IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 December 2024, accepted 23 December 2024, date of publication 3 January 2025, date of current version 15 January 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3525812

RESEARCH ARTICLE

Exploring the Side-Information Fusion for Sequential Recommendation

SEUNGHWAN CHOI[1], DONGHOON LEE[1], HYEOUNGGUK KANG[1], AND HYUNSOUK CHO[1,2]

[1]Department of Artificial Intelligence, Ajou University, Suwon-si 16499, Republic of Korea
[2]Department of Software, Ajou University, Suwon-si 16499, Republic of Korea

Corresponding author: Hyunsouk Cho (hyunsouk@ajou.ac.kr)

This work was supported in part by Institute of Information and communications Technology Planning and Evaluation (IITP) under the Artificial Intelligence Convergence Innovation Human Resources Development (IITP-2025-RS-2023-00255968) grant funded by the Korea government (MSIT), and in part by the National R&D Program through the National Research Foundation of Korea (NRF) funded by Ministry of Science and ICT (RS-2024-00407282).

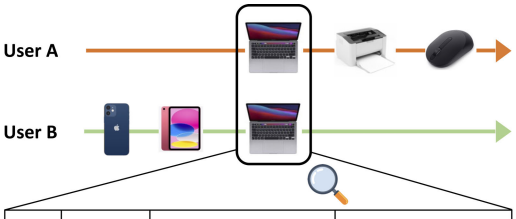
ABSTRACT Side information fusion for sequential recommendation aims to mitigate the data sparsity problems by leveraging the additional knowledge besides item ID. While most state-of-the-art methods devised elaborate fusion methods to incorporate side-information, they overlooked that there are distinct characteristics of the side-information, which can be grouped into two types: *item attribute* (e.g., category and brand) and *user behavior* (e.g., position and rating). In this paper, we argue that *attribute* information and *behavior* information are fundamentally different in relation to the item. The former is inherent to the item, whereas the latter is not. Based on this intuition, we systematically analyzed the previous fusion approach and introduced a comprehensive framework for two types of side information. Finally, we devise self-supervised objectives fitting for each type of side-information in a multi-task training scheme. To validate the effectiveness of our proposed method, we conduct experiments across various domains.

INDEX TERMS Side-information fusion, self-supervised learning, sequential recommendation.

I. INTRODUCTION

A sequential recommendation (SR) system aims to capture users' evolving preferences based on their past behaviors. Given its extensive practical applications across online platforms such as e-commerce and streaming services, this research topic has emerged as a significant area of research. Many scholars have strived to identify users' dynamic preferences within user interaction sequences. Followed by the remarkable success of Transformer [1] in NLP, the self-attention-based methods [2], [3], [4] showed impressive performance in finding the users' preferences. However, the traditional SR models, which rely on item ID in item sequence, have suffered from data sparsity problems [5], [6].

A common solution to the data sparsity problem is integrating additional knowledge (referred to as side-information) into a sequential recommendation system. The primary assumption is that the additional information will help the system understand the contextual meaning of both the item and user besides collaborative signal from user-item interaction, ultimately leading to better recommendations.



User	Item ID	Item attribute		User behavior	
		Category	Brand	Rating	Position
A	253	Electronics	Apple	★★	1st
B				★★★★	3rd

FIGURE 1. An illustration of two types of side-information, where both users purchased the same item (i.e., MacBook). The item attribute remains static relative to the item ID, whereas user behavior can vary.

The associate editor coordinating the review of this manuscript and approving it for publication was Chien-Ming Chen.

To effectively incorporate the side-information many researchers [7], [8], [9], [10], [11], [12] have proposed fusion strategies. This strategy often involves maintaining an independent representation for individual side-information and later fusing them with item ID and other side-information within the neural network. While this branch was explored thoroughly, they have not yet been fully explored in the direction of the side-information's characteristics.

In this paper, we argue that the side-informationcan be categorized into two types[1] (item attribute and user behavior), each having a different relationship with the item. Figure 1 illustrates these properties. For example, item attributes, such as brand and categories, demonstrate item ID homogeneity. This homogeneity implies that the item ID will always have the same information when the item ID is the same. Meanwhile, user behavior, including rating and position, exhibits item ID heterogeneity. This heterogeneity signifies that the type contains unique semantic information even when the item IDs are identical.

To exploit the property, we propose **COM**prehensive framework for side-**I**nformation **F**usion for Sequential Recommendation (**COMIF**). We analyze state-of-the-art fusion strategies and categorize them into *embedding-level* and *attention-level* fusion. Specifically, we found that *embedding-level fusion* architecturally causes the correlation between item ID and involved side-information while *attention-level fusion* does not. Based on the findings, we select the optimal fusion for each type of side-information. We utilize the embedding-level fusion for item attributes to apply its homogeneous nature. In contrast, we use the attention-level fusion for user behavior to employ its heterogeneous nature. Also, to effectively learn user representation from the side-information, we devise a self-supervised objective fitting for each type of side-information and construct a multi-task training environment.

To validate the proposed method's effectiveness, we conducted extensive experiments on three real-world recommendation datasets in different domains. Our experimental results show that our method outperforms several competitive methods and helps alleviate the cold-start problem.

In summary, our key contributions are as follows:

1) We provide a novel view of distinguishing item attributes and user behavior in terms of how they are related to the item ID. We also analyze the existing fusion approaches and select optimal fusion for each type of side-information in terms of correlation between item ID and side-information.
2) Based on this analysis, we propose a holistic fusion approach to exploit the nature of side-information. Furthermore, we devise two distinct self-supervised objectives fitting for the separation of side-information.
3) We have conducted experiments on various datasets of different domains, showing our method's superiority. We have also found that our strategy exhibits notable strength in alleviating cold-start problems compared to the state-of-the-art baselines.

[1]Some previous studies [13], [14] have covered another type of side-information, *user demographic*. However, this information (such as age, occupation, and sex) is omitted due to privacy issues.

II. PRELIMINARIES

This section formalizes our research problem, provides the base architecture required for our method, and categorizes the state-of-the-art fusion methods into two types.

A. PROBLEM DEFINITION

We first elaborate on the main objective of sequential recommendation with side-information. Let $\mathcal{U}$, $\mathcal{I}$, and $\mathcal{V}$ denote a set of users, items, and interactions. Each user $u \in U$ has her/his chronological sequence: $[v_u^1, v_u^2, \ldots, v_u^n]$, where v_u^j represents the j-th interaction that the user u has made. This interaction becomes: $v_u^k = [id^k, s^k]$ where id^k denotes item ID of k-th interaction, and s^k represents side-information of k-th interaction. Given n historical interactions of user u, our goal is to predict the item id that the user u will likely interact with.

B. SELF-ATTENTION FOR SR

Since our framework is built on the self-attention mechanism, we briefly introduce it, taking the SASRec [2] as an example. This model consists of three parts: 1) embedding layer, 2) stacked self-attention blocks, and 3) prediction layer.

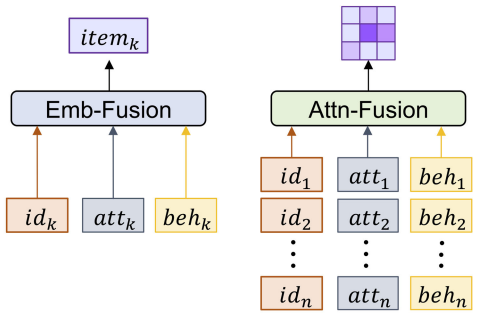
At the embedding layer, item ID look-up embedding table $M_I \in \mathbb{R}^{|I|\times d}$ is maintained. When given a user interaction sequence of length n, an item ID embedding matrix $E_{id} \in \mathbb{R}^{n\times d}$ is retrieved by look-up operation. The learnable position embedding matrix $P \in \mathbb{R}^{n\times d}$ is added to the item ID embedding matrix to construct an integrated embedding matrix $E_i = E_{id} + P$, which enters the first self-attention block.

