FISSA: Fusing Item Similarity Models with Self-Attention Networks for Sequential Recommendation

Jing Lin$^{1,2,3}$, Weike Pan$^{1,2,3,*}$ and Zhong Ming$^{1,2,3,*}$
linjing2018@email.szu.edu.cn, {panweike, mingz}@szu.edu.cn

1 National Engineering Laboratory for Big Data System Computing Technology
Shenzhen University, China

2 Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ)
Shenzhen University, China

3 College of Computer Science and Software Engineering
Shenzhen University, China
Motivation

Two issues of existing sequential recommendation methods:

- As deep learning (DL) based methods being widely adopted to model the local and dynamic preferences beneath users’ behavior sequences, the modeling of users’ global and static preferences tends to be underestimated that usually, only some simple and crude users’ latent representations are introduced.

- Most existing methods hold an assumption that users’ intention can be fully captured by considering the historical behaviors, while neglect the possible uncertainty of users’ intention in reality, which may be influenced by the appearance of the candidate items to be recommended.
In this paper, we propose a novel solution named **fusing item similarity models with self-attention networks** (or FISSA in short) for sequential recommendation.

Specifically, our model contains three main components, i.e., a **local** representation learning module, a **global** representation learning module, and a gating module to **balance** these two kinds of representations.
Focusing on the aforementioned two issues, i.e., the imperfect modeling of users’ global preferences in most DL-based sequential recommendation methods and the uncertainty of users’ intention brought by the candidate items, our FISSA not only joins the effective global representation learning to the well-established method, i.e., self-attentive sequential recommendation (SASRec) [Kang and McAuley, 2018], but also balances a user’s short-term and long-term interest for each candidate item.

Empirical studies on five commonly used datasets show that our FISSA significantly outperforms eight state-of-the-art baselines. In particular, our FISSA surpasses SASRec by 10.11% and 10.05% on average in terms of Rec@10 and NDCG@10, respectively.
Related Work (1/2)

**General Recommendation**

- **Factored item similarity model (FISM)** [Kabbur et al., 2013] regards the predicted rating as the factored similarity between the user’s historical items and the candidate item.

- **Neural attentive item similarity (NAIS)** [He et al., 2018] applies the attention mechanism to distinguish more important items for the candidate item rather than for the user in other works.

In this paper, we achieve an attentive form of FISM to obtain the global representation of a user’s behavior sequence, and design an item similarity gating for balancing the local and global representations by modeling the relationship among the candidate item, the recently interacted item and the global preference of the user.
Sequential Recommendation

Self-attentive sequential recommendation (SASRec) [Kang and McAuley, 2018] is found to be an outstanding sequential recommendation model with satisfactory conciseness and efficiency.

Different from other works that improve SASRec by introducing graph neural networks [Xu et al., 2019] or bidirectional structure [Sun et al., 2019], which still focus on the local and dynamic preference modeling, our proposed FISSA aims at combining SASRec with an effective global and static preference learning model in a balanced way.

For dealing with the uncertainty of users’ intention in sequences, existing works mainly focus on distinguishing the importance of the items in sequences. Our FISSA additionally models the short-term and long-term preferences of a user and then balances them according to different candidate items.
**Problem Definition**

- **Input:** Without loss of generality, we have a recommendation system with implicit feedback given by a set of users $U$ to a set of items $I$. For sequential recommendation, we denote the records of each user $u \in U$ as an item sequence (ordered by the interaction time) as $S^u = \{s^u_1, s^u_2, \ldots, s^u_{|S^u|}\}$, $s^u \in I$.

