Hierarchically Fusing Long and Short-Term User Interests for Click-Through Rate Prediction in Product Search

Qijie Shen*
Alibaba Group
Hangzhou, China
qijie.sqj@alibaba-inc.com

Hong Wen*
Alibaba Group
Hangzhou, China
qinggan.wh@alibaba-inc.com

Jing Zhang
The University of Sydney
Darlington NSW 2008, Australia
jing.zhang1@sydney.edu.au

Qi Rao
Alibaba Group
Hangzhou, China
rq145781@alibaba-inc.com

ABSTRACT

Estimating Click-Through Rate (CTR) is a vital yet challenging task in personalized product search. However, existing CTR methods still struggle in the product search settings due to the following three challenges including how to more effectively extract users' short-term interests with respect to multiple aspects, how to extract and fuse users' long-term interest with short-term interests, how to address the entangling characteristic of long and short-term interests. To resolve these challenges, in this paper, we propose a new approach named Hierarchical Interests Fusing Network (HIFN), which consists of four basic modules namely Short-term Interests Extractor (SIE), Long-term Interests Extractor (LIE), Interests Fusion Module (IFM) and Interests Disentanglement Module (IDM). Specifically, SIE is proposed to extract user's short-term interests by integrating three fundamental interests encoders within it namely query-dependent, target-dependent and causal-dependent interest encoder, respectively, followed by delivering the resultant representation to the module LIE, where it can effectively capture user long-term interests by devising an attention mechanism with respect to the short-term interests from SIE module. In IFM, the achieved long and short-term interests are further fused in an adaptive manner, followed by concatenating it with original raw context features for the final prediction result. Last but not least, considering the entangling characteristic of long and short-term interests, IDM further devises a self-supervised framework to disentangle long and short-term interests. Extensive offline and online evaluations on a real-world e-commerce platform demonstrate the superiority of HIFN over state-of-the-art methods.

CCS CONCEPTS

• **Information system → Information retrieval**.

*denotes equal contribution, and Shen is the 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 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.
CIKM '22, October 17–21, 2022, Atlanta, GA, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557351

KEYWORDS

Personalized Product Search, Click-Through Rate Prediction

ACM Reference Format:
Qijie Shen, Hong Wen, Jing Zhang, 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 and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3511808.3557351

1 INTRODUCTION

With the explosive emergence of products in e-commerce platforms, product search serves an indispensable role in discovering desirable items that satisfy users. When a user submits a query, the product search will deliver a small yet well ranked product set from billions of candidates to end-users through a typical multi-stage pipeline of "match→prerank→rank→rerank" [14, 15, 30]. In this paper, we only focus on the Click Through-Rate (CTR) prediction task in the *rank* stage, which aims to predict the probability of users clicking items and has a great impact on improving user experience and boosting the revenue of e-commerce platforms.

The prerequisite for a CTR prediction task in personalized product search is to accurately extract the user preferences from historical behaviors and effectively integrate them with the current query [10]. Besides, it has been well recognized that there are two types of user preferences [4], *i.e.*, long and short-term ones, where the former exhibits users' inherent and relatively stable (evolving slowly) preferences, such as preferred color, fitting size, price preferences, imperceptibly influenced by the user's family background, age, marital status, education, *etc*. While the short-term interests convey user preference intention in a relatively short period, which can be inferred from their recent behaviors and also affected by incidentally transient events, such as new product release, season change and special personal occasions like birthday [26]. In a nutshell, users' short-term interests evolve in a more high-frequently and drastically manner compared with the long-term ones.

A typical paradigm of existing personalized product search models is to represent the user intents and items with embedding vectors explicitly, followed by matching them in the latent space with dot product or feeding them to neural layers to yield the prediction score [5]. For example, HEM [2] fuses user vector and query vector to represent user intent with a convex combination. AEM [1] is

the first attention-based embedding model which constructs query-dependent user embeddings by employing attention mechanism to users' historical purchased items with respect to the query. ZAM [1] devises a zero attention mechanism to determine when and how to conduct personalization under various scenarios. TEM [5] dynamically control the influence of personalization by encoding the sequence of query and user's historical behaviors with a transformer architecture. Guo [10] proposes a novel Attentive Long Short-Term Preference model (abbreviated as ALSTP) to learn and integrate the long and short-term user preferences with respect to the current query in a typical personalized product search scenario. Despite its effectiveness, we argue that the existing product search paradigm has several obvious limitations.

First, existing methods always extract users' interest representation by aggregating user historical behaviors with respect to the query submitted by users. We argue that the strategy can work efficiently only when users exhibit their demands with submitted queries explicitly. For example, when a user submits a query "red canvas shoes", it can express the user's requirements on color, type, and material clearly, resulting in the effectiveness of extracting query-dependent interest by straightforwardly employing attention mechanism to user behaviors with respect to the submitted query. However, users sometimes could express their demands with obscure intention. For example, when a user issues a query "gift", which indicates the user does not specify the brand, category, style, price, *etc*. he (she) is desiring at this moment, resulting in the various types of candidate items delivered by previous stage *match*. Therefore, we need to extract user's current precious interest delivered from historical behaviors with respect to individual target item. In other words, an alternative solution is to employ attention mechanism with respect to target product instead of query in such situation to extract target-dependent user interest.

Second, there exists various types of historical behaviors, such as *click*, *favorite*, or *purchase*, which are inadvertently neglected by previous works or just regarded as certain a kind of behavior features integrated into original framework. We argue multiple kinds of behaviors contain abundant information that is worth investigating deeply for superior user interests representation. However, it is not trivial to integrate them into original framework. For example, the purchase behaviors delivered by some users denotes their recent desired demands have been satisfied eventually, which implies a smaller weight on the purchase behavior contributing to the final representation of user interests. On the contrary, some users probably click, or favorite after a certain click behavior on a special category, which denotes the intense of users' interests are continuously increasing, resulting in a higher weight should be given to this kind of behavior for final representation of user interest. In this paper, we regard the phenomenon, *i.e.*, the causality of multiple types of user behaviors, as causal-dependent user interest.

Third, although the state-of-the-art long and short- term interests method, *i.e.*, ALSTP, has achieved a significant performance, it still struggle in achieving better performance since it only considers the query-dependent user interest, neglecting the target-dependent interest and the causal-dependent interest. Therefore, it will lead to inferior performance in some cases. Besides, ALSTP just concatenates the long and short-term interests and entangles both aspects together for final prediction without explicitly distinguishing the importance of long and short-term interests to final prediction, which will lead to inferior accuracy and interpretability.

