

# Deep Interest Highlight Network for Click-Through Rate Prediction in Trigger-Induced Recommendation

Qijie Shen<sup>1</sup>, Hong Wen<sup>1</sup>, Wanjie Tao<sup>1</sup>, Jing Zhang<sup>2</sup>, Fuyu Lv<sup>1</sup>, Zulong Chen<sup>1</sup>, Zhao Li<sup>3\*</sup>

<sup>1</sup>Alibaba Group, <sup>2</sup>The University of Sydney, <sup>3</sup>Zhejiang University

{qijie.sqj,qinggan.wh,wanjie.twj,zulong.czl,fuyu.lfy}@alibaba-inc.com

jing.zhang1@sydney.edu.au,zhao\_li@zju.edu.cn

## ABSTRACT

In many classical e-commerce platforms, personalized recommendation has been proven to be of great business value, which can improve user satisfaction and increase the revenue of platforms. In this paper, we present a new recommendation problem, Trigger-Induced Recommendation (TIR), where users' instant interest can be explicitly induced with a trigger item and follow-up related target items are recommended accordingly. TIR has become ubiquitous and popular in e-commerce platforms. In this paper, we figure out that although existing recommendation models are effective in traditional recommendation scenarios by mining users' interests based on their massive historical behaviors, they are struggling in discovering users' instant interests in the TIR scenario due to the discrepancy between these scenarios, resulting in inferior performance. To tackle the problem, we propose a novel recommendation method named Deep Interest Highlight Network (DIHN) for Click-Through Rate (CTR) prediction in TIR scenarios. It has three main components including 1) User Intent Network (UIN), which responds to generate a precise probability score to predict user's intent on the trigger item; 2) Fusion Embedding Module (FEM), which adaptively fuses trigger item and target item embeddings based on the prediction from UIN; and (3) Hybrid Interest Extracting Module (HIEM), which can effectively highlight users' instant interest from their behaviors based on the result of FEM. Extensive offline and online evaluations on a real-world e-commerce platform demonstrate the superiority of DIHN over state-of-the-art methods. Our code is available <sup>1</sup>.

## CCS CONCEPTS

• Information system → Information retrieval.

## KEYWORDS

Recommender System, Click-Through Rate Prediction, Trigger-Induced Recommendation, Users' Behavior Modelling

\*Q. Shen and H. Wen share the co-first authorship.

<sup>1</sup><https://github.com/EzailShen/WWW-22-DIHN>

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 ACM 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](mailto:permissions@acm.org).

WWW '22, April 25–29, 2022, Virtual Event, Lyon, France

© 2022 Association for Computing Machinery.

ACM ISBN 978-1-4503-9096-5/22/04...\$15.00

<https://doi.org/10.1145/3485447.3511970>

## ACM Reference Format:

Qijie Shen, Hong Wen, Wanjie Tao, Jing Zhang, Fuyu Lv, Zulong Chen, Zhao Li. 2022. Deep Interest Highlight Network for Click-Through Rate Prediction in Trigger-Induced Recommendation . In *Proceedings of the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Virtual Event, Lyon, France*. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3485447.3511970>

## 1 INTRODUCTION

In traditional recommender system (RS), users can only receive information in a passive manner, lacking of instant feedback mechanisms for interacting with the system. However, users sometimes may intend to actively access more related items with the clicked item just now. To achieve this goal, in this paper, we present a new recommendation problem, Trigger-Induced Recommendation (TIR), where users' instant interest can be explicitly induced with a trigger item (*i.e.*, the last clicked item) and follow-up related items are recommended accordingly. For example in our online scenario (one of the largest e-commerce platform in the world), multiple cards with each three recommended items are displayed first as shown in Fig. 1(a). Then, once an item is clicked, a new page, called Item Feeds Flow Page (IFFP), will be triggered and displayed as shown in Fig. 1(b), where the clicked item just now, called trigger item, is always located at the first position (highlighted by the red rectangle), and other items are recommended accordingly. Next, if users click one interested item within IFFP, the corresponding detailed page will be displayed as shown in Fig. 1(c), where users can directly purchase or add it into the Shopping Cart. We refer to this recommendation scenario within IFFP as TIR scenario, which has become ubiquitous and popular in e-commerce platforms. In our online App, TIR has already been the standard recommended scenario, serving an essential way for users to enter the detail page for further purchase. Besides, TIR scenarios have contributed more than 60% of the item page view (IPV) among all recommendation scenarios from our app, which indicates the significance value of the proposed task and the performance improvements. The similar scenario can also be found in messaging APPs, *e.g.*, recommendation suggestion for relevant passages in WeChat Top Stories [23]. In this paper, we focus on the Click-Through Rate (CTR) prediction task for TIR in e-commerce scenarios, which can improve user experience and increase the revenue of the platform.

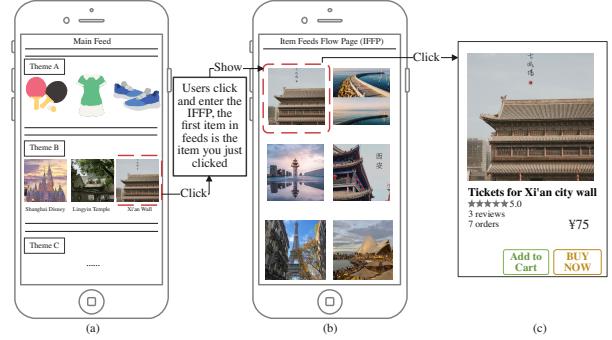
As we know, CTR is playing a crucial role in recommendation [2, 12, 15, 16, 29, 30], which aims to predict the probability of users clicking items. Recently, inspired by the success of deep learning in various research fields, *e.g.*, natural language processing [5, 11, 27] and computer vision [6, 9, 20], deep learning based

methods also have been proposed for the CTR prediction task, such as PNN [14], DeepFM [4], and DCN [21]. These methods, from the perspective of feature interactions, firstly map large-scale sparse features into fixed low dimensional embedding vectors and then feed them into fully connected layers to learn feature representations, neglecting of capturing users' interests from their historical behaviors. Alternatively, increasing novel methods have been proposed to extract users' interests from their historical behaviors. Deep Interest Network (DIN) [30], the first attention based work in the CTR area, employs attention mechanism to dynamically reweigh users' historical behaviors with respect to the target item. Deep Interest Evolution Network (DIEN) [29] is further proposed to model users' interests evolving process in e-commerce system. In Search-based Interest Model (SIM) [13], a kind of hard-search mode is proposed to extract users' interests with respect to the target item, which selects and aggregates only behaviors with the same category as the target item into a sub behavior sequence.