- **Goal:** To provide a recommendation list for each user $u$, in which we expect the real next interacted item $s^u_{|S^u|+1} \in I \setminus S^u$ to appear and be ranked as high as possible.
Our FISSA contains three main components, including a local representation learning module, a global representation learning module, and a gating module to balance these two kinds of representations. In this paper, we use capital letters in bold to denote matrices and their lowercase form to denote the corresponding row vectors.
Figure: The network architecture of our proposed FISSA. At the beginning, the input sequence (in the upper left corner) is represented as an embedding matrix $E$, in which each item embedding $m_i$ is from the item embedding matrix $M \in \mathbb{R}^{|I| \times d}$ that works as a dictionary. The local representation learning module (in the top half) consists of $n$ series of stacked self-attention blocks $SAB(\cdot)$ (see Eqs. (2~5)). The global representation learning module (in the bottom left corner) is actually a location-based attention layer $LBA(\cdot)$ (see Eq. (7)). In the gating module (in the bottom right corner), the item similarity gating function $ISG(\cdot, \cdot, \cdot)$ (see Eq. (9)) outputs the balanced weights of the local and global representations by taking the representations of the candidate item $m_i$, the recently interacted item $m_{s_{n-1}}$ and the user’s global preference $y$ as inputs.
First of all, we fix the input sequence of each user \( u \) by extracting his/her latest \( L \) behaviors, which is abbreviated as \( S^u = \{s_1, s_2, \ldots, s_L\} \) (usually a relatively large value of \( L \), e.g., \( L = 50 \) for our studied datasets, is chosen to reserve the whole sequences of most users, and padding items are appended at the beginning of the sequences when needed).

Let \( M \in \mathbb{R}^{|I| \times d} \) denote the learnable item embedding matrix with \( d \) as the latent dimensionality. We can then represent the input sequence as an embedding matrix \( E = [m_{s_1}; m_{s_2}; \ldots; m_{s_L}] \in \mathbb{R}^{L \times d} \).
Local Representation Learning (1/3)

Following SASRec [Kang and McAuley, 2018]

- In order to capture the influence of the position, we add a learnable position embedding matrix \( P = [p_1; p_2; \ldots; p_L] \in \mathbb{R}^{L \times d} \) to the input embedding matrix \( E \in \mathbb{R}^{L \times d} \), and obtain an input matrix \( X^{(0)} = [x_1; x_2; \ldots; x_L] \in \mathbb{R}^{L \times d} \) for the self-attention network:

\[
x^{(0)}_\ell = m_{s_\ell} + p_\ell, \ell \in \{1, 2, \ldots, L\}.
\] (1)

- Then, we feed the sequence \( X^{(0)} \in \mathbb{R}^{L \times d} \) into a series of stacked self-attention blocks (SABs). The output of the \( b \)th block is as follows:

\[
X^{(b)} = SAB^{(b)}(X^{(b-1)}), \ b \in \{1, 2, \ldots, B\}.
\] (2)
The self-attention block $SAB^{(b)}(\cdot)$ is first introduced in [Vaswani et al., 2017] and can be simply viewed as a self-attention layer $SAL(\cdot)$ followed by a feed-forward layer $FFL(\cdot)$ as follows:

$$SAB(X) = FFL(SAL(X)), \quad (3)$$

$$X' = SAL(X) = \text{softmax}(\frac{QK^T}{\sqrt{d}})\Delta \cdot V, \quad (4)$$

$$FFL(X') = \text{ReLU}(X'W_1 + 1^Tb_1)W_2 + 1^Tb_2, \quad (5)$$

where $X \in \mathbb{R}^{L \times d}$ is the position-aware input matrix, $Q = XW_Q$, $K = XW_K$ and $V = XW_V$ with $W_Q$, $W_K$, $W_V \in \mathbb{R}^{d \times d}$ are the projected query, key and value matrices, respectively, to improve the flexibility. Note that $W_1$, $W_2 \in \mathbb{R}^{d \times d}$ and $b_1$, $b_2 \in \mathbb{R}^{1 \times d}$ are weights and biases for the two layers of convolution, $1$ is a unit row vector of size $1 \times L$ and $\Delta$ is the causality mask, i.e., a unit lower triangular matrix of size $L \times L$, to preserve the transitions from previous steps only. The normalization and dropout layers we use in this module are the same with that in [Vaswani et al., 2017].
In this module, we take the output vector \( x^{(B)}_{\ell} \in \mathbb{R}^{1 \times d} \) from the top self-attention block as the local representation, which stands for the dynamic preference at the \( \ell \)th step in the user behavior sequence. It is shown in [Kang and McAuley, 2018] that the hierarchical structure is important for the local representation. Specifically, the bottom self-attention block (i.e., \( SAB^{(1)}(\cdot) \)) tends to capture the long-term dependencies, while higher blocks may pay attention to more recent ones.
Global Representation Learning (1/4)

