Denoising User-aware Memory Network for Recommendation

Zhi Bian*
bianzhi.bz@alibaba-inc.com
Alibaba Group
Beijing, China

Shaojun Zhou*
zsj148798@alibaba-inc.com
Alibaba Group
Beijing, China

Hao Fu
hfu@mail.ustc.edu.cn
Alibaba Group
Beijing, China

Qihong Yang
xiaokui.yqh@alibaba-inc.com
Alibaba Group
Beijing, China

Zhenqi Sun
sunzhenqi.szq@alibaba-inc.com
Alibaba Group
Beijing, China

Junjie Tang†
lixi.tjj@alibaba-inc.com
Alibaba Group
Beijing, China

Guiquan Liu
gqliu@ustc.edu.cn
University of Science and Technology of China
Anhui, China

Kaikui Liu
damon@alibaba-inc.com
Alibaba Group
Beijing, China

Xiaolong Li
xl.li@antfin.com
Alibaba Group
Beijing, China

ABSTRACT

For better user satisfaction and business effectiveness, more and more attention has been paid to the sequence-based recommendation system, which is used to infer the evolution of users' dynamic preferences, and recent studies have noticed that the evolution of users' preferences can be better understood from the implicit and explicit feedback sequences. However, most of the existing recommendation techniques do not consider the noise contained in implicit feedback, which will lead to the biased representation of user interest and a suboptimal recommendation performance. Meanwhile, the existing methods utilize item sequence for capturing the evolution of user interest. The performance of these methods is limited by the length of the sequence, and can not effectively model the long-term interest in a long period of time. Based on this observation, we propose a novel CTR model named denoising user-aware memory network (DUMN). Specifically, the framework: (i) proposes a feature purification module based on orthogonal mapping, which use the representation of explicit feedback to purify the representation of implicit feedback, and effectively denoise the implicit feedback; (ii) designs a user memory network to model the long-term interests in a fine-grained way by improving the memory network, which is ignored by the existing methods; and (iii) develops a preference-aware interactive representation component to fuse the long-term and short-term interests of users based on gating to understand the evolution of unbiased preferences of users. Extensive experiments on two real e-commerce user behavior datasets show that DUMN has a significant improvement over the state-of-the-art baselines.

CCS CONCEPTS

• **Information systems → Recommender systems**; **Personalization**; • **Computing methodologies → Learning latent representations**.

KEYWORDS

recommender systems, neural networks, denoising

ACM Reference Format:
Zhi Bian, Shaojun Zhou, Hao Fu, Qihong Yang, Zhenqi Sun, Junjie Tang, Guiquan Liu, Kaikui Liu, and Xiaolong Li. 2021. Denoising User-aware Memory Network for Recommendation. In *Fifteenth ACM Conference on Recommender Systems (RecSys '21), September 27-October 1, 2021, Amsterdam, Netherlands.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3460231.3474237

1 INTRODUCTION

Large-scale e-commerce platforms such as Taobao and Amazon have hundreds of millions of users' interaction data every day. Click-Through Rate (CTR) prediction plays an important role in the personalized recommendation system [13, 29, 49, 50]. It can recommend items consistent with users' interests and preferences by analyzing users' historical behavior data, thus greatly improving users' satisfaction and reducing information overload.

The key to CTR prediction is to understand the evolution of user preferences through historical behavior data. Traditional CTR prediction methods such as matrix factorization (MF) [21] and collaborative filtering (CF) [8] try to learn low-order cross features and capture the similarity between users and items through the user-item rating matrix constructed from user feedback data. With the rapid development of deep learning, CTR prediction method based on deep learning such as DeepFM [11] and Deep&Cross [41] and xDeepFM [25] can effectively capture the high-order cross features of users and items, and capture the evolution of users' interests through LSTM/GRU network modeling click sequence

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

RecSys '21, September 27-October 1, 2021, Amsterdam, Netherlands

ACM ISBN 978-1-4503-8458-2/21/09…$15.00
https://doi.org/10.1145/3460231.3474237

[18, 22, 38]. Methods such as DIEN [49] and DSIN [6], regard the user's click feedback data as sequence signals, and use RNN-based networks to summarize the user's preference. Recently, researchers point out that the modeling of click sequence can only focus on what users are interested in, but ignore the modeling of what users are not interested in, which leads to the captured user preferences are biased [44]. In view of this, the interaction data is subdivided into implicit and explicit feedback [10, 44]. Among them, explicit feedback is defined as precise but relatively rare feedback that can directly indicate users' positive/negative preferences in the view page, such as rating and tagging like/dislike, while implicit feedback refers to rich but noisy feedback that contains noise and cannot directly indicate the user's preferences. Implicit feedback includes click and unclick. Click feedback indicates that the user clicks an item on the view page, while unclick feedback indicates that the user slides down but does not click. In general, click feedback may come from users accidentally clicking some wrong items; unclick feedback also includes items that the user may be interested in, but it scrolls too fast to notice. [26] and [12] model users' explicit negative feedback information through collaborative filtering (CF) and multi-task learning, which improves the performance of the model to a certain extent. [28], [44] and [47] consider modeling the unclicked sequence in the user's implicit negative feedback information.

Despite these methods have achieved significant performance by modeling both implicit and explicit feedback to understand the evolution of users' unbiased preferences, we argue that the inherent noises of implicit feedback are not dealt with effectively. The existing noise purification methods use attention mechanism [44] and autoencoders [35] to increase the attention of related items, but the attention value itself may be inaccurate, which leads to room for improvement in the purification of noise features. Compared with the implicit feedback with noise, the explicit feedback can accurately indicate the user's preference. Therefore, the noise in the implicit feedback representation can be purified by the explicit feedback representation, and the user's preference can be more clearly described by the purified implicit representation and explicit representation, which has not been considered by existing methods.

In addition to the lack of noise purification, the existing methods are insufficient to represent the long-term preferences of users. Long-term preference refers to the behavior preference of users over a long period of time, which is usually relatively stable. The sequential recommendation methods, such as SDM [27], DIEN [49] and DSIN [6], try to increase the length of users' click sequence to capture users' stable long-term preferences. However, they have the following problems. First, the existing models of capturing long-term interest are all item-based methods, and the length of the item sequence used in these methods is often limited by the memory and computing resources, which leads to the gap between the model preference and the user's stable long-term preference. On the contrary, we argue that users' long-term interests should be characterized from the user level. Specifically, long-term interests should be sequence-independent and mainly related to the user profile. For example, people who own a pet cat may purchase certain cat food at intervals, which may not be related to their recent behavior sequences. Furthermore, the user's long-term interests characterized from a user-based perspective enables sequence decoupling, so as to ensure that the model is not limited to the length of the sequence. Second, the feedback information such as rating, click/unclick, like/dislike can reflect users' preference, and the preferences formed by different types of feedback are distinctive. Only by fine-grained modeling multiple types of sequential feedback can we better understand the long-term preferences of the user. MIMN [31] uses NTM [9] to maintain the latest interest state for each user and its update depends on the real-time user's click behavior to trigger events. However, all the methods mentioned above only use the click sequence to describe users' preferences, which leads to the incomplete description of user preferences. Based on the above analysis, it can be seen that user's long-term preference modeling and noise feature purification are both unsolved problems in CTR prediction, and without loss of generality, we define the preference representation learned from the limited length sequence as short-term interest.

