Dual-interest Factorization-heads Attention for Sequential Recommendation

Guanyu Lin
Department of Electronic Engineering, Beijing National Research Center for Information Science and Technology, Tsinghua University

Chen Gao[*]
Department of Electronic Engineering, Beijing National Research Center for Information Science and Technology, Tsinghua University

Yu Zheng
Department of Electronic Engineering, Beijing National Research Center for Information Science and Technology, Tsinghua University

Jianxin Chang
Beijing Kuaishou Technology Co., Ltd.

Yanan Niu
Beijing Kuaishou Technology Co., Ltd.

Yang Song
Beijing Kuaishou Technology Co., Ltd.

Zhiheng Li
International Graduate School, Tsinghua University

Depeng Jin
Department of Electronic Engineering, Beijing National Research Center for Information Science and Technology, Tsinghua University

Yong Li
Department of Electronic Engineering, Beijing National Research Center for Information Science and Technology, Tsinghua University

ABSTRACT

Accurate user interest modeling is vital for recommendation scenarios. One of the effective solutions is the sequential recommendation that relies on click behaviors, but this is not elegant in the video feed recommendation where users are passive in receiving the streaming contents and return skip or no-skip behaviors. Here skip and no-skip behaviors can be treated as negative and positive feedback, respectively. With the mixture of positive and negative feedback, it is challenging to capture the transition pattern of behavioral sequence. To do so, FeedRec has exploited a shared vanilla Transformer, which may be inelegant because head interaction of multi-heads attention does not consider different types of feedback. In this paper, we propose **D**ual-interest **F**actorization-heads **A**ttention for Sequential **R**ecommendation (short for DFAR) consisting of feedback-aware encoding layer, dual-interest disentangling layer and prediction layer. In the feedback-aware encoding layer, we first suppose each head of multi-heads attention can capture specific feedback relations. Then we further propose factorization-heads attention which can mask specific head interaction and inject feedback information so as to factorize the relation between different types of feedback. Additionally, we propose a dual-interest disentangling layer to decouple positive and negative interests before performing disentanglement on their representations. Finally, we evolve the positive and negative interests by corresponding towers whose outputs are contrastive by BPR loss. Experiments on two real-world datasets show the superiority of our proposed method against state-of-the-art baselines. Further ablation study and visualization also sustain its effectiveness. We release the source code here: https://github.com/tsinghua-fib-lab/WWW2023-DFAR.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Computing methodologies** → *Neural networks.*

KEYWORDS

Sequential recommendation, User feedback, Contrastive Learning

ACM Reference Format:
Lin, et al. 2023. Dual-interest Factorization-heads Attention for Sequential Recommendation. In *Proceedings of the ACM Web Conference 2023 (WWW '23), April 30–May 04, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3543507.3583278

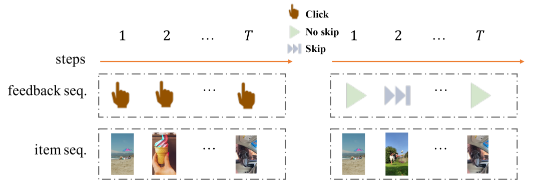
1 INTRODUCTION

Online sequential recommendation [32] has achieved great success for its time-aware personalized modeling and has been widely applied in Web platforms, including micro-video, news, e-commerce, etc. Especially in today's video feed recommendation, users are attracted immensely by video streaming which can be treated as a sequence of items. Formally speaking, the sequential recommendation is defined as predicting the next interacted item by calculating the matching probability between historical items and the target item. As shown in Figure 1 (a), existing sequential recommendation models often exploit click behaviors of users to infer their dynamic interests [11, 14, 31, 41, 42], the optimization of which samples un-clicked items as negative feedback. However, such an approach only inputs positive items into the sequential model, and negative items are sampled as target items, ignoring the transition pattern between historical positive and negative items.

In the video feed recommendation where a single item is exposed each time, users either skip or do not skip the recommended items,

[*] Chen Gao is the corresponding author (chgao96@gmail.com).

WWW '23, April 30–May 04, 2023, Austin, TX, USA
© 2023 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9416-1/23/04.
https://doi.org/10.1145/3543507.3583278



(a) Click sequence (b) Skip and no skip sequence

Figure 1: Illustration of click-based sequential recommendation and our dual-interest sequential recommendation which is hybrid with positive and negative feedback.

as illustrated in Figure 1 (b). Skip can be treated as a kind of negative feedback which means users don't want to receive certain items, while no-skip can be treated as a kind of positive feedback. That is to say, users are passive in receiving the recommended items without providing active click behaviors [10, 18, 22] in such video feed recommendations. However, the existing click-based sequential recommendation does not consider the transition pattern between positive and negative items. Indeed, there are two key challenges when modeling positive and negative feedback in one sequence.

- **Complex transition between positive and negative feedback.** The transition pattern among interacted items has become far more complex due to negative feedback. A user may provide negative feedback only because she has consumed a very similar item before, which makes accurate modeling of transition essential and challenging.
- **Mixed interest in one behavioral sequence.** The negative feedback in the behavioral sequence brings significant challenges to interest learning. The traditional methods of sequential recommendation always conduct a pooling operation on user sequence to obtain the users' current interest, which will fail when the sequence is hybrid with positive and negative signals.

To address the above challenges, in this work, we propose a model named **D**ual-interest **F**actorization-heads **A**ttention for Sequential **R**ecommendation (short for DFAR), further extracting the transition pattern and pair-wise relation between positive and negative interests. To address the first challenge, in the feedback-aware encoding layer, we assume each head of multi-head attention [28] tends to capture specific relations of certain feedback [30]. As different heads of multi-head attention [28] are independent, it may fail to capture the transition pattern between different feedback when positive feedback and negative feedback are indeed not independent of each other. Thus we exploit talking-heads attention [25] to implicitly extract the transition pattern between positive and negative historical items. However, talking-heads attention may mix different heads too much without sufficient prior knowledge. To explicitly extract the transition pattern between positive and negative historical items, we further propose feedback-aware factorization-heads attention which can even incorporate the feedback information into the head interaction. To address the second challenge, we propose a dual-interest disentangling layer and prediction layer, respectively, to disentangle and extract the pair-wise relation between positive and negative interests. Specifically, we first mask and encode the sequence hybrid with positive feedback and negative feedback into two single interest representations before performing disentanglement on them to repel the dissimilar interests. Then we perform a prediction of each interest with the corresponding positive or negative tower and apply contrastive loss on them to extract their pair-wise relation.

In general, we make the following contributions in this work.

- We have taken the pioneering step of fully considering the modeling of negative feedback, along with its impact on transition patterns, to enhance sequential recommendation.
- We propose a feedback-aware encoding layer to capture the transition pattern, dual-interest disentangling layer and prediction layer to perform disentanglement and capture the pair-wise relation between positive and negative historical items.
- We conduct experiments on one benchmark dataset and one collected industrial dataset, where the results show the superiority of our proposed method. A further ablation study also sustains the effectiveness of our three components.

2 PROBLEM FORMULATION

Click-based Sequential Recommendation. Given item sequence $\mathcal{I}_u = (i_1, i_2, \ldots, i_t)$ with only positive feedback, the goal of traditional click-based sequential recommendation is accurately predicting the probability that **given user** u will click the target item *i.e.*, i_{t+1}. The traditional click-based sequential recommendation can be formulated as follows.

Input: Item sequence $\mathcal{I}_u = (i_1, i_2, \ldots, i_t)$ with only positive feedback for a **given user** u.

Output: The predicted score that the **given user** u will click the target item i_{t+1}.

Dual-interest Sequential Recommendation. Given item sequence $\mathcal{I}_u = (i_1, i_2, \ldots, i_t)$ with both positive and negative feedback, the dual-interest sequential recommendation aims to better predict the probability that **given user** u will skip or not skip the target item *i.e.*, i_{t+1}. The dual-interest sequential recommendation with both positive and negative feedback can be formulated as follows.

