been done to improve and optimize the model in terms of wide networks and deep networks, such as Deep&Cross [29], DeepFM [12], AFM [38], and DFM [21]. Recently, a series of works have been proposed to capture the user's interest from the rich history of interaction sequences with different neural network architectures such as RNN [43, 44], CNN [20], and Transformer [4, 7, 11]. Traditional user interest methods take a straightforward way of learning the representation of user behavior and then concatenate or sum/mean pooling them together to generate user interest without considering the relative importance of different user interactions. Deep Interest Network (DIN) [44] is the first work to employ an attention mechanism to capture users' diverse interests in the online advertising field. Hereafter, DIEN [43] employs a modified GRU to model the evolution of users' interests and proposes an auxiliary loss to capture latent interest from the network's hidden state. Deep Session Interest Network (DSIN) [7] first splits the user's behavior sequence into different sessions and then employs transformer and rnn to model the user's interests intra- and inter-sessions separately. Moreover, Fi-GNN [19] is designed to model sophisticated interactions among the feature fields on the graph-structured features. DIHN [30] emphasizes modeling users' interest in trigger-induced recommendation scenarios. However, these approaches focus on mining interests through users' behavior sequences while ignoring the explicit expression (local interactions) of the user's current intent and the dynamic interactions between the user's global- and local-level interaction behavior.

2.2 Sequential Recommendation

Researchers have been studying sequential recommendation for many years, which aims to predict the next item based on the user's historical behavior sequences. Different from the CTR prediction task, which serves the ranking stage in large-scale e-commerce recommendations, the sequential recommendation is mainly deployed in the stage of recall. Besides, as an essential branch of recommender systems, there has been a lot of research in recent years [17, 18, 23, 41].

Earlier studies mainly focus on modeling the item-item transitions with Markov Chains, which assumes that the next action is highly dependent on the previous interaction sequence. For example, FPMC [27] proposes combining matrix factorization with Markov Chains to improve recommendation performance. Despite their effectiveness, these methods are incapable of capturing long-term dependency relationships between various items. Hereafter, neural network-based approaches are proposed to enhance the ability to extract the user's interest. Caser [34] employs CNN to model the user's preference representation regarding the embedding of the recent item embeddings as an "image." Hereafter, to capture the dynamic time series information of the user behavior sequence, SHAN [39] employs the RNN network and attention mechanism to model the users' long and short-term interests. Inspired by the success of Transformer, SASRec [16], and BERT4Rec [33] are proposed to utilize the self-attention mechanism for different item relation modeling. Recently, SR-GNN [37], which models the session sequences as graph-structured data, is proposed to capture the complex transitions of the different items with GNN in session-based recommendation.

Furthermore, TiSASRec [17] designs a novel time interval-aware self-attention mechanism to learn the weight of different items, absolute positions, and time intervals to predict future items. In practice, many research works, such as SURGE [3], MRIF [18], and CLSR [41], are proposed to capture users' long- and short-term interests due to the dynamic and evolving nature of their interests. Specifically, CLSR [41] first employs two different encoder networks to capture the user interests of different time scales and then designs the pseudo labels for user interests, finally proposing a contrastive learning framework to separately capture users' long- and short-term interests. Following the success of GNN, SURGE [3] is proposed to aggregate implicit signals into explicit ones from user behaviors by designing graph neural network-based models on constructed item-item interest graphs.

Despite their effectiveness, these approaches model users' long- and short-term preferences separately, which limits the performance of recommendations. In our paper, we employ the CAN to model the co-dependent relations between them.

3 METHODOLOGY

In this section, we first introduce the problem formulation. After that, we present the overview of IUI. Finally, we describe the individual components of IUI in detail.

3.1 Problem Formulation

We first introduce some background concepts. Without losing generality, we denote the set of items as $V = \{v_1, v_2, \ldots, v_N\}$, and the set of users as $U = \{u_1, u_2, \ldots, u_M\}$, where N and M are the number of items and users, respectively. For each user $u \in U$, we use $S_u = \left\{v_1^u, v_2^u, \ldots, v_n^u\right\}$ to denote the history behavior sequence, where n is the length of the behavior sequence. Note that behavior sequences may have variable lengths. Furthermore, we denote user attributes such as gender and age as $A_u = \left\{a_1^u, a_2^u, \ldots, a_K^u\right\}$, item attributes like item_id, brand as $A_v = \left\{a_1^v, a_2^v, \ldots, a_L^v\right\}$, and contextual features as $C = \{c_1, c_2, \ldots, c_P\}$, including location, timestamp, and so on, where K, L, and P are the field numbers of the user, item, and context, respectively. For simplicity, if there is no specific description in the following paragraphs, it is for a single user u. But, it can be extended to all users directly. Hereafter, we combine all these features in a predefined order, and then one example can be represented as follows:

$$\boldsymbol{X} = \{v, u, S_u, A_u, A_v, C\}, \tag{1}$$

where v and u denote $item\_id$ and $user\_id$, respectively. An encoding example of item_id, item attribute, and user behavior feature is presented as:

$$\underbrace{[1,0,0,\ldots,0]}_{v:item\_id}\ldots\quad\underbrace{[0,1,0,\ldots,1]}_{A_v:branch\&color}\quad\ldots\quad\underbrace{[1,1,0,\ldots,1]}_{S_u:click}$$

The representations of other features are similar, so we omit them for simplicity. Besides, the embedding representation for one-hot features u, v is a single vector, whereas the embedding representation for multi-hot features S_u, A_u, B_v, C is a list of vectors.

In our problem, we aim to build a prediction model $\hat{y} = f(\boldsymbol{X})$ for each user u to estimate the likelihood of a user clicking the candidate items in a conditioned context.

3.2 Overview of IUI

We propose a novel intent-aware approach named **IUI** for CTR prediction. The whole architecture is illustrated in Figure 2, which comprises seven main components: 1) Embedding Layer. The function of this layer is to obtain the embedding of various features, such as user, item, context, etc. 2) Transformer Encoder Layer. It employs the transformer encoder network to yield the representation of global interactions. 3) Intent-Aware Network (IAN). This module consists of self-attention networks, which utilize the user's local interactions as implicit intent to obtain a stable embedding of the user's local interests. 4) Co-Attention Network (CAN). CAN is designed to capture the co-dependence relationship between users' global and local-level interests. 5) Self-Supervised Layer (SSL). It employs self-supervised learning to purify the representation of user interests representation. 6)Interest Aggregation Layer. It leverages the aggregation function to aggregate the user's interest representation across different aspects. 7) Prediction Layer. It outputs the probability of the user clicking candidate items. Next, we present the seven components in detail.

