

# Deep Feedback Network for Recommendation

Ruobing Xie\*, Cheng Ling\*, Yalong Wang, Rui Wang, Feng Xia and Leyu Lin

WeChat Search Application Department, Tencent, China

ruobingxie@tencent.com

## Abstract

Both explicit and implicit feedbacks can reflect user opinions on items, which are essential for learning user preferences in recommendation. However, most current recommendation algorithms merely focus on implicit positive feedbacks (e.g., click), ignoring other informative user behaviors. In this paper, we aim to jointly consider explicit/implicit and positive/negative feedbacks to learn user unbiased preferences for recommendation. Specifically, we propose a novel Deep feedback network (DFN) modeling click, unclick and dislike behaviors. DFN has an internal feedback interaction component that captures fine-grained interactions between individual behaviors, and an external feedback interaction component that uses precise but relatively rare feedbacks (click/dislike) to extract useful information from rich but noisy feedbacks (unclick). In experiments, we conduct both offline and online evaluations on a real-world recommendation system WeChat Top Stories used by millions of users. The significant improvements verify the effectiveness and robustness of DFN. The source code is in <https://github.com/qqxiaochongqq/DFN>.

## 1 Introduction

Personalized recommendation systems aim to provide customized items for users according to their preferences. They have been widely used in various fields including video [Covington *et al.*, 2016] and E-commerce [Feng *et al.*, 2019].

There are plenty of recommendation systems personalized with user-item interactions. Such informative signals are categorized into two types, namely the explicit feedback and the implicit feedback [Liu *et al.*, 2010]. The **explicit feedback** comes from **user direct opinions** on items (e.g., star ratings or like/dislike). It could precisely indicate users' real preferences, while it is rather challenging to collect such feedback. In contrast, the **implicit feedback** mainly derives from user behaviors that **imply indirect opinions** (e.g., click or unclick). It is much easier to collect such implicit feedbacks from enormous numbers of user behaviors in real-world recommenda-

tion systems. However, implicit feedbacks struggle with their inherent noises and the natural scarcity of negative feedbacks, which gravely harm the accuracy in learning user's unbiased preferences [Hu *et al.*, 2008].



Figure 1: An example of multiple feedbacks in WeChat Top Stories.

Recently, recommendation systems usually regard the personalized recommendation as a Click-Through Rate (CTR) prediction task. Therefore, it is natural that most recommendation algorithms mainly concentrate on the implicit positive feedbacks such as clicks, which could be easily collected in practice. These models are directly optimized with click behaviors and **CTR-oriented objectives**, which will inevitably result in the following problems. First, CTR-oriented objectives usually **concentrate on what users like, ignoring what users dislike**. Simply relying on these implicit positive feedbacks will make models tend to provide homogeneous and myopic results, which will eventually harm user experiences [Zhao *et al.*, 2018a]. **Therefore, negative feedbacks should be considered in recommendation**. Second, besides passively receiving information chosen by models, users also need effective and efficient feedback mechanisms to actively interact with recommendation systems. Moreover, there are also gaps between **users' implicit feedbacks and their real preferences** (click does not always mean like) [Wang *et al.*, 2018]. It confirms the necessity of **explicit feedbacks** in recommendation.

**Multiple explicit/implicit and positive/negative feedbacks** can complement each other and reflect user unbiased preferences in recommendation. There are some efforts jointly con-

\* indicates equal contribution

sider both explicit and implicit feedbacks with Collaborative filtering [Liu *et al.*, 2010] and multi-task learning [Hadash *et al.*, 2018]. However, the negative feedbacks in these works are usually ignored or only locate in explicit feedbacks, which are precise but rare. Some works consider unclick or missing behaviors as implicit negative feedbacks to multiply negative signals [Zhao *et al.*, 2018b]. Unfortunately, the noises in such implicit negative feedbacks crucially limit the performances, since these implicit negative feedbacks may be caused by various reasons besides dislike [He *et al.*, 2016].

In this paper, we concentrate on improving recommendation performances with different types of explicit/implicit and positive/negative feedbacks. To address the problems in conventional methods, we propose a novel **Deep feedback network (DFN)**, which jointly considers multiple feedbacks and their interactions in deep model based recommendation. Fig 1 shows a brief example of different types of feedbacks used in DFN, including **implicit positive feedback (i.e., click)**, **implicit negative feedback (i.e., unclick)** and **explicit negative feedback (i.e., dislike)**. Specifically, we first conduct transformer over the target item and behaviors separately in each feedback sequence to capture internal behavior-level interactions. Next, we utilize high-quality but relatively rare click and dislike behaviors to denoise rich but noisy unclick behaviors with external feedback-level interactions. These distilled feedback features are combined with other features and then fed into the feature interaction module with Wide, FM and Deep components. The main advantage of DFN is that it successfully combines multiple feedbacks to learn user unbiased positive and negative preferences for recommendation, which solves the dilemma of quality and quantity in feedbacks.