To address aforementioned challenges, we proposed a novel method named Hierarchical Interests Fusing Network (HIFN) for CTR prediction in product search, which consists of four basic modules namely Short-term Interests Extractor (SIE), Long-term Interests Extractor (LIE), Interests Fusion Module (IFM) and Interests Disentanglement Module (IDM). Specifically, SIE is proposed to extract user's short-term interests by employing three interest encoders, *i.e.*, query-dependent interest encoder (QDIE), target-dependent interest encoder (TDIE) and causal-dependent interest encoder (CDIE) to extract query-dependent, target-dependent and causal-dependent interests, respectively, followed by fusing these three interests in an adaptive manner. In LIE, we capture user long-term interest by devising an attention mechanism with respect to the output representation of SIE module, followed by fusing the long and short-term interest in module IFM. Next, all the representation features are concatenated with original raw context features and fed into fully connected layers to generate the final prediction results. Last but not least, motivated by DLSR [32], IDM injects pseudo labels for both long and short-term interest and employs an self-supervised framework to disentangle long and short-term interests. But different from DLSR, we devise an updating strategy to set long-term interest proxy, which is more suitable and achieves better performance than the settings in DLSR.The contribution of this paper is three-fold:

- We propose a novel method named HIFN for CTR prediction in product search, which facilitates learning long and short-term user interests representations in a hierarchical manner.
- We devise four key modules including SIE, LIE, IFM and IDM to jointly address the aforementioned challenges in the context of product search, which can be implemented efficiently in a multi-task end-to-end learning framework.
- We conduct extensive experiments on real-world offline datasets and online A/B test. Both results demonstrate the effectiveness of HIFN over state-of-the-art methods.

2 PROPOSED METHOD

In this paper, we propose a novel method named HIFN, as depicted in Figure 1. It consists of four basic modules namely Short-term Interest Extractor (SIE), Long-term Interest Extractor (LIE), Interest Fusion Module (IFM) and Interest Disentanglement Module (IDM). We will detail them as follows.

2.1 Problem Definition

Given user profiles u_p, long-term user behaviors u_{b_l}, short-term user behaviors u_{b_s}, target item i, search query q, context feature c, where $u_{b_l} = \{i_1, i_2, ...i_{T_l}\}$, $u_{b_s} = \{i_1, i_2, ...i_{T_s}\}$, and i_t from set u_{b_l} (resp. u_{b_s}) represents user's t-th interacted item from long-term (resp. short-term) behaviors. Besides, T_l (resp. T_s) denotes the length of user's long-term (resp. short-term) behaviors. Next, we firstly encode them into one-hot high-dimensional vectors, followed by transforming these vectors into low-dimensional dense vectors by ways of multiplying corresponding embedding matrices [24, 25, 28]. After concatenating various types of dense vectors, we obtain the embedding vectors e_{u_p}, $x_{u_{b_l}}$, $x_{u_{b_s}}$, e_i, e_q, e_c for u_p, u_{b_l}, u_{b_s}, i, q, c,

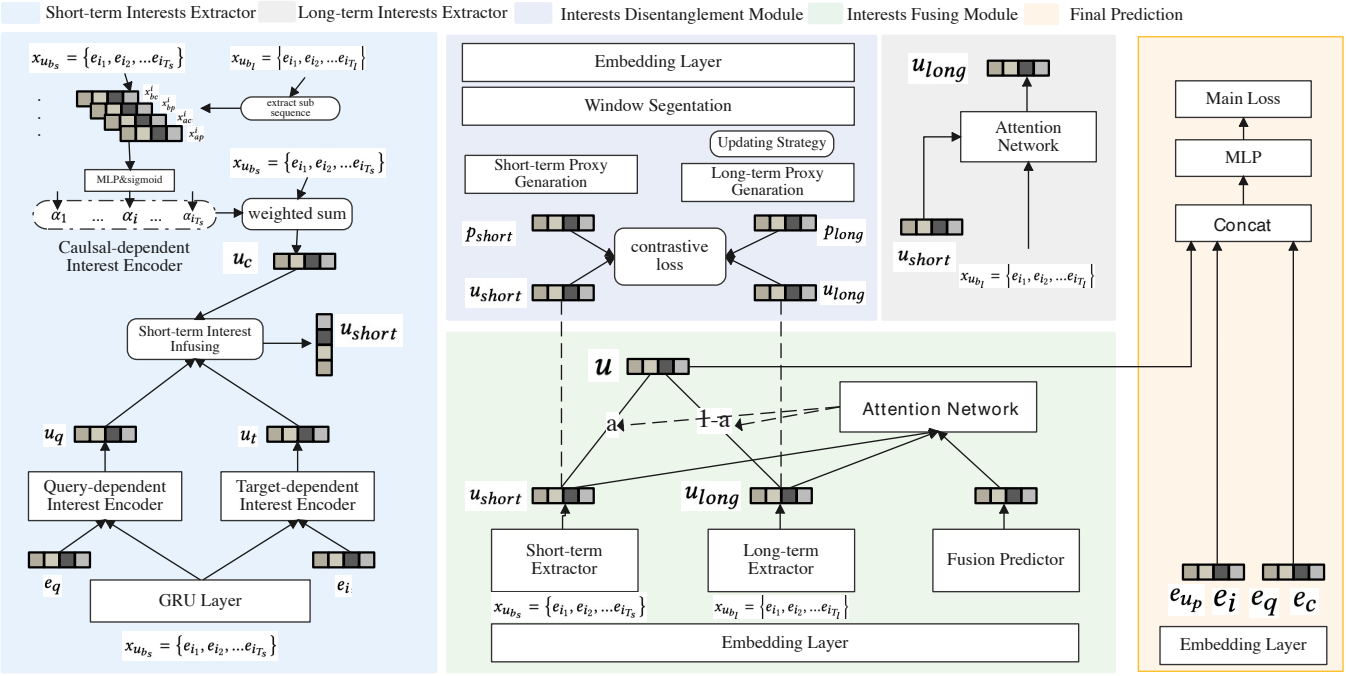

Figure 1: The overview architecture of the proposed HIFN. It consists of several modules, i.e., SIE, LIE, IFM and IDM.

respectively, where $x_{u_{b_s}} = \{e_{i_1}, e_{i_2}, ...e_{i_{T_s}}\}$, $x_{u_{b_l}} = \{e_{i_1}, e_{i_2}, ...e_{i_{T_l}}\}$, and e_{i_t} is the embedding vector of user's t-th interacted item. Our goal is to devise a model to predict the probability of user u clicking target item i given search query c and corresponding relevant features, formally defined as: $\hat{y}_{u,i} = \mathcal{F}(e_{u_p}, x_{u_{b_l}}, x_{u_{b_s}}, e_i, e_q, e_c; \theta)$, where $\mathcal{F}$ is the learning objective with model parameters θ.