3.2.1 Embedding Layer. In CTR prediction, the distribution of the input data (user id u, item id v, etc.) is typically sparse and high-dimensional. Therefore, it is common to employ an embedding layer to transfer them into a dense low-dimension vector by looking up the embedding table $\boldsymbol{W} \in R^{f \times d}$, where f is the field number of $\boldsymbol{X}$, d is the dimension size of each field. Furthermore, the embedding table $\boldsymbol{W}$ is randomly initialized and jointly learned with our model before concatenating these dense vectors into a single embedding vector. Without losing generality, we take the item v_i representation $\boldsymbol{e}_i$ for example:

$$\boldsymbol{e}_i = \text{Concat}[\boldsymbol{e}_v, \boldsymbol{e}_{A_v}], \tag{2}$$

where $\boldsymbol{e}_{A_v}$ is the embedding of item attribute features like the brand, color, and so on. Consequently, the representation collection of the user behavior sequence can be denoted as $\boldsymbol{E}_{S_u} = \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\}$, where n is the length of the sequence. For simplicity, let $\boldsymbol{E}_{global} = \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\}$ denote the item representation collection of global interactions, $\boldsymbol{E}_{local} = \{\boldsymbol{e}_k, \ldots, \boldsymbol{e}_n\}$ denote the item representation collection of local interactions, and T denote the length of the local interaction sequence.

3.2.2 Transformer Encoder Layer. Inspired by [11], we employ the transformer encoder network to capture the user's global interest and mitigate the impact of irrelative interactions. Generally, the transformer encoder network consists of a Multi-head Self-attention Network (MSN) and a point-wise Feed-Forward Network (FFN).

The self-attention network uses the scaled dot-product attention defined by the following equation:

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d}})\boldsymbol{V}, \tag{3}$$

$$\boldsymbol{Q} = \boldsymbol{K} = \boldsymbol{V} = \boldsymbol{E}_{global}, \tag{4}$$

where $\boldsymbol{Q}$, $\boldsymbol{K}$ and $\boldsymbol{V}$ present the queries, keys, and values, respectively, d is the hidden dimension size of the queries, keys, and values. And then, we employ multi-head attention to capture the relationships between queries and keys from different aspects:

$$\boldsymbol{S} = \text{Multi-head}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\boldsymbol{W}_H, \tag{5}$$

$$\text{head}_i = \text{Attention}(\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i), \tag{6}$$

where $\boldsymbol{W}_H \in R^{d \times d}$ is weight matrix, and h is the number of heads. Moreover, we combine FFN with MSN to increase the capability of the model representation. The main idea can be summarized as follows:

$$\boldsymbol{G}_{global} = \text{FFN}(\boldsymbol{S}), \tag{7}$$

$$\text{FFN}(\boldsymbol{x}) = \text{ReLU}(\boldsymbol{x}\boldsymbol{W}_1 + b_1)\boldsymbol{W}_2 + b_2, \tag{8}$$

where $\boldsymbol{G}_{global}$ is the collection of user global-level interest representation, $\boldsymbol{x} \in \boldsymbol{S}, d_k = d/h, \boldsymbol{W}_1 \in R^{d \times d_k}, \boldsymbol{W}_2 \in R^{d_k \times d}$ are the weight matrix, $b_1 \in R^{d_k}, b_2 \in R^d$ are the bias.

3.2.3 Intent-Aware Module. According to psychological theory[1, 2], user actions are driven by a set of intents, and the lifetime of different intents may be quite different. Therefore, we consider the user's local interaction behaviors to be inspired by the user's current implicit intent and then leverage the user's local interactions to model the user's current intent. Finally, we utilize the user's current intent to obtain the corresponding local interest representation.

Owing to the fact that users' local interactions convey users' current intent in a relatively short period of time and are also influenced by incidentally transient events (e.g., holidays, birthdays, etc.), users' local interaction behaviors evolve more frequently than global interaction behaviors. As a result, we distinguish between a user's local and global interactions.

Different from the existing approaches, we treat items in the local interactions as the **Q**ueries and items in global interactions as **K**eys and **V**alues in order to generate a stable user intent representation $\hat{\boldsymbol{L}}_{int}$. Then, we transfer $\hat{\boldsymbol{L}}_{int}$ into the interest embedding space through 2-layer MLP. Similar to equation 3, the process can be formulated as follows:

$$\hat{\boldsymbol{L}}_{int} = \text{Attention}(\boldsymbol{Q}_l, \boldsymbol{K}_g, \boldsymbol{V}_g) = \text{softmax}(\frac{\boldsymbol{Q}_l \boldsymbol{K}_g^T}{\sqrt{d}})\boldsymbol{V}_g, \tag{9}$$

$$\boldsymbol{L}_{local} = \text{MLP}(\hat{\boldsymbol{L}}_{int}), \tag{10}$$

where $\boldsymbol{L}_{local} \in R^{d \times T}$ is the collection of user local-level interest representation, $\boldsymbol{Q}_l$ is the item representation collection of local interactions $\boldsymbol{E}_{local}$, and $\boldsymbol{K}_g, \boldsymbol{V}_g$ are the collections of user global-level interest representation $\boldsymbol{G}_{global}$.

3.2.4 Co-attention Network. As mentioned before, it is common for a user's global-level interests and local-level interests not to be completely separated. In other words, the relative importance of items in a user's global-level interaction history depends on items in their local-level interaction history and vice versa. Particularly when users' local interactions are sparse, global interactions will serve as a supplement to generate local interest.