Despite effective, we figure out that those methods are not optimal in a trigger-induced recommender due to the discrepancy between TIR and traditional scenarios. Users probably have multiple interests in their historical behaviors, such as electronics, clothing, and snacks, resulting in diverse items recommended accordingly in non-TIR scenarios, *e.g.*, the scenario in Fig. 1(a). However, in TIR scenarios, users' instant interest can be explicitly induced with a trigger item. For example, given the category of the trigger item *Electronic*, it implies that the user is only interested in items related to *Electronic* category at that transient moment. Therefore, if directly employing existing methods, such as DIN [30] or DIEN [29], the performance will be seriously degraded because they do not explicitly model the instant interest induced by the trigger item. An intuition is that if we can leverage the trigger item to discover users' instant interest, we can recommend more relevant target items that will be probably clicked, *i.e.*, improving the CTR performance. However, it is not trivial due to the following challenges.

- Challenge 1: Users' instant interest induced from a trigger item are inherently noisy, because there are some accidental clicks on wrong items in users' behaviors. How to evaluate users' real intent for the trigger item remains challenging.
- Challenge 2: Users always show multiple interests from their historical behaviors. However, in TIR scenarios, users usually show their instant interest on the trigger item. Therefore, how to extract user's interest from their historical behaviors with respect to the clicked trigger item and the target item simultaneously is unexplored.

To address these challenges, we propose a novel model named Deep Interest Highlight Network (DIHN) for CTR prediction in TIR scenarios. Specifically, it consists of a User Intent Network (UIN), a Fusion Embedding Module (FEM), and a Hybrid Interest Extracting Module (HIEM). UIN responds to generate a precise probability score to predict user's intent on the trigger item. FEM adaptively fuses trigger item and target item embeddings based on the prediction from UIN. For example with two extreme cases, one is that if users have no any interest on trigger item but only click it casually, the results of FEM will degrade to the target item embedding. The other is that if users have exactly intense interest on the trigger item, the FEM will degrade to the trigger item embedding. Actually,



**Figure 1: An illustration of the online TIR scenario in a popular e-commerce platform, where users' instant interest can be explicitly induced by the trigger item (clicked item) and relevant target items are recommended accordingly.**

the result predicted by UIN reflects the intensity of users' instant interest, which is leveraged in FEM to fuse both embeddings in an adaptive manner for better users' interest extraction. In addition, HIEM can effectively highlight users' exact interest from their behaviors by leveraging two kinds of modelling paradigms named Soft Sequential Modelling (SSM) and Hard Sequential Modelling (HSM). SSM adaptively extracts the representation vector of users' interests by taking into consideration the relevance of historical behaviors with respect to the FEM. While HSM, borrowing the idea of hard-search mode from SIM [13], firstly selects the behaviors with the same category as the trigger item, and then aggregates these behaviors as a sub behavior sequence, followed by employing it to capture users' corresponding interests. Finally, all the representation features are concatenated with several raw context features and fed into fully connected layers to generate the final prediction results.

The main contributions of our paper are as follows:

- To the best of our knowledge, this is the first work to study the important recommendation problem named trigger-induced recommendation in e-commerce, which poses new challenges beyond existing recommender system.
- We propose a novel CTR model named DIHN for TIR scenarios, which can learn more expressive user interest representation and achieve more precise CTR prediction based on three carefully devised components.
- We conduct extensive experiments on the real-world offline datasets, and the results demonstrate the effectiveness of the proposed DIHN than representative state-of-the-art solutions. It is notable that DIHN has been deployed in our online recommender system and delivers significant improvement, further confirming its value in industrial applications.

The rest of our paper is organized as follows. Section 2 presents a brief review of related works, followed by the details of the proposed model in Section 3. Experiment setups as well as the corresponding results and analysis are presented in Section 4. We finally conclude the paper and discuss the future work in Section 5.

## 2 RELATED WORK

In this section, we will review the related work from following three respects briefly: Feature Interaction, User Behavior Modelling and Trigger-Induced Recommendation.

**Feature Interaction:** Nowadays, academic and industrial communities have paid more and more attention on capturing feature interactions instead of exhausting feature engineering works. DCN [21] and Wide&Deep [1] creatively replace the manual features transformation with neural networks for better memorization and generalization abilities. NCF [7], DeepFM [4] and DMF [24] impose a neural network with multiple MLPs to model the feature interactions between users and items. AutoInt [17] and CAN [28] further propose the self-attention mechanism for comprehensive feature interactions. Moreover, GNNs [3] realize feature interaction from the perspective of graphs and achieve great success. In our DIHN, we also explore higher-order feature interactions from all the feature embeddings as well as raw context features.

**User Behavior Modelling:** Recently, a series of works are proposed to capture users' interests from their rich historical behavior data with different neural network architecture such as Transformer [2, 18], CNN [19, 25], Capsule [10], RNN [8]. For example, DIN [30] emphasizes that users' interests are diverse and an attention mechanism is introduced to capture users' diverse interests on the different target items. DIEN [29] refines GRU to model evolution of interest and proposes an auxiliary loss to capture latent interest from users' behaviors. Pi et al. [13] proposes a novel memory based architecture named MIMN to capture users' interests from long sequential behavior data, which is the first industrial solution that is capable of handling long sequential user behavior data with length scaling up to thousands. These existing recommendation models, despite effective, are struggling in discovering users' interests in the TIR scenario due to the absence of explicitly modelling the trigger item.