In seeking to address these challenges, we propose DUMN for recommendation. First, we design a feature purification (FP) module based on vector orthogonal mapping [32] for short-term preference modeling. Specifically, we construct two sets of contrast pairs $< click, dislike >$ and $< unclick, like >$, and map the first representation vector $click$ and $unclick$ of each pair to the vertical direction of the second representation vector $dislike$ and $like$, and the mapped vector is used as the purified representation. Then, a user memory network (UMN) is proposed to understand the stable long-term preference of users in a fine-grained way. UMN improves the memory network used in NTM [9] to represent all types of feedback, and designs a novel triple loss to regularize the memory network, so that the content written in the memory network can truly express the user's preferences. Finally, a preference-aware interactive representation (PAIR) module is proposed to obtain the cross representation that can simultaneously aggregate user long-term and short-term preferences. Our contributions are summarized as follows:

- We introduce a novel denoising user-aware memory network for CTR prediction task, which can understand the unbiased evolution of users' preferences by fine-grained modeling of users' feedback information.
- We design a FP module for implicit feature purification. Through a novel vector orthogonal mapping method, the FP module can effectively extract the differences in two sets of contrast pairs $< click, dislike >$ and $< unclick, like >$.
- We propose UMN module and PAIR module for fine-grained long-term preference modeling and cross representation of long-term and short-term interests, respectively.
- We conduct experiments on two real-world datasets. The experimental results show that our DUMN is superior to the existing state-of-the-art baselines, which verifies the effectiveness of our DUMN.

2 RELATED WORK

2.1 Recommendation with Implicit Feedback

User feedback data contains both precise but relatively rare explicit feedback and rich but noisy implicit feedback. As mentioned in

Introduction, we can obtain the user's unbiased preference presentation by modeling both explicit feedback and implicit feedback at the same time. A large number of existing works have also proved the improvement of system performance by using various feedback information such as implicit and explicit through experiments. [14] treats all unobserved items of users as negative instances and expresses the value of implicit feedback information as confidential. [20, 26, 35, 46] fuses CF and various feedback information, and [16] uses weak supervision method to model feedback information. BINN [24] constructs a sequence of multiple interactive data of each user, and tries to use RNN to model the unbiased behavior of users. Furthermore, more algorithms use multi-task learning frameworks to jointly solve ranking and rating tasks by combining various explicit feedback and implicit feedback [12]. Recently, [42] constructs the multi-relational item graph for learning global item-to-item relations and develops the novel graph model MGNN-SPred to learns global item-to-item relations through graph neural network. GCN-based GRCN [43] is proposed to implement pruning of noisy edges in the graph constructed by users' feedback information on items. FAWMF [2] applies variational autoencoder to realize adaptive weight assignment of implicit feedback and effective model learning. To the best of our knowledge, we are the first to use the representation of explicit feedback to purify the representation of implicit feedback by orthogonal mapping.

2.2 CTR Model

Early methods of CTR prediction construct the interactive data of users and items into a user-item rating matrix, and the user-based or item-based CF method [8] is used for rating prediction. In these methods, the user and item evaluation vectors are regarded as the presentation vectors of each user and item. Later, based on the co-occurrence matrix of CF algorithm, MF adds the concept of hidden vector to reduce the representation dimension of vector and enhance the ability of the model to deal with sparse matrix. MF based methods, such as singular value decomposition [21], non-negative matrix factorization [7] and probabilistic matrix factorization [32], are widely used in recommendation system.

With the rapid development of deep learning in many fields, such as computer vision [15, 34], natural language processing [5], there are more and more researches on personalized recommendation of jobs [1], music [39], news [30] and video [4] based on deep neural network. Different from the traditional CTR prediction models such as MF and CF, which capture the similarity between users and items through feature engineering, the CTR prediction method based on deep learning uses neural networks to capture the interaction between features, which can better capture high-order interactive features [23, 25]. Wide&Deep [3] considers the wide part for memory and the deep part for generalization together, making the model have the advantages of both logistic regression and deep neural network. DIN [50] proposes an attention mechanism to capture the relative interest of candidates and obtain adaptive interest representation. Inspired by the success of the Transformer network in neural machine translation [40], ATRANK [48] invents an attention-based framework for CTR prediction. SASRec [18] uses self-attention to capture long-term semantics and makes the predictions based on relatively few actions. DIEN [49] proposes a two-layer RNN structure with an attention mechanism, which uses attention weights to control the second-layer RNN to activate the interests most relevant to the candidates. Furthermore, DSIN [6] captures the evolution of user preferences in the session by dividing the user's interactive behavior by session.

In recent years, inspired by the development of neural network graph embedding algorithm [19], more and more attention has been paid to graph structure development for various recommended scenarios [17, 36]. [45] use different types of entity relationships in heterogeneous information networks to make use of the personalized recommendation framework. Recently, in order to address the early summarization issue on heterogeneous information networks, NIRec [17] is designed to capture and aggregate rich interactive patterns in both node-level and path-level.

3 METHOD

In this section, we present the technical details of DUMN. As shown in Fig. 1, DUMN mainly consists of four modules, namely embedding layer, FP layer, UMN layer and PAIR layer. First, the embedding layer takes $user$, ad, user's $click$, $unclick$, $like$ and $dislike$ sequences as input, and implements embedding for all input data. Secondly, FP uses the multi-head interaction-attention component shown in the lower right side of Fig. 1 to model the user's various implicit/explicit feedback sequences, and uses the representation of explicit feedback to purify the representation of implicit feedback by orthogonal mapping. Third, the UMN module captures users' fine-grained long-term interests by improving the memory network to all types of feedback. Finally, PAIR combines the short-term and the long-term interest representation to obtain the cross representation, and then uses the fully connected layer for CTR prediction.

3.1 Problem Definition

The DUMN network is represented by the function $\mathcal{F}(x;\theta)$, where θ is the parameters and x is the input, which contains the initial representation of the user, target item and four sequences of user's click, unclick, like and dislike feedback. Our goal is to make the predicted value $\hat{y}$ of $\mathcal{F}(x;\theta)$ as close to the user's real click preference $y \in \{0,1\}$ as possible by minimizing the specific loss $\mathcal{L}$, so as to realize the purpose of CTR prediction.

3.2 Embedding Layer

The input of DUMN can be divided into three parts: $user\ profile$, ad and $user\ behavior\ sequences$. The attribute fields of $user\ profile$ includes user_id, gender, age, etc. ad refers to the target item for CTR prediction, which includes item_id and brand_id. $user\ behavior\ sequence$ is a list of items. In this paper, we model four kinds of behavior sequences, which are click sequence $\mathbf{C} = [C_1, C_2, ..., C_T]$ and unclick sequence $\mathbf{U} = [U_1, U_2, ..., U_T]$ of implicit feedback, dislike sequence $\mathbf{D} = [D_1, D_2, ..., D_T]$ and like sequence $\mathbf{L} = [L_1, L_2, ..., L_T]$ of explicit feedback, where T is the maximum sequence length input to the model. As previously did [6, 49], we use the embedding layer to transform the high dimensional sparse ids into low dimensional dense representations. The concatenate of different fields' embedding output from $user\ profile$ and ad form e_{user} and e_{item} respectively. Accordingly, after embedding $\mathbf{C}$, $\mathbf{U}$, $\mathbf{D}$ and $\mathbf{L}$, the outputs are $\mathbf{e}_c \in \mathbb{R}^{T\times E}$, $\mathbf{e}_u \in \mathbb{R}^{T\times E}$,