• Though applying the attention mechanism to avoid rigorous ordering of the previous items, the local representation still ignores the variable ordering of the current item and its subsequent items. A simple way to deal with this issue is to generate a global and non-causal representation of each user’s behavior sequence, so that more available information from the future can be utilized for the prediction at each step in the sequence during training.

• In FISM [Kabbur et al., 2013], the preference of user $u$ on the interacted item $s_{\ell+1}^u$ at the $(\ell + 1)$th step is generated as the uniform aggregation of the representation of the other interacted items:

$$\tilde{y}_{\ell+1}^u = \frac{1}{\sqrt{|S^u\backslash\{s_{\ell+1}^u\}|}} \sum_{i' \in S^u\backslash\{s_{\ell+1}^u\}} m_{i'}.$$  

(6)

so that the predicted rating $r_{\ell+1}^u, s_{\ell+1}^u = \tilde{y}_{\ell+1}^u m_{s_{\ell+1}^u}^T$ can be regarded as a factored similarity between the historical items of user $u$ and the candidate item $s_{\ell+1}^u$. 
Another understanding of FISM is that with the benefit of utilizing the shared item representation, sequences with similar items tend to have similar representations. We believe that this effect can be enhanced if more representative items are noticed.

So instead of an aggregation with average weighting, we introduce a learnable query vector $q^S \in \mathbb{R}^{1 \times d}$ shared by all sequences to figure out the most representative items in the sequences. Omitting the superscript $u$, the global representation of the sequence can then be formalized as follows:

$$y = LBA(E) = \text{softmax}(q^S(EW'_K)^T)EW'_V, \quad (7)$$

where $E \in \mathbb{R}^{L \times d}$ is the initial input matrix, $W'_K, W'_V \in \mathbb{R}^{d \times d}$ are projection matrices to be learned, similar to $W_Q, W_K, W_V$ in Eq.(4). Such an attention layer is also known as a location-based attention layer $LBA(\cdot)$ described in [Luong et al., 2015]. Note that different from the local representation learning module, the position information $P$ and the causality constraint $\Delta$ are abandoned here.
It is worth mentioning that in our case, the global representation of the sequence is the same to all steps, which means that the corresponding parameters in Eq.(7) are updated only once in a training epoch even for predictions in many (e.g., $L$) steps.

So a dropout layer is very important during training to generalize the global representation to all steps, i.e., we have the global representation matrix $Y \in \mathbb{R}^{L \times d}$ as follows:

$$y_{\ell} = Dropout(y), \ell \in \{1, 2, \ldots, L\}. \quad (8)$$
In this way, we propose an attentive version of FISM for global representation learning in sequential recommendation, which is well adapted to the parallelization training process of the self-attention framework (i.e., including all the steps of a sequence in one training sample). Note that the global representation learning module is independent of the local one.
To combine the local representation and the global representation, we may naturally think of concatenation or summation. Our early attempts have shown that **summation is always better than concatenation**.

In [He et al., 2020], the authors suggest a weighted summation to balance the two representations by considering the consistency of the item sequences. However, these approaches of combination are still based on the **historical information only**, which may be **idealistic** because users’ intention can be uncertain and a proper way to tell whether a new item can attract a user is to consider how it can arouse different parts of the user’s interest (i.e., the short-term one and the long-term one).
Inspired by neural attentive item similarity (NAIS) [He et al., 2018], we propose an item similarity gating module, which calculates the weight of the local representation and global representation by modeling the item similarity between the candidate item \( i \in I \) and the recently interacted item \( s_\ell \), as well as the item similarity between the candidate item \( i \) and the aggregation of the historical items \( S^u \).
Item Similarity Gating (3/4)