2.2 Short-Term Interest Extractor

In general, users' short-term behaviors play a vital role to infer their recent interests. Although existing works towards extracting query-dependent interest in product search have showed significant performance, they still struggle in explicitly delivering users' interests due to the confusion of users' multiple interests in some conditions. Therefore, besides query-dependent interest extraction, an additional supplement is to extract corresponding interest representation from users' behaviors with respect to individual candidate product, called *target-dependent* interest extraction. In addition, users always exhibit various types of behaviors, *i.e.*, *click* or *purchase*, while neglected by previous works when extracting users' interests from their behaviors or just regarded as certain a kind of behavior features. And we argue users' multiple types of behaviors are very essential for users' final interest extraction, called *causal-dependent* interest extraction.

Based on the above considerations, we first adopt a GRU [7] to model the short-term interests, as it has been successfully applied in the session-based recommendation [12]. The RNN module takes the recent s product representations $x_{u_{b_s}} = \{e_{i_1}, e_{i_2}, ...e_{i_{T_s}}\}$ as inputs. We thus can obtain a set of high dimensional hidden representations $x'_{u_{b_s}} = \left\{e'_{i_1}, e'_{i_2}, ...e'_{i_{T_s}}\right\}$, denoting the short-term user preference.

Then, We feed $x'_{u_{b_s}}$ into three well-designed interest encoder named Query-dependent Interest Encoder, Target-dependent Interest Encoder, Causal-dependent Interest Encoder, respectively.

2.2.1 Query-dependent Interest Encoder. Given users express their intention with submitted queries explicitly, we can straightforwardly devise a module named *Query-Dependent Interest Encoder* (QDIE) to accurately extract users' current interests from their behaviors with respect to the submitted query. After considering the varied relevance of historical behaviors with respect to the current submitted queries, we resort to the attention mechanism in QDIE module to extract query-dependent interest, which can be formulated as:

$$u_q = \sum_{i \in x'_{u_{b_s}}} \frac{exp(f(e_q, e_i))}{\sum_{i' \in x'_{u_{b_s}}} exp(f(e_q, e_{i'}))} e_i \tag{1}$$

where $f(q, i)$ is an attention function that defines the weight of each item i with respect to the current query q, defined as:

$$f(q, i) = (i \cdot tanh(W_f \cdot q + b_f)) \cdot W_h \tag{2}$$

where $W_h \in \mathbb{R}^{\beta}$, $W_f \in \mathbb{R}^{\alpha \times \beta \times \alpha}$, $b_f \in \mathbb{R}^{\alpha \times \beta}$.

2.2.2 Target-dependent Interest Encoder. Practically, users sometimes could not clearly express their search intention with submitted queries. Under such circumstances, we employ a module named *Target-Dependent Interest Encoder* (TDIE) to extract users' current interest from their behaviors with respect to individual target item. Take the travel scenario as example, when a user has many online clicking behaviors on the scenic spots from city *Hangzhou* and after he(she) submitted a query named *Hangzhou*, we can probably infer

the user prefers the well-known spot *West Lake* more compared with *snack* in *Hangzhou*. Similar with Query-dependent Interest Encoder, we directly employ an attention mechanism on $x'_{u_{b_s}}$ with respect to target item e_i to achieve target-dependent interest u_t.

2.2.3 Causal-dependent Interest Encoder. Users usually exhibit diverse and continuous interests by ways of various types of behaviors, *i.e.*, *click* or *purchase*. For example, if users are impressed by *clothes*, they will naturally deliver continuous behaviors on *clothes* related products for a period of time until their needs are satisfied, *i.e.*, desired clothes have been bought eventually. Mathematically, let e_i^t in user behaviors $x_{u_{b_s}}$ denote the category of the i-th behavior item at time t. In fact, we can extract four auxiliary sub-sequences from $x_{u_{b_l}}$ for the behavior item at time t according to whether click (or purchase) behaviors before(or after) time t have the same category with e_i^t, *i.e.*, click sub-sequence before time t, x_{bc}^i, purchase sub-sequence before time t, x_{bp}^i, click sub-sequence after time t, x_{ac}^i, purchase sub-sequence after time t, x_{ap}^i. We argue that some users would express intense interests since they continuously exhibit click behaviors with the same category to the one at $t-$th behavior, whether before or after t, especially when users have finished purchase behavior at time t. Meanwhile, some users may have continuous click behaviors before time t with the same category to the one at time t, while they show the decrement of interests after they finished purchase behaviors. Therefore, we employ the four auxiliary sub-sequences to portray users' precious intention at time t, formulated as:

$$\alpha_i = Sigmoid(FFN(Sum(x_{bc}^i)||Sum(x_{bp}^i)||Sum(x_{ac}^i)||Sum(x_{ap}^i))) \tag{3}$$

where $Sum(\cdot)$ indicates sum pooling operator, the reason for using sum pooling is that it can express the density of behavior and achieving great performance, however the computational complexity is lower than others (e.g. GRU, LSTM). Then, causal-dependent Interest is calculated as follows:

$$u_c = \sum_i \alpha_i e_i, \tag{4}$$

2.2.4 Short-term Interest Fusing. As mentioned earlier, the three aspects both play an important role under different user intentions, we employ a gate network to fusing three aspects of short-term interest. The structure of the gate network is based on a single-layer feed-forward network with a SoftMax activation function. It acts as a selector to calculate the weighted sum of the selected vector.

$$u_{short} = SoftMax(FFN(u_t, u_q, u_c))^T [u_t, u_q, u_c], \tag{5}$$

2.3 Long-Term Interests Extractor

Generally, users' behaviors are categorized into two disjoint groups, *i.e.*, long-term behaviors and short-term ones. When users' short-term behaviors are sparse or even absent, the long-term behaviors will be as a kind of supplement cues to discover their interests. Moreover, users' long-term behaviors are regarded as the reflection of their intrinsic tastes while evolving slowly over time, which further affects their short-term behaviors in a relatively recent period. Motivated by this, we propose a Long-Term Interest Extractor (LTIE) module to extract users' long interests. Specially, we employ an attention mechanism to long-term behaviors with respect to the generated short-term interest u_{short} like Eq. (1) and Eq. (2) to deliver users' long-term interests u_{long}.