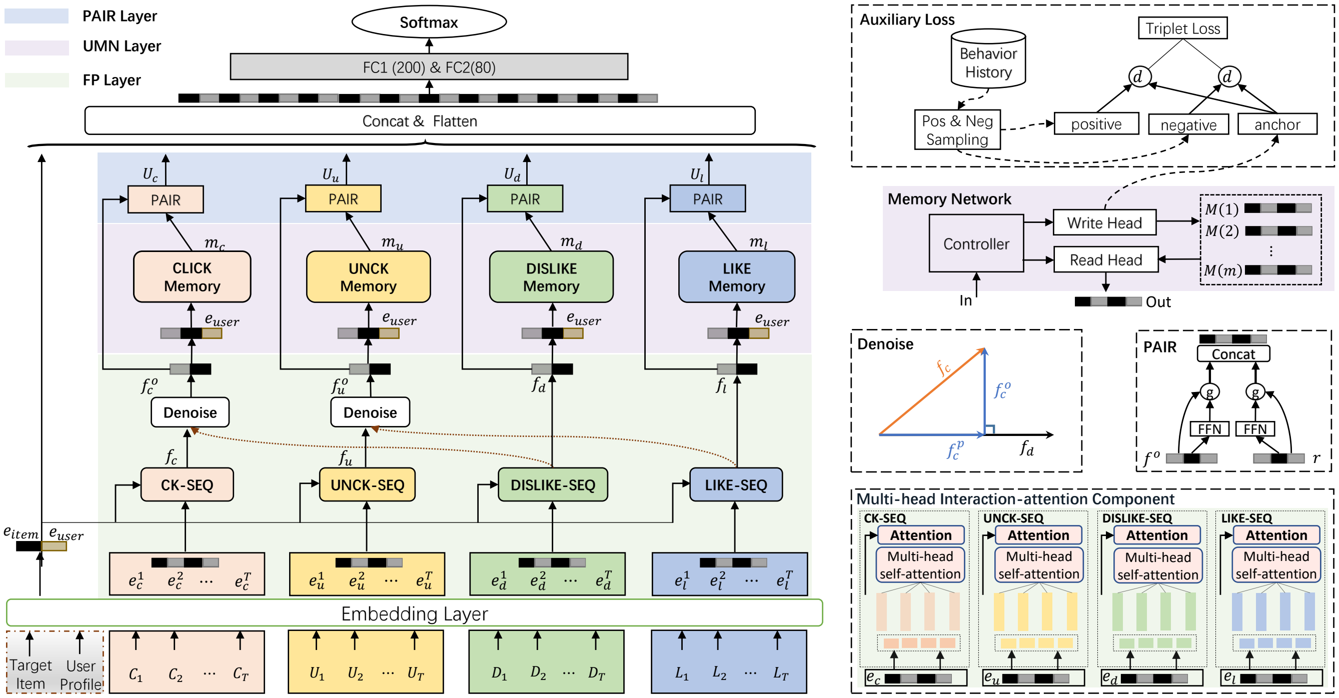

Figure 1: The overall architecture of denoising user-aware memory network.

$\mathbf{e}_d \in \mathbb{R}^{T\times E}$ and $\mathbf{e}_l \in \mathbb{R}^{T\times E}$, where E is the unified dimension of embedding.

3.3 Feature Purification Layer

The feature purification layer takes the embedding lists $\mathbf{e}_c$ and $\mathbf{e}_u$ of implicit feedback, $\mathbf{e}_l$ and $\mathbf{e}_d$ of explicit feedback, $\mathbf{e}_{user}$ of the user and $\mathbf{e}_{item}$ of the target item as inputs. Specially, the feature purification layer uses two components to learn the short-term interest expressed by the feedback sequences and purify the feature of implicit feedback by proposing a novel vector orthogonal mapping method.

3.3.1 Multi-head Interaction-attention Component. Inspired by the potential of self-attention mechanism in data correlation learning [40], we model various feedback behaviors of users on the framework of multi-head self-attention network to obtain fine-grained preference representation, and the whole process is shown in the lower right side of Fig. 1. We use $\mathbf{e}_c$ as an example to introduce the working of multi-head self-attention network. Mathematically, we construct $\mathbf{e}_c$ as a form with H heads, that is $\mathbf{e}_c = [\mathbf{e}_{c,1}, ..., \mathbf{e}_{c,h}, ..., \mathbf{e}_{c,H}]$, where $\mathbf{e}_{c,h} \in \mathbb{R}^{T\times \frac{1}{H}E}$ is the h-th head of $\mathbf{e}_c$, H is the number of heads. The output of the multi-head self-attention network is calculated as follows:

$$\mathbf{head}_h = \text{softmax}\left(\frac{\mathbf{e}_{c,h}\mathbf{W}^Q_{c,h}(\mathbf{e}_{c,h}\mathbf{W}^K_{c,h})^{\mathrm{T}}}{\sqrt{T}}\right)\mathbf{e}_{c,h}\mathbf{W}^V_{c,h} \quad h = 1, 2, ..., H, \tag{1}$$

$$\mathbf{O}_c = \text{Concat}(\mathbf{head}_1, \mathbf{head}_2, ..., \mathbf{head}_H)\mathbf{W}^F \tag{2}$$

where $\mathbf{W}^Q_{c,h}$, $\mathbf{W}^K_{c,h}$ and $\mathbf{W}^V_{c,h}$ are trainable linear matrices. Then, we calculate the attention value between the target *ad* item and the item representation of each position in the sequence representation through the full connection layer, as shown in the following formulas:

$$\alpha_j = \text{ReLU}(\text{Concat}(\mathbf{e}_{user}, \mathbf{e}_{item}, \mathbf{o}_c^j)\mathbf{W}_c) \quad j = 1, 2, ..., T, \tag{3}$$

$$\tilde{\alpha}_j = \frac{\exp(\alpha_j)}{\sum_{j'=1}^{T}\exp(\alpha_{j'})} \quad j = 1, 2, ..., T, \tag{4}$$

where $\mathbf{o}_c^j$ is the output representation of the j-th item through the multi-head self-attention network. $\mathbf{W}_c$ is trainable linear matrix. ReLU$(\cdot)$ is the activation function. Then, the representation of the click sequence fused with the target item can be calculated as:

$$\mathbf{f}_c = \sum_{j=1}^{T}\tilde{\alpha}_j\mathbf{o}_c^j. \tag{5}$$

Similarly, we can get the representations of unclick, like, and dislike sequences as $\mathbf{f}_u$, $\mathbf{f}_l$, and $\mathbf{f}_d$ by using formulas (1)-(5).

3.3.2 Feature Orthogonal Mapping Component. The representations $\mathbf{f}_c$ and $\mathbf{f}_u$ of the implicit feedback contain inherent noise, and the goal of the feature orthogonal mapping component is to purify the representations $\mathbf{f}_c$ and $\mathbf{f}_u$ of implicit feedback. We argue that the explicit feedback representation which can definitely indicate user preferences can be used to purify the implicit feedback representation which can not directly indicate user preferences. So we extract two groups of orthogonal mapping pairs $< click, dislike >$ and $< unclick, like >$ with differences from implicit and explicit feedback, and their corresponding sequence representations are

$< \mathbf{f}_c, \mathbf{f}_d >$ and $< \mathbf{f}_u, \mathbf{f}_l >$ respectively. Taking $< \mathbf{f}_c, \mathbf{f}_d >$ as an example, in order to purify the representation of implicit feedback sequence with noise, we project the first element $\mathbf{f}_c$ of the sequence representation pair onto the orthogonal direction of the second element $\mathbf{f}_d$. The original feature vector $\mathbf{f}_c$ is projected into the orthogonal feature space to eliminate the noise features. Compared with the original vector, the orthogonal mapping vector contains pure and efficient user preferences.