The sequence encoder is stacked with several self-attention blocks. Each block consists of a self-attention layer and a feed-forward network. The self-attention layer aggregates items of different relevance in sequence, using dot-products between the items to calculate their relevance. The relevance calculations are done as follows:

$$\begin{aligned} \text{Attn} &= (HW_Q)(HW_K), \\ \text{SA} &= \text{softmax}\left(\frac{\text{Attn}}{\sqrt{d}}\right)(HW_V) \end{aligned} \tag{1}$$

where $W_Q, W_K.W_V \in \mathbb{R}^{d\times d}$ are learnable weight matrices, and $H \in \mathbb{R}^{n\times d}$ is the input for the self-attention block. This self-attention layer is followed by a feed-forward network, which introduces the non-linearity into the self-attention block. The self-attention mechanism of Eq.1 can be extended to multi-head attention [1], while other components in the Transformer are also used, including residual connection, layer normalization, and dropout.

At the prediction layer, relevance scores $r_{(u,i)}$ get calculated with the user u's interaction sequence of length n and the



(a) Embedding-level fusion (b) Attention-level fusion

FIGURE 2. Two types of side-information fusion.

candidate item i. We take the hidden states of the final block and use n-th hidden state h_n as user sequence representation. This can be formulated as:

$$r_{(u,i)} = e_i \cdot {h_n}^\top \tag{2}$$

C. SIDE-INFORMATION FUSION FOR SR

Since side-informationcan help the recommendation system understand user preferences, various methods were proposed to introduce them, including fusion approaches. We investigated previous side-information fusion approaches and categorized them into *embedding-level* and *attention-level fusion* in terms of the moment of fusion in an attention-based neural network.

1) EMBEDDING-LEVEL FUSION

Embedding-level fusion integrates side-information embedding with the item ID embedding. Intuitively, combining them can enhance item representation by injecting the contextualized semantic into the representation. In this case, side-information look-up embedding tables additionally need to be retained.

Specifically, given user interactions with item ID and its side-information, it constructs item ID embedding matrix E_{id} and side-information embedding matrices E_s by applying a look-up operation. Those embedding matrices get integrated with the fusion operation described in Figure 2.a. The fusion operation $\mathcal{F}_{\text{emb}}$ consists of addition, concatenation, and gating, which are studied in [15] and can be described as: $E_i = \mathcal{F}_{emb}\left(E_{id}, E_s\right)$. The fused embedding matrix will be fed into the sequence encoder.

The *embedding-level fusion* can be regarded as extension of positional encoding used in Transformer [1]. This is because it becomes equivalent to the positional encoding when the involved side-information is limited to the position of an item within a sequence and the fusion operation is addition.

The recent researches [11], [12] have expanded this fusion method to the prediction layer. *Embedding-level fusion* at the layer integrates side-information with the target item ID before calculating the relevance score. This process can be defined as: $r_{(u,i)} = e_i \cdot {h_n}^\top$ and $e_i = \mathcal{F}_{emb}(e_{id}, e_s)$.

2) ATTENTION-LEVEL FUSION

Attention-level fusion separately computes attention score matrices for each side-information and item ID, which will then be aggregated as shown in Figure 2.b. This method claims that the separate calculation constructs elaborate attention matrices. Specifically, DIF-SR [10] observed that *embedding-level fusion* limits the expressiveness of the attention score matrices and demonstrated that it causes compound embedding space, where combined embeddings from distinct information sources inevitably attend to unrelated information. Although these various fusion methods show effectiveness for incorporating the side-information, they do not distinguish the types of side-information: homogeneous and heterogeneous information. Thus, naively choosing one among the two fusion methods will result in missing opportunities to explore the potential of the two factions.

III. THEORETICAL ANALYSIS FOR FUSION

To fully understand which fusion method can exploit the best of both types of side-information, we first analyze the two fusion methods in terms of how correlations[2] are made between the item ID and its side-information. Then, according to the homogeneous and heterogeneous nature of side-information, we confirm the appropriate fusion for both types of side-information.

A. CORRELATION AT SELF-ATTENTION LAYER

We can derive that *embedding-level fusion* method strengthens the correlation between item ID and side-information based the finding of Ke et al. [16]. They found that positional encoding in vanilla Transformer [1] brings a correlation between word embedding and positional embedding during attention calculation. Extending this finding to *embedding-level fusion* methods, we can obtain that the fusion method plays a similar role.

Specifically, consider the calculation of the attention score α_{jk} between j-th and k-th interaction using *embedding-level fusion*:

$$\begin{aligned}\alpha_{jk} &= \frac{\left(\left(e^j_{id} + e^j_s\right) W_Q\right)\left(\left(e^k_{id} + e^k_s\right) W_K\right)^T}{\sqrt{d}} \\ &= \frac{\left(e^j_{id} W_Q\right)\left(e^k_{id} W_K\right)^T}{\sqrt{d}} + \frac{\left(e^j_{id} W_Q\right)\left(e^k_s W_K\right)^T}{\sqrt{d}} \\ &\quad + \frac{\left(e^j_s W_Q\right)\left(e^k_{id} W_K\right)^T}{\sqrt{d}} + \frac{\left(e^j_s W_Q\right)\left(e^k_s W_K\right)^T}{\sqrt{d}}\end{aligned} \tag{3}$$

where e^k_{id}, e^k_s are the item ID embedding and the side-information embedding of k-th interaction. For simplicity, we choose addition as the fusion operation in this section.[3]

[2]The term "correlation" mainly refers to the dot product between item and side-information.

[3]We also derive the same analysis for the other two fusion operations in the Appendix.

When investigating the equation 3, we can infer that the *embedding-level fusion* architecturally reinforces the correlation. The second and the third terms in equation 3 represent the dot product of the item ID and the side-information. Thus, these two terms enable the correlation between an item and side-information to participate in the self-attention operation. In contrast, the *attention-level fusion* method weakens the correlation because it removes the correlation by separately computing the attention scores.

B. CORRELATION AT PREDICTION LAYER

Moreover, we have found that *embedding-level fusion* also can increase the correlation at the prediction layer. Taking inspiration from the findings of Yang et al. [17], we derive the followings from equation 2:

$$
\begin{aligned}
r_{(u,i)} &= e_i \cdot h_n^\top \\
&= (e_{id} + e_s) \cdot \left(\sum_{k=1}^{m} e_{id}^k W_X^k + \sum_{k=1}^{m} e_s^k W_X^k + \sum_{l=1}^{L} b^l W_B^l \right)^\top \\
&= e_{id} \cdot (\sum_{k=1}^{m} e_{id}^k W_X^k)^\top + e_{id} \cdot (\sum_{k=1}^{m} e_s^k W_X^k)^\top \\
&\quad + e_s \cdot (\sum_{k=1}^{m} e_{id}^k W_X^k)^\top + e_s \cdot (\sum_{k=1}^{m} e_s^k W_X^k)^\top \\
&\quad + (e_{id} + e_s) \cdot (\sum_{l=1}^{L} b^l W_B^l)^\top
\end{aligned} \tag{4}
$$

where b^l denotes the parameter of additive bias at the l-th layer of the model, e_{id}^k denotes k-th item ID embedding, and e_s^k denotes embedding of the k-th side-information. W_X^k and W_B^l is the linear transformation matrix for k-th item embedding and additive bias at l-th layer, respectively. We can see that *embedding-level fusion* increases correlation between item ID and side-information in prediction layer because the dot-product similarity between item ID and side-information exists in relevance score calculation.

We have analyzed that *embedding-level fusion* reinforce correlation between item ID and side-information while *attention-level fusion* does not. Thus, to improve the homogeneity, we should employ *embedding-level fusion* with item attributes while *attention-level fusion* should be applied for user behavior to handle the heterogeneous information.

IV. METHODOLOGY

In this section, we propose a comprehensive fusion method based on our analysis of optimal fusion approaches for each type of side-information. We introduce the method into SASRec [2], a representative example of self-attention-based architecture. Initially, we apply the *embedding-level fusion* approach for the item attribute, fusing the side-information with the item ID at both the embedding and prediction layers. We adopt the *attention-level fusion* approach for user behavior information, computing separate attention scores for item ID and user behavior and aggregating them.