Input: Item sequence $\mathcal{I}_u = (i_1, i_2, \ldots, i_t)$ with positive and negative feedbacks for a **given user** u.

Output: The predicted score that the **given user** u will skip or do not skip the target item i_{t+1}.

3 METHODOLOGY

Our model captures the relation between positive feedback and negative feedback at the transition level and interest level of sequential recommendation, respectively, by the proposed Feedback-aware Encoding Layer, Dual-interest Disentangling Layer and Prediction Layer, as shown in Figure 2.

- **Feedback-aware Encoding Layer**. We build item embeddings by item IDs and label embeddings by item feedback and further propose feedback-aware factorization-heads attention to capture the transition pattern between different feedback.
- **Dual-interest Disentangling Layer**. We mask the sequence hybrid with both positive and negative feedback into two sequences with solely positive or negative feedback. After encoding two split sequences with independent factorization-heads attention to extract the positive and negative interests, we then disentangle them to repel the dissimilar interests.
- **Dual-interest Prediction Layer**. We further extract the positive and negative interests with independent towers and then perform contrastive loss on them to extract the pair-wise relation.

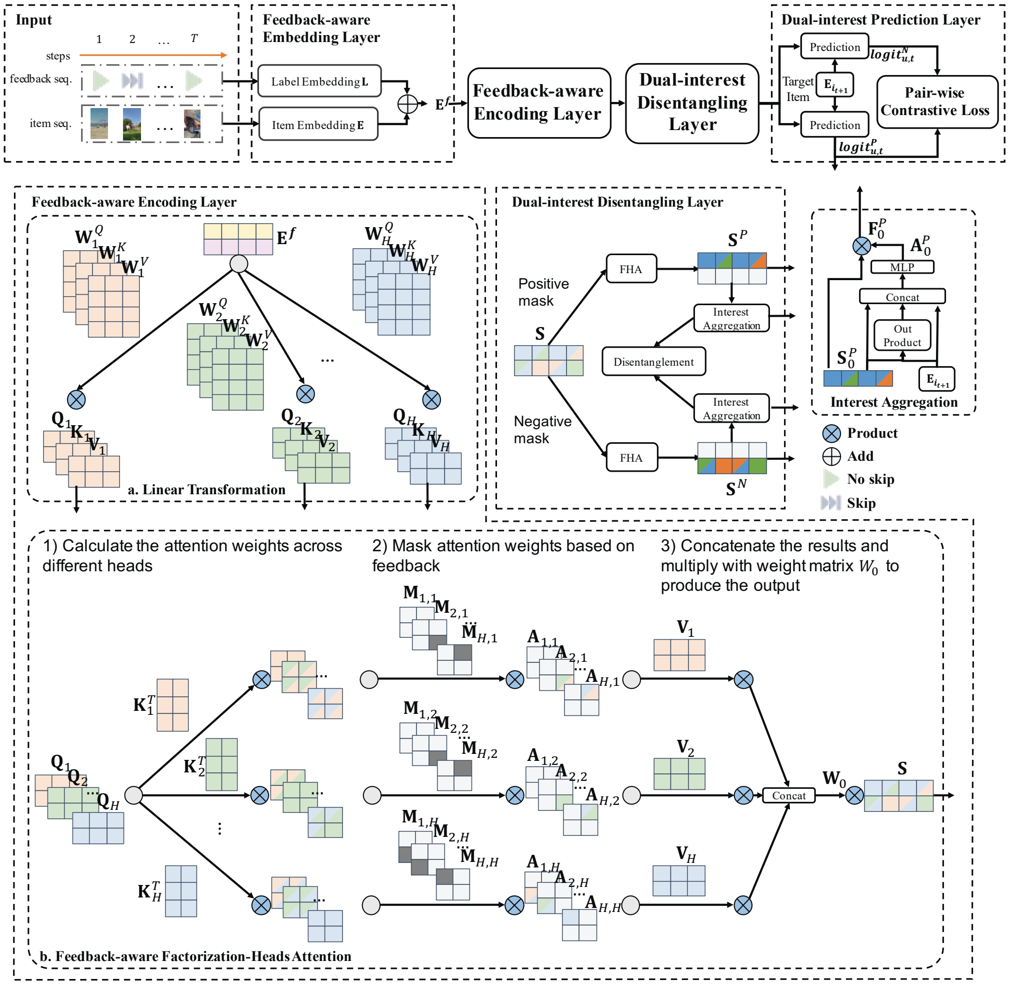

Input
steps
1 2 … T
feedback seq.
item seq.
Feedback-aware Embedding Layer
Label Embedding L
Item Embedding E
E^f
Feedback-aware Encoding Layer
Dual-interest Disentangling Layer
Dual-interest Prediction Layer
Prediction
$logit^N_{u,t}$
Target Item
$E_{i_{t+1}}$
Prediction
$logit^P_{u,t}$
Pair-wise Contrastive Loss
Feedback-aware Encoding Layer
W^Q_1 W^K_1 W^V_1
E^f
W^Q_H W^K_H W^V_H
W^Q_2 W^K_2 W^V_2
Q_1 K_1 V_1
Q_2 K_2 V_2
Q_H K_H V_H
a. Linear Transformation
Dual-interest Disentangling Layer
Positive mask
S
Negative mask
FHA
S^P
Interest Aggregation
Disentanglement
Interest Aggregation
FHA
S^N
F^P_0
A^P_0
MLP
Concat
Out Product
S^P_0
$E_{i_{t+1}}$
Interest Aggregation
Product
Add
No skip
Skip
1) Calculate the attention weights across different heads
2) Mask attention weights based on feedback
3) Concatenate the results and multiply with weight matrix W_0 to produce the output
K^T_1
K^T_2
K^T_H
Q_1 Q_2 Q_H
$M_{1,1}$ $M_{2,1}$ $M_{H,1}$
$M_{1,2}$ $M_{2,2}$ $M_{H,2}$
$M_{1,H}$ $M_{2,H}$ $M_{H,H}$
$A_{1,1}$ $A_{2,1}$ $A_{H,1}$
$A_{1,2}$ $A_{2,2}$ $A_{H,2}$
$A_{1,H}$ $A_{2,H}$ $A_{H,H}$
V_1
V_2
V_H
Concat
W_0
S
b. Feedback-aware Factorization-Heads Attention

Figure 2: Illustration of DFAR. (*i*) Feedback-aware Encoding Layer is linked after the Feedback-aware Embedding Layer where each historical item is injected with a label embedding according to the corresponding feedback; It consists of linear transformation and feedback-aware factorization-heads attention. In the linear transformation, input embeddings are transformed into query, key and value matrices. In feedback-aware factorization-heads attention, the transition relation between different items is factorized into different heads which are masked according to the positive or negative feedback. (*ii*) Dual-interest Disentangling Layer decouples positive and negative interests and performs disentanglement to repel the dissimilar representations of different feedback; (*iii*) Dual-interest Prediction Layer evolves positive and negative interests with corresponding towers and perform BPR loss to capture the pair-wise relation.

3.1 Feedback-aware Encoding Layer

In the feedback-aware encoding layer, we first inject each historical item embedding with corresponding feedback embeddings to incorporate the feedback information into each historical item embedding. Then we further propose talking-heads attention and feedback-aware factorization-heads attention to capture the transition pattern between positive and negative historical items.