In experiments, we conduct both offline and online evaluations on a well-known recommendation system *WeChat Top Stories* widely used by hundreds of millions of users. We also conduct parameter analyses and ablation tests to show the effectiveness and robustness of our model. The main contributions of DFN are concluded as follows:

- To the best of our knowledge, we are the first to combine **implicit positive feedbacks**, **implicit negative feedbacks**, **explicit negative feedbacks** and their interactions in deep neural recommendation.
- We propose a novel deep feedback network, which creatively uses both internal and external feedback interactions to learn user unbiased preferences. We also jointly consider **multiple feedback losses** in optimization.
- The significant improvements in both offline and online evaluations confirm the effectiveness and robustness of DFN in real-world recommendation systems.

## 2 Related Works

**Recommendation System.** Conventional recommendation algorithms such as Collaborative filtering (CF) [Sarwar *et al.*, 2001] capture the similarities between users and items. Factorization machine (FM) [Rendle, 2010] utilizes factorized parameters to model second-order feature interactions. With the blooming of deep learning, Wide&Deep [Cheng *et al.*, 2016] jointly considers memorization in Wide component

and generalization in Deep component. DeepFM [Guo *et al.*, 2017] improves Wide&Deep by replacing the wide part with FM layer, while NFM [He and Chua, 2017] and AFM [Xiao *et al.*, 2017] combine FM with DNN and attention serially. DCN [Wang *et al.*, 2017] and AutoInt [Song *et al.*, 2019] further consider high order feature interactions. In session-based recommendation, DIN [Zhou *et al.*, 2018], DSIN [Feng *et al.*, 2019] and ATRank [Zhou *et al.*, 2018] conduct attention over different behaviors. BERT4Rec [Sun *et al.*, 2019] also uses BERT for sequence modeling. In this paper, we model multiple feedbacks with transformer and attention, and combine Wide, FM and DNN components for feature interactions.

**Implicit and Explicit Feedbacks.** Both implicit and explicit feedbacks are beneficial in recommendation [Jawaheer *et al.*, 2010]. There are plenty of efforts that jointly consider multiple feedbacks with CF [Koren, 2008; Liu *et al.*, 2010; Zhang *et al.*, 2018], bayesian ranking model [Liu *et al.*, 2017] and weak supervision [Jadidinejad *et al.*, 2019]. Some works aim to conduct feature mapping or transfer learning to build relations between explicit and implicit feedbacks [Pan *et al.*, 2016]. Most algorithms combine explicit and implicit feedbacks in multi-task learning framework [Hadash *et al.*, 2018; Jadidinejad *et al.*, 2019] to jointly solve ranking and rating tasks. In DFN, we use high-quality but relatively rare explicit feedbacks to guild feature extraction in rich but noisy implicit negative feedbacks for CTR and dislike prediction.

**Negative Feedbacks.** Negative feedbacks are essential for modeling user preferences but hard to collect [Jawaheer *et al.*, 2010]. Conventional methods usually regard all missing or unclicked data as negative feedbacks in CF-based models [Hu *et al.*, 2008]. However, it also brings in large numbers of noises, since unclick does not always indicate dislike [Wang *et al.*, 2018]. To distill the real negative signals in implicit feedbacks, some models use exposure variables [Liang *et al.*, 2016] or popularity [He *et al.*, 2016]. Zhao *et al.* [2018b] conducts reinforcement learning with both click and unclick sequences as features. In contrast, explicit negative feedbacks could directly reflect user’s negative opinions [Jawaheer *et al.*, 2010; Zhao *et al.*, 2018a], while their scarcity limits their usage in deep-based models. To the best of our knowledge, we are the first to encode **click, unclick, dislike behaviors** and their interactions into deep neural recommendation, considering negative signals in both implicit and explicit feedbacks.

## 3 Methodology

We aim to jointly consider multiple explicit/implicit and positive/negative feedbacks to learn user unbiased preferences for recommendation. Specifically, we conduct the DFN model on a real-world recommendation system, and collect three types of feedbacks in user historical behaviors as follows:

- **Implicit positive feedbacks.** The implicit positive feedbacks are the most widely-used feedbacks in large-scale recommendation, which are satisfactory in both quantity and quality. Following most conventional models, we consider the **click behavior sequence**  $\{c_1, \dots, c_{n_1}\}$  as the implicit positive feedback used in DFN.
- **Explicit negative feedbacks.** Explicit feedbacks are high-quality but rare in read-world recommendation. We