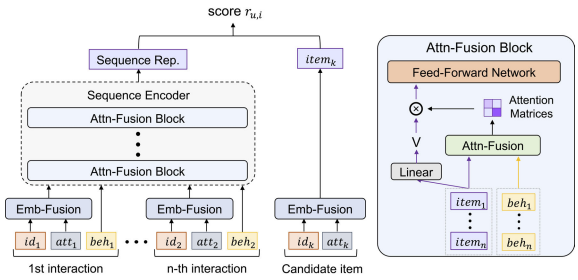


FIGURE 3. Overall framework.

Moreover, we introduce two self-supervised learning tasks specific to these two types of side-information.

A. SEQUENCE ENCODER

According to our intuition in Fig. 1, we split side-information s into two parts: attribute a and behavior r. For simplicity, we utilize a single type of attribute and behavior. We can extend a single type to multiple types. Separate look-up embedding matrices are maintained for the item ID id, item attribute a, and user behavior r. Given a user interaction sequence, we construct the embedding matrix for the item ID and each type of side-information associated with the user sequence.

1) EMBEDDING-LEVEL FUSION FOR ITEM ATTRIBUTE

Integrating item attributes with *embedding-level fusion* can be demonstrated as follows:

$$E_i = \mathcal{F}_{emb}(E_{id}, E_a) \tag{5}$$

After the integrated item embedding matrix E_i is constructed, it enters the sequence encoder with the behavior embedding matrices. However, while the former gets an update at each transformer block, the latter remains unchanged.

2) ATTENTION-LEVEL FUSION FOR USER BEHAVIOR

Instead of the original self-attention layer of SASRec, we introduce the *attention-level fusion* method to the self-attention layer but restrict its participant to the user behavior. The relevant equations are:

$$
\begin{aligned}
\text{Attn}_i &= (H_i W_i^Q)(H_i W_i^K)^\top, \\
\text{Attn}_r &= (E_r W_r^Q)(E_r W_r^K)^\top, \\
\text{Attn}_{fusion} &= \mathcal{F}_{attn}(\text{Attn}_i, \text{Attn}_r) \\
\text{SA}_{\text{fusion}} &= \text{softmax}\left(\frac{\text{Attn}_{fusion}}{\sqrt{d}} \right)(H_i W_V)
\end{aligned} \tag{6}
$$

where H_i, E_r stands for integrated item representation matrix and behavior embedding matrices. At first transformer blocks, we set $H_i = \hat{E}_i$. As mentioned earlier, applying multi-head attention to *attention-level fusion* is also possible.

B. NEXT-ITEM PREDICTION

Owing to the use of *embedding-level fusion* with the item attribute information, the target items are integrated with their

(a) Item-Attribute Alignment (b) User Preference Alignment

FIGURE 4. Illustration of positive pair and negative pair in both self-supervised learning tasks.

attributes as:

$$r_{(u,i)} = e_i \cdot h_n{}^{\top}$$
$$e_i = \mathcal{F}_{emb}(e_{id}, e_a) \tag{7}$$

Therefore, the cross-entropy loss $\mathcal{L}_{rec}$ for recommending ground-truth item i to each user u can be expressed as:

$$\hat{y}_{(u,i)} = \frac{\exp(r_{(u,i)})}{\sum_{i' \in \mathcal{I}} \exp(r_{(u,i')})}$$
$$\mathcal{L}_{rec} = -\mathbb{E}_{u \in \mathcal{U}}[y_{(u,i)} \log(\hat{y}_{(u,i)})] \tag{8}$$

C. SELF-SUPERVISED LEARNING FOR SIDE-INFORMATION

Even though side-information fusion aids SR models in understanding user preference, the models may still suffer from data sparsity problems. Since next-item prediction, the only training task, depends on interaction density, the models might struggle to capture user preference when the training data is sparse. To address this issue, we have employed two additional self-supervised learning tasks.

1) ITEM-ATTRIBUTE ALIGNMENT

The first task focuses on the item attributes, which possess a homogeneous trait to the item ID as illustrated in Figure 1. We propose Item-Attribute Alignment (IAA) to strengthen further the correlation between the item ID and its associated attribute in representation space. Unlike prior work [8], which utilized the output of transformer encoder as item representation, we use pure item embedding because noisy items(e.g., accidental interactions or false positive interactions) may exist in the same sequence [18], [19], [20], [21]. For a given item *id* and its corresponding attributes a, their embeddings are derived by inputting item i into the designated embedding layer. The loss function for IAA is defined as:

$$L_{IAA} = \mathbb{E}_{i \in I}\left[-\log \frac{\exp\left(e_{id} \cdot e_a^{\top}\right)}{\sum_{\tilde{a} \in \mathcal{A}} \exp\left(e_{id} \cdot e_{\tilde{a}}^{\top}\right)}\right] \tag{9}$$

where $\mathcal{A}$ denotes attribute set of entire items.

2) USER PREFERENCE ALIGNMENT

To utilize the user behavior in self-supervision tasks, we propose User Preference Alignment (UPA). A fundamental intuition behind our approach is that behavior information contains crucial knowledge for user preference, affecting next-item interaction [9], [22], [23]. Therefore, when behavior information in an item sequence changes from the original one, it may show different preferences accordingly. Specifically, We treat the original sequence as a positive sample and the random sequence as a negative. Our objective then becomes predicting the positive sample within the set of k random behavior sequences plus the original sequence based on the subsequent item. The formulation for this self-supervised loss is given by:

$$L_{UPA} = \mathbb{E}_{u \in \mathcal{U}}\left[-\log \frac{exp(e_i \cdot h_{pos}{}^{\top})}{\exp(e_i \cdot h_{pos}{}^{\top}) + \sum_{j=1}^{k} \exp(e_i \cdot h_j{}^{\top})}\right] \tag{10}$$

where e_i denotes integrated representation of next-item, and h^{pos}, h_j denote representation of original sequence and representation of the j-th sequence among k random sequences, respectively. In conclusion, our training schema involves a total of three loss functions, consisting of recommendation loss $\mathcal{L}_{rec}$ and two SSL losses ($\mathcal{L}_{IAA}$, $\mathcal{L}_{UPA}$). We define our joint objective $\mathcal{L}$ with the balancing parameters (λ_1, λ_2):

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda_1 \mathcal{L}_{IAA} + \lambda_2 \mathcal{L}_{UPA} \tag{11}$$

V. EXPERIMENTS

In this section, we discuss the following research questions:

- RQ1: (Empirical fusion analysis) Does our fusion analysis have an empirical background?
- RQ2: (Overall performance) Does our method outperform the state-of-the arts basic SR methods and SR methods with side-information?
- RQ3: (Ablation study) How do the different components and hyperparameters affect our model?
- RQ4: (Cold-start problem) Could our method alleviate the cold-start problem and show consistent effectiveness on different groups of users?
- RQ5: (Computational Efficiency) Does our method accomplish state-of-the-arts performance with comparable computational efficiency?

A. DATASET

Our experiments were conducted with the publicly available and widely used recommendation dataset.

- **Amazon Beauty, Sports:** These datasets [24] were created with the reviews posted over the famous e-commerce platform Amazon. For the two datasets, we used user-item interaction and two side information: brands for item attribute information and position for user behavior information.
- **Yelp:** Yelp is a collection of reviews for businesses such as restaurants, cafeterias, etc. Because of its large size, we experimented on only the transaction records after Jan. 1st, 2019. we utilize the city of businesses as item attribute information and position as user behavior information.

Following the pre-processing method of [8] and [10], items and users with interaction of less than five were removed. We also treated all interactions as implicit feedback. The

TABLE 1. Statistics of dataset after preprocessing.