Therefore, we design a Co-Attention Network to capture the dynamic interactions between global and local interactions and then obtain the co-dependent representation of global and local interests. To capture the dynamic interactions, as shown in Figure 3, we define an affinity matrix $\boldsymbol{M}_A$:

$$\boldsymbol{M}_A = \tanh(\boldsymbol{G}_{global}^T \boldsymbol{W}_3 \boldsymbol{L}_{local}), \tag{11}$$

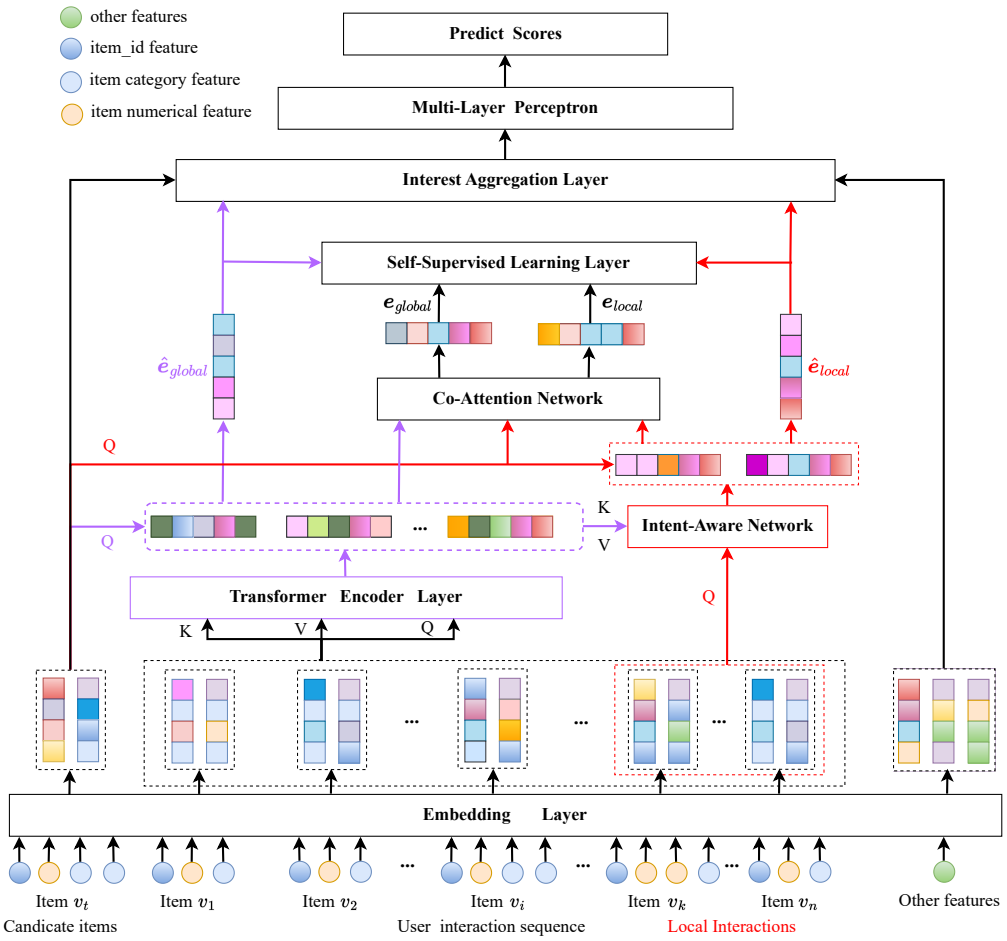

Figure 2: Architecture of the proposed framework IUI. It mainly consists of Transformer Encoder Layer, Intent-Aware Network, Co-Attention Network, and Self-Supervised Learning Layer, where the red line represents the process of local interest generation and the blue line represents global interest. Other features include the collection of user attributes and contextual features.

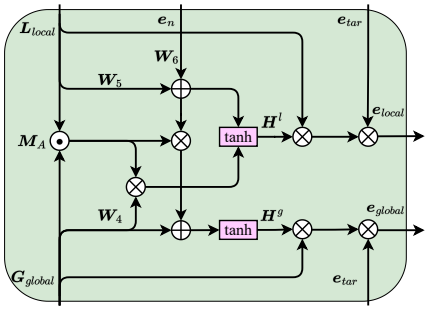

Figure 3: Architecture of Co-Attention Network.

where $W_3 \in R^{d\times d}$ is the weight matrix, $G_{global} \in R^{d\times \mathcal{N}}$, $L_{local} \in R^{d\times T}$, $M_A \in R^{\mathcal{N}\times T}$, $\mathcal{N}$ and T are the number of items in global and local interactions. And then, we employ the affinity matrix M_A to transfer L_{local} into the global attention space:

$$H^g = \tanh(W_4 G_{global} + (W_5 L_{local} + W_6 e_n) M_A^T), \tag{12}$$

$$\alpha_g = softmax(w_g^T H^g), \tag{13}$$

$$\bar{e}_{global} = \sum_{n=1}^{\mathcal{N}} \alpha_g^n G_{global}, \tag{14}$$

$$e_{global} = \text{Concat}(e_{tar}^T, \bar{e}_{global}) W_7^T, \tag{15}$$

where $W_4, W_5, W_6 \in R^{K\times d}$, $W_7 \in R^{d\times 2d}$, and $w_g \in R^K$ are the weight matrix. Besides, e_{tar} is the target item vector, $e_n \in R^d$ is the last item representation of the local interactions, and K is the hidden embedding size of w_g. Finally, we obtain the co-dependent representation of global-level user interest e_{global} for the target item e_{tar}. In addition, vice versa:

$$H^l = \tanh(W_5 L_{local} + W_6 e_n + (W_4 G_{global}) M_A), \tag{16}$$

$$\alpha_l = softmax(w_l^T H^l), \tag{17}$$

$$\bar{\boldsymbol{e}}_{local} = \sum_{t=1}^{T} \alpha_l^t \boldsymbol{L}_{local}, \tag{18}$$

$$\boldsymbol{e}_{local} = \text{Concat}(\boldsymbol{e}_{tar}^T, \bar{\boldsymbol{e}}_{local})\boldsymbol{W}_8^T \tag{19}$$

where $\boldsymbol{H}^l \in R^{K\times T}, \boldsymbol{w}_l \in R^K, \boldsymbol{W}_8 \in R^{d\times 2d}$ are the weight matrix and $\boldsymbol{e}_{local}$ is the co-dependent local-level user interest representation.