2.4 Interests Fusing Module

Given users' long- and short-term interests, how to fuse them is not trivial due to the different contributions with respect to individual users. One straightforward solution is employing concatenation operator to integrate them, while it still struggles in how to effectively distinguish the contribution of each interest. For example, basketball fans may continuously click basketball shoes when a query "shoes" submitted due to their long-term interest even they have delivered a lots of canvas shoes related behaviors in a recent period. Alternatively, we devise a module named Interest Fusing Module to seamlessly integrate both interest representation together with submitted query, historical behaviors, target item. Formally, the process can be formulated as :

$$\alpha = sigmoid(FFN(e_i||e_q||u_{short}||u_{long}||\tau_{u_{b_l}})) \tag{6}$$

$$\tau_{u_{b_l}} = GRU(x_{u_{b_l}}) \tag{7}$$

where, α denotes the estimated fusion weight based on aforementioned information. Then, the final user interest representation can be formulated as follows.

$$u = \alpha \cdot u_{short} + (1 - \alpha) \cdot u_{long} \tag{8}$$

Finally, we concatenate u, e_i, e_q and e_c to form a comprehensive vector, followed by feeding it to a MLP layer with *Relu* as activation function to get the final predicted CTR score $\hat{y}$. Finally, we define the objective function as follows:

$$Loss_t = -\frac{1}{N} \sum_{(x,y)\in S} (ylog\hat{y} + (1 - y)log(1 - \hat{y})), \tag{9}$$

where S denotes the training set with total size N, and $y \in \{0, 1\}$ is the ground truth label representing whether click or not.

2.5 Interests Disentanglement Module

Desired by Zheng [32] who introduces a self-supervised framework for disentanglement, we adapt it for product search. We take the mean representation of short-term behaviors as the proxy of short-term interests. As for long-term interests, we argue that directly mean representation of entire long-term behaviors applied by Zheng [32] will lead to sub-optimal performance. Long-term user preference is relatively stable and updates slowly, if we mean representations simply, the temporal variability of long-term interests would be ignored. Since short-term interests are based on the most recent s interacted items, the impact of this drawback is minimal. We apply an updating mechanism to the extraction of long-term interests proxy to capture the slowly evolving characteristic of long-term interests. For simplicity, we use the first Ts products to set it up, and then update it with every Ts product interacted (evolving gradually). In this setting, p is updated based on a session of Ts

products, which is also the window size of modeling the short-term user preference. It is illustrated as follows:

$$p_{long} = (1-\beta)p_{long} + \beta FFN(h_{t-1}) \tag{10}$$

where, FFN is multiple layers of feed forward network, h_{t-1} is the mean representation of previous Ts interacted items. As for short-term proxy, it is calculated as follows:

$$p_{short} = MeanPooling\left\{e_{i_1}, e_{i_2}, ...e_{i_{T_s}}\right\} \tag{11}$$

With proxies serving as labels, we can utilize them to supervise the disentanglement of long and short-term interests. Specifically, we perform contrastive learning [16] between the outputs of interest extractors and proxies,which requires the learned representations of long and short-term interests to be more similar to their corresponding proxies than the opposite proxies. Formally, there are four contrastive tasks as follows,

$$sim(u_{long}, p_{long}) > sim(u_{long}, p_{short}) \tag{12}$$

$$sim(p_{long}, u_{long}) > sim(p_{long}, u_{short}) \tag{13}$$

$$sim(u_{short}, p_{short}) > sim(u_{short}, p_{long}) \tag{14}$$

$$sim(p_{short}, u_{short}) > sim(p_{short}, u_{long}) \tag{15}$$

where Eqn (12)-(13) supervise long-term interests, and Eqn (14)-(15) supervise short-term interests, and $sim(\cdot,\cdot)$ measures embedding similarity. we add self-supervision on long and short-term interests modeling which can achieve stronger disentanglement compared with existing unsupervised approaches. We implement a pairwise loss functions based on Bayesian Personalized Ranking (BPR) to accomplish contrastive learning, are computed as follows:

$$\pounds_{bpr}(a,p,q) = softplus(< a,q > - < a,p >) \tag{16}$$

$$\pounds_{con} = \pounds_{bpr}(u_{long}, p_{long}, p_{short}) + \pounds_{bpr}(p_{long}, u_{long}, u_{short}) + \pounds_{bpr}(u_{short}, p_{short}, p_{long}) + \pounds_{bpr}(p_{short}, u_{short}, u_{long}). \tag{17}$$

We train the model in an end-to-end manner with multi-task learning on two objectives. Specifically, the joint loss function with a hyper-parameter λ to balance objectives, formulated as follows:

$$\pounds = Loss_t + \lambda \pounds_{con} \tag{18}$$

3 EXPERIMENT

To verify the effectiveness of HIFN, we conduct extensive experiments to compare it with representative SOTA methods on both offline datasets and online deployment.

3.1 Dataset

We use two datasets to conduct experiments, including a public dataset and an industrial dataset namely Amazon and Fliggy, respectively, which are both with million users scale and collected from real-world applications. The details of the adopted datasets are introduced as follows. Statistics of datasets are listed in Table 1.

Fliggy[1] The Fliggy dataset which contains properties of users, queries, user historically behaviors (including click, favorite, and purchase), and target product, is generated based on user logs collected from October 1th to October 31th, 2021. Note that users with fewer than 20 historical interactions are removed. User logs of the first 30 days in October 2021 is used as the training data, while reserve the last day in October for validation (before 12 pm) and test (after 12 pm). Negative samples in the training dataset are set to those impressed products but not clicked by users, and posivtie samples are clicked.

Table 1: Statistics of the offline dataset. (QL indicates Query Length, PL indicated Product Length.)

Dataset	Clothing, shoes&jewelry	Electronics	Toys&Gift	Fliggy
Users	36421	34489	76234	2029216
Products	22521	18397	63323	920487
Queries	1998	1499	645	82409
Avg.QL	8.23	8.74	6.21	13.41
Avg.PL	75.36	95.36	196.3	53.67
#Query-User Pairs				
Train	456521	273890	1493132	26379808
Valid/Test	4006/4151	753/890	10210/10131	83129/89936

Amazon[2] As there are no large-scale public datasets in the personalized product search area, we use Amazon dataset as our experiment corpus, consistent with existing approaches [1, 10]. In our experiments, we adopted the five-core version provided, whereby the remaining users and products have at least five reviews, respectively. Besides, we selected three categories with different sizes: Clothing, Shoes & Jewelry, Toys & Games, Electronics. These datasets both contain several categories so that users may have different interests. Following the strategy in References [11], we extracted the users' product purchasing behaviors based on their reviews, i.e., the products they reviewed are the ones they purchased. Our model uses the previously purchased products in a neighboring window size to model the short-term user interests. We further filtered the dataset to make sure each user has at least 20 purchased products (i.e., 20 reviews).