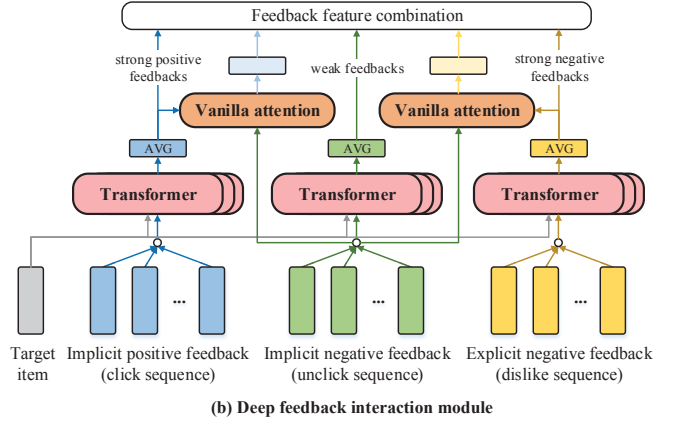
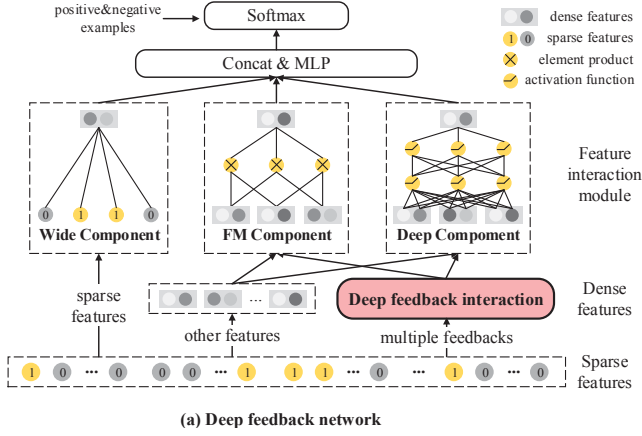


Figure 2: The overall architecture of Deep feedback network and deep feedback interaction module.

use the **dislike** button attached to each item to collect explicit **negative feedback sequence** as  $\{d_1, \dots, d_{n_2}\}$ .

- **Implicit negative feedbacks.** We regard the *impressed but unclick* behavior sequence  $\{u_1, \dots, u_{n_3}\}$  as the implicit negative feedbacks. This unclick behavior is the vast majority of all types of feedbacks, while it seriously struggles with noises and false-negative signals.

DFN attempts to use high-quality click and dislike behaviors as instructors to extract useful information from unclick behaviors. It is also easy to add other feedbacks in DFN.

### 3.1 Overall Architecture

The Deep feedback network mainly consists of two modules, namely the deep feedback interaction module and the feature interaction module. First, the deep feedback interaction module takes multiple feedbacks as inputs to extract user unbiased positive and negative preferences, with the help of internal and external feedback interactions. Second, the refined feedback features are combined with other informative features such as user profiles, item features and recommendation contexts. We implement Wide, FM and Deep components for feature aggregation. Finally, the outputs of feature interaction module are fed into fully connected and Softmax layers for model optimization with both positive and negative losses. Fig 2 (a) illustrates the overall architecture of DFN.

### 3.2 Deep Feedback Interaction Module

The deep feedback interaction module in Fig 2 (b) takes implicit positive (click), explicit negative (dislike) and implicit negative (unclick) feedbacks with target item as inputs. We conduct two components to learn from the interactions inside and between different types of feedbacks.

#### Internal Feedback Interaction Component

This component focuses on the interactions between target item and individual behaviors within a certain type of feedback. We conduct a multi-head self-attention over behaviors following Vaswani *et al.* [2017]. All behavior features consist of their item embeddings and position embeddings, and are projected into a joint semantic space to form the behavior

embeddings. Taking the click behavior for instance, we combine the target item  $t$  with the behavior embeddings of click sequence to form the input matrix  $B_c = \{t, c_1, \dots, c_{n_1}\}$ . The query, key, value matrices are calculated as:

$$Q = W^Q B_c, \quad K = W^K B_c, \quad V = W^V B_c, \quad (1)$$

where  $W^Q, W^K, W^V$  are projection matrices. We then calculate the self-attention as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{Q^\top K}{\sqrt{n_h}}\right)V, \quad (2)$$

where  $n_h$  is the dimension of query, key and value. The  $i$ -th head of the total  $h$  multi-heads is calculated as:

$$\text{head}_i = \text{Attention}(W_i^Q Q, W_i^K K, W_i^V V). \quad (3)$$

$W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{n_h \times n_h/h}$  are weighting matrices for the  $i$ -th head. The final output matrix of self-attention is:

$$F_c = \text{concat}(\text{head}_1, \dots, \text{head}_h) \cdot W^O, \quad (4)$$

$W^O \in \mathbb{R}^{n_h \times n_h}$  is a projection matrix. Finally, we conduct an average pooling over all  $n+1$  output embeddings in  $F_c$  to generate the implicit positive feedback embedding  $f_c$  as:

$$f_c = \text{Average\_pooling}(F_c), \quad f_c \in \mathbb{R}^{n_h}. \quad (5)$$

We also use the same transformer with type-specific hyper-parameters to generate the explicit negative feedback embedding  $f_d$  and the implicit negative feedback embedding  $f_u$  from dislike and unclick behaviors respectively. The internal feedback interaction component well captures behavior-level interactions between target item and behaviors in each type of feedback sequence. It can provide user positive and negative preferences related to the target item.