Formally, we describe the orthogonal mapping process of sequence pair $< \mathbf{f}_c, \mathbf{f}_d >$ in a two-dimensional space shown in the middle of the right side of Fig. 1. The noise representation vector $\mathbf{f}_c^p$ in $\mathbf{f}_c$ can be obtained by projecting the vector $\mathbf{f}_c$ onto the vector $\mathbf{f}_d$:

$$\mathbf{f}_c^p = \text{project}(\mathbf{f}_c, \mathbf{f}_d), \tag{6}$$

where $\text{project}(a, b) = \frac{a \cdot b}{|b|} \frac{b}{|b|}$ represent the projection function. a and b are vectors with the same dimension. Then, we can get the vector representation purified by orthogonal mapping as follows:

$$\mathbf{f}_c^o = \mathbf{f}_c - \mathbf{f}_c^p. \tag{7}$$

Obviously, according to formula (6), the representation of implicit feedback contains a mixture of user's click and dislike noise $\mathbf{f}_c^p$, which is filtered from the original representation $\mathbf{f}_c$, and the new pure feature $\mathbf{f}_c^o$ can effectively represent the user's pure click preference in orthogonal space, which is also in line with the assumption that the user's click and dislike representation should be distinctive. Similarly, according to formulas (6)-(7), we can get the orthogonal mapping vector $\mathbf{f}_u^o$ between the representation $\mathbf{f}_u$ of unclick sequence and the representation $\mathbf{f}_l$ of like sequence, and the purified vectors $\mathbf{f}_c^o$ and $\mathbf{f}_u^o$ can better describe the unbiased preferences of users.

3.4 User Memory Network Layer

In order to get a more stable and fine-grained representation of long-term preference from the perspective of users than the item-based methods, we improve the memory network used in NTM [9]. Specifically, the memory network of NTM contains multiple slots to model the user's click sequence, and uses a controller to generate the key for reading or writing of the user's click sequence representation, so as to complete the operation of **memory read** and **memory write** for the memory network. Considering that memory network can store feature representation, and each slot has the characteristic of aggregating the same feature representation, we extend it to store user-level long-term interests, so that the user feature representation in the same slot can reflect the similar interests between users, and the long-term interest representation obtained in this way is more generalized than using only user_id embedding.

We improve the basic memory network as follows. First, in order to capture users' fine-grained unbiased long-term interests, we use four memory network $\mathbf{M}_c$, $\mathbf{M}_u$, $\mathbf{M}_l$ and $\mathbf{M}_d$ to save users' click, unclick, like and dislike preferences respectively, and each memory network contains m slots whose output dimension is Z. Second, the input of the controller is replaced by the concatenate of users' short-term representation obtained by FP and the embedding of $user\ profile$ from the representation of item to ensure that the model can learn the user level long-term interest representation. Taking $\mathbf{M}_c$ as an example, **memory read** and **memory write** of $\mathbf{M}_c$ are as follows.

Memory read. Input the concatenate of $\mathbf{f}_c^o$ and $\mathbf{e}_{user}$, and the controller generates a read key $\mathbf{k}_c$ to address the memory $\mathbf{M}_c$ through a fully connected layer.

$$\mathbf{k}_c = \text{FFN}(\text{Concat}(\mathbf{f}_c^o, \mathbf{e}_{user})), \tag{8}$$

where FFN$(\cdot)$ denotes the feed-forward network. Then, by traversing all memory slots, a weight vector $\mathbf{w}_c^r$ is generated:

$$\mathbf{w}_c^r(j) = \frac{\exp(\text{K}(\mathbf{k}_c, \mathbf{M}_c(j)))}{\sum_{j'=1}^{m} \exp(\text{K}(\mathbf{k}_c, \mathbf{M}_c(j')))} \quad j = 1, 2, ..., m, \tag{9}$$

$$\text{K}(\mathbf{k}_c, \mathbf{M}_c(j)) = \frac{\mathbf{k}_c^{\text{T}} \mathbf{M}_c(j)}{\|\mathbf{k}_c\| \, \|\mathbf{M}_c(j)\|}, \tag{10}$$

finally, the weighted memory summary is calculated as the output $\mathbf{r}_c \in \mathbb{R}^Z$:

$$\mathbf{r}_c = \sum_{j=1}^{m} \mathbf{w}_c^r(j) \mathbf{M}_c(j). \tag{11}$$

Memory write. The generation of the weight vector $\mathbf{w}_c^w$ for memory write is similar to the **memory read** operation in equation (9). The controller also generates two additional keys, add vector $\mathbf{add}_c$ and erase vector $\mathbf{erase}_c$, to control the update of memory.

$$\mathbf{M}_c = (1 - \mathbf{E}_c) \odot \mathbf{M}_c + \mathbf{A}_c, \tag{12}$$

where $\mathbf{E}_c = \mathbf{w}_c^w \otimes \mathbf{erase}_c$ is the erase matrix. $\mathbf{A}_c = \mathbf{w}_c^w \otimes \mathbf{add}_c$ is the add matrix. $\odot$ and $\otimes$ means dot product and outer product respectively.

Accordingly, we can get four representations of users' long-term preferences: $\mathbf{r}_c$, $\mathbf{r}_u$, $\mathbf{r}_l$ and $\mathbf{r}_d$.

3.5 Preference-aware Interactive Representation Component

In order to get the cross representation of users' long-term and short-term interests, we design a gating mechanism to fuse the long-term and short-term representations of users' preferences with the same type. Similarly, we use the representation of click preference as an example to get the cross representation $\mathbf{U}_c$ of long-term and short-term click preference through the following formula:

$$\mathbf{U}_c = \text{Concat}(\mathbf{f}_c^o * \text{sigmoid}(\mathbf{f}_c^o \mathbf{W}_1), \mathbf{r}_c * \text{sigmoid}(\mathbf{r}_c \mathbf{W}_2)), \tag{13}$$

where $*$ is the Hadamard product. sigmoid$(\cdot)$ is the activation function and $\mathbf{W}$ is the dimension conversion matrix to ensure that the dimensions of the two vectors are consistent. The cross representation $\mathbf{U}_u$, $\mathbf{U}_l$ and $\mathbf{U}_d$ corresponding to unclick, like and dislike can also be obtained. Therefore, we get the deep cross representation interest representation of users:

$$\mathbf{R}_{cross} = \text{Concat}(\mathbf{U}_c, \mathbf{U}_u, \mathbf{U}_l, \mathbf{U}_d). \tag{14}$$

Finally, we concatenate the representations of user, target item, long-term interests, short-term interests and cross features as deep representations, and then use a fully connected layer with sigmoid$(\cdot)$ function to generate the predicted CTR as:

$$\hat{y} = \text{sigmoid}(\text{FFN}(\text{Concat}(\mathbf{e}_{user}, \mathbf{e}_{item}, \mathbf{R}_{cross}))). \tag{15}$$

3.6 Loss Function

In our proposed model DUMN, we have two goals: 1) the predicted CTR of the target item should be as close to the true label as possible; and 2) we need to ensure that the content written into these 4 memory networks can truly express the user's long-term preferences of click, unclick, like and dislike.