3.1.1 *Feedback-aware Embedding Layer.*

To fully distinguish positive and negative feedback, we build a label embedding matrix $\mathbf{L} \in \mathbb{R}^{2\times D}$, besides the item embedding matrix $\mathbf{E} \in \mathbb{R}^{m\times D}$. Here m denotes the number of items, and D is the dimensionality for the hidden state. Then we inject the feedback information into the item embedding and obtain the feedback-aware input embeddings as the model input. Therefore, given item sequence $\mathcal{I}_u = (i_1, i_2, \ldots, i_t)$, we can obtain the feedback-aware item embeddings $\mathbf{E}^f \in \mathbb{R}^{T\times D}$ as:

$$\mathbf{E}^f = [\mathbf{E}_{i_1}, \mathbf{E}_{i_2}, \ldots, \mathbf{E}_{i_t}] + [\mathbf{L}_{y_{u,i_1}}, \mathbf{L}_{y_{u,i_2}}, \ldots, \mathbf{L}_{y_{u,i_t}}], \tag{1}$$

where $\{y_{u,i_1}, y_{u,i_2}, \cdots, y_{u,i_t}\}$ are feedback of items $\{i_1, i_2, \cdots, i_t\}$. Here $y_{u,i_1} = 1$ if i_1 is the no-skip item, and $y_{u,i_1} = 0$ if i_1 is the skip item. Note that if the sequence length is less than t, we can pad $\mathbf{E}^f$ with zero embedding [14].

3.1.2 *Talking-Heads Attention.*

After obtaining the input embeddings for positive and negative historical items, we then capture the transition pattern between them. The existing work, FeedRec [37], exploits vanilla Transformer to roughly capture this transition pattern, of which multi-head attention [14] is the essential part, having the following equation:

$$\mathbf{S} = \mathrm{MHA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \left[\mathbf{A}^{MHA}_1 \mathbf{V}_1, \ldots, \mathbf{A}^{MHA}_H \mathbf{V}_H\right] \mathbf{W}_0, \tag{2}$$

$$\mathbf{A}_h^{MHA} = \text{softmax}\left(\frac{\mathbf{Q}_h{\mathbf{K}_h}^T}{\sqrt{d}}\right), \tag{3}$$

$$\mathbf{Q}_h = \mathbf{Q}\mathbf{W}_h^Q, \mathbf{K}_h = \mathbf{K}\mathbf{W}_h^K, \mathbf{V}_h = \mathbf{V}\mathbf{W}_h^V, \tag{4}$$

where $h \in \{1, 2, \cdots, H\}$ is the number of heads. $\mathbf{W}_0 \in \mathbb{R}^{HD \times D}$ and $\mathbf{W}_h^Q, \mathbf{W}_h^K, \mathbf{W}_h^V \in \mathbb{R}^{D \times D}$ are parameters to be learned. MHA means multi-heads attention [28]. However, different heads of multi-head attention are independent of each other, sharing no information across heads. If assuming different heads capture specific relations between different feedback, then this means there is no information sharing across different feedback. Thus we first propose talking-heads attention [25] to address this issue as below.

$$\mathbf{S} = \text{THA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \left[\mathbf{A}_1^{THA}\mathbf{V}_1, \ldots, \mathbf{A}_H^{THA}\mathbf{V}_H\right]\mathbf{W}_0, \tag{5}$$

$$\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_{H'} \end{bmatrix} = \mathbf{W}_{THA} \begin{bmatrix} \frac{\mathbf{Q}_1{\mathbf{K}_1}^T}{\sqrt{d}} \\ \frac{\mathbf{Q}_2{\mathbf{K}_2}^T}{\sqrt{d}} \\ \vdots \\ \frac{\mathbf{Q}_H{\mathbf{K}_H}^T}{\sqrt{d}} \end{bmatrix}, \tag{6}$$

$$\begin{bmatrix} \mathbf{A}_1^{THA} \\ \mathbf{A}_2^{THA} \\ \vdots \\ \mathbf{A}_H^{THA} \end{bmatrix} = \mathbf{W}_{THA}^S \begin{bmatrix} \text{softmax}\left(\mathbf{A}_1\right) \\ \text{softmax}\left(\mathbf{A}_2\right) \\ \vdots \\ \text{softmax}\left(\mathbf{A}_{H'}\right) \end{bmatrix}, \tag{7}$$

where $\mathbf{W}_{THA} \in \mathbb{R}^{H' \times H}$, $\mathbf{W}_{THA}^S \in \mathbb{R}^{H \times H'}$ and $\mathbf{W}_0 \in \mathbb{R}^{HD \times D}$ are parameters to be learned. Here THA refers to talking-heads attention. However, the interaction between different heads in talking-heads attention is implicit, which may confuse the task for each head and result in overfitting. Not to mention, the two additional linear transformations (i.e. Eq.(6) and Eq.(7)) of talking-heads attention will increase the computation cost.

3.1.3 *Feedback-aware Factorization-heads Attention*.

In this part, we factorize the interaction between positive and negative feedback. Traditional multi-heads attention assigns similar items with higher attention weights. However, in our problem with both positive and negative feedback, two similar items may have different attention weights due to the feedback they have. For example, an NBA fan skips the recommended video about basketball when he/she has watched a lot of basketball videos. But he/she engages in the video about basketball when he/she only has watched a few videos about basketball. In the first case we should repel the representations between historical basketball videos and target basketball videos, while in the second case we should attract them. That is to say, it is necessary to inject the user's feedback into the transition pattern between different feedback. Here we suppose different heads can represent different transition patterns for different feedback [30]. To explicitly factorize interaction across different heads, we further propose factorization-heads attention as:

$$\mathbf{S} = \text{FHA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \left[\mathbf{A}_{1,1}^{FHA}\mathbf{V}_1, \ldots, \mathbf{A}_{H,H}^{FHA}\mathbf{V}_H\right]\mathbf{W}_0, \tag{8}$$

$$\mathbf{A}_{h_1,h_2}^{FHA} = \text{softmax}\left(\frac{\mathbf{Q}_{h_1}{\mathbf{K}_{h_2}}^T}{\sqrt{d}}\right), \tag{9}$$

where $h_1, h_2 \in \{1, 2, \cdots, H\}$. $\mathbf{W}_0 \in \mathbb{R}^{HD \times D}$ are parameters to be learned. Here FHA is our proposed factorization-heads attention.

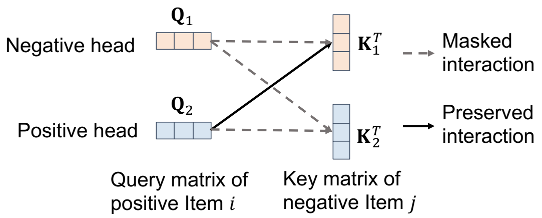


Figure 3: Illustration of label mask $\mathbf{M}_{h_1,h_2}$ on head interaction. Here we show the comprehensible case with two heads, where the first half of heads, i.e. head 1, represents negative head and second half of heads, i.e. head 2, represents positive head.

The factorization-heads attention can represent $H \times H$ relations by H heads. That is to say, our factorization-heads attention can reduce $\sqrt{H}$ times parameters if we want to represent H head interaction relations like talking-heads attention or multi-heads attention. Besides, to further inject the prior feedback knowledge into the factorization-heads attention, we propose feedback-aware factorization-heads attention with a label mask $\mathbf{M}_{h_1,h_2} \in \{0, 1\}^{t \times t}$ as:

$$\mathbf{S} = \text{FFHA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \left[\mathbf{A}_{1,1}^{FFHA}\mathbf{V}_1, \ldots, \mathbf{A}_{H,H}^{FFHA}\mathbf{V}_H\right]\mathbf{W}_0, \tag{10}$$

$$\mathbf{A}_{h_1,h_2}^{FFHA} = \text{softmax}\left(\mathbf{M}_{h_1,h_2}\frac{\mathbf{Q}_{h_1}{\mathbf{K}_{h_2}}^T}{\sqrt{d}}\right), \tag{11}$$

where $\mathbf{M}_{h_1,h_2,i,j} = 1$, if $h_1 \in \{\frac{y_{u,i}H}{2}+1, \frac{y_{u,i}H}{2}+2, \cdots, \frac{(y_{u,i}+1)H}{2}\}$, $h_2 \in \{\frac{y_{u,j}H}{2}+1, \frac{y_{u,j}H}{2}+2, \cdots, \frac{(y_{u,j}+1)H}{2}\}$, $i \in \{1, 2, \cdots, t\}$, $j \in \{1, 2, \cdots, t\}$ and $\mathbf{M}_{h_1,h_2,i,j} = 0$, otherwise. Here the first half of heads w.r.t. $\{1, 2, \cdots, \frac{H}{2}\}$ represent negative heads and second half of heads w.r.t. $\{\frac{H}{2}+1, \frac{H}{2}+2, \cdots, H\}$ represent positive heads. For example, as shown in Figure 3, if item i is positive and item j is negative (i.e., $y_{u,i} = 1$ and $y_{u,j} = 0$), h_1 in positive half and h_2 in negative half will be preserved, i.e., $\mathbf{M}_{2,1,i,j} = 1$, and $\mathbf{M}_{1,1,i,j}, \mathbf{M}_{1,2,i,j}, \mathbf{M}_{2,2,i,j} = 0$.