Following the same paradigm used in [10], we construct queries for each purchased item with category information. This strategy is based on the finding that directed product search is users' search for a producer's name, a brand, or a set of terms describing product category. We partitioned each of the four datasets into three sets: training, validation and testing sets. We first extracted the user-product pairs from users' reviews, and then extracted the queries for these products, getting triplets. For each dataset, the last purchasing transaction of each user is held for the testing set, the second last for the validation set, and the rest for the training set. Moreover, we hid the reviews of the validation and testing sets in the training phase to simulate the real-world scenarios.

Since Amazon datasets do not provide negative samples, we adopt a negative sampling method to evaluate it. In detail, for each positive user-query-item triplet, we generate negative products with the same category as query randomly to simulate a real search

[1]One of the most popular online travel platforms in china, www.fliggy.com

[2]http://jmcauley.ucsd.edu/data/amazon

scenario. Then, the generated products, query and user form negative triplet samples. We generate ten negative samples for each positive triplet.

3.2 Competitors

To verify the effectiveness of the proposed method, we compare it with following methods:

- **AEM** [1]: The Attention-based Embedding Model (AEM) constructs query-dependent user embeddings by attending to users' historical purchased items with the query.
- **ZAM** [1]: The Zero Attention Model (ZAM) extends AEM with a zero vector and conducts differentiated personalization by allowing the query to attend to the zero vector.
- **TEM** [5]: The Transformer-based Embedding Model (TEM) [6] is a state-of-the-art model that encodes query and historically purchased items with a transformer and does item generation based on the encoded query-user information.
- **DeepFM** [9]: It combines the explicit high-order interaction module with deep MLP module and traditional FM module, and requires no manual feature engineering.
- **DIN** [34]: It utilizes attention mechanism to activate relevant users' behaviors with respect to corresponding targets and learns an adaptive representation vector for users' interests.
- **DIEN** [33]: 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.
- **ZAM+TDIE** : Compared with ZAM, we add a Target-dependent Interest Encoder (TDIE) to model the target-dependent user interest. The target-dependent user interest is concatenated with user representation to form a new user interest vector.
- **TEM+TDIE** : Do the same operation to TEM as ZAM+TDIE.
- **ALSTP** [10]: It represent long and short-term user preferences with an attention mechanism applied to users' recent interacted products and their global vectors. It is the sota long and short-term modeling method for personalized product search.

3.3 Metrics

In the experiments, we use AUC and logloss (or cross-entropy) [21, 22] to evaluate the ranking ability of all comparison models on the Fliggy dataset. The larger AUC and smaller Logloss mean better ranking performance. On public datasets, namely Amazon, the NDCG@10, MRR and MAP are used [10], since these datasets do not provide negative samples and the computation of AUC requires both positive and negative samples in a dataset.

3.4 Implement details

We use the Adam optimizer. Embedding size d is set as 16. Batch normalization is enabled for the MLP, and the activation function is ReLU. We use grid-search to find the best hyper-parameters. The optimal settings for our proposed implementation are: Batchsize is 512. Learning rate is 0.001. Feed forward networks (FFN) in IDM is set [128, 64, 32, 16], and FFN in short term interest fusing is set [128, 64, 32, 3]. λ is 0.1 in both datasets. Ts is 10 for Fliggy, 8 for Clothing,shoes&jewelry, 14 for Electronics, 12 for Toys&gift. β is 0.55 for Fliggy, 0.35 for Clothing,shoes&jewelry, 0.45 for Electronics, 0.6 for Toys&gift. Value of each experimental result is the average of 10 repeated tests.

3.5 Main Comparison Results

We illustrate the overall performance on two adopted datasets in Table 2. From the results, we have the following observations:

- **It is important to extract the target-dependent user interest in product search.** AEM, ZAM, and TEM all model the correlation between query and user behaviors to represent user interests. However, the user interests are not always focused. For example, when a user issues a query "clothing", he does not express preferences for colors, brands, and styles. Users will click navy blue clothings instead of other colors in the result page, because the navy blue color is the most common among the products purchased by users in the history. If we only model the relationship between query and historical behaviors to extract user interest vectors, we will ignore this important point. It can be seen from the table that the methods of modeling query and user historical behavior such as AEM, ZAM, TEM and the methods of modeling target item and user historical behavior such as DIN and DIEN have different performances in different datasets. However, after adding the module for modeling the relationship between target item and user behavior sequence (TDIE), the performance is significantly improved, which shows the importance of modeling user interests from two perspectives at the same time.
- **Joint modeling of long and short-term interests does not always bring performance gains for product search.** Although ALSTP is the SOTA long and short-term interests method in product search. However, there are many issues have not been addressed carefully. First, short-term interests play an important role for final prediction, but ALSTP only considers query-dependent interest, neglecting the target-dependent and causal-dependent interests. Second, simply concatenating long and short-term interests neglect the fact that long and short-term interests have varying degrees of importance in different situations. Third, modeling long and short-term interests entangled with each other will increase model redundancy and leads to inferior accuracy. Results demonstrate that ALSTP is not consistently effective across different metrics and datasets. For example, ALSTP is the best baseline on Clothing dataset, but its NDCG performance is poorer than ZAM by about 5.35% on Electronics. It indicates that without carefully address the above issues, modeling long and short-term interests can not get consistent improvements.
- **HIFN can achieve significant improvements over all datasets and metrics.** HIFN outperforms all competitors with significant progress. Specifically, HIFN improves NDCG by about 0.07 on Clothing dataset and improves AUC by about 0.017 on Fliggy dataset, agaisnt SOTA methods. The consistent and significant progress indicate that HIFN models long and short-term user interests in a more proper manner than previous works.

3.6 Study on Short-term Interests Extractor

In this part, we will discuss how each module of short-term interests extractor impact the effectiveness of HIFN. The base model is HIFN*, which using a mean pooling operator (No Attention) to aggregate the output of GRU layer. In particular, we consider the following

Table 2: Overall performance on Fliggy and Amazon datasets.