**Trigger-Induced Recommendation:** The most relevant work to ours is the R3S [23], *i.e.*, Real-time Relevant Recommendation Suggestion, which can be regarded as a research for TIR to some extent. R3S proposes a novel recommendation suggestion task for extended reading in recommendation, aiming to predict users' intent on extended reading and recommend appropriate relevant items given the current clicked item. Following the definition of our work, the item a user has just clicked can be regarded as the trigger item. Specifically, R3S extract users' interests from multiple aspects including feature interactions, semantic similarity and information gain between clicked item (trigger item) and relevant candidate items. However, R3S fails to capture users' interests from their historical behaviors given clicked item (trigger item) and candidate items. Alternatively, in our DIHN, we devise a FEM module to fuse trigger item and target item embeddings seamlessly, which is further utilized to extract users' interests from their behaviors.

## 3 PROPOSED METHOD

In this paper, we propose a novel recommendation model named DIHN for CTR prediction in TIR scenarios. The overall architecture of DINH is depicted in Fig. 2, which consists of four main components including 1) Feature Representation Layer (FRL), which transforms all kinds of high dimension sparse one-hot vectors into fixed-length low dimension dense vectors; 2) User Intent Network

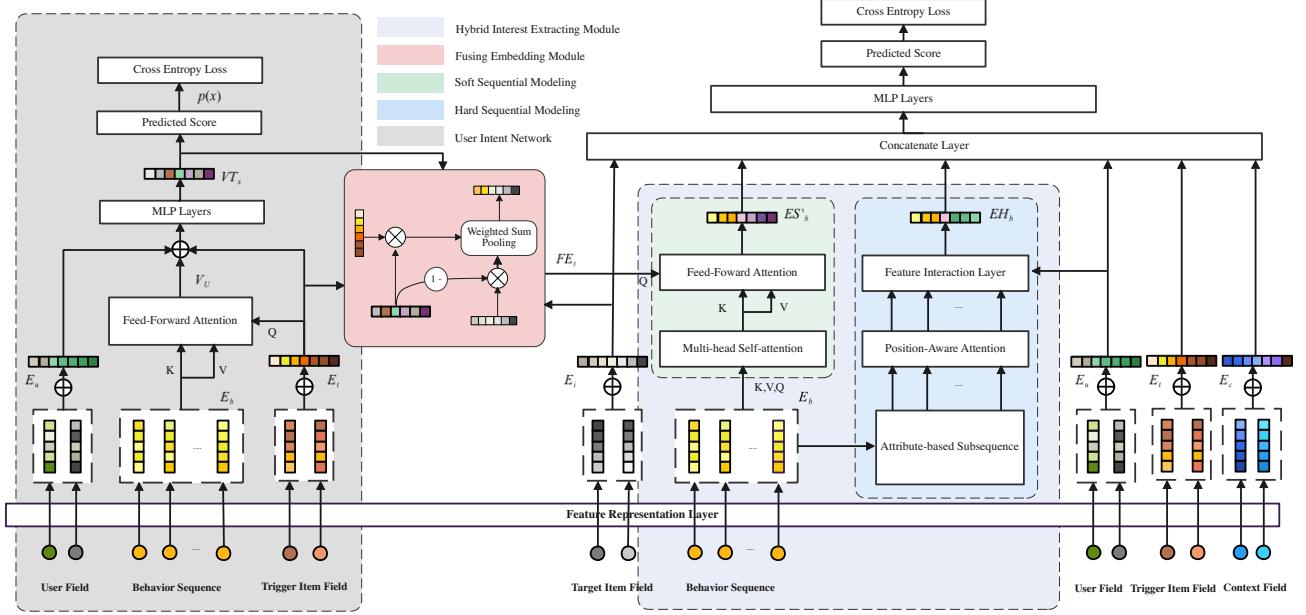
(UIN), which responds to generate a precise probability score to predict user's intent on the trigger item; 3) Fusion Embedding Module (FEM), which adaptively fuses trigger item and target item embeddings based on the result of UIN; and (4) Hybrid Interest Extracting Module (HIEM), which can effectively highlight users' instant interest from their behaviors based on the result of FEM. Finally, all the representation features are concatenated with several raw context features and fed into multi-layer perceptron (MLP) layers for final CTR prediction. In the remaining of this section, we will introduce them in detail.

### 3.1 Motivation

Generally speaking, users could show multiple interests from their historical behaviors, such as electronics, clothing, and snacks, resulting in diverse items recommended accordingly in non-TIR scenarios, *e.g.*, the scenario in Fig. 1(a). Several sequential modelling methods are proposed to capture user's dynamic interests from their historical behaviors with respect to different target items. For example, Deep Interest Network (DIN) [30], as the excellent representative work, is designed to activate relevant users' behaviors with respect to corresponding targets and obtain adaptive representation vectors for users' interest extraction. Despite effective, we figure out that those existing methods are not so effective in a trigger-induced recommender due to the absence of explicitly modelling the instant interest induced by a trigger item. For example, given the category of the clicked trigger item *Electronic*, it implies that the user is only interested in items related to *Electronic* category at that transient moment. If directly employing DIN without explicitly modeling users' instant interest, the performance of the model will be seriously degraded. Alternatively, we can leverage the trigger item together with target items, to suitably model users' sequential behaviors and explore their exact interests. However, there are several challenges as introduced in previous section should be carefully addressed. In the following, we will introduce how to address these challenges with the elaborate designed components in DIHN.

### 3.2 Feature Representation Layer

In DIHN, we mainly employ four categories of features namely *User Profile*, *User Behaviors*, *Trigger*, *Target Item*, and *Context* for users' interest representation, where each category feature consists of several fields. For example, *User Profile* contains *age*, *sex*, *purchase level*, etc. *User Behaviors* contains the sequential list of users visiting items. *Trigger* as well as *Target Item* contain *Id*, *Category*, etc. And *Context* contains *time*, *weather*, etc. In addition, feature in each field is normally transformed into high-dimensional sparse one-hot features via encoding. For example, the feature value *male* from *sex* field of *User Profile* category is encoded as [0,1]. Assuming the concatenation results of different fields' one-hot vectors from *User Profile*, *User Behaviors*, *Trigger*, *Target Item* and *Context* are denoted as  $X_u$ ,  $X_b$ ,  $X_t$ ,  $X_i$ , and  $X_c$ , respectively, they can be further transformed into low dimensional dense representations by utilizing embedding layers, named as  $E_u$ ,  $E_b$ ,  $E_t$ ,  $E_i$ , and  $E_c$ , respectively. Besides, in the sequential CTR prediction task,  $X_b$  usually contains a list of users behaviors, mathematically denoted as  $X_b = [b_1; b_2; \dots; b_T] \in \mathbb{R}^{K \times T}$ ,  $b_t \in \{0, 1\}^K$ , where  $T$ ,  $K$ , and  $b_t$  represent the length of users' behaviors, the total number of all



**Figure 2: The overview architecture of our proposed DIHN model, which consists of Feature Representation Layer (FRL), User Intent Network (UIN), Fusion Embedding Module (FEM), and Hybrid Interest Extracting Module (HIEM) modules.**

candidate items, and the one-hot vector of the  $t$ -th behavior, respectively. Similarly,  $E_b$  is denoted as  $E_b = [e_1; e_2; \dots; e_T] \in \mathbb{R}^{D \times T}$ , where  $D$  and  $e_t$  represent the dimension of dense feature transformed by embedding layer, and the transformed embedding feature of the  $t$ -th behavior, respectively.