Besides, FFHA is our proposed feedback-aware factorization-heads attention. Apart from the advantage of explicit interaction between different heads, unlike talking-heads attention, our factorization-heads attention also improves the multi-heads attention without high computation cost. We feed the input embedding into the feedback-aware factorization attention module as:

$$\mathbf{S} = \text{FFHA}(\mathbf{E}^f, \mathbf{E}^f, \mathbf{E}^f), \tag{12}$$

where S are the obtained feedback-aware sequential representations. We put the pseudocode of FHA at Appendix A.1 and compare its complexity with MHA and THA at Appendix A.1.5.

3.2 Dual-interest Disentangling Layer

Though feedback-aware factorization-heads attention has factorized the transition relation between positive feedback and negative feedback, their interest-level relations require further extracting. In this part, we decouple the positive and negative interests and then perform disentanglement on them to repel the dissimilar interests.

3.2.1 *Dual-interest Decoupling Attention*.

After capturing the transition pattern between positive feedback and negative feedback, we then filter out each feedback by a corresponding feedback mask as follows,

$$\mathrm{S}^P = [\mathbf{S}_{i_1}, \mathbf{S}_{i_2}, \ldots, \mathbf{S}_{i_t}] * [y_{u,i_1}, y_{u,i_2}, \ldots, y_{u,i_t}],$$
$$\mathrm{S}^N = [\mathbf{S}_{i_1}, \mathbf{S}_{i_2}, \ldots, \mathbf{S}_{i_t}] * (1 - [y_{u,i_1}, y_{u,i_2}, \ldots, y_{u,i_t}]), \tag{13}$$

which are then fed into the corresponding factorization-heads attention modules to enhance the transition pattern learning for each feedback as:

$$\mathrm{S}^P = \mathrm{FHA}(\mathrm{S}^P, \mathrm{S}^P, \mathrm{S}^P), \mathrm{S}^N = \mathrm{FHA}(\mathrm{S}^N, \mathrm{S}^N, \mathrm{S}^N), \tag{14}$$

where S^P (or S^N) are the single-feedback sequential representations for positive feedback (or negative feedback). In the subsequent section, we will exploit these filtered representations to further extract the interest-level relations.

3.2.2 ***Dual-interest Aggregation and Disentanglement.*** The positive and negative interests of a given user should be distinguished from each other. Hence we aim to repel the positive and negative representations of corresponding interests. Specifically, we assume the target item is possibly either positive or negative. Then we assign the target item with positive and negative label embeddings, respectively, in positive and negative assumed cases. To calculate the attention scores of positive and negative historical items, we fuse them with the target item in assumed positive and negative cases as below.

$$\mathbf{A}^P = \mathrm{MLP}\left((\mathbf{E}_{i_{t+1}} + \mathbf{L}_1)\|\mathrm{S}^P\right), \mathbf{A}^N = \mathrm{MLP}\left((\mathbf{E}_{i_{t+1}} + \mathbf{L}_0)\|\mathrm{S}^N\right), \tag{15}$$

where $\mathbf{A}^P$ and $\mathbf{A}^N \in \mathbb{R}^{t \times D}$ are the positive and negative attention scores. MLP is the multi-layer perceptron. Here $\mathbf{L}_1$ and $\mathbf{L}_0$ are the label embeddings for positive and negative feedback, respectively. With the calculated attention scores by (15), we can then obtain the single-feedback aggregated representations for positive and negative items, respectively, as,

$$\mathrm{F}^P = \mathbf{softmax}\left(\mathrm{A}^P\right)\mathrm{S}^P, \mathrm{F}^N = \mathbf{softmax}\left(\mathrm{A}^N\right)\mathrm{S}^N, \tag{16}$$

$$\mathbf{f}^P = \sum_{j=1}^{t} \mathbf{F}_j^P, \mathbf{f}^N = \sum_{j=1}^{t} \mathbf{F}_j^N, \tag{17}$$

which are then further disentangled with cosine distance as:

$$\mathcal{L}^D = \frac{\mathbf{f}^P \cdot \mathbf{f}^N}{\left\|\mathbf{f}^P\right\| \times \left\|\mathbf{f}^N\right\|}. \tag{18}$$

where $\| \cdot \|$ is the L2-norm. By this disentangling loss, we can repel the aggregated positive and negative representations so as to capture the dissimilar characteristics between them.

3.3 Dual-interest Prediction Layer

In this section, we predict the next item of different interests by positive and negative towers. Finally, we further perform contrastive loss on the outputs of positive and negative towers so as to extract the pair-wise relation between them.

3.3.1 ***Dual-interest Prediction Towers.*** To extract the positive and negative interests, we fuse the feedback-aware sequential representations, single-feedback sequential representations, and single-feedback aggregated representations into the corresponding positive or negative prediction tower. Before feeding different representations into the final prediction towers, we first aggregate part of them by the sum pooling as:

$$\mathbf{s} = \sum_{j=1}^{t} \mathbf{S}_j, \mathbf{s}^P = \sum_{j=1}^{t} \mathbf{S}_j^P, \mathbf{s}^N = \sum_{j=1}^{t} \mathbf{S}_j^N,$$

which are then finally fed into the positive and negative prediction towers as:

$$logit_{u,t}^P = \mathbf{MLP}\left(\mathbf{s}\|\mathbf{s}^P\|\mathbf{f}^P\|(\mathbf{E}_{i_{t+1}} + \mathbf{L}_1)\right), \tag{19}$$

$$logit_{u,t}^N = \mathbf{MLP}\left(\mathbf{s}\|\mathbf{s}^N\|\mathbf{f}^N\|(\mathbf{E}_{i_{t+1}} + \mathbf{L}_0)\right). \tag{20}$$

where $logit_{u,t}^P$ and $logit_{u,t}^N$ are positive and negative predicted logits for user u on time step t, aiming to capture the positive and negative interests, respectively. Here $\mathbf{f}^P$ and $\mathbf{f}^N$ are pooled at Eq.(17).

3.3.2 ***Pair-wise Contrastive Loss.*** When the target item is positive, the prediction logit of the positive tower will be greater than that of the negative tower, and vice versa. After obtaining the positive and negative prediction logits, we then perform BPR loss [23] on them as:

$$\mathcal{L}^{BPR} = \begin{cases} -\log(\sigma(logit_{u,t}^P - logit_{u,t}^N)), & y_{u,t} = 1, \\ -\log(\sigma(logit_{u,t}^N - logit_{u,t}^P)), & y_{u,t} = 0. \end{cases} \tag{21}$$

where σ denotes the sigmoid function. With this BPR loss, we can extract the pair-wise relations between positive and negative logits.