#### External Feedback Interaction Component

Implicit negative feedbacks are sufficient but extremely noisy. In general, unclick behaviors seem to imply negative signals, while items exposed to users are carefully chosen by certain strategies, which may also contain user interests from coarse-grained aspects. The external feedback interaction component aims to distinguish what users really like and dislike in

unclick behaviors, according to strong feedbacks in click and dislike behaviors. Specifically, we conduct two vanilla attentions, which considers implicit positive and explicit negative feedback embeddings  $\mathbf{f}_c$  and  $\mathbf{f}_d$  as instructors to guild positive and negative preference extractions from unclick sequences  $\{\mathbf{u}_1, \dots, \mathbf{u}_{n_3}\}$ . We formalize the unclick-dislike interaction embedding  $\mathbf{f}_{ud}$  with dislike and unclick behaviors as:

$$\mathbf{f}_{ud} = \sum_{i=1}^{n_3} \alpha_i \mathbf{u}_i, \quad \alpha_i = \frac{f(\mathbf{f}_d, \mathbf{u}_i)}{\sum_{j=1}^{n_3} f(\mathbf{f}_d, \mathbf{u}_j)}, \quad (6)$$

where the weighting score function  $f(\mathbf{a}, \mathbf{b})$  is defined as:

$$f(\mathbf{a}, \mathbf{b}) = \text{MLP}(\text{concat}(\mathbf{a}, \mathbf{b}, \mathbf{a} - \mathbf{b}, \mathbf{a} \odot \mathbf{b})). \quad (7)$$

We regard  $\odot$  as the element-wise product and use a 2-layer Multi-layer perceptron (MLP).  $\mathbf{f}_d$  contains user's strong negative preferences refined from explicit negative feedbacks related to target item. It helps vanilla attention to extract items that users truly dislike in unclick behaviors. We also amplify the positive voices in unclick behaviors with implicit positive feedback embedding  $\mathbf{f}_c$  similarly as follows:

$$\mathbf{f}_{uc} = \sum_{i=1}^{n_3} \beta_i \mathbf{u}_i, \quad \beta_i = \frac{f(\mathbf{f}_c, \mathbf{u}_i)}{\sum_{j=1}^{n_3} f(\mathbf{f}_c, \mathbf{u}_j)}. \quad (8)$$

At last, we combine all five feedback features to generate the final refined feedback feature  $\mathbf{f}_{Feed}$  as follows:

$$\mathbf{f}_{Feed} = \{\mathbf{f}_c, \mathbf{f}_d, \mathbf{f}_u, \mathbf{f}_{uc}, \mathbf{f}_{ud}\}. \quad (9)$$

The implicit positive and explicit negative feedbacks  $\mathbf{f}_c$  and  $\mathbf{f}_d$  are regarded as strong positive and negative signals, while the rest unclick-related feedbacks are regarded as weak signals.

### 3.3 Feature Interaction Module

In feature interaction, we combine the refined feedback feature with other features including user profiles, item features and recommendation contexts. Following Guo *et al.* [2017], we group these sparse features into  $m$  fields  $\{x_1, \dots, x_m\}$  including continuous fields (e.g., age) and categorical fields (e.g., location). All fields are represented as one-hot embeddings. A lookup table is used to generate the dense feature of all fields as  $\{\mathbf{f}_1, \dots, \mathbf{f}_m\}$ . We implement Wide, FM and Deep components for feature interaction.

**Wide Component** The Wide component is a generalized linear model widely used in recommendation [Cheng *et al.*, 2016]. The output of the Wide component  $\mathbf{y}^{Wide}$  is a  $m$ -dimensional vector, where the  $i$ -th element is calculated as:

$$y_i^{Wide} = \mathbf{w}_i^\top \mathbf{x}_i + b_i, \quad \mathbf{w}_i, \mathbf{x}_i \in \mathbb{R}^{n_{fi}}. \quad (10)$$

$\mathbf{w}_i$  is the weighting vector of the  $i$ -th one-hot field embedding  $\mathbf{x}_i$ , and  $b_i$  is the bias.  $n_{fi}$  is the dimension of  $\mathbf{x}_i$ .