For goal 1), we achieve it by minimizing the average logistic loss as:

$$\mathcal{L}_1 = -\frac{1}{N} \sum_{(x,y)\in\mathcal{D}} (y\log(\hat{y}) + (1-y)\log(1-\hat{y})), \tag{16}$$

where $\mathcal{D}$ is the training set of size N. x is the input of DUMN. $y \in \{0, 1\}$ represents whether the user clicks the target item.

For goal 2), we propose an auxiliary loss based on triplet loss to help complete the **memory write** operation. Specifically, for each content update, we randomly sample an item from each of the four feedback sequences of click, unclick, like and dislike in each batch, and record their output $\mathbf{s}_c$, $\mathbf{s}_u$, $\mathbf{s}_l$ and $\mathbf{s}_d$ in the embedding layer. It is worth noting that the samples are sampled from the data of all user interactions, not only from the constructed sequence for short-term interest representation. Then, we construct positive and negative sample pairs $< \mathbf{s}_c, \mathbf{s}_u >$, $< \mathbf{s}_u, \mathbf{s}_c >$, $< \mathbf{s}_l, \mathbf{s}_d >$ and $< \mathbf{s}_d, \mathbf{s}_l >$ for $\mathbf{M}_c$, $\mathbf{M}_u$, $\mathbf{M}_l$ and $\mathbf{M}_d$ respectively. Without loss of generality, we use $\mathbf{M}_c$ as an example to explain the proposed triplet loss:

$$\mathcal{L}_c = \max(\mathrm{d}(\mathbf{q}_c, \mathbf{s}_c) - \mathrm{d}(\mathbf{q}_c, \mathbf{s}_u) + margin, 0), \tag{17}$$

$$\mathbf{q}_c = \sum_{j=1}^{m} \mathbf{w}_c^w(j)\mathbf{M}_c(j), \tag{18}$$

$$\mathrm{d}(a, b) = 1 - \frac{a \cdot b}{\|a\| \, \|b\|}, \tag{19}$$

where $\mathrm{d}(\cdot, \cdot)$ is the function to calculate the similarity, and cosine similarity is used in this paper. $\mathbf{s}_c$, $\mathbf{s}_u$ and $\mathbf{q}_c$ are positive sample, negative sample and anchor corresponding to the upper right side of Fig. 1, respectively. Similarly, we can get the triplet losses $\mathcal{L}_u$, $\mathcal{L}_l$ and $\mathcal{L}_d$ of $\mathbf{M}_u$, $\mathbf{M}_l$ and $\mathbf{M}_d$, respectively. Furthermore, goal 2) can be achieved by accumulating these 4 auxiliary losses as:

$$\mathcal{L}_2 = \mathcal{L}_c + \mathcal{L}_u + \mathcal{L}_l + \mathcal{L}_d. \tag{20}$$

then, the final loss function of DUMN is expressed as $\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2$.

4 EXPERIMENTS

In this section, we conduct experiments on different datasets to prove the effectiveness of our DUMN by comparing with several state-of-the-art methods. We start with 3 research questions (RQ) to guide the experiment and the following discussion:

- **RQ1:** Compared with state-of-the-art methods, can DUMN achieve better performance?
- **RQ2:** What is the impact of modules designed in DUMN? Are the proposed feature purification layer and user memory network layer modules necessary for improving performance?
- **RQ3:** What is the impact of hyper-parameter settings on CTR prediction performance in DUMN?

4.1 Experimental Setups

4.1.1 Data Description. We evaluate the model on two real-world e-commerce datasets, namely Alibaba dataset and Industrial dataset. For the Alibaba dataset, it is a public dataset, collecting ad display/feedback logs of 1.14 million users in Taobao's recommendation system. We have made statistics on the training set and test set of Alibaba dataset, including 26 million pieces of feedback records from 1.14 million users in 0.8 million items, and these items can be divided into 12,960 categories. For Industrial dataset, it contains 5.5 billion feedback records from 1.1 billion users and 91.0 million items in 30 days. Records from 2020-12-25 to 2021-01-18 are for training, and records from 2021-01-19 to 2021-01-24 are for testing. In particular, these two datasets contain a variety of implicit and explicit feedback data, we record the items purchased, added to the shopping cart and labeled with like in the log feedback as explicit like feedback, the items marked with dislike as explicit dislike feedback, the items simply clicked by user as implicit click feedback, the items displayed but not operated as implicit unclick feedback. Since we focus on the CTR prediction tasks, we treat all valid click interactions with the label of 1.

4.1.2 Compared Methods. We compared DUMN with following mainstream CTR prediction methods:

- **Wide&deep:** Wide&Deep [3] consists of two parts: wide module of memory and deep module of generalization. It can effectively capture the high-order cross features between users and items.
- **PNN:** PNN [33] uses the product layer containing inner product and outer product to capture the interactive patterns between interfield categories.
- **DeepFM:** DeepFM [11] is achieved by replacing the wide component in Wide&Deep with FM layer, which can also capture the cross features of users and items.
- **DIN:** DIN [50] uses attention mechanism to activate related users' historical behaviors, which can fully exploits the relationship between users' historical behaviors and the target item.
- **DIEN:** DIEN [49] uses two layers of GRU to extract latent temporal interests from user behaviors and models interests evolving process.
- **DSIN:** DSIN [6] divides the user's historical click sequence into sessions, and then apply Bi-LSTM to model how users' interests evolve and interact among sessions.
- **AutoInt:** AutoInt [37] introduces self-attentive neural network to find low-dimensional representations of the sparse and high-dimensional raw features and their meaningful combinations.
- **DFN:** DFN [44] jointly consider explicit and implicit feedback to learn user unbiased preferences for recommendation. Among them, the explicit feedback includes dislike sequence, and the implicit feedback includes click sequence and unclick sequence.
- **DMT:** DMT [10] uses multiple Transformers to model users' multiple types of behavior sequences, including click sequence of items, cart sequence of items, and order sequence of items.

We divide the above baselines into the following two categories: the first is the non-sequence method that does not construct feedback sequence to obtain cross features between users and items, including Wide&Deep, PNN and DeepFM; the second is the sequence-based method that uses feedback sequence information to capture

Table 1: Performance comparison with different baselines.

Model	Alibaba	Industrial
Wide&Deep	0.6326	0.7526
PNN	0.6328	0.7602
DeepFM	0.6347	0.7612
DIN	0.6330	0.7653
DIEN	0.6343	0.7803
DSIN	0.6375	0.7873
AutoInt	0.6360	0.7708
DFN	0.6368	0.7884
DMT	0.6378	0.8039
DUMN	**0.6496**	**0.8136**

Table 2: Effect of user memory network layer.

Model	Alibaba	Industrial
DUMN-UMN	0.6434	0.7966
DUMN+UMN1	0.6446	0.8073
DUMN	**0.6496**	**0.8136**

Table 3: Effect of NTM.

Model	Alibaba	Industrial
AutoInt	0.6360	0.7708
AutoInt+UMN	0.6396	0.7802
DFN	0.6368	0.7884
DFN+UMN	0.6437	0.7973

users' interest evolution, including DIN, DIEN, DSIN, AutoInt, DMT and DFN. In particular, DMT and DFN are models based on implicit feedback and explicit feedback.