3.4 Joint Optimization

Though we have positive and negative towers, in the optimization step, we only need to optimize the next item prediction loss with the positive tower as:

$$\mathcal{L} = -\frac{1}{|\mathcal{R}|} \sum_{(u,i_t) \in \mathcal{R}} \left(y_{u,t} \log \hat{y}_{u,t}^P + (1 - y_{u,t}) \log\left(1 - \hat{y}_{u,t}^P\right)\right), \tag{22}$$

where $\hat{y}_{u,t}^P = \sigma(logit_{u,t}^P)$ and $\mathcal{R}$ is the training set. The negative prediction tower $\hat{y}_{u,t}^N$ indeed will be self-supervised and optimized by the contrastive loss of Eq.(21). After obtaining the main loss for the next item prediction, disentangling loss for repelling representations and BPR loss for pair-wise learning, we can then jointly optimize them as:

$$\mathcal{L}^J = \mathcal{L} + \lambda^{BPR}\mathcal{L}^{BPR} + \lambda^D \mathcal{L}^D + \lambda\|\Theta\|, \tag{23}$$

where λ^{BPR} and λ^D are hyper-parameters for weighting each loss. Here λ is the regularization parameter, and Θ denotes the model parameters to be learned.

4 EXPERIMENTS

In this section, we experiment on a public dataset and an industrial dataset, aiming to answer the following research questions (RQ):

- **RQ1**: Is the proposed DFAR effective when compared with the state-of-the-art sequential recommenders?
- **RQ2** : What is the effect of our proposed feedback-aware encoding layer, dual-interest disentangling layer and prediction layer?
- **RQ3** : How do the heads of proposed feedback-aware factorization-heads attention capture the transition pattern between different feedback?

Table 1: Micro-video and Amazon data statistics.

Dataset		Micro-video	Amazon
#Users		37,497	6,919
#Items		129,092	28,695
#Records	Positive	6,413,396	99,753
	Negative	5,448,693	20,581
Avg. records per user		316.35	17.39

- **RQ4**: How does the proposed method perform compared with the sequential recommenders under different sequence lengths?

We also look into the question: "how do the auxiliary loss for disentanglement and pair-wise contrastive learning perform under different weights?" in Appendix A.4.

4.1 Experimental Settings

4.1.1 ***Datasets.*** The data statistics of our experiments are illustrated in Table 1 where Micro-video is a collected industrial dataset and Amazon is the public benchmark dataset which is widely used in existing work for sequential recommendation [19]. The detailed descriptions of them are as below.

Micro-video This is a popular micro-video application dataset, which is recorded from September 11 to September 22, 2021. In this platform, users passively receive the recommended videos, and their feedbacks are mostly either skip or no-skip. Skip can be treated as a form of negative feedback, and no-skip can be treated as a form of positive feedback. That is to say, we have hybrid positive and negative feedback in this sequential data which is very rare in modern applications.

Amazon[1] This is Toys domain from a widely used public e-commerce dataset in recommendation. The rating score in Amazon ranges from 1 to 5, and we treat the rating score over three and under two as positive and negative feedback, respectively, following existing work DenoisingRec [33] which is not for the sequential recommendation.

For the Micro-video dataset, interactions before and after 12 pm of the last day are split as the validation and test sets, respectively, while interactions before the last day are used as the training set. For the Amazon dataset, we split the last day as the test set and the second last day as the validation set, while other days are split as the training set.

4.1.2 ***Baselines and Evaluation Metrics.*** We compare our DFAR with the following state-of-the-art methods for sequential recommender systems.

- **DIN** [42]: It aggregates the historical items via attention score with the target item.
- **Caser** [27]: It captures the transition between historical items via convolution.
- **GRU4REC** [11]: It captures the transition between historical items via GRU [5].
- **DIEN** [41]: It captures the transition between historical items via interest extraction and evolution GRUs [5].
- **SASRec** [14]: It captures the transition between historical items via multi-heads attention [28].
- **THA4Rec**: It means talking-heads attention [25] for the sequential recommendation, which is firstly applied in the recommendation by us.
- **DFN** [38]: It purifies unclick (weak feedback) by click (strong positive feedback) and dislike (strong positive feedback).
- **FeedRec** [37]: It further performs disentanglement on the weak positive and negative feedback.

Besides, Widely-used AUC and GAUC [9] are adopted as accuracy metrics here while MRR@10 and NDCG@10 [19] are used as ranking metrics for performance evaluation. The detailed illustration of them is in Appendix A.2.

4.1.3 ***Hyper-parameter Settings.*** Hyper-parameters are generally set following the default settings of baselines. We strictly follow existing work for sequential recommendation [19] and leverage Adam [15] with the learning rate of 0.0001 to weigh the gradients. The embedding sizes of all models are set as 32. We use batch sizes 20 and 200, respectively, on the Micro-video and Amazon datasets. We search the loss weights for pair-wise contrastive loss in $[10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}]$.

4.2 Overall Performance Comparison(RQ1)

We compare our proposed method with eight competitive baselines, and the results are shown as Table 2, where we can observe that:

- **Our method achieves the best performance.** The results on two datasets show that our DFAR model achieves the best performance compared with these seven baselines on all metrics. Specifically, GAUC is improved by about 2.0% on the Micro-video dataset and 0.5% on the Amazon dataset and when comparing DFAR with other baselines. Please note that 0.5% improvement on GAUC could be claimed as significant, widely acknowledged by existing works [42]. Besides, the improvement is more significant in the Micro-video with more negative feedback, which means incorporating the negative feedback into the historical item sequence can boost the recommendation performance.
- **Existing work roughly captures the relation between positive feedback and negative feedback**. FeedRec and DFN even underperform some traditional sequential recommendation models like GRU4REC and Caser in Amazon dataset. Besides, though they outperform other baselines in Micro-video dataset, the improvement is still slight. In other words, their designs fail to capture the relation between positive feedback and negative feedback, which motivates us to further improve them from transition and interest perspectives.

4.3 Ablation Study (RQ2)

We further study the impact of four proposed components as Table 3, where FHA represents the factorization-heads attention, the MO represents the mask operation on factorized heads for factorization-heads attention, IDL means the interest disentanglement loss on the positive and negative interest representations, and IBL means the interest BPR loss on the positive and negative prediction logits. From this table, we can have the following observations.

- **Factorization of heads for transition attention weights is important**. Removing FHA and MO both show significant performance drops, which means these two components are both

[1] https://www.amazon.com

Table 2: Overall evaluations for DFAR against baselines under Micro-video and Amazon datasets on four metrics. Here Improv. is the improvement. Bold is the highest result and underline is the second highest result.

Models		DIN	Caser	GRU4REC	DIEN	SASRec	THA4Rec	DFN	FeedRec	Ours	Improv.
Micro-video	AUC	0.7345	0.8113	0.7983	0.7446	0.8053	0.8104	0.8342	0.8119	**0.8578**	2.83%
	MRR	0.5876	0.6138	0.5927	0.5861	0.6046	0.6080	0.6321	0.6095	**0.6568**	3.91%
	NDCG	0.6876	0.7079	0.6916	0.6861	0.7009	0.7035	0.7222	0.7047	**0.7410**	2.60%
	GAUC	0.7703	0.8211	0.8041	0.7753	0.8120	0.8138	0.8362	0.8180	**0.8545**	2.19%
Amazon	AUC	0.6595	0.7192	0.7278	0.6688	0.6903	0.7069	0.6998	0.7037	**0.7333**	0.76%
	MRR	0.4344	0.4846	0.4901	0.4547	0.4604	0.4599	0.4743	0.4675	**0.4980**	1.61%
	NDCG	0.5669	0.6073	0.6114	0.5832	0.5883	0.5879	0.5990	0.5938	**0.6175**	1.00%
	GAUC	0.6618	0.7245	0.7266	0.6859	0.7029	0.7021	0.7120	0.7079	**0.7305**	0.54%

Table 3: Effectiveness study of our proposed components. FHA means factorization-heads attention; MO means label mask operation on heads; IDL means interest disentangling loss on positive and negative representations; IBL means interest BPR loss on positive and negative logits.