Dataset	Beauty	Sports	Yelp
# Users	22,363	35,598	30,499
# Items	12,101	18,357	20,038
# Categories	355	1,876	1,607
# Brands (Cities)	2,076	2,411	327
# Avg. Interaction Length / User	8.9	8.3	10.4
# Avg. Interaction Size / Item	16.4	16.1	15.8
# Interaction	198,502	296,337	317,182
Sparsity	99.93%	99.95%	99.95%

detailed statistics of all three datasets that went through pre-processing are summarized in Table 1.

B. EVALUATION METRICS

We used the leave-one-out strategy for evaluation, following the prior works [2]. This strategy selects the final two items of the interaction sequence for validation and testing data, while the rest are used to train the recommendation models. The baselines are scored and compared by top-K Recall (R@K) and top-K Normalized Discounted Cumulative Gain (NDCG@K) with K chosen from {10, 20}. We calculated the ranking of the ground-truth item in an item set ordered by its relevance score. Following [15], which observed that a negative sampling strategy might result in severe bias, we evaluate our model performance in a full-ranking manner.

C. IMPLEMENTATION DETAILS

All the baselines and ours, excluding STOSA and DLFS-Rec, are implemented based on the popular recommendation framework RecBole [25]. For STOSA[4] and DLFS-Rec,[5] we use the code provided by the corresponding authors. We train all the methods, excluding STOSA, with Adam optimizer for 200 epochs. Exceptionally, we train STOSA with Adam optimizer for 500 epochs because of an underfitting issue. We adopt the early stopping strategy that model optimization stops when the validation Recall@20 does not increase for ten epochs, with a batch size of 512 and a learning rate 1e-4. The fusion operation is chosen from addition, concatenation, and gating from NOVA [15]. We choose addition for *embedding-level fusion* and concatenation for *attention-level fusion* based on empirical experiments. The hidden size of our model and other baselines are all set to 128. For other hyper-parameters, we apply grid-search to find the best configuration for our model and baseline that involves the following hyper-parameters. The search space is: number of layers $\in \{2, 3, 4\}$, number of heads $\in \{2, 4, 8\}$ and balance parameter $\lambda_1, \lambda_2 \in \{0.4, 0.8, 1.2, 1.6\}$.

D. BASELINES

We compare our method with two groups of Sequential Recommendation(SR) baselines: SR models without side-information and SR models with side-information. Taxonomy for baselines with side-information and our proposed method are represented in Table 2.

TABLE 2. Taxonomy of the baselines and proposed method.

Method	Attribute	Behavior	SSL
S3-Rec [8]	-	-	Attribute
SASRecF [2]	**Emb**	Emb	-
NOVA [15]	**Emb**	Emb	-
DIF [10]	Attn.	**Attn**	Attribute
DLFS [12]	**Emb**	Emb	-
COMIF (Ours)	**Emb**	**Attn**	**Attribute + Behavior**

- **SR models without side-information**
  - **GRU4Rec [26]**: A session-based recommendation system that uses Gated Recurrent Units to capture sequential patterns.
  - **SASRec [2]**: A method that firstly adopts a self-attention mechanism for sequential recommendation.
  - **BERT4Rec [3]**: A sequential recommendation method that utilizes bi-directional self-attention employing Cloze task.
  - **STOSA [27]**: A self-attention-based method for sequential recommendation task, which represents each item as a Gaussian distribution and utilizes Wasserstein self-attention to characterize item transition patterns.
- **SR models with side-information**
  - **S3-Rec [8]**: A self-supervised sequential recommendation model to learn relationships among items, attributes, subsequences, and sequences with four auxiliary self-supervised objectives.
  - **SASRecF**: An *embedding-level fusion* method, which extends SASRec. Using concatenation operation, it integrated item attributes with item ID at the embedding layer.
  - **NOVA [15]**: An *embedding-level fusion* model where the item attribute is fused at the embedding layer before the attention score calculation. The fused representation is used for Query and Key, while the item ID representation is used for Value.
  - **DIF-SR [10]**: A *attention-level fusion* model that operates decoupled self-attention, where attention score for item embedding and side information gets calculated separately.
  - **DLFS-Rec [12]**: An *embedding-level fusion* model, which embeds item ID and side-information as stochastic distribution.
  - **MSSR [28]**: An *embedding-level fusion* model, which exploits association between item ID sequence and side-information sequence by calculating multi-sequence integrated attention scores.

E. EMPIRICAL FUSION ANALYSIS (RQ1)

We theoretically analyzed each fusion method's traits and concluded that user behaviors are appropriate for

[4]https://github.com/zfan20/STOSA
[5]https://github.com/zxiang30/DLFS-Rec

TABLE 3. Overall performance. The best and second-best results are bold and underlined, respectively. "Improve." is the relative improvement against the second-best baseline performance.

Dataset	Metric	GRU4Rec	SASRec	BERT4Rec	STOSA	SASRecF	S3-Rec	NOVA	DIF	DLFS-Rec	MSSR	Ours	Improv.
Beauty	R@10	0.0537	0.0859	0.0501	0.0669	0.0797	0.0869	0.0872	0.0876	0.0891	0.0916	**0.0979**	9.87%
	R@20	0.0808	0.1222	0.0763	0.0944	0.1096	0.1263	0.1240	0.1256	0.1295	0.1293	**0.1449**	11.89%
	NDCG@10	0.0282	0.0420	0.0264	0.0379	0.0480	0.0442	0.0435	0.0431	0.0472	0.0453	**0.0474**	0.42%
	NDCG@20	0.0351	0.0512	0.0330	0.0448	0.0555	0.0542	0.0527	0.0526	0.0574	0.0549	**0.0592**	3.14%
Sports	R@10	0.0286	0.0507	0.0217	0.0344	0.0426	0.0503	0.0501	0.0521	0.0494	0.0530	**0.0573**	9.98%
	R@20	0.0459	0.0732	0.0379	0.0515	0.0616	0.0749	0.0744	0.0766	0.0751	0.0785	**0.0847**	10.57%
	NDCG@10	0.0144	0.0238	0.0108	0.0196	0.0250	0.0244	0.0234	0.0242	0.0262	0.0247	**0.0268**	2.29%
	NDCG@20	0.0187	0.0294	0.0149	0.0239	0.0298	0.0306	0.0295	0.0304	0.0327	0.0311	**0.0337**	3.06%
Yelp	R@10	0.0361	0.0637	0.0377	0.0291	0.0462	0.0652	0.0658	0.0645	0.0371	0.0659	**0.0704**	6.99%
	R@20	0.0601	0.0934	0.0610	0.0483	0.0754	0.0968	0.0959	0.0955	0.0616	0.0965	**0.1038**	7.23%
	NDCG@10	0.0177	0.0394	0.0201	0.0148	0.0236	0.0394	0.0402	0.0398	0.0183	0.0403	**0.0419**	4.23%
	NDCG@20	0.0237	0.0469	0.0259	0.0196	0.0309	0.0473	0.0477	0.0475	0.0245	0.0480	**0.0503**	5.45%

TABLE 4. Performance of fusion method for both types of side-information. The metric is Recall@20.

Side-info.	Fusion	Beauty	Sports	Yelp
Attribute	Emb	**0.1400**	**0.0814**	**0.0987**
	Attn	0.1234	0.0740	0.0931
Behavior	Emb	0.1223	0.0751	0.0960
	Attn	**0.1251**	**0.0758**	**0.0968**

TABLE 5. Performance comparison of each side-information.

Side-info.	Beauty		Sports		Yelp	
	R@20	NDCG@20	R@20	NDCG@20	R@20	NDCG@20
None	0.1224	0.0527	0.0739	0.0294	0.0944	0.0477
P	0.1236	0.0534	0.0768	0.0304	0.1038	0.0507
$P+B$	0.1449	0.0592	0.0847	0.0337	0.1038	0.0503
$P+B+C$	**0.1482**	**0.0607**	**0.0911**	**0.0363**	**0.1132**	**0.0537**

P: position, B: brand (city), C: category

attention-level fusion while item attributes are for *embedding-level fusion*. To validate our statement, we experiment with three datasets. For the item attributes, we have conducted experiments using categories and brands. We use cities instead of brands in the Yelp dataset. For user behaviors, we used the position of interaction within its sequence.