**FM Component** The FM component captures the second-order feature interactions between all features. The input embeddings of FM is the combination of all dense features and final refined feedback features as  $\mathbf{F}' = \{\mathbf{f}_1, \dots, \mathbf{f}_m, \mathbf{f}_{Feed}\}$ . We follow the Bi-interaction layer in He and Chua [2017] and generate the output embedding  $\mathbf{y}^{FM}$  as follows:

$$\mathbf{y}^{FM} = \sum_{i=1}^{m+5} \sum_{j=i+1}^{m+5} \mathbf{f}'_i \odot \mathbf{f}'_j, \quad \mathbf{f}'_i, \mathbf{f}'_j \in \mathbf{F}'. \quad (11)$$

**Deep Component** In Deep component, we implement a 2-layer MLP to learn high-order feature interactions. The input is the concatenation of dense features and feedback features represented as  $\mathbf{f}^{(0)} = \text{concat}(\mathbf{f}_1, \dots, \mathbf{f}_m, \mathbf{f}_{Feed})$ . We have:

$$\mathbf{y}^{Deep} = \mathbf{f}^{(2)}, \quad \mathbf{f}^{(i+1)} = \text{ReLU}(\mathbf{W}^{(i)} \mathbf{f}^{(i)} + \mathbf{b}^{(i)}), \quad (12)$$

where  $\mathbf{f}^{(i)}$  is the output embedding of the  $i$ -th layer.  $\mathbf{W}^{(i)}$  is the weighting matrix and  $\mathbf{b}^{(i)}$  is the bias of the  $i$ -th layer.

Finally, we concatenate all outputs from three components to generate the aggregated feature embedding  $\mathbf{y}$  as:

$$\mathbf{y} = \text{concat}(\mathbf{y}^{Wide}, \mathbf{y}^{FM}, \mathbf{y}^{Deep}). \quad (13)$$

### 3.4 Optimization Objective

We utilize click, unclick and dislike behaviors for supervised training. The predicted click probability is calculated with the aggregated feature embedding  $\mathbf{y}$  as follows:

$$p(x) = \sigma(\mathbf{w}_p^\top \mathbf{y}). \quad (14)$$

$\mathbf{w}_p$  is the weighting vector, and  $\sigma(\cdot)$  is the sigmoid function. The loss function of DFN consists of three parts corresponding to click, unclick and dislike behaviors as:

$$L = -\frac{1}{N} (\lambda_c \sum_{S_c} \log p(x) + \lambda_u \sum_{S_u} \log(1 - p(x)) + \lambda_d \sum_{S_d} \log(1 - p(x))). \quad (15)$$

The train set has  $N$  instances grouped into click set  $S_c$ , dislike set  $S_d$  and unclick set  $S_u$ .  $\lambda_c, \lambda_d, \lambda_u$  are weights of different losses to measure the importances of different feedbacks.

## 4 Experiments

### 4.1 Datasets

Since there are few large-scale datasets having click, unclick and dislike behaviors, we build a new dataset MultiFeed from a real-world recommendation system *WeChat Top Stories* after data masking. Precisely, we randomly collect 448 million user behaviors from 20.3 million users on 3.1 million items, considering the behaviors in the first few days as train set and the rest as test set. These user behaviors include implicit positive (click), implicit negative (unclick) and explicit negative (dislike) feedbacks. In test set, MultiFeed has nearly 222 million instances for CTR and dislike prediction, containing 33 million click behaviors and 328 thousand dislike behaviors.

#user	#item	#click	#dislike	#unclick
20.3M	3.10M	66.0M	0.65M	381M

Table 1: Statistics of the MultiFeed dataset.

### 4.2 Competitors and Experimental Settings

**Competitors** We implement eight classical models as baselines for evaluation. All models (DFN and baselines) use the same features *including all feedbacks* for fair comparisons.

- **FM [Rendle, 2010]**. Factorization machine (FM) models second-order feature interactions for CTR prediction. FM is considered as the base model in evaluation.
- **Wide&Deep [Cheng *et al.*, 2016]**. Wide&Deep consists of a Wide part that handles raw features, and a Deep part that extracts high-order feature interactions.
- **NFM [He and Chua, 2017]**. NFM uses a bi-interaction layer before DNN layers for feature interaction.
- **AFM [Xiao *et al.*, 2017]**. AFM brings in attention over feature interactions from the bi-interaction layer.
- **DeepFM [Guo *et al.*, 2017]**. DeepFM replaces the Wide component in Wide&Deep with a FM layer.
- **DCN [Wang *et al.*, 2017]**. DCN captures the bounded-degree feature interactions with its cross network.
- **DIN [Zhou *et al.*, 2018]**. DIN is a classical model for session-based recommendation. It considers the weights of items in user historical behaviors with attention.
- **AutoInt [Song *et al.*, 2019]**. AutoInt introduces self-attentive neural network for feature interactions.

We do not compare with other models like Hadash *et al.* [2018] and Jadidinejad *et al.* [2019], for these models usually rely on customized feedbacks or multi-task learning, which are hard to be adapted to our CTR and dislike prediction tasks.