### 3.3 User Intent Network

In TIR scenarios, users can explicitly express their instant interests via the trigger item. However, users' instant interest induced from a trigger item are inherently noisy, because there are some accidental clicks on wrong items in users behaviors. Therefore, how to evaluate user's real intent on the trigger item remains challenging. Here, we proposed a User Intent Network (UIN) to address the challenge, which responds to generate a precise probability score accounting for user's real intent on the trigger item.

Referring to the UIN module in Fig.2, where we utilize three categories of features, i.e., *User Profile*, *User Behaviors*, *Trigger* to estimate the probability. To adaptively calculate user sequential representation of *User Behaviors* with respect to the trigger item, we refer to the architecture of DIN [30]. It can be formulated as:

$$V_U = f(E_b; E_t) = \sum_{i=1}^T a(e_i, E_t) e_i = \sum_{i=1}^T w_i e_i, \quad (1)$$

where  $V_U$  denotes the user representation feature with respect to  $E_t$ , and  $a(\cdot)$  is a feed-forward network whose output is the activation weight  $w_i$ , as illustrated in DIN [30].

Now, given the user representation feature  $V_U$ , we firstly concatenate it with other dense representation vector  $E_u$  and  $E_t$ , i.e.,

$x = [V_U; E_u; E_t]$ . It is then fed into multiple fully connected layers to further learn the high-order feature interactions. Next, after the activation function, we obtain the output of the UIN, denoted as  $p(x)$ , which represents the predicted probability of the trigger item being clicked in TIR scenarios (e.g., depicted in Fig.1(b)). Finally, we define the objective function of this module as follows:

$$\text{Loss}_t = -\frac{1}{N} \sum_{(x, y) \in S} (y \log p(x) + (1 - y) \log(1 - p(x))), \quad (2)$$

where  $S$  denotes the training set with total size  $N$ , and  $y \in \{0, 1\}$  is the ground truth label representing whether users clicking the trigger item or not. Additionally, in TIR scenarios, each sample has several additional information from a trigger item, including features of the trigger item, the label of whether the trigger item being clicked. Referring to the IFFP depicted in Fig. 1(b), if users clicking the trigger item, the samples from the same IFFP with the trigger item will have the same auxiliary label (e.g., positive), which can be used to supervise the learning of UIN.

Besides, the representation vector from last MLP layer is denoted as  $VT_x$ , which has the same dimension with  $E_t$  and  $E_i$ , will be utilized in following module.

### 3.4 Fusion Embedding Module

In non-TIR scenarios, existing representative methods, e.g., DIN [30], always employ attention mechanism to dynamically re-weight users historical behaviors with respect to target items. However, in TIR scenarios, we figure out that if directly employing DIN without explicitly modeling users' instant interest induced from the trigger

item, the performance of the model will be seriously degraded. Alternatively, a suitable users' behavior modelling solution could be adaptively fuse the trigger item with target item. Specifically, we proposed a Fusion Embedding Module (FEM), which adaptively fuses trigger item and target item embeddings based on the result of UIN. For example with two extreme cases, one is that if users have no any interest on trigger item but only click it casually, the results of FEM module should degrade to the target item embedding, which indicates only behaviors related to the target item will be selected to extract users' interest. The other is that if users have exact interest on the trigger item, the FEM module will degrade to the trigger item embedding, indicating only behaviors related to the trigger item will be selected for users' interest extraction. Actually, the result predicted by UIN reflects the intensity of users' instant interest, which is used (*e.g.*, element-wise products) in FEM to fuse both embeddings in an adaptive manner for better users' interest extraction. It is formulated as:

$$FE_t = VT_x E_t + (\mathbf{1} - VT_x) E_i, \quad (3)$$

where  $FE_t$ ,  $VT_x$ ,  $E_t$ , and  $E_i$  represent the fusion embedding, the representation vector from last MLP layer in UIN, the embedding of trigger item, and the embedding of target item, respectively. Here,  $\mathbf{1}$  is a vector with the same dimension as  $VT_x$ ,  $E_t$ , and  $E_i$ .

### 3.5 Hybrid Interest Extracting Module

In TIR scenarios, users usually show their instant interest on the trigger item. How to extract user's interest from their historical behaviors with respect to the trigger item and the target item simultaneously is unexplored. In this section, given users' instant interest via a trigger item and the fusing result from FEM, we devise a novel module named Hybrid Interest Extracting Module (HIEM), which can effectively highlight users' exact interest from their behaviors. Specifically, from two different behavior modelling perspectives, we proposed the Hard Sequential Modelling (HSM) and Soft Sequential Modelling (SSM), which are detailed as follows.

**3.5.1 Hard Sequential Modelling.** The trigger item can reflect users' instant interests. For example, given the category of the trigger item *Electronic*, it implies that the user is only interested in items related to *Electronic* category at that transient moment. Following the hard-search mode from SIM [13], we propose the Hard Sequential Modelling (HSM), indicating that only behaviors with the same attribute (*e.g.*, category, destination) as the trigger item will be firstly selected and aggregated as a sub behavior sequence, followed by employing it to capture users' corresponding interests.