Specifically, the representation of the candidate item $i \in \mathcal{I}$ and the interacted item at the most recent step $s_\ell$ are taken from the primitive item embedding matrix $\mathbf{M}$, i.e., $\mathbf{m}_i$ and $\mathbf{m}_{s_\ell}$, respectively. The global preference is represented as the aggregated representation of the historical items $S^u$, which is exactly the learned global representation $\mathbf{y}$. The individual-level gating is then written as an MLP as follows:

$$g = \sigma(ISG(\mathbf{m}_{s_\ell}, \mathbf{y}, \mathbf{m}_i)) = \sigma([\mathbf{m}_{s_\ell}, \mathbf{y}, \mathbf{m}_i] \mathbf{W}_G + b_G),$$

(9)

where $ISG(\cdot, \cdot, \cdot)$ is the item similarity gating function, $[\cdot, \cdot, \cdot]$ denotes the ternary concatenation operation, $\mathbf{W}_G \in \mathbb{R}^{3d \times 1}$ and $b_G \in \mathbb{R}$ are the weights and bias to be learned, respectively. We use the sigmoid function $\sigma(\xi) = 1/(1 + e^{-\xi})$ as the activation function, so that $g$ is restricted to $(0, 1)$. 
Item Similarity Gating (4/4)

The final representation of the sequence at the $\ell$th step is obtained by the weighed sum of the corresponding local representation $x^{(B)}_{\ell}$ and global representation $y$ as follows:

$$z_{\ell} = x^{(B)}_{\ell} \otimes g + y \otimes (1 - g),$$

(10)

where $\otimes$ denotes the element-wise product operation with the broadcasting mechanism in TensorFlow. Note that the weight of the local representation in our FISSA can be extended from $(0, 1)$ to $[0, 1]$ to include both SASRec ($z^L_{\ell} = x^{(B)}_{\ell}$ when $g = 1$, i.e., our local representation learning module) and the attentive version of FISM ($z^G_{\ell} = y$ when $g = 0$, i.e., our global representation learning module) as special cases.
We predict the preference of item $i$ being the $(\ell + 1)$th item in the sequence as follows:

$$r_{\ell+1,i} = z_{\ell}(m_i)^T.$$  \hfill (11)

We train our FISSA by minimizing the binary cross-entropy loss with the Adam optimizer. The loss function is as follows:

$$\mathcal{L} = -\sum_{u \in \mathcal{U}} \sum_{\ell=1}^{L-1} \delta(s_{\ell+1}^u)[\log(\sigma(r_{\ell+1,u}^u,s_{\ell+1}^u)) + \log(1 - \sigma(r_{\ell+1,u}^u,j))],$$  \hfill (12)

where $j \in \mathcal{I}\setminus S^u$ is a negative item randomly sampled for each prediction. The indicator function $\delta(s_{\ell+1}^u) = 1$ only if $s_{\ell+1}^u$ is not a padding item, otherwise 0.
Research Questions

RQ1: Does our FISSA achieve the state-of-the-art performance?

RQ2: What is the impact of different components in our FISSA?

RQ3: How does the key parameters such as the dimensionality $d$ and the number of blocks $B$ affect the performance of our FISSA?

RQ4: What is the impact of some options for the design of the global representation module (e.g., the consideration of causality) and the gating module (e.g., the input and output of the MLP layer) in our FISSA?
Datasets

We preprocess the datasets as follows: 1) we treat the presence of review, check-in and purchase behaviors as positive feedback and order them by the timestamps; 2) we discard later duplicated user-item pairs in order to predict new items; 3) we successively discard items and users with fewer than 5 records to maintain sequentiality; and 4) we adopt the leave-one-out evaluation by splitting each dataset into three parts, i.e., the last interaction of each user for test, the penultimate one for validation and the rest for training. Note that during testing, the input sequences contain training actions and the validation action. Note that cold-start items in the test and validation data are also removed.