**Experimental Settings** In DFN, the max length of all three behavior sequences is 30 and the feature field number is 47. The dimension of each feature embeddings  $n_h = 64$ , and the dimension of 2-layer MLP in Deep is 32 and 16. In training, we utilize Adam with the batch size to be 64. The weights of click, unclick and dislike losses  $\lambda_c : \lambda_u : \lambda_d = 1 : 1 : 10$ . We conduct the grid search for parameters. All models follow the same experimental settings for fair comparisons.

### 4.3 CTR Prediction

We first evaluate DFN on the classical CTR prediction to verify its capability in modeling user positive preferences.

**Evaluation Protocol** In CTR prediction task, we utilize a widely-used metric Area Under Curve (AUC) for evaluation. Following Yan *et al.* [2014], we further bring in RelaImpr to measure the relative improvements over base model (i.e., FM in our setting). Since AUC is 0.5 from a random strategy, the RelaImpr in this task is formalized as:

$$\text{RelaImpr} = \frac{\text{AUC}(\text{measured model}) - 0.5}{\text{AUC}(\text{base model}) - 0.5} - 1. \quad (16)$$

We do not use Logloss as evaluation metric, for the loss functions of DFN and other baselines are different.

**Experimental Results** Table 2 shows the results of CTR prediction on MultiFeed, from which we can find that:

(1) DFN significantly outperforms all baselines on AUC and achieves 11.85% relative improvement over base model. We also conduct a significance test to verify that DFN outperforms baselines with the significance level  $\alpha = 0.01$ . Note that all baselines also use multiple feedbacks as features. The impressive improvements over strong baselines indicate that DFN could well capture informative messages in implicit and

model	AUC	RelaImpr
FM [Rendle, 2010]	0.7591	0.00%
Wide&Deep [Cheng <i>et al.</i> , 2016]	0.7728	5.29%
AFM [Xiao <i>et al.</i> , 2017]	0.7601	0.35%
NFM [He and Chua, 2017]	0.7627	1.39%
DeepFM [Guo <i>et al.</i> , 2017]	0.7718	4.90%
DCN [Wang <i>et al.</i> , 2017]	0.7735	5.56%
DIN [Zhou <i>et al.</i> , 2018]	0.7797	7.95%
AutoInt [Song <i>et al.</i> , 2019]	0.7726	5.21%
DFN (ours)	<b>0.7898</b>	<b>11.85%</b>

Table 2: Results of CTR prediction on MultiFeed dataset.

explicit feedbacks, which are essential for modeling user unbiased positive preferences in recommendation.

(2) The advantages of DFN mainly derive from the deep feedback interaction module. First, the internal feedback interaction component successfully captures fine-grained interactions between the target item and individual behaviors with transformer. It could extract user preferences from behavior-level interactions inside different types of feedbacks. Second, the external feedback interaction component uses precise but relatively rare feedbacks to denoise rich but noisy unclick behaviors with vanilla attention. Therefore, DFN can solve the dilemma of quantity and quality. In ablation test, we will give detailed analyses on different components of DFN.

### 4.4 Dislike Prediction

The significant improvements in CTR prediction have shown that DFN could well learn user positive preferences. In this subsection, we further propose a new dislike prediction task to evaluate DFN in modeling user negative preferences.

**Evaluation Protocol** The dislike behavior usually indicates a strong negative signal. A timely feedback mechanism could rapidly capture user’s instant preferences from dislike behaviors and improve user experience. We propose the dislike prediction task, which aims to predict what users dislike in recommended items and avoid disappointing users. Following CTR prediction, we also use AUC and RelaImpr as metrics, regarding  $1 - p(x)$  as the predicted dislike probability.

model	AUC	RelaImpr
FM [Rendle, 2010]	0.6979	0.00%
Wide&Deep [Cheng <i>et al.</i> , 2016]	0.6803	-8.89%
DeepFM [Guo <i>et al.</i> , 2017]	0.6784	-9.85%
DCN [Wang <i>et al.</i> , 2017]	0.6884	-4.80%
NFM [He and Chua, 2017]	0.7042	3.18%
AFM [Xiao <i>et al.</i> , 2017]	0.6988	4.55%
AutoInt [Song <i>et al.</i> , 2019]	0.6761	-11.02%
DIN [Zhou <i>et al.</i> , 2018]	0.7147	8.49%
DIN+ (DIN + dislike loss)	0.7749	38.91%
DFN (ours)	<b>0.8804</b>	<b>92.22%</b>

Table 3: Results of dislike prediction on MultiFeed dataset.

**Experimental Results** Table 3 demonstrates the results of dislike prediction. We can observe that:



(1) DFN achieves the best performance on AUC with the significance level  $\alpha = 0.01$ . It indicates that DFN could learn both user positive and negative preferences and respond timely to explicit negative feedbacks. Note that dislike prediction needs algorithms to make fine-grained discriminations, for all impressed items (including disliked items) are relatively good candidates selected by algorithms. Currently, dislike behavior only accounts for 0.15% of all feedbacks in our system, while it has already shown its power in CTR and dislike predictions. The improvements will be more significant with the mutual promotion of more negative feedbacks and better models.