Taking the attribute *category* as example, mathematically, let  $E_{bc} = [e_{1_c}; e_{2_c}; \dots; e_{T_c}] \in \mathbb{R}^{D \times T_c}$  denotes the sub-sequence from  $E_b = [e_1; e_2; \dots; e_T] \in \mathbb{R}^{D \times T}$ , each element within  $E_{bc}$  has the same category as the trigger item and ordered by their occurring time. As we all known, user's interest may change with time, where more recent behaviors reflect user's temporal interest better, we apply an attention mechanism with positional encoding as query to adaptively learn the weight for each behavior, where the position of user behavior is the serial number in the behavior sequence. It can be formulated as follows:

$$\alpha_t = z^T \tanh(W_p p_t + W_e e_t + b), \quad (4)$$

$$\alpha_t = \frac{\exp(\alpha_t)}{\sum_{i=1}^T \exp(\alpha_i)}, \quad (5)$$

where  $p_t \in \mathbb{R}^{d_p}$  is the t-th position embedding,  $e_t \in \mathbb{R}^{d_e}$  is the feature vector for the t-th behavior,  $W_p \in \mathbb{R}^{d_h \times d_p}$ ,  $W_e \in \mathbb{R}^{d_h \times d_e}$ ,  $b \in \mathbb{R}^{d_h}$ , and  $z \in \mathbb{R}^{d_h}$  are learnable parameters, and  $\alpha_t$  is the normalized weight for the t-th behavior. By weighted-sum pooling, the feature vector list of user's sub behaviors is mapped into fixed-length user representation vector  $u_c$ :

$$u_c = \sum_{i=1}^T \alpha_i e_i. \quad (6)$$

In the same way, the above processing method can be applied in other attributes (*i.e.*, destination, tag) of trigger item, which reflect users' interests from different aspects, getting the *destination* based vector  $u_d$  and *tag* based representation  $u_t$ , respectively. For further capturing the high-order interaction between these representation vectors and original user profile features, we propose the Feature Interaction Module (FIM). To be specific, we first concatenate features into one representation vector as:

$$r_g = [E_u, u_c, u_d, u_t]. \quad (7)$$

Then, the combination of all the features is taken as the input of the FIM, which utilizes the global attention unit to extract the relationship among different parts of the input features. The output is calculated by:

$$EH_b = \sum_{l=1}^n \frac{\exp(\tanh(r_l \cdot W_l + b_l))}{\sum_{l=1}^L \exp(\tanh(r_l \cdot W_l + b_l))} r_l, \quad (8)$$

where  $W_l$  and  $b_l$  are weight and bias matrix, respectively,  $r_l$  is the representation of each individual feature,  $L$  is the number of total features in  $r_g$ , and  $EH_b$  is the output vector of FIM.

**3.5.2 Soft Sequential Modelling.** Different from users' interest extraction in non-TIR scenarios that directly calculating the relevant weight between target item and users' behaviors, while in TIR scenarios, users usually can show their instant interest with the clicked trigger item. An alternative modeling solution is to extract users' interests with respect to the clicked trigger item and the target item simultaneously. Therefore, from another modelling perspective, we proposed the Soft Sequential Modelling (SSM), which adaptively calculates the representation vector of users' behaviors with respect to the fusing result from FEM. In addition, since attention mechanism, specifically the Multi-Head Self-Attention (MHSA), which can capture the dependencies between representation pairs despite their distance within the sequence, has become a key ingredient for sequential modeling, we also adopt the MHSA to effectively extract users' interest representation. Mathematically, it is defined as:

$$MHSA(F^l) = [head_1, head_2, \dots, head_h]W^O, \quad (9)$$

$$head_i = Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d/h}}V\right), \quad (10)$$

where  $F^l$  denotes the  $l$ -th layer input and  $Q = F^l W_i^Q$ ,  $K = F^l W_i^K$ ,  $V = F^l W_i^V$  denotes the linear transformations of the input  $F_l$ . The projection matrix  $W_i^Q \in \mathbb{R}^{D \times D/h}$ ,  $W_i^K \in \mathbb{R}^{D \times D/h}$ ,  $W_i^V \in \mathbb{R}^{D \times D/h}$  and

$W_O \in \mathbb{R}^{D \times D}$  represents the corresponding learnable parameters for each head.  $\sqrt{d/h}$  is the scale factor for normalization.

Next, we endow the non-linearity of the self-attention block by applying a feed-forward network, *i.e.*,

$$F^l = [FFN(F_1^l)^T; FFN(F_2^l)^T; \dots; FFN(F_n^l)^T], \quad (11)$$

$$FFN(x) = (Relu(xW_1 + b_1))W_2 + b_2, \quad (12)$$

where  $W_1, b_1, W_2, b_2$  are learnable parameters.

In this way, given users sequential behaviors  $E_b = [e_1; e_2; \dots; e_T]$ , where  $T$  and  $e_i$  represent the length of users' behaviors and the embedding of  $i$ -th user behavior, respectively, we can obtain the representation vectors  $E'_b = [e'_1; e'_2; \dots; e'_T]$  by MHSA. Next, given  $E'_b$  and the fusing result from FEM, we employ DIN [30] to extract the user representation vector, denoted as  $ES'_b$ .

Then, all the dense representation vectors are concatenated with the raw context features, defined as  $x' = [ES'_b; EH_b; E_U; E_t; E_i]$ , which are then fed into fully connected layers to learn the higher-order feature interactions, followed by a relu function to obtain the predicted probability of the target item being clicked. Similarly, the objective function of this module is the negative log-likelihood function like Eq. (2). We denote it as  $Loss_t$ . Finally, the total loss of our proposed DIHN model is formulated as:

$$Loss = \alpha Loss_t + \beta Loss_i, \quad (13)$$

where  $\alpha$  and  $\beta$  are hyper parameters to balance these two losses.

## 4 EXPERIMENTS

To evaluate the effectiveness of our proposed DIHN, in this section, we conduct extensive experiments to compare it with several representative SOTA methods on both offline datasets and online deployment. We firstly present the details of benchmark dataset and experimental setup, including the offline dataset preparation, evaluation metrics, and a brief description of representative SOTA methods. Then, the main results and analysis are presented, followed by ablation studies.