As shown in Table 4, *embedding-level fusion* method for integrating item attributes outperformed all the other three method's metrics. It achieved more than 10% higher in both R@10 and R@20. Besides the attribute part of side-information paired with *embedding-level fusion*, we can see that *attention-level fusion* is superior to other fusion methods when dealing with behavior information. This observation represents the empirical results supporting our analysis.

F. OVERALL PERFORMANCE (RQ2)

Table 3 illustrates the overall performance of all the baselines, showing the different side-information fusion methods' ability in recommendations. Within the four basic baselines, the attention-based method showed better performances in most metrics than GRU4Rec, implying the superiority of the attention mechanism when it comes to a sequential recommendation. Unlike other attention-based methods, BERT4Rec is behind GRU4Rec on many metrics, which was already seen in prior works [8], [10], [29] that BERT4Rec does not show strong performances against SASRec under the full-ranking evaluation setting.

We can also witness that the introduction of the side information overall helps find the users' sequential patterns. One notable mention is the performance difference between SASRec and SASRecF, which suggests that a simple introduction of the side information does not guarantee an increase in performance. This implies that sophisticated methods are almost mandatory.

As we have witnessed through previous experiments, the difference in the fusion approaches determines the recommendation performance. Baselines that utilized the *embedding-level fusion* method, DLFS-Rec and MSSR, showed better overall performance than those that utilized *attention-level fusion*, DIF, suggesting that using only *attention-level fusion* is less effective in recognizing the side-information. Another interesting fact is that almost all the fusion-utilizing baselines outperform S3-Rec, which indicates that direct fusion is better at leveraging side-information than self-supervised learning.

Finally, our proposed method gave the best next-item prediction results by a wide margin over others. This shows that our method of fusion, which involves splitting the side information into two, is valid. Our strategy ensured the enrichment of the item attribute while limiting the chance of causing undesirable correlations between item ID and user behavior at the attention calculation.

G. ABLATION STUDY (RQ3)

We analyze the effectiveness of the side-information types and the modules in the proposed model.

1) CONTRIBUTIONS OF DIFFERENT SIDE-INFORMATION

To discover the effect of each side-information, we experimented on four scenarios where some side-information is absent. Specifically, Table 5 shows the recommendation performance when the models are trained with position, categories, and brand information, without the brand, without brand and categories, or with no side information at all. We can see that each introduction of the side information leads to better performance. Specifically, Beauty and Sports received less improvement when introducing only the user behavior(i.e., position) than the Yelp dataset.

We hypothesize that the average interaction length of users in the dataset contributes to this occurrence. An increase in average interaction size likely enables the model to learn more various types of patterns, thereby enhancing

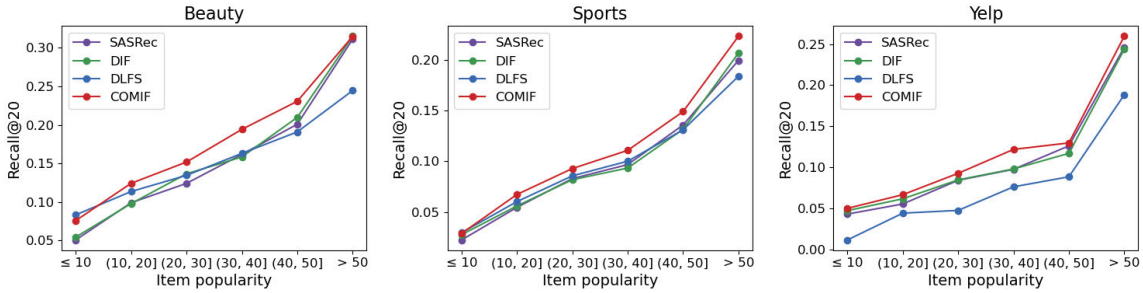


FIGURE 5. Recall@10 performance on different item popularity on all datasets.

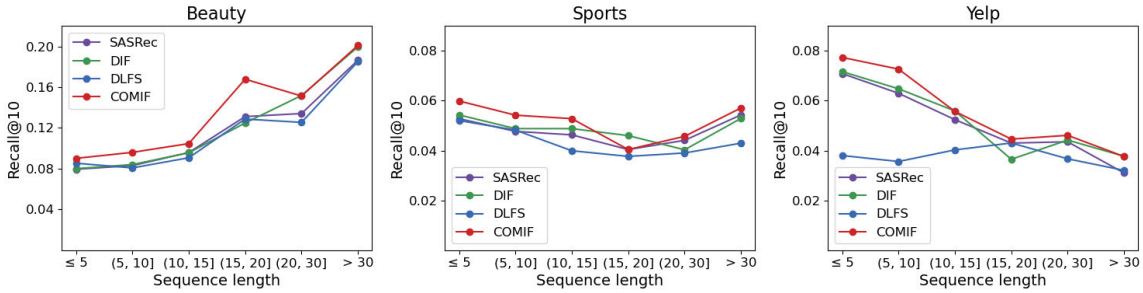


FIGURE 6. Recall@20 performance on different sequence lengths on all datasets.

TABLE 6. Performance of each module on Recall@20.

Fusion		IAA	UPA	Beauty	Sports	Yelp
Attn	Emb					
O	-	-	-	0.1238	0.0733	0.0951
O	O	-	-	0.1411	0.0825	0.1000
O	O	O	-	0.1441	0.0831	0.1014
O	O	O	O	**0.1449**	**0.0847**	**0.1038**

recommendation performance. As detailed in Table 1, the Yelp dataset exhibited the highest average interaction length compared to other datasets, supporting our hypothesis.

2) EFFECTIVENESS OF EACH COMPONENT

We experimented on models with each module removed to evaluate our modules' effectiveness. Table 6 demonstrates that the complete model outperforms all others. Most notably, the existence of the fusion strategies played a crucial role in the model's performances. Furthermore, using only *attention-level fusion* for both types of side-information resulted in the poorest performance. Proving our analysis's validity and showing that distinguishing side-information by its different properties enables the model to exploit them fully. Also, removing self-supervision tasks resulted in poor performance, making us conclude that both tasks are a great asset when using side-information.

3) HYPER-PARAMETER STUDY

Figure 7 presents the results of a grid search for loss hyperparameters λ_1 and λ_2 on three different datasets. Darker colors indicate higher performance in terms of R@20. The performance is mostly higher at below 0.8 in λ_2. We attribute this pattern to the shorter interaction length of every dataset. Because L_{UPA} is based on the negative sampling of random behavior sequence, the interaction length affects the quality of negative samples. It is worth noting that the lowest-performing settings are still comparable to or outperforming the best-performing baselines.

H. EFFECTIVENESS ACROSS DIFFERENT GROUPS (RQ4)

To see our method's abilities in different groups of users and items, we compared with three baseline methods, SASRec (with no side information), DIF-SR (*attention-level fusion*), and DLFS-Rec (*embedding-level fusion*), on their average recommendation performances with the cold item and user. We partitioned items based on their number of appearances in the training dataset, while users were partitioned based on their interaction size. To demonstrate the performances for each partition, the mean R@20 score was used.

1) PERFORMANCE IN TERMS OF ITEM POPULARITY

Figure 5 illustrates that the overall next-item prediction task becomes easier when the target items have been spotted more frequently while decreasing when dealing with unpopular items. However, compared to SASRec, the introduction of side-information showed improvements on cold items in most of the dataset. Our method also follows this trend while making comparable recommendation results at the extreme cold-item partition to the best-performing baselines.

Using side information, our method improved in every demographic on all three datasets compared to those that did not. From these observations, we successfully alleviated the cold-item problems and showed excellent stabilities, while the other side-information using baselines did not. We believe this was obtained due to our separation on side-information.

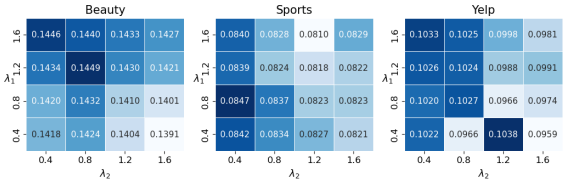


FIGURE 7. Heatmaps of hyperparameters λ_1 and λ_2 on Recall@20.