3.2.5 Self-Supervised Learning Layer. As illustrated in Section 3.2.2 and 3.2.3, we obtain the collections of user-diverse interest representations $\boldsymbol{G}_{global}$ and $\boldsymbol{L}_{local}$. According to equation 3, similar to the Transformer Decoder Network, we employ the target item v_t as a query, $\boldsymbol{G}_{global}$ and $\boldsymbol{L}_{local}$ as both keys and values to learn the relationships between the target item and the user history interactions. Following that, we obtain the unique interest vector $\hat{\boldsymbol{e}}_{global}$ and $\hat{\boldsymbol{e}}_{local}$ towards target item v_t.

Similar to previous works [35, 41], we leverage the self-supervised learning layer to purify the user interest representation. Due to the fact that there is no manually annotated label for user interests, we employ the co-dependent user global interest representation $\boldsymbol{e}_{global}$ and local interest representation $\boldsymbol{e}_{local}$ illustrated in Section 3.2.4 as pseudo interest labels. Specifically, we perform contrastive learning between the unique interest vector and the proxies vector (co-dependent user interest vector). The main idea is to maximize the mutual information between unique interest vectors and their corresponding proxies, which can be summarized as follows:

$$\begin{aligned}
\text{Sim}(\hat{\boldsymbol{e}}_{global}, \boldsymbol{e}_{global}) &> \text{Sim}(\hat{\boldsymbol{e}}_{global}, \boldsymbol{e}_{local}) \\
\text{Sim}(\boldsymbol{e}_{global}, \hat{\boldsymbol{e}}_{global}) &> \text{Sim}(\boldsymbol{e}_{global}, \hat{\boldsymbol{e}}_{local}) \\
\text{Sim}(\hat{\boldsymbol{e}}_{local}, \boldsymbol{e}_{local}) &> \text{Sim}(\hat{\boldsymbol{e}}_{local}, \boldsymbol{e}_{global}) \\
\text{Sim}(\boldsymbol{e}_{local}, \hat{\boldsymbol{e}}_{local}) &> \text{Sim}(\boldsymbol{e}_{local}, \hat{\boldsymbol{e}}_{global}),
\end{aligned} \tag{20}$$

where Sim is the function (e.g., Cosine Similarity) to estimate the similarity of two vectors. Notably, we employ the BPR loss to accomplish contrastive learning, which can be formulated as:

$$\mathcal{L}_{bpr}(a, p, q) = \sigma(\text{Sim}(a, p) - \text{Sim}(a, q)), \tag{21}$$

where σ is the activate function (e.g., Softplus). Besides, a, p, and q are the anchor, positive sample, and negative sample, respectively. Therefore, the contrastive loss for self-supervised learning can be computed as follows:

$$\begin{aligned}
\mathcal{L}_{aux} = &\mathcal{L}_{bpr}(\hat{\boldsymbol{e}}_{global}, \boldsymbol{e}_{global}, \boldsymbol{e}_{local}) + \mathcal{L}_{bpr}(\boldsymbol{e}_{global}, \hat{\boldsymbol{e}}_{global}, \hat{\boldsymbol{e}}_{local}) \\
&+\mathcal{L}_{bpr}(\hat{\boldsymbol{e}}_{local}, \boldsymbol{e}_{local}, \boldsymbol{e}_{global}) + \mathcal{L}_{bpr}(\boldsymbol{e}_{local}, \hat{\boldsymbol{e}}_{local}, \hat{\boldsymbol{e}}_{global}),
\end{aligned} \tag{22}$$

3.2.6 Interest Aggregation Layer. Generally, we employ the interest aggregation layer to aggregate user interests from different aspects. Generally, the aggregation operation can have many functions, such as sum, concatenation, attention, etc. The effectiveness of different aggregation functions on IUI will be discussed in detail in the ablation study.

$$\hat{\boldsymbol{e}}_{ideal}^{(0)} = \text{Agg}(\hat{\boldsymbol{e}}_{global}, \hat{\boldsymbol{e}}_{local}), \tag{23}$$

where $\hat{\boldsymbol{e}}_{ideal}^{(0)}$ is the ideal user-interest representation. Finally, we concatenate $\hat{\boldsymbol{e}}_{ideal}^{(0)}, \boldsymbol{e}_{tar}, \boldsymbol{e}_{A_u}$ and $\boldsymbol{e}_C$ to form a comprehensive vector $\boldsymbol{e}_{ideal}^{(0)}$, and then feed it to the MLP layer, where $\boldsymbol{e}_{tar}$, $\boldsymbol{e}_{A_u}$ and $\boldsymbol{e}_C$ are the embedding vector of the target item, user features, and contextual feature, respectively.

3.3 Prediction and Training

3.3.1 Prediction Layer. To increase the representation ability of the model, we feed $\boldsymbol{e}_{int}^{(0)}$ into the deep neural network as follows:

$$\hat{y} = \sigma(\boldsymbol{W}_l \boldsymbol{e}_{int}^{(l-1)} + b_l), \tag{24}$$

where l is the number of MLP layers, and $\boldsymbol{e}_{int}^{(l-1)}$ is equal to $\boldsymbol{e}_{int}^{(0)}$ when $l = 1$. $\boldsymbol{W}_l$ and b_l are the weight matrix and bias of l-th layer, respectively.

3.3.2 Training Strategy. To learn the trainable parameters, we optimize the proposed model with log loss [11, 43]. The objective function is formalized as follows:

$$\mathcal{L}_{main} = -\frac{1}{|\mathcal{D}|} \sum_{(x,y)\in\mathcal{D}} (y \log \hat{y} + (1-y)\log(1-\hat{y})), \tag{25}$$

where $\mathcal{D}$ is the training dataset, $|\mathcal{D}|$ is the size of the training dataset, x is the input feature of the model, and $y \in \{0, 1\}$ is the label indicating the probability of user interaction with the item.

As illustrated in Section 3.2.5, we employ self-supervised learning as the auxiliary task to enhance the CTR prediction performance. Thus, the joint loss function can be described as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{main} + \beta\mathcal{L}_{aux} + \lambda\|\Theta\|_2, \tag{26}$$

where β is the hyper-parameter, λ represents the regularization coefficient, and Θ represents the model parameters.