(2) The impressive improvement comes from two points: (i) DFN considers explicit negative feedbacks in loss function, which directly optimizes the dislike prediction. (ii) The deep feedback interaction module brings in both internal and external feedback interactions, which better extracts informative user unbiased preferences for recommendation.

(3) For fair models comparisons, we further add the dislike loss in DFN to some strong baselines (e.g., DIN+). The results are also improved but still far worse than DFN, which confirms the power of both dislike loss and the feedback interaction module of DFN. Moreover, we find that the relative performances of baselines on CTR and dislike predictions are different. It is natural since they do not specifically optimize dislike loss, and thus are unstable in dislike prediction.

#### 4.5 Ablation Tests

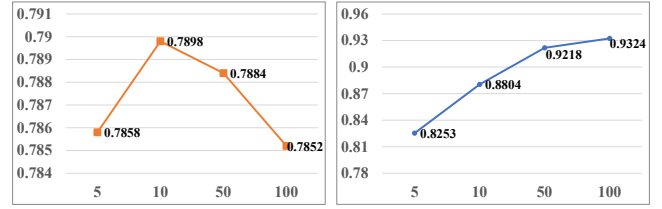
In Table 4, we conduct an ablation test to show the effectiveness and necessity of different components in deep feedback interaction module. We observe that: (1) DFN (click) performs better than DFN (w/o feedbacks), which confirms the significance of click behaviors. (2) The significant improvement from DFN (click) to DFN (internal) also verifies that unclick and dislike behaviors could provide complementary information that helps to learn user unbiased preferences. (3) Comparing with DFN (internal) and DFN (All), we can find that the external feedback interaction still makes a notable improvement, which confirms that the external feedback interaction component is beneficial in DFN.

model	AUC	RelaImpr
DFN (w/o feedbacks)	0.7742	5.75%
DFN (click)	0.7824	8.84%
DFN (internal)	0.7879	10.77%
DFN (All)	<b>0.7898</b>	<b>11.85%</b>

Table 4: Ablation tests for DFN on CTR prediction.

#### 4.6 Online A/B Test

**Online System and Evaluation Protocol** We conduct an online A/B test to evaluate DFN on *WeChat Top Stories* used by millions of users. The compared baseline is DIN with other online modules unchanged. We use four evaluation metrics including CTR, list-wise CTR (LCTR), average using time (AUT) and dislike-through rate (DTR). We conduct the A/B test with nearly 870 thousand users, and report the improvements instead of detailed values in Table 5.



(a) AUC of CTR prediction (b) AUC of dislike prediction

Figure 3: Analysis on different weights of dislike loss  $\lambda_d$ .

**Experimental Results** We find that: (1) DFN achieves consistent improvements on CTR and LCTR metrics over DIN, which confirms that DFN performs well in real-world CTR prediction. The improvement of AUT also implies that users are willing to spend more time using our system, since DFN could provide better recommended items. (2) The significant improvement in DTR shows that DFN is capable of modeling user negative preferences in recommendation, which is essential for improving user experience in practice.

model	CTR	LCTR	AUT	DTR
DFN	+1.17%	+0.65%	+0.52%	-33.17%

Table 5: Online A/B tests on WeChat Top Stories.

#### 4.7 Parameter Analysis

We further conduct a parameter analysis on different weights of dislike loss  $\lambda_d$  in Eq. (15) to measure the impact of dislike loss function on both CTR and dislike prediction. In Fig. 3, we evaluate DFN with different  $\lambda_d$  on these two tasks. We find that: (1) in CTR prediction, DFN achieves the best performance when  $\lambda_d = 10$ . The performance will get worse if  $\lambda_d$  is set too low or too high. It indicates that dislike feedbacks are useful not only as features, but also as loss function, while too many weights on dislike feedbacks will harm the learning of user positive preferences. (2) As  $\lambda_d$  grows bigger, the performance of dislike prediction also becomes better, while the AUC growth will gradually slow down when  $\lambda_d$  gets too high. It is natural that the dislike loss function could benefit dislike prediction, which has also been verified in Sec. 4.4. However, the performance growth is not endless, which confirms the importance of balancing positive and negative feedbacks. In experiments, we choose  $\lambda_d = 10$  to jointly consider both CTR (we concern more about) and dislike prediction tasks.

### 5 Conclusion and Future Work

In this paper, we propose a Deep feedback network (DFN), which considers both explicit/implicit and positive/negative feedbacks to learn user unbiased preferences. DFN uses internal behavior-level and external feedback-level interactions in multiple feedbacks. The significant improvements in offline and online verify the effectiveness and robustness of DFN. In future, we will use more sophisticated ranking models for feature interactions. Moreover, we will explore other explicit feedbacks to improve recommendation interpretability.