Dataset	Clothing,shoes&jewelry			Electronics			Toys&Games			Fliggy	
Model	MRR	NDCG	MAP	MRR	NDCG	MAP	MRR	NDCG	MAP	AUC	Logloss
AEM	0.024	0.025	0.019	0.269	0.291	0.265	0.068	0.072	0.062	0.735	0.598
ZAM	0.025	0.027	0.021	0.289	0.315	0.288	0.079	0.112	0.088	0.734	0.601
TEM	0.027	0.026	0.023	0.201	0.261	0.194	0.138	0.158	0.129	0.731	0.621
DeepFM	0.020	0.028	0.016	0.222	0.296	0.286	0.073	0.104	0.073	0.737	0.588
DIN	0.022	0.029	0.018	0.223	0.298	0.289	0.075	0.105	0.076	0.739	0.581
DIEN	0.023	0.034	0.022	0.288	0.319	0.301	0.099	0.127	0.118	0.741	0.579
ZAM+TDIE	0.031	0.036	0.026	0.311	0.318	0.322	0.106	0.144	0.123	0.746	0.552
TEM+TDIE	0.032	0.034	0.028	0.315	0.289	0.291	0.141	0.162	0.134	0.749	0.541
ALSTP	0.034	0.037	0.029	0.302	0.299	0.294	0.101	0.134	0.120	0.742	0.576
HIFN	**0.041**	**0.044**	**0.035**	**0.331**	**0.342**	**0.351**	**0.162**	**0.171**	**0.143**	**0.759**	**0.533**

Table 3: Study on Short-term Interests Extractor.

Dataset	Clothing,shoes&jewelry			Fliggy	
Model	MRR	NDCG	MAP	AUC	Logloss
HIFN*	0.033	0.032	0.023	0.739	0.581
HIFN*+QDIE	0.037	0.038	0.028	0.746	0.553
HIFN*+TDIE	0.036	0.037	0.026	0.745	0.558
HIFN*+CDIE	0.033	0.032	0.024	0.744	0.561
HIFN w/o CDIE	0.039	0.040	0.032	0.753	0.541
HIFN w/o TDIE	0.036	0.039	0.031	0.751	0.549
HIFN w/o QDIE	0.037	0.038	0.028	0.749	0.552
HIFN w/o gate	0.034	0.035	0.024	0.743	0.563
HIFN	**0.041**	**0.044**	**0.035**	**0.759**	**0.533**

settings: 1) add QDIE, 2) add TIDE, 3) add CDIE, 4) add QDIE and TIDE, i.e.,without CDIE, 5) add QDIE and CDIE, i.e., without TIDE, 6) add TIDE and CDIE, i.e, without QDIE. 7) replacing the fusing gate with mean pooling operator.

3.6.1 Ablation Study of each module. In this section, we investigate the impact of Short-term Interests Encoder. As illustrated in Table 3 , base model performs the worst, showing that useful signals could be easily buried in noise without distilling. In addition, either QDIE, TDIE or CDIE can improve the AUC compared with the base model and the former has the most gain, while the latter has the least gain, demonstrating that capturing query-, target- and causal-dependent user interest all can bring gains and query-dependent user interest helps the most in prediction accuracy, which is also consistent with the characteristics of the product search. (Note that, adding CDIE module just have a weak improvement in Clothing but strong relatively in Fliggy. This is because Amazon only contains user purchase bahavior, while FLiggy contaions purchase and click, and CDIE is proposed to extract information from user's multiple historical behavior types.) 4), 5) and 6) all achieve better performance than previous method, implying that the information extracted by these three interest extraction modules is complementary. In contrast, HIFN learns user short-term interest from three different perspective and fusing the output vector of them with a well-designed gate network, obtaining the highest gains. It shows that capturing query-, target- and causal-dependent user interest can improve the accuracy of prediction.

Table 4: Study on SIE with different query types.

Query Type	Abstract	Precise
Model	AUC	AUC
HIFN	**0.748**	**0.776**
HIFN w/o QDIE	0.745	0.759
HIFN w/o TDIE	0.739	0.772
HIFN w/o CDIE	0.742	0.768

3.6.2 Study on datasets of different query types. In this section, we investigate the impact of different interest extractors in datasets of different query types. We first divide the Fliggy test set into two sets according to the type of query, one is the abstract search query, and the other is the precise (i.e. non-abstract) search query. We define the abstract search query as 1) destination,e.g., Hangzhou, Shanghai, Miami; 2) tag, e.g., playing, surfing. Different from precise search, users do not express obvious demand mentality when issuing the abstract query. As illustrated in Table 4, there are several points worth noting. On the one hand, HIFN w/o QDIE performs worst than the other two in precise query dataset, it implies that when user has clear requirements , extracting user interest with respect to query can obtain more gains. On the other hand, HIFN w/o TDIE performs worst than the other two in abstract query dataset, implying that when user has no clear intent, taget-dependent interest plays the most important role, and extracting user interest with respect to target item is an effective way to mining the intent from the multi-interest of users. It further validates that jointly modeling query-dependent, target-dependent, causal-dependent user interest can achieve significant performance.

3.7 Study on Long-term Interests Extractor

In this section, we investigate the impact of the Long-term Interests Extractor in HIFN, the base model is HIFN†, which extracts user behaviors using mean pooling operator (No Attention). In particular, we consider the following settings: 1) Target Attention: taking the target item as the query and the behavior item as the key in the attention mechanism; 2) Query Attention: the same long-term attention mechanism with the SOTA long and short-term methods in product search, ALSTP; 3) User Attention: take the user field

Table 5: Study on Long-term Interests Extractor.

Dataset	Clothing,shoes&jewelry			Fliggy	
Model	MRR	NDCG	MAP	AUC	Logloss
HIFN†	0.035	0.036	0.029	0.745	0.554
HIFN†+Target Attention	0.036	0.037	0.03	0.746	0.553
HIFN†+Query Attention	0.039	0.042	0.032	0.754	0.558
HIFN†+User Attention	/	/	/	0.752	0.564
HIFN	**0.041**	**0.044**	**0.035**	**0.759**	**0.533**

Table 6: Study on Interests Fusing Module.

Dataset	Clothing,shoes&jewelry			Fliggy	
Model	MRR	NDCG	MAP	AUC	Logloss
concatenate	0.035	0.036	0.029	0.750	0.539
α=0.9	0.037	0.036	0.028	0.747	0.544
α=0.75	0.036	0.034	0.025	0.749	0.541
α=0.6	0.034	0.033	0.022	0.751	0.537
α=0.45	0.035	0.034	0.024	0.753	0.535
α=0.3	0.036	0.035	0.026	0.747	0.543
α=0.15	0.032	0.031	0.021	0.748	0.545
HIFN	**0.041**	**0.044**	**0.035**	**0.759**	**0.533**

feature as the query and the behavior item as the key in the attention mechanism.

As can be seen from table 5, each attention mechanism can beat base model, implying that directly meaning pooling user long-term behaviors could not fully attend and exploit the useful information. The query attention mechanism used in ALSTP and user attention mechanism both get better performance than target attention, we suppose that Long-term interests of users are stable and change slowly, while user field and query field features represent user's intrinsic characteristic, which is consistent with the user's long-term interest to some extent. (Since there is no user profile feature in Amazon dataset, we do experiments about user attention only on Fliggy dataset). HIFN, leveraging short-term interests to activate long-term interests, achieves the best performance, we suspect that this is because short-term interests are a unified vector fusing user profile, query and recent behaviors, which can extract users' long-term interests more properly.