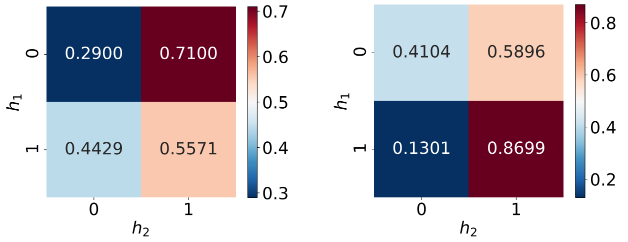
Dataset	Micro-video				
Methods	w/o FHA	w/o MO	w/o IDL	w/o IBL	Ours
AUC	0.8360	0.8473	0.8475	0.8364	**0.8578**
MRR	0.6198	0.6378	0.6377	0.6324	**0.6568**
NDCG	0.7127	0.7264	0.7264	0.7212	**0.7410**
GAUC	0.8319	0.8428	0.8436	0.8283	**0.8545**
Dataset	Amazon				
AUC	0.7133	0.7141	0.7284	0.7137	**0.7333**
MRR	0.4782	0.4883	0.4855	0.4839	**0.4980**
NDCG	0.6016	0.6095	0.6073	0.6057	**0.6175**
GAUC	0.7054	0.7137	0.7128	0.7047	**0.7305**

necessary to each other. Specifically, removing FHA means it is impossible to apply the mask on the implicit head interaction of either multi-heads attention or talking-heads attention. At the same time, removing MO on FHA will cause it to fail to exploit the prior knowledge of labels for historical items and degenerate to even as poor as multi-heads attention or talking-heads attention in the Amazon dataset.

- **Pair-wise interest is more important than disentangling interest**. Removing IDL and IBL will both drop the performance, while removing IBL is more significant. This is because contrastive learning by BPR loss can indeed inject more self-supervised signals, while disentanglement solely tends to repel the dissimilar representations of positive feedback and negative feedback.

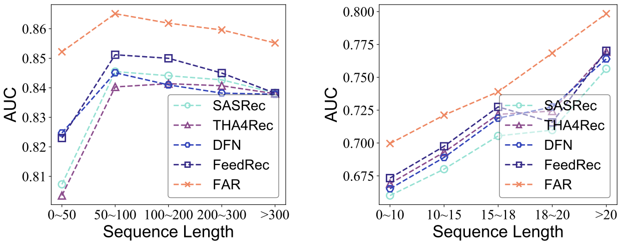
4.4 Visualization for Attention Weights of Heads (RQ3)

As illustrated in Eq.(8), our proposed factorization-heads attention can factorize the relation between different feedback, which makes it possible for us to study the attention weights between them. Therefore, we perform visualization on the attention weights between positive and negative heads in Figure 4, where h_1 and h_2 (defined at (11)) represent heads for source and target behaviors, respectively, with corresponding feedback. From this figure, we can observe that: (1) For the collected Micro-video dataset, users are still willing to watch videos even after they receive the disliked videos. This may be because the negative recommended videos are of low cost for users as they can easily skip the disliked videos, making no significant impact on their later preferred videos; (2) For the e-commerce dataset about Amazon, we can discover that when the source feedback is negative, the probability of target feedback being negative will increase sharply. This may be because the negative purchased items are of high cost in e-commerce for users as it will waste their money, increasing their unsatisfied emotion sharply.



(a) Micro-video (b) Amazon

Figure 4: Visualization of accumulated attention weights between different heads. Here h_1 and h_2 represent the heads for the source and target behaviors, respectively (i.e., if the source behavior is negative and target behavior is positive, we have $h_1 = 0$ and $h_2 = 1$). This illustrates our method can factorize and extract the relation between different feedback based on the proposed factorization-heads attention.



(a) Micro-video (b) Amazon

Figure 5: AUC performance comparisons under different sequence lengths on the Micro-video and Amazon datasets.

4.5 The Impact of Sequence Length (RQ4)

On large-scale online platforms, active users often observe a lot of items and generate very long historical item sequences, while cold-start users are recorded with very short sequences. Long historical item sequences can bring them more information but the problem of gradient vanishing will increase, while short historical item sequence brings limited information and tends to overfit the model. Thus, we divide historical item sequences into five groups based on their lengths and further study how DFAR outperforms the attention-based models under different lengths, under Micro-video and Amazon datasets, as illustrated in Figure 5. From the visualization, we can observe that:

- **DFAR is superior under different sequence lengths**. It is obvious that there is always a significant performance gap between DFAR and other methods. In the Amazon dataset, where the sequence length is relatively short, the AUC performances increase with the growth of sequence length for all methods. This means a longer sequence can bring more information. However, in the Micro-video dataset where the sequence length is relatively long, the performances of all methods improve with the increase of sequence length and reach their peak at around 50-100. But then they all decline with the further increase in length. Most importantly, our DFAR outperforms other methods significantly throughout various sequence lengths.
- **DFAR is stable under different sequence lengths**. DFAR is more stable with the sequence length increasing or decreasing, even into very long or short. In the Amazon dataset, other methods first increase with the sequence length but fluctuate at 15-20 while DFAR increases steadily with the sequence length. In the Micro-video dataset, All methods drop sharply when the sequence length is too short or long, but our DFAR is more stable and still keeps a decent AUC performance at 0.8382.

In summary, our DFAR is superior and robust under both long and short historical item sequences.

5 RELATED WORK

Sequential Recommendation Sequential Recommendation [32] predicts the next interacted item of the given user based on his/her historical items. As the early work, FPMC [24] exploits the Markov chain to capture the transition pattern of historical item sequence in the recommendation. Then some advanced deep learning methods such as RNN [5, 12] and attentive network [28] are applied in recommendation [11, 14, 41, 42] to capture the chronological transition patterns between historical items. While the evolution of RNN-based methods should forward each hidden state one by one and are difficult to parallel, attention-based methods can directly capture the transition patterns among all historical items at any time step. Furthermore, researchers also attempt to leverage convolution neural network [16] to capture the union and point levels sequential pattern in recommendation [27]. Compared with CNN-based methods, attention-based methods are more effective for their non-local view of self-attention [34]. However, the most existing sequential recommendation is based on click behavior. Recently, there have been some methods of achieving sequential recommendations beyond click behaviors [20]. For example, DFN [38] captures the sequential patterns among click, unclick and dislike behaviors by an internal module for each behavior and an external module to purify noisy feedback under the guidance of precise but sparse feedback. CPRS [36] derives reading satisfaction from the completion of users on certain news to facilitate click-based modeling. Based on them, FeedRec [37] further enhances sequential modeling by a heterogeneous transformer framework to capture the transition patterns between user feedback such as click, dislike, follow, etc. However, these works mainly focus on exploiting the auxiliary feedback to enhance the modeling in the sequential recommendation, which does not consider the most important characteristic - the transition patterns between historical positive and negative feedback. Differently from them, our approach can factorize the transition patterns between different feedback, achieving more accurate modeling for sequential recommendation with both positive and negative feedback. Additionally, our approach extracts the relation between positive and negative feedback at interest level.

Explainable Attention Attention methods are popular in many machine learning fields such as recommender systems [14, 26, 40], computer vision [7, 8, 17, 34] and natural language processing [1, 29], etc. Attention mechanisms are often explainable and have been widely used in deep models to illustrate the learned representation by visualizing the distribution of attention scores or weights under specific inputs [4, 21, 35]. Some explainable attention methods are also generalizable and can be equipped with many backbones. For example, L2X [3] exploits Gumbel-softmax [13] for feature selection by instance, with its hard attention design [39]. Moreover, VIBI [2] further propose a feature score constraint in a global prior so as to simplify and purify the explainable representation learning. As self-attention is popular [6, 28], there is also a work that explains what heads learn and concludes that some redundant heads can be pruned [30]. In this work, we propose feedback-aware factorization-heads attention to explicitly capture the transition pattern between positive and negative feedback. The feedback mask matrix in our attention module can be treated as hard attention based on feedback.