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Users</th>
<th># Items</th>
<th># Interactions</th>
<th>Avg. Length</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beauty</td>
<td>40,226</td>
<td>54,542</td>
<td>353,962</td>
<td>8.80</td>
<td>0.02%</td>
</tr>
<tr>
<td>Games</td>
<td>29,341</td>
<td>23,464</td>
<td>280,945</td>
<td>9.58</td>
<td>0.04%</td>
</tr>
<tr>
<td>Steam</td>
<td>281,428</td>
<td>13,044</td>
<td>3,488,899</td>
<td>12.40</td>
<td>0.10%</td>
</tr>
<tr>
<td>Foursquare</td>
<td>22,748</td>
<td>11,146</td>
<td>145,106</td>
<td>6.38</td>
<td>0.06%</td>
</tr>
<tr>
<td>Tmall</td>
<td>201,139</td>
<td>97,636</td>
<td>1,936,790</td>
<td>9.63</td>
<td>0.01%</td>
</tr>
</tbody>
</table>

Table: Statistics of the processed datasets.
Evaluation Metrics

- We evaluate the recommendation performance via two common metrics, i.e., recall (Rec@10, equivalent to hit ratio because there is exactly one preferred item by each user in our case) and normalized discounted cumulative gain (NDCG@10).

- Rec@10 refers to the ratio of the real next items presenting in the top-10 recommendation lists, while NDCG@10 cares more about the exact ranking positions of the target items in these lists.

- To reduce computation, we follow [Kang and McAuley, 2018] to prearrange a candidate list with 100 randomly sampled un-interacted items for each user.
Baselines

Four MF-based methods:
- BPRMF [Rendle et al., 2009].
- FISM [Kabbur et al., 2013].
- FPMC [Rendle et al., 2010].
- Fossil [He and McAuley, 2016].

Four DL-based methods:
- GRU4Rec+ [Hidasi and Karatzoglou, 2018]. An RNN-based model.
- SASRec [Kang and McAuley, 2018]. A hierarchical self-attention network for sequential recommendation, which also works as the local representation learning module in our FISSA.
- CAR [He et al., 2020]. A similar model to ours, which also improves SASRec by introducing the global preferences of users and a consistency-aware gating. Note that unlike the others, this model aims at addressing the user-generated item list continuation problem defined in [He et al., 2020].
We implement the MF-based models with the codes provided by [He and McAuley, 2016] for the research of Fossil\(^1\), and run the DL-based methods GRU4Rec\(^2\) and Caser\(^3\) with the codes released by the authors of the original papers.

Our code of FISSA\(^4\) is an adaption from the published code of SASRec\(^5\), in which the attention module is modified and a new MLP for our gating module is added. We also adapt our code for CAR. So we run the experiments of SASRec and CAR via our adapted codes rather than the original ones.

---

1. [https://cseweb.ucsd.edu/~jmcauley/](https://cseweb.ucsd.edu/~jmcauley/)
2. [https://github.com/hidasib/GRU4Rec](https://github.com/hidasib/GRU4Rec)
3. [https://github.com/graytowne/caser_pytorch](https://github.com/graytowne/caser_pytorch)
5. [https://github.com/kang205/SASRec](https://github.com/kang205/SASRec)
For fair comparison, we select the item embedding dimensionality $d$ in all models as 50. Other key parameters such as the MC orders ($\in \{1, 2, \ldots, 9\}$ for Fossil and Caser), negative sampling numbers (2048 for GRU4Rec+) , filter sizes (4 and 16 for the vertical and horizontal filters, respectively in Caser) and so on are all tuned on the validation data according to the suggestions in the corresponding papers.