Compared to our method, DIF-SR and DLFS-Rec could not maintain the upper hand against SASRec on all partitions, indicating that the different item popularity requires different side-information understanding.

2) PERFORMANCE IN TERMS OF SEQUENCE LENGTHS

We have compared how well models capture sequential patterns on different groups of users based on item interaction length. Both DIF-SR and DLFS-Rec perform better than SASRec, demonstrating that the introduction of side-information helps find the user preference in short sequences. From Figure 6, we can see that our method showed dominance over all partitions, showing the superiority of our fusion strategy even in the smallest sizes of interaction sequences. Like the cold-item case, our method again outperformed SASRec, which does not use side-information, once again showing off our method's stabilities in handling different groups of users.

I. COMPUTATIONAL EFFICIENCY (RQ5)

Table 7 presents the computational efficiency comparison of various methods evaluated on an A5000 GPU with 24GB memory. All models are configured consistently with three layers and four attention heads, maintaining identical hyper-parameters as detailed in Section V-C. The results in Table 3 and Table 7 demonstrate a clear trade-off between performance and computational resources among recent sequential recommendation models. While state-of-the-art models DLFS and MSSR achieve competitive recommendation performance, they incur substantial computational overhead. In contrast, our proposed method achieves superior recommendation performance while maintaining moderate computational requirements. This efficiency is particularly noteworthy compared to DLFS's memory usage (4.6× higher) and MSSR's training time (1.5× longer).

TABLE 7. Comparison of the computational efficiency on the Beauty dataset, which involves measuring GPU memory usage (GB), the training time per epoch (s/epoch), and the inference time for all test users in seconds (s). For DLFS, we decrease the batch size from 512 to 64 at inference because of OOM issue.

Method	Memory	Training Time	Inference Time
NOVA (AAAI21)	1.44	7.66	0.72
DIF (SIGIR22)	2.08	11.67	1.06
DLFS (Recsys23)	15.64	11.46	57.64
MSSR (WSDM24)	6.24	34.22	3.26
COMIF (Ours)	3.39	23.17	0.76

VI. RELATED WORK

A. SEQUENTIAL RECOMMENDATION

Early works [30], [31], [32], [33], [34] attempted to model the item-item transition patterns based on the Markov Chain and matrix factorization, accomplishing their superiority over the traditional top-k recommendation [33], [35]. After the success of deep learning, many attempts based on various architectures were made to capture the complex sequential patterns. CNN [36], [37] looked at the sequence's local pattern with the kernels, while RNN [26], [38], [39], [40] considered the flow of the items. However, these architectures could not capture the long-term sequential pattern, restricting their effectiveness to the locally adjacent parts of the sequence. To counter this drawback, self-attention-based methods, such as SASRec [2] and BERT4Rec [3], are proposed. Self-attention mechanism [1] could attend to the relevant items of the sequence, simultaneously solving the locality issues. Even with these dedications, sequential recommendation systems were oriented only to user-item interactions, leading them to suffer from data sparsity issues [5], [8].

B. SEQUENTIAL RECOMMENDATIONS WITH SIDE-INFORMATION

To cover the data sparsity problem, several researchers have suggested fusion strategies to incorporate side-information such as title, brand, and category. The fusion method involves managing separate embedding for each side-information, which later gets fused in the neural network. An early work, FDSA [7], suggested using two separate self-attention blocks for item ID and side-information sequences and fusing them in the prediction layer. S3-Rec [8] proposed attribute prediction tasks based on self-supervised learning to enrich the item ID representation. However, both methods make it challenging for side information to directly interact with item ID in the self-attention mechanism, leading to limited utilization of side-information in self-attention operation.

Thus, two research branches are introduced to directly leverage side-information in self-attention architecture. The first branch, *embedding-level fusion*, attempts to fuse the side-information with item ID at the embedding layer, intending to inject its contextual information into item representation. NOVA [15] calculates the attention score with Query, Key for this integrated representation but Value for pure item embedding. DLFS-Rec [12] represents the item and side information by stochastic Gaussian distribution to handle the uncertainty of item sequence. MSSR [28] exploits the association between item ID sequence and side-information sequence with intra and inter-sequence attention scores to enhance user representation. Another branch, *attention-level fusion*, suggested changing the fusion stage from the embedding layer to the self-attention layer. DIF [10] point out that *embedding-level fusion* cause degradation of expressiveness regarding the rank of attention matrix and demonstrate that *attention-level fusion* solve the problem.

VII. CONCLUSION

In conclusion, we considered that item attribute side-information and user behavior side-information show different properties with respect to item ID, while previous works would regard them as equal. We analyzed previous fusion architectures in terms of the correlation between an item and the side-information and confirmed that *embedding-level fusion* is optimal for attribute and *attention-level fusion* is for behavior. Furthermore, we suggest two self-supervised learning tasks for two types of side-information, which properly consider their properties. We validate the effectiveness of the proposed method across various domains. In future research, we aim to explore the advanced fusion method to adaptively differentiate effective and non-effective side-information.

APPENDIX A FINDING ABOUT HIDDEN REPRESENTATION OF TRANSFORMER

Yang et al. [17] found that any hidden states in the ReLU-activated Transformer are equivalent to a summation of linear projections of input vectors and additive bias of the linear layer. This is represented as:

$$\begin{aligned} h_n &= \sum_{k=1}^{m} e_i^k W_X^k + \sum_{l=1}^{L} b^l W_B^l \\ &= \sum_{k=1}^{m} e_{id}^k W_X^k + \sum_{k=1}^{m} e_s^k W_X^k + \sum_{l=1}^{L} b^l W_B^l \end{aligned} \tag{12}$$

where e_i^k denotes the k-th integrated item embedding, b^l denotes the parameter of additive bias at the l-th layer of the model, e_{id}^k denote k-th item ID embedding, and e_s^k denote embedding of the k-th side-information. W_X^k and W_B^l is the linear transformation matrix for k-th item embedding and additive bias at l-th layer, respectively

APPENDIX B MORE ANALYSIS ON FUSION

In this section, we introduce two fusion operations: gating and concatenation. Then, we demonstrate that they, beyond addition operations, also contribute to the correlation in the self-attention module. Finally, we show that these two operations, like the addition operation, increase correlation at the prediction layer.

Firstly, we explain the concatenation and gating operation. As explored in [15], we define the concatenation operation to concatenate all side information, followed by a fully connected layer to uniform the dimension:

$$\begin{aligned} \mathcal{F}_{concat}(f_1, \ldots, f_m) &= [f_1 \| \ldots \| f_m] W \\ &= [f_1 \| \ldots \| f_m][W_1 \| \ldots \| W_m]^\top \\ &= f_1 W_1^\top + \cdots + f_m W_m^\top \end{aligned} \tag{13}$$

where $f_1, \ldots, f_m \in \mathbb{R}^{1\times d}$ are input features, $W_1, \ldots, W_m \in \mathbb{R}^{d\times d}$ are learnable weight matrix, and $\|$ denotes concatenation. The concatenation operation applies a linear transformation to each input feature individually and adds up the results. We also define gating operation as:

$$\begin{aligned} \mathcal{F}_{gating}(f_1, \ldots, f_m) &= \sum_{i=1}^{m} G_{(i)} f_i \\ &= G_{(1)} f_1 + \cdots + G_{(m)} f_m \\ G &= \text{softmax}(F W^F) \end{aligned} \tag{14}$$

where F is matrix form of given feature $[f_1, \ldots, f_m] \in \mathbb{R}^{m\times d}$ and $W^F \in \mathbb{R}^{d\times 1}$ is a learnable vector. $G \in \mathbb{R}^{m\times 1}$ is a vector and $G^{(i)}$ is i-th scalar element of the vector. In short, the gating operation also results in a sum of input features being weighted by scalar transformations.