6 CONCLUSIONS AND FUTURE WORK

In this work, we considered the positive and negative feedback in the historical item sequence for the sequential recommendation, while existing works were mostly click-based and considered solely positive feedback. Such exploration addressed the challenge of current multi-head attention for different feedback interactions in one sequence. More specifically, we first applied talking-heads attention in the sequential recommendation and further proposed feedback-aware factorization-heads attention to explicitly achieve interaction across different heads for self-attention. Secondly, we proposed disentanglement and pair-wise contrastive learning to repel the dissimilar interests and capture the pair-wise relation between positive and negative feedback. In the future, we plan deploy the model in industrial applications to validate online performance.

ACKNOWLEDGMENTS

This work is supported in part by the National Key Research and Development Program of China under 2022YFB3104702, the National Natural Science Foundation of China under 62272262, 61971267, U1936217 and 61972223, BNRist, the Fellowship of China Postdoctoral Science Foundation under 2021TQ0027 and 2022M710006, and the Tsinghua University Guoqiang Institute under 2021GQG1005.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1409.0473

[2] Seojin Bang, Pengtao Xie, Heewook Lee, Wei Wu, and Eric Xing. 2020. Explaining A Black-box By Using A Deep Variational Information Bottleneck Approach. https://openreview.net/forum?id=BJlLdhNFPr

[3] Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. 2018. Learning to explain: An information-theoretic perspective on model interpretation. In *International Conference on Machine Learning*. PMLR, 883–892.

[4] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. 2016. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. *Advances in neural information processing systems* 29 (2016).

[5] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*. https://openreview.net/forum?id=YicbFdNTTy

[8] Kun Fu, Junqi Jin, Runpeng Cui, Fei Sha, and Changshui Zhang. 2016. Aligning where to see and what to tell: Image captioning with region-based attention and scene-specific contexts. *IEEE transactions on pattern analysis and machine intelligence* 39, 12 (2016), 2321–2334.

[9] Asela Gunawardana and Guy Shani. 2015. Evaluating Recommender Systems. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.). Springer US, 265–308. https://doi.org/10.1007/978-1-4899-7637-6_8

[10] Lei Guo, Hongzhi Yin, Qinyong Wang, Tong Chen, Alexander Zhou, and Nguyen Quoc Viet Hung. 2019. Streaming session-based recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 1569–1577.

[11] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. In *ICLR*.

[12] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[13] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rkE3y85ee

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

[15] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. 25 (2012), 1097–1105.

[17] Yanwei Li, Xinze Chen, Zheng Zhu, Lingxi Xie, Guan Huang, Dalong Du, and Xingang Wang. 2019. Attention-guided unified network for panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7026–7035.

[18] Yuchen Li, Dongxiang Zhang, Ziquan Lan, and Kian-Lee Tan. 2016. Context-aware advertisement recommendation for high-speed social news feeding. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*. IEEE, 505–516.

[19] Guanyu Lin, Chen Gao, Yinfeng Li, Yu Zheng, Zhiheng Li, Depeng Jin, and Yong Li. 2022. Dual Contrastive Network for Sequential Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2686–2691.

[20] Mingyuan Ma, Sen Na, Hongyu Wang, Congzhou Chen, and Jin Xu. 2022. The graph-based behavior-aware recommendation for interactive news. *Applied Intelligence* 52, 2 (2022), 1913–1929.

[21] Andre Martins and Ramon Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International conference on machine learning*. PMLR, 1614–1623.

[22] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based news recommendation for millions of users. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 1933–1942.

[23] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Schmidt-Thie Lars. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence* (Montreal, Quebec, Canada) *(UAI '09)*. AUAI Press, Arlington, Virginia, United States, 452–461. http://dl.acm.org/citation.cfm?id=1795114.1795167

[24] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *WWW*. 811–820.

[25] Noam Shazeer, Zhenzhong Lan, Youlong Cheng, Nan Ding, and Le Hou. 2020. Talking-heads attention. *arXiv preprint arXiv:2003.02436* (2020).

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

[27] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *WWW*. 565–573.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*. 5998–6008.

[29] Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. *Advances in neural information processing systems* 28 (2015).

[30] Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 5797–5808. https://doi.org/10.18653/v1/P19-1580

[31] Heyuan Wang, Fangzhao Wu, Zheng Liu, and Xing Xie. 2020. Fine-grained interest matching for neural news recommendation. In *Proceedings of the 58th annual meeting of the association for computational linguistics*. 836–845.

[32] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z. Sheng, and Mehmet Orgun. 2019. Sequential recommender systems: challenges, progress and prospects. (2019).

[33] Wenjie Wang, Fuli Feng, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2021. Denoising Implicit Feedback for Recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. ACM. https://doi.org/10.1145/3437963.3441800

[34] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. 2018. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7794–7803.

[35] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*. 606–615.

[36] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2020. User Modeling with Click Preference and Reading Satisfaction for News Recommendation.. In *IJCAI*. 3023–3029.

[37] Chuhan Wu, Fangzhao Wu, Tao Qi, Qi Liu, Xuan Tian, Jie Li, Wei He, Yongfeng Huang, and Xing Xie. 2022. Feedrec: News feed recommendation with various user feedbacks. In *Proceedings of the ACM Web Conference 2022*. 2088–2097.

[38] Ruobing Xie, Cheng Ling, Yalong Wang, Rui Wang, Feng Xia, and Leyu Lin. 2021. Deep feedback network for recommendation. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*. 2519–2525.

[39] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*. PMLR, 2048–2057.

[40] Junqi Zhang, Bing Bai, Ye Lin, Jian Liang, Kun Bai, and Fei Wang. 2020. General-purpose user embeddings based on mobile app usage. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2831–2840.

[41] 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 *AAAI*. 5941–5948.

[42] 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 *KDD*. 1059–1068.

A APPENDIX FOR REPRODUCIBILITY

A.1 Pseudocode

Listing 1: Pseudocode for Multi-heads Attention

```
1  def MultiHeadAttention (
2  X[n, d_X], # n vectors with dimensionality d_X
3  M[m, d_M], # m vectors with dimensionality d_M
4  P_q[d_X, d_k, h], # learned linear projection to produce queries
5  P_k[d_M, d_k, h], # learned linear projection to produce keys
6  P_v[d_M, d_v, h], # learned linear projection to produce values
7  P_o[d_Y, d_v, h]): # learned linear projection of output
8      Q[n, d_k, h] = einsum (X[n, d_X], P_q[d_X, d_k, h])
9      K[m, d_k, h] = einsum (M[m, d_M], P_k[d_M, d_k, h])
10     V[m, d_v, h] = einsum (M[m, d_M], P_v[d_M, d_v, h])
11
12     L[n, m, h] = einsum (Q[n, d_k , h], K[m, d_k , h]) # logits h*n*m* d_k
13
14     W[n, m, h] = softmax (L[n, m, h], reduced_dim =m) # weights
15
16     O[n, d_v , h] = einsum (W[n, m, h], V[m, d_v , h]) # h*n*m* d_v
17     Y[n, d_Y ] = einsum (O[n, d_v , h], P_o[d_Y , d_v , h]) # output h*n* d_Y * d_v
18     return Y[n, d_Y]
```

We follow talking-heads attention [25] and present the following notation and pseudocode.

A.1.1 *Notation*.

In our pseudocode, we follow talking-heads attention [25] and have a notation as below.

- The capital letters represent the variable names, and lower-case letters represent the number of dimensions. Each variable of a tensor is presented with its dimensions. For example, a tensor for an item sequence with batch size b, sequence length n, hidden state d is written as: X[b, n, d] [25].
- The einsum represents the generalized contractions between tensors without any constraint on their dimension. Its computation process is: (1) Broadcasting each input to have the union of all dimensions, (2) multiplying component-wise, and (3) summing across all dimensions not in the output. The dimensions are identified by the dimension-list annotations on the arguments and on the result instead of being identified by an equation, as in TensorFlow and NumPy. For example, multiplying two matrices is written as: Z[a, c] = einsum (X[a, b], W[b, c]) [25].

A.1.2 *Multi-heads Attention*.