For our FISSA, following [Kang and McAuley, 2018], we set the sequence length $L$ to 50, the batch size to 128, the learning rate to 0.001 and the dropout rate to 0.5, and use single-head self-attention layers. The number of blocks $B$ is important for the performance of SASRec, CAR and our FISSA, which is searched from $\{1, 2, 3\}$. 
### Performance Comparison (RQ1) (1/3)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>MF-based</th>
<th></th>
<th>DL-based</th>
<th></th>
<th>FISSA vs. SASRec</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>BPRMF</td>
<td>FISM</td>
<td>FPMC</td>
<td>Fossil</td>
<td>GRU4Rec+</td>
</tr>
<tr>
<td>Beauty</td>
<td>Rec@10</td>
<td>0.2498</td>
<td>0.3533</td>
<td>0.2397</td>
<td>0.3570</td>
<td>0.2729</td>
</tr>
<tr>
<td></td>
<td>NDCG@10</td>
<td>0.1148</td>
<td>0.1942</td>
<td>0.1093</td>
<td>0.2108</td>
<td>0.1683</td>
</tr>
<tr>
<td>Games</td>
<td>Rec@10</td>
<td>0.3454</td>
<td>0.4791</td>
<td>0.3665</td>
<td>0.4800</td>
<td>0.4825</td>
</tr>
<tr>
<td></td>
<td>NDCG@10</td>
<td>0.1981</td>
<td>0.2631</td>
<td>0.2115</td>
<td>0.2708</td>
<td>0.2906</td>
</tr>
<tr>
<td>Steam</td>
<td>Rec@10</td>
<td>0.1023</td>
<td>0.3183</td>
<td>0.1735</td>
<td>0.2926</td>
<td>0.3177</td>
</tr>
<tr>
<td></td>
<td>NDCG@10</td>
<td>0.0468</td>
<td>0.1703</td>
<td>0.0849</td>
<td>0.1546</td>
<td>0.1707</td>
</tr>
<tr>
<td>Foursquare</td>
<td>Rec@10</td>
<td>0.2659</td>
<td>0.3977</td>
<td>0.3641</td>
<td>0.4223</td>
<td>0.4324</td>
</tr>
<tr>
<td></td>
<td>NDCG@10</td>
<td>0.1287</td>
<td>0.2025</td>
<td>0.1873</td>
<td>0.2303</td>
<td>0.2375</td>
</tr>
<tr>
<td>Tmall</td>
<td>Rec@10</td>
<td>0.1744</td>
<td>0.2149</td>
<td>0.1739</td>
<td>0.2303</td>
<td>0.3526</td>
</tr>
<tr>
<td></td>
<td>NDCG@10</td>
<td>0.0825</td>
<td>0.1043</td>
<td>0.0821</td>
<td>0.1142</td>
<td>0.2072</td>
</tr>
</tbody>
</table>

**Table:** Recommendation performance of our FISSA and eight baselines on five datasets.

- We can see that our FISSA achieves the best performance on all of the five datasets compared with all the baselines, which clearly demonstrates the superiority of our proposed model.
On average of the five datasets, our FISSA improves SASRec by 10.11% in terms of Rec@10 and 10.05% in terms of NDCG@10.

The second best performance is obtained by SASRec or CAR, which shows the advantage of the self-attention network for dynamic preference modeling.

CAR does not improve much over SASRec. Some possible reasons are as follows: 1) the global preference representation learned in CAR still maintains the causality constraint, which makes it redundant to the local one; and 2) the consistency-aware gating is designed for the user-generated item lists rather than for the interaction sequences, which means that it is more suitable for the prediction of longer sequences with different consistencies.
For the four MF-based methods, we observe that:

- FISM beats BPRMF on all the five sparse datasets, which indicates the effectiveness of the item similarity model for generating users’ global representations on these sparse datasets;

- FPMC surpasses BPRMF on three of the five datasets and Fossil performs the best among these four MF-based methods except on Steam, which shows the rationality to consider the high-order sequential information, as well as the importance to balance the dynamic short-term preferences with the static long-term preferences.

- Moreover, we notice that though having achieved very promising performance, GRU4Rec+ and Caser are still challenged by Fossil and FISM in some cases. This actually justifies our motivation to generate better global representation for DL-based sequential recommendation models.
Ablation Study (RQ2) (1/3)

We compare the **separate effect** of

- the local representation learning module (i.e., SASRec, $z^L_\ell = x^{(B)}_\ell$, denoted as ‘L_1’ for $B = 1$ and ‘L_3’ for $B = 3$)
- and the global representation learning module (i.e., $z^G_\ell = y$, denoted as ‘G’).