4 EXPERIMENTS

In this section, we first describe the experimental settings and then conduct experiments on three benchmark datasets to evaluate our proposed model by answering the following questions:

- **RQ1**: How does our model outperform state-of-the-art baselines in terms of CTR prediction accuracy?
- **RQ2**: How does each component of IUI perform in predicting the CTR?
- **RQ3**: How do the hyper-parameters affect the performance of IUI?

4.1 Dataset

We leverage both public and production datasets to evaluate the effectiveness of our proposed model.

- **Alibaba**[1] is a public dataset released by Alibaba, which is an online advertising platform in China. It randomly sampled 1.14 million users' online ad displays and click logs for 8 days to form the original data. We use the same sampled data as DIN [44].
- **Amazon(Electro)**[2] is a public dataset provided by Amzon.com that contains the transactions about the users' shopping details, such as item_id, cate_id, time_stamp, etc. The size of the training and testing sets is 23,249,296 and 3,308,665 respectively. It has been extensively used in the previous work [14, 44].

[1] https://tianchi.aliyun.com/dataset/dataDetail?dataId=56&userId=1#1
[2] http://jmcauley.ucsd.edu/data/amazon/

Table 1: Statistics of the datasets used in the experiments

Dataset	#train	#test	#fields	#users	#items	#pos_ratio
Alibaba	23,249,296	3,308,665	14	1,141,729	846,811	5.44%
Amazon(Electro)	2,608,764	384,806	5	192,403	63,001	50.00%
JD	54,405,829	2,735,903,	8	10,666,914	4,937,026	6.36%

Table 2: The overall performance comparison with other baseline methods over three datasets

Category	Methods	Alibaba		Amazon(Electro)		JD	
		AUC	Logloss	AUC	Logloss	AUC	Logloss
Con-CTR	LR	0.5921(0.00012)	0.1981(0.00005)	0.7776(0.00007)	0.5568(0.00011)	0.6104(0.00012)	0.2349(0.00002)
	W&D	0.5982(0.00019)	0.1988(0.00011)	0.8261(0.00006)	0.5290(0.00191)	0.6218(0.00019)	0.2335(0.00014)
	DeepFM	0.5978(0.00030)	0.1977(0.00010)	0.8309(0.00026)	0.5188(0.00994)	0.6208(0.00005)	0.2339(0.00010)
DIM-CTR	DIN	0.5995(0.00013)	0.1979(0.00008)	0.8372(0.00038)	0.5146(0.00201)	0.6428(0.00081)	0.2314(0.00011)
	DIEN	0.6013(0.00022)	0.1995(0.00032)	0.8379(0.00043)	0.5139(0.00105)	0.6451(0.00027)	0.2310(0.00008)
	DMT	0.6007(0.00015)	0.1990(0.00030)	0.8375(0.00014)	0.5128(0.00313)	0.6470(0.00071)	0.2310(0.00004)
	DSIN	0.6039(0.00024)	0.1986(0.00021)	0.8396(0.00050)	0.5120(0.00348)	0.6497(0.00059)	0.2309(0.00006)
	MIAN	0.6037(0.00028)	0.1977(0.00008)	0.8398(0.00039)	0.5119(0.00254)	0.6497(0.00041)	0.2307(0.00007)
LS-SR	STAMP	0.5617(0.00033)	0.2131(0.00015)	0.7483(0.00039)	0.5732(0.00201)	0.5892(0.00037)	0.2593(0.00011)
	SHAN	0.5976(0.00021)	0.1981(0.00009)	0.8278(0.00039)	0.5239(0.00801)	0.6209(0.00041)	0.2334(0.00009)
	CLSR	0.6039(0.00017)	0.1977(0.00011)	0.8397(0.00042)	0.5120(0.00363)	0.6498(0.00051)	0.2301(0.00003)
Ours	IUI	**0.6111(0.00031)**	**0.1965(0.00012)**	**0.8496(0.00097)**	**0.4901(0.00630)**	**0.6609(0.00081)**	**0.2261(0.00014)**

- **JD**[3] is extracted from JD.com, one of China's two massive B2C online retailers. This dataset is composed of users' interaction details. Furthermore, it is worth noting that all the data has been desensitized.

We selected the above datasets because they were collected and sampled from real-world interactions in production. Specifically, users and items appearing less than 10 times are filtered across all the datasets, and the maximum length of the behavior sequence is 150. The statistics of the datasets are illustrated in Table 1.

4.2 Experiment setting

4.2.1 Evaluation metrics. Following the previous work [30, 42], we evaluate the ranking result using two widely used metrics: AUC and Logloss.

- **AUC** (Area Under the ROC Curve) represents the probability that a random positive example is positioned in front of a random negative example. The higher, the better.
- **Logloss** is also called binary cross-entropy loss, as defined in Equation 25. The lower, the better.

In the online A/B test, we also employ UCTR (clicks per user per day) and UCVR (orders per user per day) to evaluate the performance of our model. Moreover, it is worth noting that when the user base is large enough (billion), an improvement of AUC/CTR at 0.1% is generally considered significant for CTR in the industry, such as Google, Alibaba, and JD.

$$UCTR = \frac{\#clicks}{\#users}$$
$$UCVR = \frac{\#orders}{\#users}, \tag{27}$$

[3]https://www.jd.com/

4.2.2 Baselines. We compared our model with the following representative methods: the conventional methods (Con-CTR: LR, Wide&Deep, DeepFM), deep interest modeling methods (DIM-CTR: DIN, DIEN, DSIN, DMT, MIAN), and sequential recommendation methods (LS-SR: SHAN, CLSR). In this paper, we focus on comparing our methods with existing classical user interest modeling methods. Other lifelong search-based CTR approaches, such as SIM [25], are not the focus of this paper.

- **LR** [28]: This is a classical and widely applied algorithm for CTR prediction before the eve of deep learning-based methods in the industry.
- **Wide&Deep** [5]: Wide&Deep (W&D) trains a wide linear model and a deep neural model simultaneously for CTR prediction, combining the benefits of memorization and generalization.
- **DeepFM** [12]: To combine the power of traditional FM and deep MLP, DeepFM replaces the LR in the wide network of Wide&Deep with FM to model the 2-order feature interactions.
- **STAMP** [22]: STAMP considers the user's current interests to improve the effectiveness of session-based recommendation.
- **SHAN** [39]: SHAN utilizes a hierarchical attention network to capture users' dynamic interests.
- **DIN** [44]: DIN is a deep model that employs an attentive neural network to activate related user behaviors with respect to corresponding targets.
- **DIEN** [43]: DIEN employs a GRU encoder to capture the dependencies of user behaviors and another modified GRU model for evolving user interests.