Now, we show that the two operations introduce the correlation in the self-attention module. When we apply concatenation operation for *embedding-level fusion*, attention score α_{jk} is described as:

$$\begin{aligned} &\alpha_{jk} \\ &= \frac{\left(\mathcal{F}_{concat}(e_{id}^j, e_s^j) W_Q\right) \left(\mathcal{F}_{concat}(e_{id}^k, e_s^k) W_K\right)^\top}{\sqrt{d}} \\ &= \frac{\left(\left(e_{id}^j W_1^\top + e_s^j W_2^\top\right) W_Q\right) \left(\left(e_{id}^k W_1^\top + e_s^k W_2^\top\right) W_K\right)^\top}{\sqrt{d}} \\ &= \frac{\left(e_{id}^j W_1^\top W_Q\right) \left(e_{id}^k W_1^\top W_K\right)^\top}{\sqrt{d}} + \frac{\left(e_{id}^j W_1^\top W_Q\right) \left(e_s^k W_2^\top W_K\right)^\top}{\sqrt{d}} \\ &\quad + \frac{\left(e_s^j W_2^\top W_Q\right) \left(e_{id}^k W_1^\top W_K\right)^\top}{\sqrt{d}} + \frac{\left(e_s^j W_2^\top W_Q\right) \left(e_s^k W_2^\top W_K\right)^\top}{\sqrt{d}} \end{aligned} \tag{15}$$

where $e_{id}^k, e_s^k \in \mathbb{R}^d$ are the item ID embedding and the side-information embedding of k-th interaction, $W_Q, W_K \in \mathbb{R}^{d\times d}$ are learnable weight matrix. When we apply gating operation for *embedding-level fusion*, attention score α_{jk} is described as:

$$\begin{aligned} \alpha_{jk} &= \frac{\left(\mathcal{F}_{gating}(e_{id}^j, e_s^j) W_Q\right) \left(\mathcal{F}_{gating}(e_{id}^k, e_s^k) W_K\right)^\top}{\sqrt{d}} \\ &= \frac{\left(\left(G_{(1)}^j e_{id}^j + G_{(2)}^j e_s^j\right) W_Q\right) \left(\left(G_{(1)}^k e_{id}^k + G_{(2)}^k e_s^k\right) W_K\right)^\top}{\sqrt{d}} \\ &= \frac{\left(G_{(1)}^j e_{id}^j W_Q\right) \left(G_{(1)}^k e_{id}^k W_K\right)^\top}{\sqrt{d}} \\ &\quad + \frac{\left(G_{(1)}^j e_{id}^j W_Q\right) \left(G_{(2)}^k e_s^k W_K\right)^\top}{\sqrt{d}} \\ &\quad + \frac{\left(G_{(2)}^j e_s^j W_Q\right) \left(G_{(1)}^k e_{id}^k W_K\right)^\top}{\sqrt{d}} \\ &\quad + \frac{\left(G_{(2)}^j e_s^j W_Q\right) \left(G_{(2)}^k e_s^k W_K\right)^\top}{\sqrt{d}} \end{aligned} \tag{16}$$

We can see that the second and third term in both equations introduces the correlation between item ID and side-information.

Furthermore, we show that *embedding-level fusion* with the two operations increases the correlation between item ID and side-information like the one with addition operation. The relevance score with concatenation operation is computed as:

$$
\begin{aligned}
r_{(u,i)} &= e_i \cdot {h_n}^\top \\
&= e_i \cdot \left(\sum_{k=1}^{n} e_i^k W_X^k + \sum_{l=1}^{L} b^l W_B^l \right)^\top \\
&= \mathcal{F}_{concat}(e_{id}, e_s) \\
&\quad \cdot \left(\sum_{k=1}^{n} \mathcal{F}_{concat}(e_{id}^k, e_s^k) W_X^k + \sum_{l=1}^{L} b^l W_B^l \right)^\top \\
&= (e_{id} W_1^\top + e_s W_2^\top) \cdot \left(\sum_{k=1}^{m} e_{id}^k W_1^\top W_X^k \right. \\
&\quad \left. + \sum_{k=1}^{m} e_s^k W_2^\top W_X^k + \sum_{l=1}^{L} b^l W_B^l \right)^\top \\
&= e_{id} W_1^\top \cdot (\sum_{k=1}^{m} e_{id}^k W_1^\top W_X^k)^\top + e_{id} W_1^\top \\
&\quad \cdot (\sum_{k=1}^{m} e_s^k W_2^\top W_X^k)^\top \\
&\quad + e_s W_2^\top \cdot (\sum_{k=1}^{m} e_{id}^k W_1^\top W_X^k)^\top + e_s W_2^\top \\
&\quad \cdot (\sum_{k=1}^{m} e_s^k W_2^\top W_X^k)^\top \\
&\quad + (e_{id} W_1^\top + e_s W_2^\top) \cdot (\sum_{l=1}^{L} b^l W_B^l)^\top
\end{aligned}
$$

The relevance score with the gating operation is computed as:

$$
\begin{aligned}
r_{(u,i)} &= e_i \cdot {h_n}^\top \\
&= e_i \cdot \left(\sum_{k=1}^{n} e_i^k W_X^k + \sum_{l=1}^{L} b^l W_B^l \right)^\top \\
&= \mathcal{F}_{gating}(e_{id}, e_s) \\
&\quad \cdot \left(\sum_{k=1}^{n} \mathcal{F}_{gating}(e_{id}^k, e_s^k) W_X^k + \sum_{l=1}^{L} b^l W_B^l \right)^\top \\
&= (G_{(1)} e_{id} + G_{(2)} e_s) \cdot \left(\sum_{k=1}^{m} G_{(1)}^k e_{id}^k W_X^k \right. \\
&\quad \left. + \sum_{k=1}^{m} G_{(2)}^k e_s^k W_X^k + \sum_{l=1}^{L} b^l W_B^l \right)^\top \\
&= G_{(1)} e_{id} \cdot (\sum_{k=1}^{m} G_{(1)}^k e_{id}^k W_X^k)^\top + G_{(1)} e_{id} \\
&\quad \cdot (\sum_{k=1}^{m} G_{(2)}^k e_s^k W_X^k)^\top \\
&\quad + G_{(2)} e_s \cdot (\sum_{k=1}^{m} G_{(1)}^k e_{id}^k W_X^k)^\top + G_{(2)} e_s \\
&\quad \cdot (\sum_{k=1}^{m} G_{(2)}^k e_s^k W_X^k)^\top \\
&\quad + (G_{(1)} e_{id} + G_{(1)} e_s) \cdot (\sum_{l=1}^{L} b^l W_B^l)^\top
\end{aligned}
$$

We can see that *embedding-level fusion* with above two operations also increases the correlation between item ID and side-information in prediction layer since the first four terms in relevance score calculation are dot-product similarity terms between them.

CONFLICTS OF INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this article.

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[2] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2018, pp. 197–206.

[3] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 1441–1450.

[4] L. Wu, S. Li, C.-J. Hsieh, and J. Sharpnack, "SSE-PT: Sequential recommendation via personalized transformer," in *Proc. 14th ACM Conf. Recommender Syst.*, Sep. 2020, pp. 328–337.

[5] W. Song, C. Shi, Z. Xiao, Z. Duan, Y. Xu, M. Zhang, and J. Tang, "AutoInt: Automatic feature interaction learning via self-attentive neural networks," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 1161–1170.

[6] T. Yao, X. Yi, D. Z. Cheng, F. Yu, T. Chen, A. Menon, L. Hong, E. H. Chi, S. Tjoa, J. J. Kang, and E. Ettinger, "Self-supervised learning for large-scale item recommendations," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2021, pp. 4321–4330.

[7] T. Zhang, P. Zhao, Y. Liu, V. S. Sheng, J. Xu, D. Wang, G. Liu, and X. Zhou, "Feature-level deeper self-attention network for sequential recommendation," in *Proc. 28th Int. Joint Conf. Artif. Intell. (IJCAI)*, Aug. 2019, pp. 4320–4326.