### 4.1 Dataset and Experimental Setup

To the best of our knowledge, there are no TIR scenario based public datasets tailored for the new proposed problem. To fill this gap, we establish an offline dataset by collecting data from our real-world e-commerce TIR scenarios, named Industrial Dataset. Besides, to demonstrate the generalization of our proposed model, we also carry out experiments on a public dataset, *i.e.*, Alimama Dataset, where we manually construct the trigger items to suit for the TIR problem. Table 1 shows the statistics of all datasets. We will introduce them in detail.

**Industrial Dataset.** We firstly establish the offline dataset by collecting the users' behaviors and feedback logs from our online e-commerce platform, which is one of the largest third-party retail platforms in the world. Each sample contains *User Profile/User Behaviors/Trigger/Target Item/Context* features described in previous section, as well as feedback logs (*i.e.*, whether users click the target item or not). Besides, *Trigger* also has extra information, including features of the trigger item and the auxiliary label indicating

whether the trigger item has been clicked or not in IFFP. Referring to the IFFP depicted in Fig. 1(b), if users click the trigger item within IFFP, the auxiliary label is set to 1 (*i.e.*, positive), otherwise 0. Samples collected from the same IFFP with the same trigger item have the identical auxiliary label, which can be used to supervise the learning of UIN.

**Public Dataset.** Alimama Dataset<sup>2</sup> is a public dataset released by Alimama, which is an online advertising platform of China. It contains 26 million records from ad display/click logs with 1 million users and 800 thousand ads in 8 days. However, it is not collected from TIR scenarios, due to the lack of trigger items. Therefore, we manually define the trigger items as follows. For each sample (one display of a user and an target ad at time  $t$ ), we search for the latest clicked advertising of the user within 4 hours before  $t$ . It is worth noting that not every sample can be associated with a trigger, since users behaviors are sparse. For instance, some users may have no click behavior in the dataset. So we only select those samples that can be associated with a trigger item. If the trigger item has the same category with the target advertising, the auxiliary label is set to 1, otherwise 0.

**Table 1: Statistics of the offline datasets.**

Dataset	#User	#Item	#Impression
Industrial.	22,269,532	201,004	436,240,250
Public.	26,557,962	846,812	1,366,056

**Experimental Setup.** We implement all the competitive methods in TensorFlow using the Adam optimizer with a exponential decay learning rate schedule. The initial learning rate is set as 0.001 and decay rate is set as 0.9. The hyper-parameters  $\alpha$  and  $\beta$  in Eq. (13) are set 1 and 0.8, respectively. We take AUC [30] as the main metric to evaluate model's performance, which is widely adopted in the field of CTR prediction task. We run each method 10 times and report the average result.

### 4.2 Competitors

To demonstrate the effectiveness and superiority of DIHN, we compare it with several representative SOTA methods in TIR scenarios, which can be grouped into two categories.

Group 1: Methods without explicitly modelling users' instant interest. Classical models are WDL, DeepFM, and DIN, etc., which do not include information from the trigger item. We show them in detail.

- **WDL** [1]: It jointly trains a wide linear model and a deep neural model, which combines the benefits of memorization and generalization for CTR prediction.
- **DeepFM** [4]: It emphasizes both low- and high-order feature interactions by combining the power of traditional FM module and deep MLP module.
- **DIN** [30]: It utilizes attention mechanism to activate relevant users' behaviors with respect to corresponding targets and learns an adaptive representation vector for users' interests.

<sup>2</sup><https://tianchi.aliyun.com/dataset/dataDetail?dataId=56>

**Table 2: Comparison results of methods in Group 1.**

Model	Industrial (mean ± std)	Public (mean ± std)
WDL	0.7231 ± 0.00019	0.6351 ± 0.00034
DeepFM	0.7271 ± 0.00031	0.6359 ± 0.00039
DIN	0.7351 ± 0.00021	0.6382 ± 0.00024
DIEN	0.7364 ± 0.00041	0.6388 ± 0.00033
MIAN	0.7379 ± 0.00038	0.6391 ± 0.00039
<b>DIHN</b>	<b>0.7506 ± 0.00025</b>	<b>0.6421 ± 0.00019</b>

**Table 3: Comparison results of methods in Group 2.**

Model	Industrial (mean ± std)	Public (mean ± std)
DIN+2TA	0.7401 ± 0.00029	0.6394 ± 0.00029
DIEN+2TA	0.7389 ± 0.00042	0.6391 ± 0.00036
R3S	0.7378 ± 0.00031	0.6392 ± 0.00017
R3S+2TA	0.7415 ± 0.00027	0.6403 ± 0.00024
<b>DIHN</b>	<b>0.7506 ± 0.00025</b>	<b>0.6421 ± 0.00019</b>

- **DIEN** [29]: It adopts an interest extractor layer to capture temporal interests from users' historical behaviors and integrates GRUs with attention mechanism for further capturing the involved interests with respect to the target item.
- **MIAN** [26]: It contains a multi-interactive layer to capture multiple representations of user preference from sequential behaviors. Besides, it utilizes a global interaction module to learn the high-order interactions and balances the different impacts of multiple features.

Group 2: Methods with explicitly modelling users' instant interest. As aforementioned, there are no prior works in TIR, thus we equip some SOTA CTR models with the capacity of utilizing the trigger item for fair and solid comparison.

- **DIN+2TA**: It utilizes the attention mechanism to extract users' interest representation not only with respect to the target item, but also to the trigger item inducing users' explicit instant interests.
- **DIEN+2TA**: It captures users' evolved interests from users' historical behaviors by using the similar attention structure in **DIN+2TA**.
- **R3S**[23]: It extracts users' interest from multiple aspects including feature interactions, semantic similarity and information gain between clicked item and target items.
- **R3S+2TA**: Since the model R3S does not utilize users' sequential behaviors, in this model, we add it as input and capture users' interests with respect to the trigger item and the target item simultaneously.