We also examine the **joint effect** ($B = 1$) with different approaches of combination, i.e.,

- normal summation (‘L+G’, $z^L_\ell + G = x^{(B)}_\ell + y$),
- weighted summation with consistency-aware gating as in CAR [He et al., 2020] (‘L+G+C’),
- and weighted summation with our proposed item similarity gating (‘L+G+I’, i.e., our FISSA).
### Ablation Study (RQ2) (2/3)

<table>
<thead>
<tr>
<th>Architecture \ Dataset</th>
<th>Beauty</th>
<th>Games</th>
<th>Steam</th>
<th>Foursquare</th>
<th>Tmall</th>
</tr>
</thead>
<tbody>
<tr>
<td>L_1</td>
<td>0.3268</td>
<td>0.5646</td>
<td>0.3699</td>
<td>0.4650</td>
<td>0.3874</td>
</tr>
<tr>
<td>L_3</td>
<td>0.3609</td>
<td>0.6009</td>
<td>0.3886</td>
<td>0.4808</td>
<td>0.4204</td>
</tr>
<tr>
<td>G</td>
<td>0.3407</td>
<td>0.5447</td>
<td>0.2981</td>
<td>0.4830</td>
<td>0.3423</td>
</tr>
<tr>
<td>L + G</td>
<td>0.3851</td>
<td>0.6301</td>
<td>0.4010</td>
<td><strong>0.5267</strong></td>
<td>0.4197</td>
</tr>
<tr>
<td>L + G + C</td>
<td>0.3727</td>
<td>0.6152</td>
<td>0.3973</td>
<td>0.5199</td>
<td>0.4198</td>
</tr>
<tr>
<td>L + G + I</td>
<td><strong>0.4046</strong></td>
<td><strong>0.6712</strong></td>
<td><strong>0.4305</strong></td>
<td>0.4972</td>
<td><strong>0.4237</strong></td>
</tr>
</tbody>
</table>

**Table**: Recommendation performance (Rec@10) in ablation studies with different architectures on five datasets.
Ablation Study (RQ2) (3/3)

We have the following observations.

- **G vs. L.** ‘L_3’ wins on most of these datasets, while ‘L_1’ performs worse on two datasets, which demonstrates the importance of the hierarchical structure and the competitive effectiveness of our global representation learning module.

- **‘L + G’ vs. L or G.** ‘L + G’ always significantly outperforms L or G (except on Tmall), which shows the complementary effect between the local and global representations in our FISSA.

- **‘L + G + I’ or ‘L + G + C’ vs. ‘L + G’**. The consistency-aware gating ‘C’ does not work well in our cases, e.g., it pulls down the results from normal summation ‘L + G’ on almost all the datasets. In contrast, adopting our item similarity gating ‘I’ improves the recommendation accuracy on four of the five datasets. These results show that our gating network that concerns about the candidate item is more effective in balancing the local and global representations for sequential recommendation.
Quantitative Study (RQ3) (1/3)

Figure: Recommendation performance and our FISSA with different dimensionalities $d$ on five datasets ($B = 1$).
Figure: Recommendation performance (Rec@10) of SASRec and our FISSA with different numbers of blocks $B$ on five datasets ($d = 50$).
Ablation Study (RQ2) (3/3)

- From the results with different values of $d$, we can see that our FISSA achieves better results as the dimensionality $d$ gets bigger on Games and Steam, but is more easier to become overfitting on Beauty, Foursquare and Tmall, for which $d = 40$, $d = 30$ and $d = 40$ perform the best, respectively.