3.8 Study on Disentanglement of Long and Short-Term Interests

In order to prove the effectiveness of the disentanglement framework, we designed two kinds of experiments. Firstly, we will verify whether the framework can disentangle short-term and long-term interests well. Secondly, we will verify the impact of different proxy settings on the results.

3.8.1 Counterfactual Evaluation. If the importance of different factors change over time, learning disentangling representations of underlying factors is beneficial to final accuracy [20]. In e-commerce product search, users will have various types of behaviors, such as purchase, favorite, click, the user preferences revealed by each type of behavior range from strong to weak, i.e purchase expresses strong preferences, while click expresses relatively weak preferences. Grbovic [8] acknowledged that behaviors of higher costs tend to be more driven by users' long-term interests, and behaviors of lower costs such as click indicate more about short-term interests. As we known, purchase (cost of money) have higher costs than favorite (cost of double clicks), while favorite have higher costs than click (only one click). Therefore, a set of experiments were designed to verify the effectiveness of the disentanglement. Because Amazon dataset only contains user purchase behavior, we verify the effect on Fliggy dataset. We firstly train a model with exposure-click dataset, and predicting on both clicks, favorites and purchases test dataset of Fliggy. Intuitively, it is expected that the weight for short-term interests is smaller in predicting purchase than predicting click when fusing the two aspects. We compare our proposed model HIFN with HIFN without self-supervised framework, and the SOTA long and short-term methods for product search, ALSTP.

Table 7 illustrates the AUC, Logloss and the average α for clicked items, favorited items and purchased items. We have the following findings. First, HIFN w/o IDM achieves a better performance than ALSTP, demonstrating that HIFN extracts and fuses long and short-term interests in a more proper and effective way than ALSTP. Second, although predicting purchase/favorite with model trained on click data is challenging, HIFN also gets a good results and beats other competitors in favorite test set and purchase test set. Third, the average α of HIFN in click is larger than favorite, and favorite is larger than purchase, which is consistent with the previous argument, that behaviors of higher costs tend to be more driven by users' long-term interest, and behaviors of lower costs such as click indicate more about short-term interest. However the average α of HIFN w/o IDM in favorite is samller than purchase and the average α of ALSTP in click is samller than purchase/favorite. It demonstrates that HIFN successfully disentangle the long and short-term interest to a certain degree, while HIFN w/o IDM and ALSTP entangles two aspects with each other.

3.8.2 The effectiveness of long-term interests proxy updating. Although Zheng [32] proposes a simple strategy for proxy setting to disentangling user interests, we argue that this strategy ignores the nature that long-term interests will evolve slowly, so we propose an updating strategy for long-term interests proxy to address the issue. In this section, we study the effectiveness of the long-term interests proxy updating strategy on the four datasets with respect to different metrics. As shown in table 8, the performance with the long-term user preference updating strategy surpasses the one without updating. In detail, updating strategy improves NDCG by about 0.003, 0.002, and 0.003 in three Amazon datasets respectively, and improves AUC by about 0.007 in Fliggy dataset. This is in accordance with our assumption that the long-term user preference should update gradually.

3.9 Study on Interests Fusing Module

Althogh ALSTP achieves the significant performance, we argue that concatenating long and short-term interests will lead to inferior performance. In HIFN, we propose to aggregate them adaptively. We compare HIFN with two different settings: 1) concatenation, 2) using a fixed α when combining the two aspects. The result is illustrated in Table 6 , we can discover that HIFN with IFM outperforms all

Table 7: Study on Interest Disentanglement Module.

Dataset	Click			Favorite			Purchase		
Model	AUC	Logloss	AVG (α)	AUC	Logloss	AVG (α)	AUC	Logloss	AVG (α)
HIFN	**0.759**	**0.533**	0.783	**0.746**	**0.569**	0.721	**0.741**	**0.573**	0.699
HIFN w/o IDM	0.753	0.549	0.637	0.741	0.578	0.526	0.737	0.586	0.672
ALSTP	0.742	0.576	0.433	0.733	0.591	0.621	0.729	0.598	0.596

Table 8: Study on Different Proxy Settings.

Dataset	Clothing			Electronics			Toys&Games			Fliggy	
Strategy	MRR	NDCG	MAP	MRR	NDCG	MAP	MRR	NDCG	MAP	AUC	Logloss
without updating	0.038	0.041	0.032	0.329	0.340	0.347	0.159	0.168	0.137	0.752	0.542
with updating	**0.041**	**0.044**	**0.035**	**0.331**	**0.342**	**0.351**	**0.162**	**0.171**	**0.143**	**0.759**	**0.533**

different settings. These results verify the necessity of adaptive fusion of long and short-term interests.

3.10 Online A/B Test

We also deploy HIFN on our platform for A/B test, where baseline model is TEM+TDIE (Previously, TEM+TDIE had demonstrated online that its results were superior to other competitors in Section 3.5). The proposed method achieves average **5.13%** overall CTR gains over the baseline model in successive fifteen days, which is consistent with the offline evaluation results. Now, HIFN has been deployed online and serves the main traffic.

4 RELATED WORK

4.1 Product Search

Aware of the importance of personalization in product search, Ai et al. [2] proposed a hierarchical embedding model where they use a convex combination of the query and user vector to predict purchased items. Guo et al. [10] represent long and short-term user preferences with an attention mechanism applied to users' recent purchases and their global vectors. Recently, from the analysis of commercial search logs, Ai et al. [1] observed that personalization does not always have a positive effect. They further proposed a zero-attention model (ZAM) that can control the influence of personalization. However, the maximal effect personalization can have is equal to the query. Bi et al. [5] found this limitation and proposed a transformer model to encode the query and historically purchased items where personalization can have none to full effect. These models construct user profiles by capturing dynamic user interests with respect to query. However, they ignore the target-dependent interest and causal-dependent interest, which will bring additional information for more accurate prediction.

4.2 CTR Prediction

Existing CTR prediction works mainly focus on single scenario modeling from the following aspects: 1) feature interaction (e.g., FM [18], DeepFM [9]); 2) user historical behavior (e.g., DIN[34], DIEN [33]); Factorization Machine (FM) is proposed to model feature interactions explicitly, while previous generalized linear models such as Logistic Regression (LR) [19] and Follow-The-Regularized-Leader (FTRL) [17] lack the ability to solve interaction issue. Wide&Deep [6] and DeepFM [9] combine low-order and high-order features to improve the performance. FmFM [23] makes each field feature have different embedding dimensions, so as to reduce the amount of model parameters and avoid over fitting problem. DIN [34] utilizes the attention mechanism to capture relative interests from the user behavior sequence with regard to the candidate item. DIEN [33] further uses a GRU structure to capture the evolution of user interest. Considering a single vector might be insufficient to capture complicated user patterns, DMIN [27] models user's multiple interests by a special designed extractor layer.