### 4.3 Main Offline Comparison Results

4.3.1 *The Comparison Results with Competitors in Group 1.* We start off reporting the AUC of all the competitive methods in Group 1. It can be seen that DeepFM achieves 0.55% and 0.13% AUC improvement over the WDL for Industrial and Public datasets, respectively. It demonstrates the remarkable effectiveness of both low- and high-order feature interactions, attributing to the combination of traditional FM and deep MLP modules. Different from WDL and DeepFM without employing users' behaviors, DIN, DIEN,

and MIAN performance better on both datasets, benefiting from capturing users' interests by modelling users' historical behaviors. For example, DIN achieves AUC gains of 1.10% and 1.66% over DeepFM and WDL on the Industrial dataset, respectively. As for the MIAN, it can efficiently learn the multiple fine-grained interactions from historical behaviors, user-specific and contextual interactions. While DIEN only focuses on the evolving interests within users' behaviors. Consequently, MIAN achieves 0.21% and 0.05% improvement over DIEN on the two datasets, respectively. By contrast, our proposed DIHN significantly outperforms above state-of-the-art competitors, which mainly benefits from explicitly modeling users' instant interest induced from the trigger item. Specifically, compared with MIAN, the improvement on AUC is 1.72% and 0.47% on the two datasets, respectively. It is worth mentioning that the gain of 0.01 on the offline dataset always means significant increment for online CTR tasks [22]. The experimental comparison results also reveal that explicitly utilizing users' instant interests in TIR scenarios can achieve the best CTR performance.

4.3.2 *The Comparison Results with Competitors in Group 2.* For the competitive methods in Group 2, we utilize the attention mechanism to activate relevant users' behaviors not only with respect to target item but also the trigger item. First, DIN+2TA (or DIEN+2TA) simply employs two attention mechanism with respect to the trigger item and target items individually, while R3S+2TA employs a fusion of them. Specifically, R3S+2TA considers semantic relevance and information gain between the trigger item and target items, resulting in 0.19% (or 0.35%) and 0.14% (or 0.18%) AUC gains over DIN+2TA (or DIEN+2TA) on the two datasets, respectively. Despite effective, it has no elaborated modelling between users' historical behaviors and the fusion results. As for the proposed DIHN, it uses the FEM module to supervise the extraction of users' interests and the HIEM module to highlight users' exact interest from their behaviors, achieving the best performance among all the competitors. For example, the AUC gains over R3S+2TA are 1.23% and 0.28% on the two datasets, respectively. It is worth noting that the improvement on the Alimama dataset is not as obvious as on the Industry dataset due to the reason that the Alimama dataset is not directly collected from online TIR scenario but established in a synthetic way where we manually mine the information of trigger items according to self-defined rules. Consequently, it may mismatch with the real online TIR scenario and limit our model's performance. Additionally, compared with competitors in Group 1, which are not explicitly modelling users' instant interest, DIHN also achieves significant gains, which further demonstrates the effectiveness of our method.

### 4.4 Ablation Study

To investigate the effectiveness of each component in our model, in this subsection, we present ablation studies on the offline datasets.

4.4.1 *Effectiveness of HSM module.* To evaluate the impact of HSM module for users' interest extraction, we compared DIHN with DIHN w/o HSM, indicating DIHN without the HSM module. The results are shown in Table 4.4. Since HSM module can help filter irrelevant noise so that the model focuses on the most relevant behaviors for users' interest extraction, especially when the users

**Table 4: Results of ablation study on offline datasets.**

Model	Industrial (mean $\pm$ std)	Public (mean $\pm$ std)
DIHN w/o HSM	$0.7447 \pm 0.00009$	$0.6390 \pm 0.00020$
DIHN(scalar)	$0.7484 \pm 0.00011$	$0.6409 \pm 0.00013$
DIHN w/o SSM+m	$0.7443 \pm 0.00031$	$0.6388 \pm 0.00034$
DIHN w/o SSM+target	$0.7469 \pm 0.00013$	$0.6401 \pm 0.00022$
DIHN w/o SSM+trigger	$0.7475 \pm 0.00019$	$0.6403 \pm 0.00012$
DIHN w/o SSM+concat	$0.7459 \pm 0.00041$	$0.6399 \pm 0.00027$
<b>DIHN</b>	<b><math>0.7506 \pm 0.00025</math></b>	<b><math>0.6421 \pm 0.00019</math></b>

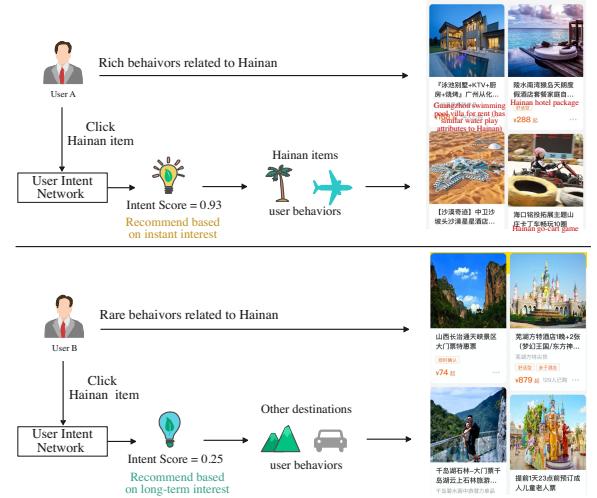
have intense interest on trigger item, it is a straightforward but effective solution to improve the performance.

**4.4.2 Effectiveness of UIN module.** The module UIN is designed to predict users' instant interests on the trigger item. We now investigate whether this module can strengthen users' interests representation. To be specific, we evaluate DIHN without the module while employing four kinds of SSM methods, *e.g.*, *DIHN w/o SSM+m*, *DIHN w/o SSM+target*, *DIHN w/o SSM+trigger*, and *DIHN w/o SSM+concat*, denoting that DIN where mean pooling for the representation of users' sequential behaviors, DIN where the query is target item, DIN where the query is the trigger item, DIN where the query is the concatenation of the trigger item and target items, respectively. The offline comparison results in Table 4.4 show the effectiveness of UIN module and the benefit of using supervised auxiliary label from whether users clicking the trigger item. Users' interest on the trigger item obtained by the UIN module can help extract users' interests with respect to the target item and bring further performance improvement.

**4.4.3 Effectiveness of FEM module.** Table 4.4 also shows the results of different embedding fusing methods in the FEM module, where DIHN (scalar) denotes the model with scalar fusing operator, *i.e.*, replacing the vector  $VT_x$  with a scalar  $p(x)$  in Eq.3, while DIHN uses an element-wise embedding fusing operator. As can be seen, DIHN obtains 0.3% and 0.19% gains over DIHN (scalar) on the two offline datasets, respectively. Although DIHN (scalar) can directly employ the predicted score from UIN to control the fusion of two embeddings, it only uses the scalar  $p(x)$ , which ignores the importance at different dimensions. Therefore, we propose to use the element-wise fusion operator in the FEM module.