- **DSIN** [7]: This is a transformer-based model that uses the transformer and RNN to model the user's intra- and inter-session interests separately.
- **DMT** [11]: DMT is a multi-task learning model that exploits multiple transformers to model users' diverse behavior sequences.
- **MIAN** [40]: MIAN is a deep CTR model that contains a multi-interaction and transformer layer to extract multiple representations of user behavior.
- **CLSR** [41]: CLSR proposes a contrastive learning framework of sequential recommendation with self-supervision.

4.2.3 Parameter Setting. For fair comparisons, we implement all the models with TensorFlow [4]. The batch size for min-batch is set to 1000, 5000, and 10000 for Amazon, Alibaba, and JD datasets, depending on the data size. Moreover, the user and item embedding size are set to 64, and the contextual feature embedding size is 16. For a fair comparison, we maintain the hyper-parameter consistency for each model. Besides, L_2 regularization is 10^{-5} and all parameters are initialized using a Gaussian distribution with a mean of 0 and a deviation of 0.1. We use the Adam optimizer with an initial learning rate of 0.001, which will decay by 0.1 after every 3 epoch. The length of the local interaction sequence T is set to 3, 5, and 4 for Alibaba, Amazon, and JD datasets.

4.3 Performance Comparison (RQ1)

Table 2 reports the experimental results of the baseline methods and our proposed model on three real-world industry datasets, in which the best results are highlighted in boldface. It can be observed that IUI achieves the best performance across all three datasets in terms of two metrics, which demonstrates the effectiveness of our proposed model. Besides, we also have the following observations:

- Among the conventional methods, LR performs the worst, as it only captures a shallow linear combination of different category features without considering the 2-order feature interactions. DeepFM and W&D perform better than LR, which demonstrates the importance of modeling high-order feature interactions. In summary, all these approaches only model the user's general taste and ignore the user's implicit interest contained in their behavior sequences.
- In general, compared with conventional methods, neural network-based methods have better performance for CTR prediction. This verifies the necessity of modeling the user's interests. DIEN outperforms the DIN and DMT among deep interest modeling-based approaches in Alibaba and Amazon datasets, demonstrating the fact that users' interests are complex and dynamic, as well as the importance of users' current interests. The transformer-based model DMT outperforms DIN and DIEN in JD datasets, perhaps because when the user behavior data is large enough, the transformer is more capable of modeling user interest than GRU. Moreover, DSIN outperforms DMT, owing to its effectiveness in extracting users' historical behaviors into session interests and modeling the dynamic evolution of session interests. Among DIM-CTR, the Transformer-based model MIAN outperforms all the aforementioned baseline methods generally, owing to the more fine-grained interaction modeling among user sequence behavior, items, and context information.
- Among sequential recommendation methods, CLSR outperforms SHAN, STAMP, and other baselines, indicating the effectiveness of contrastive learning in purifying the representation of user interests. IUI performs better than SHAN and STAMP since, on the one hand, the last item is not stable to model users' current interests, and on the other hand, co-dependent relations are vital to model user interests.
- Our approach IUI outperforms CLSR on all three datasets. Though CLSR shows great progress in user interest modeling, it cannot capture the user's current intent precisely or the dynamic interactions between the user's global- and local-level interaction behavior. Compared with CLSR, IUI can gracefully model these factors. Specifically, taking the JD dataset as an example, IUI outperforms CLSR by 1.71% on AUC and 1.74% on Logloss.

[4]The source code of IUI is available here: https://github.com/JD-SRT/IUI

Table 3: The performance of contrast models in terms of AUC and Logloss

Methods	Alibaba		Amazon(Electro)		JD	
	AUC	Logloss	AUC	Logloss	AUC	Logloss
Base	0.6019	0.1989	0.8386	0.5127	0.6483	0.2306
Base+IAN	0.6029	0.1983	0.8401	0.5126	0.6508	0.2309
Base+IAN+CAN	0.6071	0.1976	0.8462	0.5015	0.6576	0.2289
Base+IAN+Att	0.6047	0.1982	0.8429	0.5123	0.6519	0.2305
IUI	**0.6111**	**0.1965**	**0.8496**	**0.4901**	**0.6609**	**0.2261**

4.4 Ablation Study (RQ2)

In this section, we conduct experiments to evaluate the effectiveness of different components in IUI. Specifically, we design four contrast models:

- Base: IUI without Co-Attention Network, Intent-Aware Network, and Self-Supervised Learning Layer (SSL). This base model is the DMT variant.
- Base+IAN: Base model with Intent-Aware Network (IAN).
- Base+IAN+CAN: IUI without SSL.
- Base+IAN+Att: IUI without SSL and employing the normal attention network instead of Co-Attention Network (CAN).
- Base+CAN+IAN+SSL (IUI): This is our proposed model IUI.

Table 3 shows the comparison of different contrast models. It is obvious that IUI achieves the best performance in terms of AUC and Logloss, which proves the effectiveness of each component of our model. Furthermore, we also have the following observations:

- We notice that IUI outperforms the four contrast models above. This suggests that IAN, CAN, and SSL, which can model the user's current intent, capture the interactions between the user's global and local interaction behaviors, and purify users' interests, respectively, play a vital role in improving the performance of IUI.
- We can find that Base+IAN outperforms the Base model, indicating the importance of users' current intent in modeling their local interests. Moreover, Base+IAN+CAN performs

Table 4: Effectiveness of different aggregation operations.

Methods	Alibaba		Amazon(Electro)		JD	
	AUC	Logloss	AUC	Logloss	AUC	Logloss
Concat	0.6085	0.1971	0.8473	0.4998	0.6589	0.2283
Sum_pooling	0.6049	0.1982	0.8439	0.5047	0.6543	0.2303
Avg_pooling	0.6054	0.1976	0.8449	0.5090	0.6549	0.2297
Attention	**0.6111**	**0.1965**	**0.8496**	**0.4901**	**0.6609**	**0.2261**

better than Base+IAN, which demonstrates the users' local interactions inspired by current intent can activate the corresponding interactions in global interactions. Besides, Base+IAN+Att performs worse than Base+IAN+CAN, which indicates the necessity of modeling the co-dependent relationships between users' global and local interactions. In addition, global interactions can serve as a supplement to generate stable local interest when local interaction is sparse.