4.1.3 Parameter Settings. The DUMN model is implemented in Python based on the Tensorflow framework. All the experiments are conducted on a server machine equipped with a 16 GB NVIDIA Tesla V100 GPU. For hyper-parameters, the maximum length T of each feedback sequence is set to 100. The output dimension of the embedding layer is 16. The dimension of the feed-forward network used in the memory network is set to 512. The number of the slot in the memory network is set to 256 and the dimension of each slot is set to 64. During the training phase, we set the learning rate to 0.005 respectively, and use Adam as the optimizer. It is worth noting that the optimal parameters of DUMN are obtained by grid search, and the parameter sensitivity experiments of some important parameters are recorded in Section 4.4. For the performance of Alibaba dataset, the baseline methods such as AutoInt, DMT and DFN are implemented per their GitHub settings, and the rest baseline methods use the performance recorded in [6].

4.2 Result Analysis (RQ1)

We evaluated the CTR prediction task on the datasets mentioned above. Specifically, the Area Under Curve (AUC) widely used in binary classification problems is used as the metric. The results of the evaluation on two datasets are recorded in Table 1. By comparing with several state-of-the-art baselines, the following three conclusions can be drawn: **1)** Our proposed DUMN outperforms all baselines on these two datasets. The experimental results show that our proposed model is effective on the CTR prediction task, and the recommendation performance can be significantly improved by feature purification of implicit feedback representation and fusion of short-term and long-term interests. Compared with the best experimental results of baselines, the performances of DUMN on Alibaba dataset and Industrial dataset are improved by 1.18% and 0.97%, respectively. **2)** Among the two kinds of baselines, most sequence-based methods (DIN, DIEN, DSIN, AutoInt, DMT and DFN) are better than non-sequence methods (Wide&Deep, PNN and DeepFM) in most cases, and an intuitive explanation is that these sequential methods can better capture the evolution of users' interests over time than non-sequential methods. It is worth noting that our DUMN also uses user feedback sequence, and uses attention mechanism to understand the evolution of user interest. **3)** In the sequence-based method, DFN achieves relatively good results, which is close to the experimental results of DSIN. Specifically, it can better get the unbiased preference evolution of users by describing the user feedback data in a more fine-grained way. It is also worth noting that our DUMN further divides user interests into long-term interests and short-term interests, which can obtain more accurate and deeper unbiased user preference representation.

4.3 Ablation Study (RQ2)

To investigate the effectiveness of components in DUMN, we conduct extensive ablation studies on the two datasets.

4.3.1 Effect of Feature Purification. The purpose of feature purification can purify the presentation of implicit feedback sequences with noise. In order to explore the effectiveness of using explicit feedback representation to denoise implicit feedback representation by orthogonal mapping in the FP component, we design DUMN-FP which removes the FP component for denoising. The evaluation results of DUMN-FP in Alibaba dataset and Industrial dataset are 0.6417 and 0.7981 respectively. Comparing the results with those of DUMN reported in Table 1, we can find that: It is effective to purify the noise by using the proposed orthogonal mapping method. An intuitive explanation is that the explicit representation can accurately represent a user's preference, and the orthogonal space obtained by it should also be a noise-free space, which can better describe the user's unbiased preferences.

4.3.2 Effect of User Memory Network Layer. The user memory network layer is a fine-grained long-term interest characterization module based on user profile. In order to verify its effectiveness, we designed the following two models: DUMN-UMN which removes the user memory network layer and DUMN+UMN1 which replaces four memory networks with one memory network. Table 2 shows the evaluation results. It can be seen from the result that removing

Table 4: Effect of PAIR.

Model	Alibaba	Industrial
DUMN$_{CONCAT}$	0.6438	0.8023
DUMN$_{CROSS}$	0.6487	0.8049
DUMN$_{FFN}$	0.6486	0.8037
DUMN$_{ATTE}$	0.6433	0.7966
DUMN	**0.6496**	**0.8136**

the representation of long-term interest or not distinguishing various feedback information will significantly reduce the prediction performance of the model, which is also in line with our previous hypothesis that only by fine-grained modeling multiple types of sequential feedback can we better understand the long-term preferences of the user.

Then, we further explore whether the capture of long-term interest can improve other existing models. Therefore, we introduce the user memory network which can capture users' long-term interest into AutoInt and DFN to get models AutoInt+UMN and DFN+UMN. Table 3 shows the evaluation results, AutoInt+UMN and DFN+UMN beats AutoInt and DFN respectively, showing that the performance of existing methods has been improved after adding NUM module which can capture users' long-term interests, indicating that the integration of long-term and short-term interests can better understand the evolution of users' preferences.

4.3.3 Effect of Preference-aware Interactive Representation Component. The purpose of the preference-aware interactive representation component is to fuse short-term and long-term interests. In order to explore the impact of different fusion methods on CTR prediction performance, we design a variety of fusion methods and get the following four methods: DUMN$_{CONCAT}$ which use concatenate operation, DUMN$_{CROSS}$ which use cross operation, DUMN$_{FFN}$ which use the feed-forward network to represent first and then concatenate, and DUMN$_{ATTE}$ which use attention operation. Among them, concatenate operation represents the function Concat$(\cdot)$, and cross operation refers to the concatenate after the addition, subtraction and multiplication of the elements at the corresponding positions of two vectors. According to Table 4, the DUMN uses the gate-based method proposed in Section 3.5 can achieve the best performance on the fusion of long-term and short-term interest representation, which proves the feasibility of our proposed gate-based fusion method. In addition, we find that the cross and attention mechanism operations that can get high-order crossover features are better than the direct concatenate operation. The intuitive explanation is that the user's preference for the target item needs to be represented by both long-term interest and short-term interest, and the dependence of different items on the long-term and short-term is also different.

4.3.4 Effect of the Proposed Triplet Loss. The triplet loss is to ensure that the content written into the 4 memory networks can truly express the user's preferences. To evaluate the proposed triplet loss, we design a variety of comparison methods: DUMN-TL which removes the triplet loss from the loss function and DUMN$_{RS}$ which uses random sampling to sample positive and negative samples. It should be noted that our DUMN uses cosine similarity to find the hardest positive and negative samples in each batch. According to the results shown in Table 5, we can find that the performance of our DUMN is significantly reduced by removing the limit of triple loss, and different sampling samples also have an impact on the performance. Specifically, the more dissimilar the representations of the sample pairs are, the more effective it is to ensure that the contents written in the memory network are reliable.

Table 5: Effect of the proposed triplet loss.

Model	Alibaba	Industrial
DUMN-TL	0.6467	0.7969
DUMN$_{RS}$	0.6472	0.7991
DUMN	**0.6496**	**0.8136**

Table 6: Effect of implicit feedback.