- From the results with different values of $B$, we can see that unlike SASRec, setting the number of blocks $B = 2$ is enough for our FISSA to achieve the best results in most cases (except on Tmall), and adopting more blocks may backfire. This is because that though the hierarchical structure is still useful, the global representation learned in our FISSA is actually a novel substitute for the long-term transitions learned in the bottom block of SASRec.
Figure: Recommendation performance of having and not having the causality constraint for the global representation leaning ($B = 1, d = 50$). Note that ‘G’ and ‘L+G’ denote the global representation learning module ($z^G_\ell = y$) and the combination of the local and global representation modules ($z^{L+G}_\ell = x^{(B)}_\ell + y$), respectively, and the other two architectures are achieved by replacing the proposed global representation module ‘G’ with its causality-constrained version as ‘G w/ causality’, i.e., $y'_\ell = LBA(E_{1:\ell})$. 
Though a relatively global representation that only considers the historical interactions may be more suitable than a time unaware one (i.e., ‘G w/ causality’ is better than ‘G’ on Beauty, Games, Steam and Tmall), when joined with a well-established local representation learning model (also with causality), the causality consideration for global representation learning becomes redundant, i.e., ‘L+G’ is better than ‘L + (G w/ causality)’ on all the five datasets. This also demonstrates the advantage of introducing the future information for global representation learning in our FISSA.
Exploratory Study (RQ4) (3/4)

Figure: Recommendation performance (Rec@10) of some changes to the gating MLP (see Eq.(9), $B = 1$, $d = 50$). Note that $y$ and $m_s$ denote the representation of the user’s global preference and the recently interacted item, respectively, which are two parts of the inputs of Eq.(9), and $g$ is an output in the form of a 1-D vector.

Note that $g \in \mathbb{R}^{1 \times d}$ for a feature-level gating is obtained by setting $W_G \in \mathbb{R}^{3d \times d}$ and $b_G \in \mathbb{R}^{1 \times d}$ in Eq.(9), which provides different weights for different dimensions.
Experiments

Exploratory Study (RQ4) (4/4)

Introducing a single type of historical representation (i.e., the global preference $y$ or the recent interaction $m_{s\ell}$) into the gating function is usually enough to achieve the excellent results. Specifically, on Beauty and Steam, introducing $y$ only is more effective, while on the other three datasets, introducing $m_{s\ell}$ only is helpful. Considering that sometimes (e.g., on Games) introducing both $y$ and $m_{s\ell}$ still increases the performance, we keep both $y$ and $m_{s\ell}$ in the standard item similarity gating function for universality.

A feature-level gating brings worse results on four datasets (except on Foursqaure), though it is expected to refine the weights for different dimensions. Actually, in our experiments we find that a feature-level gating makes the model more unstable, which tends to be trapped into local optimal.
Conclusions

- We propose a novel solution named **fusing item similarity models with self-attention networks (FISSA)** for sequential recommendation, which contains a local representation learning module, a global representation learning module and a gating module to balance these two kinds of representations.

- We design an attentive version of FISM for **global representation learning** to fill the gap of the deficient consideration on global preference learning in most DL-based sequential recommendation methods.

- We design a **gating** network, which takes the relationship among the candidate item, the recent interaction and the global preference of each user into consideration, to deal with the possible uncertainty of users’ intention.

- Extensive empirical studies on five public datasets show that our FISSA achieves the **state-of-the-art performance** compared with several very competitive baselines. Some ablation and quantitative studies showcase the rationality of our design of the global and gating modules.
In the future, we plan to explore the application of federated machine learning on sequential recommendation and generalize our FISSA to a privacy-aware version. Additionally, more contextual information such as knowledge graph of the items is also worth being incorporated into our FISSA.
Thank you!

- We thank the anonymous reviewers for their expert and constructive comments and suggestions.
- We thank the support of National Natural Science Foundation of China Nos. 61872249, 61836005 and 61672358.
- If you have any questions, please feel free to contact us.
Fusing similarity models with Markov chains for sparse sequential recommendation.

NAIS: Neural attentive item similarity model for recommendation.
*IEEE Transactions on Knowledge and Data Engineering*, 30(12):2354–2366.

Consistency-aware recommendation for user-generated item list continuation.
In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM ’20, pages 250–258.

Recurrent neural networks with top-k gains for session-based recommendations.

FISM: Factored item similarity models for top-N recommender systems.
In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’13, pages 659–667.

Self-attentive sequential recommendation.
In *Proceedings of the 18th IEEE International Conference on Data Mining*, ICDM ’18, pages 197–206.

Effective approaches to attention-based neural machine translation.

BPR: Bayesian personalized ranking from implicit feedback.