## 4.5 Online A/B Testing Results

To further validate the effectiveness of our proposed model, we also conducted online A/B testing on our e-commerce platform. However, it is not an easy job to deploy the proposed model in our online recommender system since it serves at the scale of tens of millions of users every day. Besides, the traffic is very expensive from the business view. Considering this fact, we only deployed the best offline method as our baseline method, *i.e.*, R3S+2TA model. Simultaneously, to make the online evaluation fair, confident, and comparable, each deployed method for an A/B test has involved the same number of users, *i.e.*, millions of users. Careful online A/B testing was conducted from 2021-06 to 2021-07. DIHN contributes up to 6.5% CTR promotion, which is a significant improvement and demonstrates the effectiveness of our proposed method. Now, DIHN has been deployed online and serves the main traffic.



**Figure 3: Two cases of DIHN in the travel scenario, where two different users clicked the same trigger item (Hainan vacation item). User A (top) has rich behaviors related to Hainan, while user B (bottom) has rare behaviors.**

## 4.6 Case Study

Taking the travel scene as an example, we present two real-world cases shown in Figure 3 to illustrate the effectiveness of DIHN. When user A comes into the TIR scenario, the UIN module predicts A's intent score for the trigger item, *i.e.*, an item from Hainan province, as 0.93, which indicates the user has intense interest on items related to Hainan province. Therefore, more products related to Hainan or similar scenic spots will be recommended to satisfy the user's instant intention, as shown in Figure 3. Different from user A, user B has rare Hainan province related behaviors, therefore the intent score on the trigger item predicted by UIN module is very low, *i.e.*, 0.25. Such condition indicates user B has divergent interests, and the recommender system in IFFP should recommend less items related to Hainan province. As can be seen from the above two cases, DIHN can seamlessly adjust the suitable recommendation mode based on the user's instant interest on the trigger item.

## 5 CONCLUSION

In this paper, we introduce a new Trigger-Induced Recommendation (TIR) problem, where users' instant interest can be explicitly induced with a trigger item. Due to the discrepancy between TIR scenarios and non-TIR scenarios, we figure out that existing recommendation models are struggling in discovering users' instant interests in TIR scenarios. To tackle the problem, we propose a novel recommendation method DIHN, which shows great benefits of modeling users' instant interests for CTR prediction. Using a User Intent Network, DIHN can predict the extent of user's intent on the trigger item and accordingly rely on the trigger item or target item embeddings for user's interest extraction. DIHN not only achieves improvement over representative state-of-the-art methods on offline datasets, but also gains 6.5% CTR promotion on online e-commerce platform, confirming its superiority for TIR.

## REFERENCES

- [1] 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.
- [2] 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).
- [3] Marco Gori, Gabriele Monfardini, and Franco Scarselli. 2005. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, Vol. 2. IEEE, 729–734.
- [4] Hui Feng Guo, Ruiming Tang, Yunning Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).
- [5] Jian Guo, He He, Tong He, Leonard Lausen, Mu Li, Haibin Lin, Xingjian Shi, Chengguang Wang, Junyuan Xie, Sheng Zha, et al. 2020. GluonCV and GluonNLP: Deep Learning in Computer Vision and Natural Language Processing. *J. Mach. Learn. Res.* 21, 23 (2020), 1–7.
- [6] Fredrik K Gustafsson, Martin Danelljan, and Thomas B Schon. 2020. Evaluating scalable bayesian deep learning methods for robust computer vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 318–319.
- [7] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.
- [8] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).
- [9] Anastasia Ioannidou, Elisavet Chatzilari, Spiros Nikolopoulos, and Ioannis Kompatiari. 2017. Deep learning advances in computer vision with 3d data: A survey. *ACM Computing Surveys (CSUR)* 50, 2 (2017), 1–38.
- [10] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2615–2623.
- [11] Hang Li. 2017. Deep learning for natural language processing: advantages and challenges. *National Science Review* (2017).
- [12] Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Practice on long sequential user behavior modeling for click-through rate prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2671–2679.
- [13] 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.
- [14] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. 2016. Product-based neural networks for user response prediction. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 1149–1154.
- [15] Qijie Shen, Wanjie Tao, Jing Zhang, Hong Wen, Zulong Chen, and Quan Lu. 2021. SAR-Net: A Scenario-Aware Ranking Network for Personalized Fair Recommendation in Hundreds of Travel Scenarios. *arXiv preprint arXiv:2110.06475* (2021).
- [16] Qingquan Song, Dehua Cheng, Hanning Zhou, Jiyan Yang, Yuandong Tian, and Xia Hu. 2020. Towards automated neural interaction discovery for click-through rate prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 945–955.
- [17] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1161–1170.
- [18] 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.
- [19] 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.
- [20] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. 2018. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience* 2018 (2018).
- [21] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*. 1–7.
- [22] Hong Wen, Jing Zhang, Quan Lin, Keping Yang, and Pipei Huang. 2019. Multi-level deep cascade trees for conversion rate prediction in recommendation system. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 338–345.
- [23] Ruobing Xie, Rui Wang, Shaoliang Zhang, Zhihong Yang, Feng Xia, and Leyu Lin. 2021. Real-time Relevant Recommendation Suggestion. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 112–120.
- [24] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep Matrix Factorization Models for Recommender Systems.. In *IJCAI*, Vol. 17. Melbourne, Australia, 3203–3209.
- [25] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. 2019. A simple convolutional generative network for next item recommendation. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 582–590.
- [26] 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.
- [27] Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)* 11, 3 (2020), 1–41.
- [28] Guorui Zhou, Weijie Bian, Kailun Wu, Lejian Ren, Qi Pi, Yujing Zhang, Can Xiao, Xiang-Rong Sheng, Na Mou, Xinchen Luo, et al. 2020. CAN: Revisiting Feature Co-Action for Click-Through Rate Prediction. *arXiv preprint arXiv:2011.05625* (2020).
- [29] 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*.
- [30] 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*. ACM, 1059–1068.