Model	Alibaba	Industrial
DUMN$_{IF}$	0.6392	0.8072
DUMN$_{AF}$	0.6394	0.8091
DUMN	**0.6496**	**0.8136**

4.3.5 Effect of Implicit/Explicit Feedback. The intention of the fine-grained description of various implicit and explicit feedback is to obtain unbiased user preferences. We design two models to evaluate the improvement of CTR prediction brought by fine-grained interest representation: DUMN$_{IF}$ which uses only implicit feedback and removes $\mathbf{M}_l$ and $\mathbf{M}_d$ in user memory network layer, DUMN$_{AF}$ which connects all types of feedback into a sequence through time. It should be noted that these new models do not use orthogonal mapping to purify the representation of implicit feedback, and DUMN$_{AF}$ constructs a single sequence of all feedback information, which makes it difficult to capture the fine-grained interest representation formed by different types of user feedback. Table 6 shows the performance of the comparison methods, and we can find that DUMN$_{AF}$ that uses all types of the user's feedback information for interest characterization has better performance than the DUMN$_{IF}$ that only use implicit feedback. Furthermore, DUMN which fine-grained description interest of users with various feedback can further improve the recommendation performance when compared with DUMN$_{AF}$. It is intuitive that the user preferences indicated by different feedback are inconsistent, and the corresponding representation space should also be different. Accordingly, DUMN$_{AF}$ without fine-grained modeling of interest will lead to the relative confusion of multi-interest representation.

4.4 Parameter Sensitivity (RQ3)

In this subsection, we investigate the impact of two hyper-parameters in our developed DUMN framework: the number of slots m and dimension Z of slots. For conciseness, we report experimental results on Alibaba dataset.

We make the number of slots m in DUMN varies in a range of $\{16, 32, 64, 128, 256\}$ and let the dimension Z of each slot varies in a

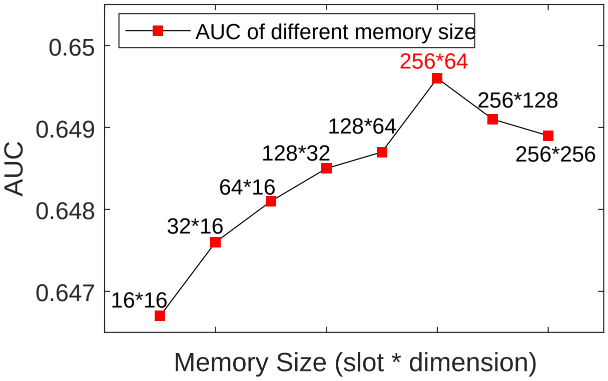

Figure 2: Impact of m and Z.

Table 7: Results of dislike prediction on Alibaba dataset.

Model	Alibaba-dislike	Industrial-dislike
DFN	0.7609	0.7394
DUMN	**0.8062**	**0.7601**

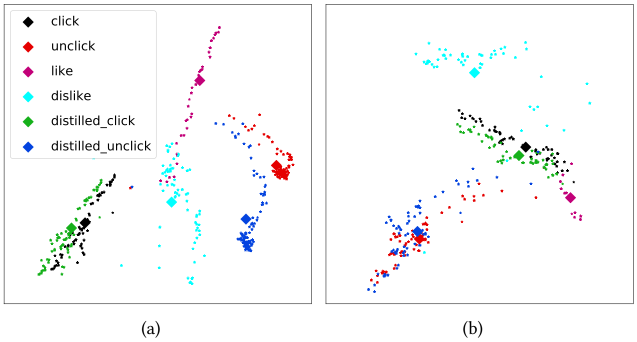

Figure 3: Visualization of purification characteristics. Different color represent the representation results of different types of feedback in FP module, each ⋆ represents the specific feedback generated by the user and item, and ⧫ represents the center of the interest representation cluster.

range of {16, 32, 64, 128, 256}. From Fig. 2, we can find that when m and Z in each memory network are 256 and 64 respectively, the performance of DUMN is the best. When m and Z are too large or too small, the effect of the model will be worse. The potential explanation for this phenomenon is that the parameters m and Z can directly reflect the storage capacity of the memory network. When the values of these parameters are small, the storage capacity is too weak to effectively store the long-term interests of users. If the values are too large, it may lead to overfitting.

4.5 Dislike Prediction

The above experiments show that the user's click preferences can be effectively predicted through DUMN, which is to predict the items that the user is interested in. In this subsection, we further use DUMN to do the prediction task that users are not interested in, which is also a new prediction task recently proposed in DFN. It is dedicated to predicting whether users rate the target item as dislike, so as to avoid the model's prediction results from frustrating users. Table 7 records the experimental results of DUMN and DFN, from which we can find that DUMN can achieve better prediction results than the baseline experimental result, which indicates that our DUMN is more sensitive to the capture of disliked preferences, and the better experimental results come from our fine-grained preference modeling of user feedback.

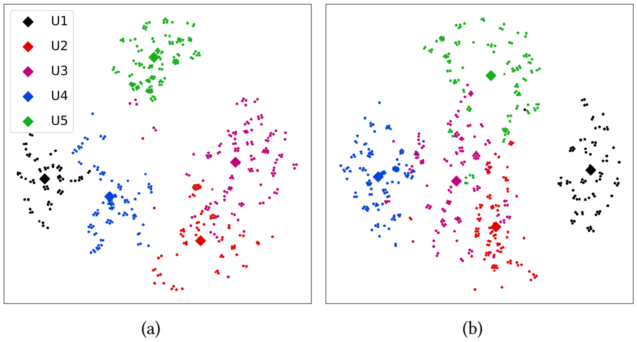

Figure 4: Visualization of long-term interest. Different colors represent the long-term interest representation of different users, and each point refers to the user's long-term interest representation read from the memory network.

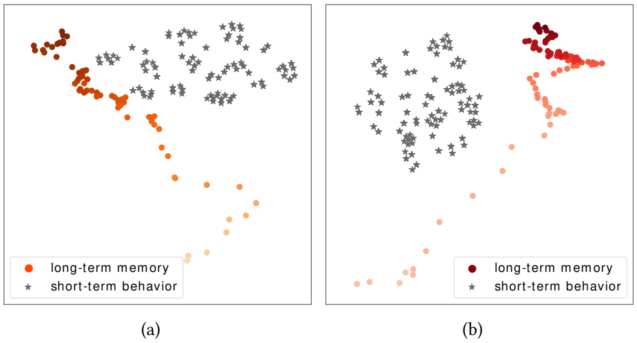

Figure 5: Visualization of long-term and short-term interests. • represents the user's long-term interest, and the deepening of color represents the change process of user's long-term interest in the training process. ⋆ denotes the user's short-term interest.

4.6 Case Study

In this section, we conduct case study to prove that the FP module FP of purifying implicit feedback representation and the PAIR module of capturing long-term interest is effective.

4.6.1 Display of Purification Characteristics. First, we show the dimensionality reduction visualization of different short-term interest representations of the same user in Fig. 3. Specifically, it includes the click and the unclick representations before and after denoising, the like representation and the dislike representation. From the figure, we can find that the noise in implicit feedback representation is effectively removed, and the distance between the denoised representation and the representation of orthogonal mapping corresponding to explicit feedback is also far away. For example, compared with before purification, the presentation of click and unclick after purification is far away from that of dislike and like respectively.

4.6.2 Display of Long-term Interest. Then, we visualized the representation of long-term interests obtained by multiple users from the memory network. It can be observed from Fig. 4 that the long-term interests of the same user are aggregated in the whole representation space, reflecting the stability of long-term interests, and the representations of long-term interests of different users can show intersection or no intersection. Among them, the non-intersection part reflects that users' interests are independent, and the intersection part indicates that users' long-term interests are partially similar.