## References

- [Cheng *et al.*, 2016] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, 2016.
- [Covington *et al.*, 2016] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of RecSys*, 2016.
- [Feng *et al.*, 2019] Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. Deep session interest network for click-through rate prediction. In *Proceedings of IJCAI*, 2019.
- [Guo *et al.*, 2017] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: a factorization-machine based neural network for ctr prediction. In *Proceedings of IJCAI*, 2017.
- [Hadash *et al.*, 2018] Guy Hadash, Oren Sar Shalom, and Rita Osadchy. Rank and rate: multi-task learning for recommender systems. In *Proceedings of RecSys*, 2018.
- [He and Chua, 2017] Xiangnan He and Tat-Seng Chua. Neural factorization machines for sparse predictive analytics. In *Proceedings of SIGIR*, 2017.
- [He *et al.*, 2016] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of SIGIR*, 2016.
- [Hu *et al.*, 2008] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Proceedings of ICDM*, 2008.
- [Jadidinejad *et al.*, 2019] Amir H Jadidinejad, Craig Macdonald, and Iadh Ounis. Unifying explicit and implicit feedback for rating prediction and ranking recommendation tasks. In *Proceedings of ICTIR*, 2019.
- [Jawaheer *et al.*, 2010] Gawesh Jawaheer, Martin Szomszor, and Patty Kostkova. Comparison of implicit and explicit feedback from an online music recommendation service. In *proceedings of HetRec*, 2010.
- [Koren, 2008] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of KDD*, 2008.
- [Liang *et al.*, 2016] Dawen Liang, Laurent Charlin, James McInerney, and David M Blei. Modeling user exposure in recommendation. In *Proceedings of WWW*, 2016.
- [Liu *et al.*, 2010] Nathan N Liu, Evan W Xiang, Min Zhao, and Qiang Yang. Unifying explicit and implicit feedback for collaborative filtering. In *Proceedings of CIKM*, 2010.
- [Liu *et al.*, 2017] Jian Liu, Chuan Shi, Binbin Hu, Shenghua Liu, and S Yu Philip. Personalized ranking recommendation via integrating multiple feedbacks. In *PAKDD*, 2017.
- [Pan *et al.*, 2016] WeiKe Pan, Shanchuan Xia, Zhuode Liu, Xiaogang Peng, and Zhong Ming. Mixed factorization for collaborative recommendation with heterogeneous explicit feedbacks. *Information Sciences*, 2016.
- [Rendle, 2010] Steffen Rendle. Factorization machines. In *Proceedings of ICDM*, 2010.
- [Sarwar *et al.*, 2001] Badrul Munir Sarwar, George Karypis, Joseph A Konstan, John Riedl, et al. Item-based collaborative filtering recommendation algorithms. In *WWW*, 2001.
- [Song *et al.*, 2019] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of CIKM*, 2019.
- [Sun *et al.*, 2019] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of CIKM*, 2019.
- [Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of NIPS*, 2017.
- [Wang *et al.*, 2017] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. Deep & cross network for ad click predictions. In *Proceedings of ADKDD*, 2017.
- [Wang *et al.*, 2018] Menghan Wang, Mingming Gong, Xiaolin Zheng, and Kun Zhang. Modeling dynamic missingness of implicit feedback for recommendation. In *Proceedings of NIPS*, 2018.
- [Xiao *et al.*, 2017] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. Attentional factorization machines: Learning the weight of feature interactions via attention networks. In *IJCAI*, 2017.
- [Yan *et al.*, 2014] Ling Yan, Wu-Jun Li, Gui-Rong Xue, and Dingyi Han. Coupled group lasso for web-scale ctr prediction in display advertising. In *Proceedings of ICML*, 2014.
- [Zhang *et al.*, 2018] Quanguai Zhang, Longbing Cao, Chengzhang Zhu, Zhiqiang Li, and Jinguang Sun. Coupledcf: Learning explicit and implicit user-item couplings in recommendation for deep collaborative filtering. In *Proceedings of IJCAI*, 2018.
- [Zhao *et al.*, 2018a] Qian Zhao, F Maxwell Harper, Gediminas Adomavicius, and Joseph A Konstan. Explicit or implicit feedback? engagement or satisfaction?: a field experiment on machine-learning-based recommender systems. In *Proceedings of SAC*, 2018.
- [Zhao *et al.*, 2018b] Xiangyu Zhao, Liang Zhang, Zhuoye Ding, Long Xia, Jiliang Tang, and Dawei Yin. Recommendations with negative feedback via pairwise deep reinforcement learning. In *Proceedings of KDD*, 2018.
- [Zhou *et al.*, 2018] Chang Zhou, Jinze Bai, Junshuai Song, Xiaofei Liu, Zhengchao Zhao, Xiushi Chen, and Jun Gao. Atrank: An attention-based user behavior modeling framework for recommendation. In *Proceedings of AAAI*, 2018.