- We also observe that IUI outperforms Base+CAN+IAN, which suggests that SSL is effective in purifying the user's interest representation.

4.5 Impact of Aggregation Function

As aforementioned in Section 3.2.6, we utilize different aggregation functions to aggregate user interest representation from different aspects. For the concatenation operation, the final representation of user interest is the concatenation of user global-level interest representation $\hat{\boldsymbol{e}}_{global}$, user local-level interest representation $\hat{\boldsymbol{e}}_{local}$.

$$\hat{\boldsymbol{e}}_{ideal}^{(0)} = \text{Concat}(\hat{\boldsymbol{e}}_{global}, \hat{\boldsymbol{e}}_{local}), \tag{28}$$

For the sum_pooling operation, we utilize the sum value of every dimension of each feature. The final ideal user interest representation $\boldsymbol{e}_{int}^{(0)}$ is presented as:

$$\hat{\boldsymbol{e}}_{ideal}^{(0)} = \text{Sum\_pooling}(\hat{\boldsymbol{e}}_{global}, \hat{\boldsymbol{e}}_{local}), \tag{29}$$

For the average_pooling operation, we employ the average value of every dimension of each feature:

$$\hat{\boldsymbol{e}}_{ideal}^{(0)} = \text{Avg\_pooling}(\hat{\boldsymbol{e}}_{global}, \hat{\boldsymbol{e}}_{local}), \tag{30}$$

For the attention mechanism, the main idea can be summarized as follows:

$$\alpha = \text{Sigmoid}(\boldsymbol{W}_9\hat{\boldsymbol{e}}_{global} + \boldsymbol{W}_{10}\hat{\boldsymbol{e}}_{local}), \tag{31}$$

$$\hat{\boldsymbol{e}}_{ideal}^{(0)} = \alpha\hat{\boldsymbol{e}}_{global} + (1-\alpha)\hat{\boldsymbol{e}}_{local}, \tag{32}$$

where $\boldsymbol{W}_9, \boldsymbol{W}_{10} \in R^{d\times d}$ are the weight matrix.

As mentioned in Section 3.2.6, Table 4 shows the effectiveness of different aggregation functions on IUI. It can be observed that IUI with attention outperforms other aggregation operations on JD and Amazon(Electro) in terms of AUC and Logloss, which indicates concatenation and pooling mechanisms may cause an overlap or dilution of the user interests.

4.6 Influence of Hyper-parameters (RQ3)

In this section, we conduct experiments to evaluate the effectiveness of different hyper-parameters. In general, we vary the length of the local interaction sequence to investigate the effectiveness of using IAN when learning the user's current interest representation. Specifically, the size of the local interaction sequence T is searched in [1, 2, 3, 4, 5, 6, 8, 10]. Figure 4 shows the impact of different local interaction sizes. For JD datasets, we can observe that when the size is less than 4, the model does not perform well, as it is hard to precisely capture the current user intent. It achieves the best performance when the number is set to 4 on JD, 3 on Alibaba, and 5 on Amazon. However, as the size increases, the performance of our model begins to deteriorate. It may be because the user's local interests, inspired by current intent, have been submerged by global interests.

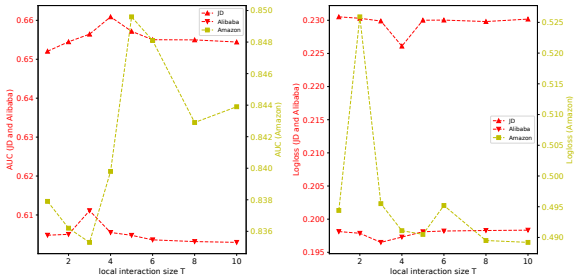

Figure 4: Impact of different local interaction size T.

4.7 Online A/B Testing

To further illustrate the effectiveness of our proposed model, we conducted online A/B testing on the JD homepage recommendation service for 15 consecutive days. The baseline model is the last deployed production model (Base model in Section 4.4: DMT's variant), a transformer-based model with over 1,000 recent user behaviors. Simultaneously, to make the online evaluation confident and fair, each method deployed for the A/B test has the same number of users. IUI contributes up to 0.91% UCTR and 1.54% UCVR promotion. Now, IUI has been deployed online and serves the main traffic. Specifically, for the cold-start users and items, we only need to set the item_id and user_id to the default and then employ other features to fulfill the embedding vectors.

5 CONCLUSION

In this paper, we propose an intent-enhanced user interest modeling network for CTR prediction, which consists of an intent-aware network to auxiliary model the representation of the user's current preference and a co-attention network to capture the dynamic interactions between the user's global- and local-level behavior sequences. Moreover, we employ self-supervised learning as an auxiliary task to maximize the mutual information between global and local interest representation. Our model can make full use of all informative user behavior interactions while also addressing the issue of user interest shifts. In the experiments, we show that our models significantly and consistently outperform the state-of-the-art approaches on real-world datasets. In future work, we plan to exploit the cross-domain information and the distribution of users' interests to better model the user's intent for further improvement.

REFERENCES

[1] Icek Ajzen. 2002. Residual effects of past on later behavior: Habituation and reasoned action perspectives. *Personality and social psychology review* 6, 2 (2002), 107–122.

[2] Dolores Albarracin and Robert S Wyer Jr. 2000. The cognitive impact of past behavior: influences on beliefs, attitudes, and future behavioral decisions. *Journal of personality and social psychology* 79, 1 (2000), 5.

[3] Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2021. Sequential recommendation with graph neural networks. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 378–387.

[4] Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, and Wenwu Ou. 2019. Behavior sequence transformer for e-commerce recommendation in alibaba. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*. 1–4.

[5] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.