4.6.3 Display of Long-term and Short-term Interests. Last, we visualize the long-term interest representation and short-term interest representation in Fig. 5, and we can draw the following conclusions: 1) In the training process of DUMN, the user's long-term interest representation is gradually aggregated into a smaller space, reaching a more stable state; and 2) Compared with the relatively stable long-term interest representation learned, the short-term interest representation of the user presents a more dispersed state under the unified feature space, which reflects the difference between long-term and short-term interests. It is worth noting that our DUMN model can effectively capture users' long-term and short-term interests of users through the FP module and the UMN module.

5 CONCLUSION

In this paper, we propose a novel denoising user-aware memory network (DUMN), which constructs four feedback sequences of users, namely click, unclick, like and dislike, to model users' preferences in a fine-grained way. DUMN uses feature purification layer and user memory network layer to purify the implicit feedback sequence representation of users and capture the stable long-term preference of users, respectively. A large number of experiments verify the effectiveness of the proposed model in the CTR prediction task. Experiments verify the effectiveness of the proposed model in the CTR prediction task. In the future, we intend to further explore the impact of noise purification and long-term interest in different recommendation scenarios, and subdivide the fine-grained representation of users on the basis of the existing to obtain more comprehensive unbiased preference representation of users.

REFERENCES

[1] Fedor Borisyuk, Liang Zhang, and Krishnaram Kenthapadi. 2017. LiJAR: A system for job application redistribution towards efficient career marketplace. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1397–1406.

[2] Jiawei Chen, Can Wang, Sheng Zhou, Qihao Shi, Jingbang Chen, Yan Feng, and Chun Chen. 2020. Fast adaptively weighted matrix factorization for recommendation with implicit feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3470–3477.

[3] 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.

[4] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT (1)*.

[6] 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. In *IJCAI International Joint Conference on Artificial Intelligence*, Sarit Kraus (Ed.). ijcai.org, 2301–2307.

[7] Cédric Févotte and Jérôme Idier. 2011. Algorithms for nonnegative matrix factorization with the β-divergence. *Neural computation* 23, 9 (2011), 2421–2456.

[8] David Goldberg, David Nichols, Brian M Oki, and Douglas Terry. 1992. Using collaborative filtering to weave an information tapestry. *Commun. ACM* 35, 12 (1992), 61–70.

[9] Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines. *arXiv preprint arXiv:1410.5401* (2014).

[10] Yulong Gu, Zhuoye Ding, Shuaiqiang Wang, Lixin Zou, Yiding Liu, and Dawei Yin. 2020. Deep Multifaceted Transformers for Multi-objective Ranking in Large-Scale E-commerce Recommender Systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2493–2500.

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

[12] Guy Hadash, Oren Sar Shalom, and Rita Osadchy. 2018. Rank and rate: multi-task learning for recommender systems. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 451–454.

[13] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *4th International Conference on Learning Representations (ICLR)*.

[14] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*. Ieee, 263–272.

[15] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4700–4708.

[16] Amir H Jadidinejad, Craig Macdonald, and Iadh Ounis. 2019. Unifying explicit and implicit feedback for rating prediction and ranking recommendation tasks. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*. 149–156.

[17] Jiarui Jin, Jiarui Qin, Yuchen Fang, Kounianhua Du, Weinan Zhang, Yong Yu, Zheng Zhang, and Alexander J Smola. 2020. An Efficient Neighborhood-based Interaction Model for Recommendation on Heterogeneous Graph. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 75–84.

[18] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 197–206.

[19] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[20] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 426–434.

[21] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[22] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1419–1428.

[23] Zeyu Li, Wei Cheng, Yang Chen, Haifeng Chen, and Wei Wang. 2020. Interpretable Click-Through Rate Prediction through Hierarchical Attention. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 313–321.

[24] Zhi Li, Hongke Zhao, Qi Liu, Zhenya Huang, Tao Mei, and Enhong Chen. 2018. Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1734–1743.

[25] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1754–1763.

[26] Nathan N Liu, Evan W Xiang, Min Zhao, and Qiang Yang. 2010. Unifying explicit and implicit feedback for collaborative filtering. In *Proceedings of the 19th ACM international conference on Information and knowledge management*. 1445–1448.

[27] Fuyu Lv, Taiwei Jin, Changlong Yu, Fei Sun, Quan Lin, Keping Yang, and Wilfred Ng. 2019. SDM: Sequential deep matching model for online large-scale recommender system. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2635–2643.

[28] Fuyu Lv, Mengxue Li, Tonglei Guo, Changlong Yu, Fei Sun, Taiwei Jin, and Keping Yang. 2020. Unclicked User Behaviors Enhanced SequentialRecommendation. *arXiv preprint arXiv:2010.12837* (2020).

[29] Yabo Ni, Dan Ou, Shichen Liu, Xiang Li, Wenwu Ou, Anxiang Zeng, and Luo Si. 2018. Perceive your users in depth: Learning universal user representations from multiple e-commerce tasks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 596–605.

[30] Kyo Joong Oh, Won Jo Lee, Chae Gyun Lim, and Ho Jin Choi. 2014. *Personalized news recommendation using classified keywords to capture user preference*.

[31] 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.

[32] Qi Qin, Wenpeng Hu, and Bing Liu. 2020. Feature projection for improved text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 8161–8171.

[33] 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.

[34] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 779–788.

[35] Ilya Shenbin, Anton Alekseev, Elena Tutubalina, Valentin Malykh, and Sergey I Nikolenko. 2020. RecVAE: A new variational autoencoder for Top-N recommendations with implicit feedback. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 528–536.

[36] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and S Yu Philip. 2018. Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering* 31, 2 (2018), 357–370.

[37] 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.

[38] 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.

[39] Aäron Van Den Oord, Sander Dieleman, and Benjamin Schrauwen. 2013. Deep content-based music recommendation. In *Neural Information Processing Systems Conference (NIPS 2013)*, Vol. 26. Neural Information Processing Systems Foundation (NIPS).

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NIPS*.

[41] 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.

[42] Wen Wang, Wei Zhang, Shukai Liu, Qi Liu, Bo Zhang, Leyu Lin, and Hongyuan Zha. 2020. Beyond clicks: Modeling multi-relational item graph for session-based target behavior prediction. In *Proceedings of The Web Conference 2020*. 3056–3062.

[43] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, and Tat-Seng Chua. 2020. Graph-Refined Convolutional Network for Multimedia Recommendation with Implicit Feedback. In *Proceedings of the 28th ACM International Conference on Multimedia*. 3541–3549.

[44] Ruobing Xie, Cheng Ling, Yalong Wang, Rui Wang, Feng Xia, and Leyu Lin. 2020. Deep Feedback Network for Recommendation. *Proceedings of IJCAI-PRICAI* (2020).

[45] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. 2014. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM international conference on Web search and data mining*. 283–292.

[46] Quangui Zhang, Longbing Cao, Chengzhang Zhu, Zhiqiang Li, and Jinguang Sun. 2018. Coupledcf: Learning explicit and implicit user-item couplings in recommendation for deep collaborative filtering. In *IJCAI International Joint Conference on Artificial Intelligence*.

[47] Xiangyu Zhao, Liang Zhang, Zhuoye Ding, Long Xia, Jiliang Tang, and Dawei Yin. 2018. Recommendations with negative feedback via pairwise deep reinforcement learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1040–1048.

[48] Chang Zhou, Jinze Bai, Junshuai Song, Xiaofei Liu, Zhengchao Zhao, Xiusi Chen, and Jun Gao. 2018. Atrank: An attention-based user behavior modeling framework for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

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

[50] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1059–1068.