4.3 Long and Short-term Interests Modeling

Long and short-term interests modeling has been widely studied in recommendation, several methods [3, 13, 31] were proposed to explicitly differentiate between long and short-term interests. For example, Zhao [31] use matrix factorization for long-term interests and use RNN for short-term interests. Yu [29] develop a variant of LSTM for short-term interests and adopt asymmetric SVD for long-term interests. However, in product search, modeling long and short-interests become difficult because of the query user issued. ALSPT, which is the sota long and short-term interests modeling method in product search, represents long and short-term preferences with attention mechanisms applied to users' recent behaviors and their global vectors, achieves a significant performance, but it still contains weakness about interests modeling and fusing.

5 CONCLUSION

In this paper, we introduce a novel CTR model named HIFN for product search, which consists of four elaborately designed modules including SIE, LIE, IFM and IDM . SIE aims to extract short-term interests from different aspects. LIE is proposed to extract long-term interests with respect to the output of SIE. Then IFM fuses the long and short-term interests with an adaptive manner. IDM is used to disentangle the two aspects of interests. These modules are integrated together in an end-to-end multi-task learning framework. Extensive offline and online experiments on real-world e-commerce platforms demonstrate the superiority of HIFN over SOTA methods.

REFERENCES

[1] Qingyao Ai, Daniel N Hill, SVN Vishwanathan, and W Bruce Croft. 2019. A zero attention model for personalized product search. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 379–388.

[2] Qingyao Ai, Yongfeng Zhang, Keping Bi, Xu Chen, and W Bruce Croft. 2017. Learning a hierarchical embedding model for personalized product search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 645–654.

[3] Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. 2019. Neural news recommendation with long-and short-term user representations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 336–345.

[4] Paul N Bennett, Ryen W White, Wei Chu, Susan T Dumais, Peter Bailey, Fedor Borisyuk, and Xiaoyuan Cui. 2012. Modeling the impact of short-and long-term behavior on search personalization. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. 185–194.

[5] Keping Bi, Qingyao Ai, and W Bruce Croft. 2020. A transformer-based embedding model for personalized product search. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1521–1524.

[6] 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.

[7] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* (2014).

[8] Mihajlo Grbovic and Haibin Cheng. 2018. Real-time personalization using embeddings for search ranking at airbnb. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 311–320.

[9] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 1725–1731.

[10] Yangyang Guo, Zhiyong Cheng, Liqiang Nie, Yinglong Wang, Jun Ma, and Mohan Kankanhalli. 2019. Attentive long short-term preference modeling for personalized product search. *ACM Transactions on Information Systems (TOIS)* 37, 2 (2019), 1–27.

[11] 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.

[12] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[13] Linmei Hu, Chen Li, Chuan Shi, Cheng Yang, and Chao Shao. 2020. Graph neural news recommendation with long-term and short-term interest modeling. *Information Processing & Management* 57, 2 (2020), 102142.

[14] Sen Li, Fuyu Lv, Taiwei Jin, Guli Lin, Keping Yang, Xiaoyi Zeng, Xiao-Ming Wu, and Qianli Ma. 2021. Embedding-based Product Retrieval in Taobao Search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3181–3189.

[15] Zhao Li, Xin Shen, Yuhang Jiao, Xuming Pan, Pengcheng Zou, Xianling Meng, Chengwei Yao, and Jiajun Bu. 2020. Hierarchical bipartite graph neural networks: Towards large-scale e-commerce applications. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 1677–1688.

[16] Yixin Liu, Zhao Li, Shirui Pan, Chen Gong, Chuan Zhou, and George Karypis. 2021. Anomaly detection on attributed networks via contrastive self-supervised learning. *IEEE transactions on neural networks and learning systems* 33, 6 (2021), 2378–2392.

[17] H Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, et al. 2013. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1222–1230.

[18] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International Conference on Data Mining*. IEEE, 995–1000.

[19] 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.

[20] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. 2021. Toward causal representation learning. *Proc. IEEE* 109, 5 (2021), 612–634.

[21] 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. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4094–4103.

[22] 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.

[23] Yang Sun, Junwei Pan, Alex Zhang, and Aaron Flores. 2021. FMˆ2: Field-matrixed Factorization Machines for Recommender Systems. *arXiv preprint arXiv:2102.12994* (2021).

[24] Wanjie Tao, Liangyue Li, Chen Chen, Zulong Chen, and Hong Wen. 2022. When Online Meets Offline: Exploring Periodicity for Travel Destination Prediction. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2366–2370.

[25] Wanjie Tao, Yu Li, Liangyue Li, Zulong Chen, Hong Wen, Peilin Chen, Tingting Liang, and Quan Lu. 2022. SMINet: State-Aware Multi-Aspect Interests Representation Network for Cold-Start Users Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 8476–8484.

[26] Liang Xiang, Quan Yuan, Shiwan Zhao, Li Chen, Xiatian Zhang, Qing Yang, and Jimeng Sun. 2010. Temporal recommendation on graphs via long-and short-term preference fusion. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. 723–732.

[27] Zhibo Xiao, Luwei Yang, Wen Jiang, Yi Wei, Yi Hu, and Hao Wang. 2020. Deep Multi-Interest Network for Click-through Rate Prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2265–2268.

[28] Jia Xu, Jin Huang, Zulong Chen, Yang Li, Wanjie Tao, and Chuanfei Xu. 2022. ODNET: A Novel Personalized Origin-Destination Ranking Network for Flight Recommendation. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 3481–3493.

[29] Zeping Yu, Jianxun Lian, Ahmad Mahmoody, Gongshen Liu, and Xing Xie. 2019. Adaptive User Modeling with Long and Short-Term Preferences for Personalized Recommendation.. In *IJCAI*. 4213–4219.

[30] Jing Zhang and Dacheng Tao. 2020. Empowering things with intelligence: a survey of the progress, challenges, and opportunities in artificial intelligence of things. *IEEE Internet of Things Journal* 8, 10 (2020), 7789–7817.

[31] Wei Zhao, Benyou Wang, Jianbo Ye, Yongqiang Gao, Min Yang, and Xiaojun Chen. 2018. PLASTIC: Prioritize Long and Short-term Information in Top-n Recommendation using Adversarial Training.. In *Ijcai*. 3676–3682.

[32] 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.

[33] 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*.

[34] 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.