The pseudocode for multi-heads attention [28] is as shown in Pseudocode 1, where different heads for Q and K do not interact with each other on line 12.

A.1.3 *Talking-heads Attention*.

The pseudocode for talking-heads attention [25] is as shown in Pseudocode 2, where different heads for Q and K achieve implicit interaction by lines 15 and 18.

A.1.4 *Factorization-heads Attention*.

The pseudocode for our proposed factorization-heads attention is as shown in Pseudocode 3, where different heads for Q and K achieve explicit interaction by line 16.

Listing 2: Pseudocode for Talking-heads Attention

```
1  def TalkingHeadAttention (
2  X[n, d_X], # n vectors with dimensionality d_X
3  M[m, d_M], # m vectors with dimensionality d_M
4  P_q[d_X, d_k, h_k], # learned linear projection to produce queries
5  P_k[d_M, d_k, h_k], # learned linear projection to produce keys
6  P_v[d_M, d_v, h_v], # learned linear projection to produce values
7  P_o[d_Y, d_v, h_v]
8  P_l [h_k , h], # talking - heads projection for logits
9  P_w [h, h_v]): # talking - heads projection for weights
10     Q[n, d_k, h_k] = einsum (X[n, d_X], P_q[d_X, d_k, h_k])
11     K[m, d_k, h_k] = einsum (M[m, d_M], P_k[d_M, d_k, h_k])
12     V[m, d_v, h_v] = einsum (M[m, d_M], P_v[d_M, d_v, h_v])
13
14     J[n, m, h_k] = einsum (Q[n, d_k, h_k], K[m, d_k, h_k]) # dot prod . n*m* d_k *h_k
15     L[n, m, h] = einsum (J[n, m, h_k], P_l [h_k, h]) # Talking - heads proj . n*m*h* h_k
16
17     W[n, m, h] = softmax (L[n, m, h], reduced_dim=m) # Attention weights
18     U[n, m, h_v] = einsum (W[n, m, h], P_w [h, h_v]) # Talking - heads proj . n*m*h* h_v
19
20     O[n, d_v, h_v] = einsum (U[n, m, h_v], V[m, d_v, h_v]) # n*m* d_v * h_v
21     Y[n, d_Y] = einsum (O[n, d_v, h_v], P_o [d_Y, d_v, h_v]) # n* d_Y * d_v * h_v
22     return Y[n, d_Y]
```

A.1.5 *Comparison*.

From these three Python pseudocodes, we can discover that our factorization-heads attention achieves head interaction at a low cost. The comparison of it with multi-heads attention and talking-heads attention are as below.

- **Comparing with Multi-heads Attention**: our factorization-heads attention incorporates the interaction between different heads with additional four lines at lines 12-14 and 17, which are transpose and reshape operations and with only $O(1)$ temporal complexity. [2].
- **Comparing with Talking-heads Attention**: our factorization-heads attention achieves explicit interaction with additional transpose and reshape operations at $O(1)$ temporal complexity while talking-heads attention achieves implicit interaction with two matrix multiplication operations at $O(m \times h_k \times h)$ and $O(m \times h \times h_v)$ temporal complexities [3], respectively.

[2] https://stackoverflow.com/questions/58279082/time-complexity-of-numpy-transpose

[3] https://en.wikipedia.org/wiki/Computational_complexity_of_matrix_multiplication

A.2 Evaluation Metrics

The detailed illustration of adopted evaluation metrics is as follows.

- **AUC**: Randomly selecting one positive item and one negative item, it represents the probability that the predicted score of the positive item is higher than that of the negative item. It tests the model's ability to classify the positive and negative items.
- **GAUC**: It weighs each user's AUC based on his/her test set size. It tests the model's personalized classification ability on each user as recommender systems indeed tend to rank preferred items for users individually.
- **MRR@K**: It is the average of the reciprocal of the first hit item ranking.
- **NDCG@K**: It assigns hit items that rank higher with more weights and thus tests the model's ability to rank the hit items in higher and more confident positions.

A.3 Implementation Details

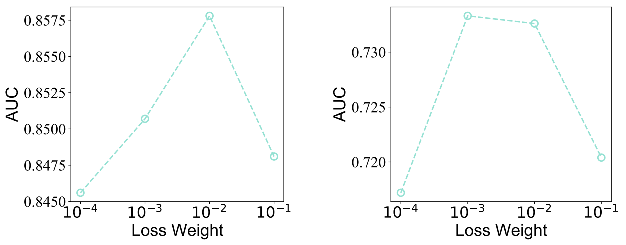
We implement all the models by a Microsoft [4] TensorFlow [5] framework in Python, which is accessible here [6]. We will publish the Micro-video dataset to benefit the community in the future, and the public Amazon dataset is accessible at this website [7].

The environment is as below.

- Anaconda 3
- Python 3.7.7
- TensorFlow 1.15.0

Besides, for other parameters, we stop the model training with early stop step 2 and leverage the MLP layer sandwiched between two normalization layers as the prediction tower for each model.

A.4 Hyper-parameter Study (RQ5)



(a) Micro-video

(b) Amazon

Figure 6: AUC performance of different auxiliary loss weights w.r.t λ^{BPR} and λ^{D} under Micro-video and Amazon datasets.

We perform hyper-parameter study on the weights for loss of disentanglement and pair-wise contrastive Learning (w.r.t. λ^{BPR} and λ^{D} at Eq.(23)) as Figure 6, varying the loss weights from 10^{-4} to 10^{-1}. From the figure, we can observe that the AUC performance reaches the peak at 10^{-3} under the Amazon dataset while that reaches the peak at 10^{-2} under the Micro-video dataset. This is because the rating for Amazon is a discrete value, but the playing time for Micro-video is a continuous value. The partition of positive and negative feedback based on continuous value is unclear and thus requires more contrastive learning. Based on the above observation, we finally choose 10^{-3} and 10^{-2} as the best values for the loss weights under Amazon and Micro-video datasets, respectively.

Listing 3: Pseudocode for Factorization-heads Attention

```
def FactorizationHeadAttention (
X[n, d_X], # n vectors with dimensionality d_X
M[m, d_M], # m vectors with dimensionality d_M
P_q[d_X, d_k, h], # learned linear projection to produce queries
P_k[d_M, d_k, h], # learned linear projection to produce keys
P_v[d_M, d_v, h], # learned linear projection to produce values
P_o[d_Y, d_v, h]): # learned linear projection of output
    Q[n, d_k, h] = einsum (X[n, d_X], P_q[d_X, d_k, h])
    K[m, d_k, h] = einsum (M[m, d_M], P_k[d_M, d_k, h])
    V[m, d_v, h] = einsum (M[m, d_M], P_v[d_M, d_v, h])

    Q[n, h, d_k] = reshape(transpose(Q, [0, 2, 1]), [n * h, d_k]) # queries h*n* d_X * d_k
    K[d_k, h, m] = reshape(transpose(K, [1, 2, 0]), [d_k, h * m]) # keys h*m* d_M * d_k
    V[m, d_v, h * h] = tile(V[m, d_v, h], [1, 1, h]) # values h*m* d_M * d_v

    L[n * h, h * m] = einsum (Q[n * h, d_k], K_[d_k, h * m])
    L[n, h * h, m] = transpose(reshape(L, [n, h * h, m]), [0, 2, 1]) # logits h*h*n*m* d_k

    W[n, m, h * h] = softmax (L[n, m, h * h], reduced_dim=m) # weights
    O[n, d_v , h * h] = einsum (W[n, m, h * h], V[m, d_v, h * h]) # h*h*n*m* d_v
    Y[n, d_Y] = einsum (O[n, d_v, h * h], P_o[d_Y, d_v, h * h]) # output h*h*n* d_Y * d_v
    return Y[n, d_Y]
```

[4] https://github.com/microsoft/recommenders

[5] https://www.tensorflow.org

[6] https://anonymous.4open.science/r/DFAR-8B7B

[7] http://jmcauley.ucsd.edu/data/amazon/index_2014.html