[6] Zhifang Fan, Dan Ou, Yulong Gu, Bairan Fu, Xiang Li, Wentian Bao, Xin-Yu Dai, Xiaoyi Zeng, Tao Zhuang, and Qingwen Liu. 2022. Modeling Users' Contextualized Page-wise Feedback for Click-Through Rate Prediction in E-commerce Search. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 262–270.

[7] Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. 2019. Deep session interest network for click-through rate prediction. *arXiv preprint arXiv:1905.06482* (2019).

[8] Kun Gai, Xiaoqiang Zhu, Han Li, Kai Liu, and Zhe Wang. 2017. Learning piece-wise linear models from large scale data for ad click prediction. *arXiv preprint arXiv:1704.05194* (2017).

[9] Suyu Ge, Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2020. Graph enhanced representation learning for news recommendation. In *Proceedings of The Web Conference 2020*. 2863–2869.

[10] Thore Graepel, Joaquin Quinonero Candela, Thomas Borchert, and Ralf Herbrich. 2010. Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. Omnipress.

[11] Yulong Gu, Zhuoye Ding, Shuaiqiang Wang, Lixin Zou, Yiding Liu, and Dawei Yin. 2020. Deep multifaceted transformers for multi-objective ranking in large-scale e-commerce recommender systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2493–2500.

[12] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).

[13] Li He, Hongxu Chen, Dingxian Wang, Shoaib Jameel, Philip Yu, and Guandong Xu. 2021. Click-Through Rate Prediction with Multi-Modal Hypergraphs. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 690–699.

[14] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*. 507–517.

[15] Yuchin Juan, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. 2016. Field-aware factorization machines for CTR prediction. In *Proceedings of the 10th ACM conference on recommender systems*. 43–50.

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

[17] Jiacheng Li, Yujie Wang, and Julian McAuley. 2020. Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th international conference on web search and data mining*. 322–330.

[18] Shihao Li, Dekun Yang, and Bufeng Zhang. 2020. MRIF: Multi-resolution interest fusion for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1765–1768.

[19] Zekun Li, Zeyu Cui, Shu Wu, Xiaoyu Zhang, and Liang Wang. 2019. Fi-gnn: Modeling feature interactions via graph neural networks for ctr prediction. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 539–548.

[20] Bin Liu, Ruiming Tang, Yingzhi Chen, Jinkai Yu, Huifeng Guo, and Yuzhou Zhang. 2019. Feature generation by convolutional neural network for click-through rate prediction. In *The World Wide Web Conference*. 1119–1129.

[21] Feng Liu, Wei Guo, Huifeng Guo, Ruiming Tang, Yunming Ye, and Xiuqiang He. 2020. Dual-attentional factorization-machines based neural network for user response prediction. In *Companion Proceedings of the Web Conference 2020*. 26–27.

[22] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: short-term attention/memory priority model for session-based recommendation. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1831–1839.

[23] Jianxin Ma, Chang Zhou, Hongxia Yang, Peng Cui, Xin Wang, and Wenwu Zhu. 2020. Disentangled self-supervision in sequential recommenders. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 483–491.

[24] Erxue Min, Yu Rong, Tingyang Xu, Yatao Bian, Da Luo, Kangyi Lin, Junzhou Huang, Sophia Ananiadou, and Peilin Zhao. 2022. Neighbour Interaction based Click-Through Rate Prediction via Graph-masked Transformer. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 353–362.

[25] Qi Pi, Guorui Zhou, Yujing Zhang, Zhe Wang, Lejian Ren, Ying Fan, Xiaoqiang Zhu, and Kun Gai. 2020. Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2685–2692.

[26] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International conference on data mining*. IEEE, 995–1000.

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

[28] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web*. 521–530.

[29] Ying Shan, T Ryan Hoens, Jian Jiao, Haijing Wang, Dong Yu, and JC Mao. 2016. Deep crossing: Web-scale modeling without manually crafted combinatorial features. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 255–262.

[30] Qijie Shen, Hong Wen, Wanjie Tao, Jing Zhang, Fuyu Lv, Zulong Chen, and Zhao Li. 2022. Deep Interest Highlight Network for Click-Through Rate Prediction in Trigger-Induced Recommendation. In *Proceedings of the ACM Web Conference 2022*. 422–430.

[31] Qijie Shen, Hong Wen, Jing Zhang, and Qi Rao. 2022. Hierarchically Fusing Long and Short-Term User Interests for Click-Through Rate Prediction in Product Search. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 1767–1776.

[32] Si Shen, Botao Hu, Weizhu Chen, and Qiang Yang. 2012. Personalized click model through collaborative filtering. In *Proceedings of the fifth ACM international conference on Web search and data mining*. 323–332.

[33] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

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

[35] Fangye Wang, Yingxu Wang, Dongsheng Li, Hansu Gu, Tun Lu, Peng Zhang, and Ning Gu. 2022. CL4CTR: A Contrastive Learning Framework for CTR Prediction. *arXiv preprint arXiv:2212.00522* (2022).

[36] Le Wu, Xiangnan He, Xiang Wang, Kun Zhang, and Meng Wang. 2021. A survey on neural recommendation: From collaborative filtering to content and context enriched recommendation. *arXiv preprint arXiv:2104.13030* (2021).

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

[38] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional factorization machines: Learning the weight of feature interactions via attention networks. *arXiv preprint arXiv:1708.04617* (2017).

[39] Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, Yanchi Liu, Guandong Xu, Xing Xie, Hui Xiong, and Jian Wu. 2018. Sequential recommender system based on hierarchical attention network. In *IJCAI International Joint Conference on Artificial Intelligence*.

[40] Kai Zhang, Hao Qian, Qing Cui, Qi Liu, Longfei Li, Jun Zhou, Jianhui Ma, and Enhong Chen. 2021. Multi-interactive attention network for fine-grained feature learning in ctr prediction. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 984–992.

[41] Yu Zheng, Chen Gao, Jianxin Chang, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2022. Disentangling Long and Short-Term Interests for Recommendation. In *Proceedings of the ACM Web Conference 2022*. 2256–2267.

[42] Zuowu Zheng, Changwang Zhang, Xiaofeng Gao, and Guihai Chen. 2022. HIEN: Hierarchical Intention Embedding Network for Click-Through Rate Prediction. *arXiv preprint arXiv:2206.00510* (2022).

[43] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 5941–5948.

[44] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1059–1068.