[8] K. Zhou, H. Wang, W. X. Zhao, Y. Zhu, S. Wang, F. Zhang, Z. Wang, and J.-R. Wen, "S3-Rec: Self-supervised learning for sequential recommendation with mutual information maximization," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 1893–1902.

[9] J. Li, Y. Wang, and J. McAuley, "Time interval aware self-attention for sequential recommendation," in *Proc. 13th Int. Conf. Web Search Data Mining*, Jan. 2020, pp. 322–330.

[10] Y. Xie, P. Zhou, and S. Kim, "Decoupled side information fusion for sequential recommendation," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2022, pp. 1611–1621.

[11] A. Rashed, S. Elsayed, and L. Schmidt-Thieme, "Context and attribute-aware sequential recommendation via cross-attention," in *Proc. 16th ACM Conf. Recommender Syst.*, Sep. 2022, pp. 71–80.

[12] H. Liu, Z. Deng, L. Wang, J. Peng, and S. Feng, "Distribution-based learnable filters with side information for sequential recommendation," in *Proc. 17th ACM Conf. Recommender Syst.*, Sep. 2023, pp. 78–88.

[13] Y. Fu, B. Liu, Y. Ge, Z. Yao, and H. Xiong, "User preference learning with multiple information fusion for restaurant recommendation," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2014, pp. 470–478.

[14] S. Yan, X. Chen, R. Huo, X. Zhang, and L. Lin, "Learning to build user-tag profile in recommendation system," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 2877–2884.
[15] C. Liu, X. Li, G. Cai, Z. Dong, H. Zhu, and L. Shang, "Noninvasive self-attention for side information fusion in sequential recommendation," in *Proc. AAAI Conf. Artif. Intell.*, May 2021, vol. 35, no. 5, pp. 4249–4256.
[16] G. Ke, D. He, and T.-Y. Liu, "Rethinking positional encoding in language pre-training," 2020, *arXiv:2006.15595*.
[17] S. Yang, S. Huang, W. Zou, J. Zhang, X. Dai, and J. Chen, "Local interpretation of transformer based on linear decomposition," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguistics, Long Papers*, vol. 1, 2023, pp. 10270–10287.
[18] G. Tolomei, M. Lalmas, A. Farahat, and A. Haines, "You must have clicked on this ad by mistake! Data-driven identification of accidental clicks on mobile ads with applications to advertiser cost discounting and click-through rate prediction," *Int. J. Data Sci. Anal.*, vol. 7, no. 1, pp. 53–66, Feb. 2019.
[19] C. Zhang, Y. Du, X. Zhao, Q. Han, R. Chen, and L. Li, "Hierarchical item inconsistency signal learning for sequence denoising in sequential recommendation," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2022, pp. 2508–2518.
[20] H. Chen, Y. Lin, M. Pan, L. Wang, C.-C.-M. Yeh, X. Li, Y. Zheng, F. Wang, and H. Yang, "Denoising self-attentive sequential recommendation," in *Proc. 16th ACM Conf. Recommender Syst.*, Sep. 2022, pp. 92–101.
[21] W. Wang, F. Feng, X. He, L. Nie, and T.-S. Chua, "Denoising implicit feedback for recommendation," in *Proc. 14th ACM Int. Conf. Web Search Data Mining*, Mar. 2021, pp. 373–381.
[22] S. Wang, L. Hu, Y. Wang, L. Cao, Q. Z. Sheng, and M. Orgun, "Sequential recommender systems: Challenges, progress and prospects," 2019, *arXiv:2001.04830*.
[23] Z. He, W. Liu, W. Guo, J. Qin, Y. Zhang, Y. Hu, and R. Tang, "A survey on user behavior modeling in recommender systems," 2023, *arXiv:2302.11087*.
[24] J. McAuley, C. Targett, Q. Shi, and A. van Den Hengel, "Image-based recommendations on styles and substitutes," in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2015, pp. 43–52.
[25] W. X. Zhao, Y. Hou, X. Pan, C. Yang, Z. Zhang, Z. Lin, J. Zhang, S. Bian, J. Tang, W. Sun, Y. Chen, L. Xu, G. Zhang, Z. Tian, C. Tian, S. Mu, X. Fan, X. Chen, and J.-R. Wen, "RecBole 2.0: Towards a more up-to-date recommendation library," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2022, pp. 4722–4726.
[26] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," 2015, *arXiv:1511.06939*.
[27] Z. Fan, Z. Liu, Y. Wang, A. Wang, Z. Nazari, L. Zheng, H. Peng, and P. S. Yu, "Sequential recommendation via stochastic self-attention," in *Proc. ACM Web Conf.*, Apr. 2022, pp. 2036–2047.
[28] X. Lin, J. Luo, J. Pan, W. Pan, Z. Ming, X. Liu, S. Huang, and J. Jiang, "Multi-sequence attentive user representation learning for side-information integrated sequential recommendation," in *Proc. 17th ACM Int. Conf. Web Search Data Mining*, Mar. 2024, pp. 414–423.
[29] Y. Li, T. Chen, P.-F. Zhang, and H. Yin, "Lightweight self-attentive sequential recommendation," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2021, pp. 967–977.
[30] R. He and J. McAuley, "Fusing similarity models with Markov chains for sparse sequential recommendation," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 191–200.
[31] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized Markov chains for next-basket recommendation," in *Proc. 19th Int. Conf. World Wide Web*, Apr. 2010, pp. 811–820.
[32] A. Zimdars, D. Maxwell Chickering, and C. Meek, "Using temporal data for making recommendations," 2013, *arXiv:1301.2320*.
[33] S. Kabbur, X. Ning, and G. Karypis, "FISM: Factored item similarity models for top-N recommender systems," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2013, pp. 659–667.
[34] Q. Liu, S. Wu, D. Wang, Z. Li, and L. Wang, "Context-aware sequential recommendation," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 1053–1058.
[35] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," 2012, *arXiv:1205.2618*.
[36] J. Tang and K. Wang, "Personalized top-N sequential recommendation via convolutional sequence embedding," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, Feb. 2018, pp. 565–573.
[37] F. Yuan, A. Karatzoglou, I. Arapakis, J. M. Jose, and X. He, "A simple convolutional generative network for next item recommendation," in *Proc. 12th ACM Int. Conf. Web Search Data Mining*, Jan. 2019, pp. 582–590.
[38] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi, "Personalizing session-based recommendations with hierarchical recurrent neural networks," in *Proc. 11th ACM Conf. Recommender Syst.*, Aug. 2017, pp. 130–137.
[39] C. Ma, P. Kang, and X. Liu, "Hierarchical gating networks for sequential recommendation," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 825–833.
[40] L. Zheng, Z. Fan, C.-T. Lu, J. Zhang, and P. S. Yu, "Gated spectral units: Modeling co-evolving patterns for sequential recommendation," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2019, pp. 1077–1080.

SEUNGHWAN CHOI received the B.S. degree in software engineering from Ajou University, Suwon-si, South Korea, in 2023, where he is currently pursuing the M.S. degree in artificial intelligence. His research interest includes recommendation systems.

DONGHOON LEE received the B.S. degree in software engineering from Ajou University, Suwon-si, South Korea, in 2023, where he is currently pursuing the M.S. degree in artificial intelligence. His research interests include recommendation systems and federated learning.

HYEOUNGGUK KANG received the B.S. degree in cyber security from Ajou University, Suwon-si, South Korea, in 2023, where he is currently pursuing the M.S. degree in artificial intelligence. He is the author of the article titled Outlier Aware Cross-Market Product Recommendation. His research interest includes recommendation systems.

HYUNSOUK CHO received the Ph.D. degree in computer science and engineering from POSTECH, South Korea. From 2018 to 2021, he was a Technical Director and a Team Leader of Knowledge AI Laboratory, NCSOFT, South Korea. Since 2021, he has been working as an Assistant Professor with the Department of Software, Ajou University, South Korea. His research interests include multimodal understanding, recommendation systems, and natural language processing.