IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 30, 2021, accepted December 20, 2021, date of publication January 5, 2022, date of current version January 10, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3140215

Learning Users' Visual Preferences for Improving Recommendations

SHOUJIN WANG, CHENLU YANG, TONG QU, KAI YANG, AND WANGGEN WAN, (Senior Member, IEEE)

School of Communication and Information Engineering, Institute of Smart City, Shanghai University, Shanghai 200444, China

Corresponding authors: Chenlu Yang (yangchenlu@shu.edu.cn) and Wanggen Wan (wanwg@staff.shu.edu.cn)

This work was supported in part by the China Postdoctoral Science Foundation under Grant 2020M681264, in part by the Anhui Natural Science Foundation under Grant 1908085MF178, and in part by the Anhui Key Research and Development Plan Project under Grant 202104b11020031.

ABSTRACT *Sequential recommender systems* (SRSs) aim to predict the next item interest to a user by learning the users' dynamic preferences over items from the sequential user-item interactions. Most of existing SRSs make recommendations by only modeling a user's main preference towards the functions of items, while ignoring the user's auxiliary visual preference towards the appearances and styles of items. Although visual preference is less significant than the main preference, it may still play an important role in most of users' choice on items. On the one hand, a user often prefers to choose the item which matches her/his visual preference well from multiple items with the same function. For example, a lady may choose one clothes whose style suits her best from multiple clothes with the same function. On the other hand, some particular users (e.g., young girls) are usually very concerned about the appearances of some special items (e.g., clothes, jewelry). Therefore, the overlook of modeling users' visual preference may generate unsatisfied recommendations which can not match a user's various types of preferences and thus reduce the consumption experience. To address this gap, in this paper, we propose modeling users' visual preferences to improve the performance of sequential recommendations. Specifically, we devise a coupled Double-chain Preference learning Network (DPN) to jointly learn a user's main preference and visual preference as well as the interactions between them. In DPN, one chain is for modeling a user's main preference by taking the IDs of items as the input and the other chain is for modeling the user's visual preference by taking appearance images of items as the input. Finally, the two types of preferences are carefully integrated with an attention module for the next item prediction. Extensive experiments on two real-world transaction datasets show the superiority of our proposed DPN over the representative and state-of-the-art SRSs.

INDEX TERMS Visual preference, recommendations, deep neural networks.

I. INTRODUCTION

In the era of the digital economy, recommender systems (RSs) are becoming increasingly popular and have been planted in almost every part of our daily life [1]. They can help us with efficient and effective decision making when facing a large number of choices on content, products and service [2]–[4]. However, most of the traditional RSs are built on static user-item interaction data generated during a long period and thus they usually ignore the intrinsic dynamics of user preferences towards items, i.e., a user's preference over items changes over time [5]. To bridge such a gap, sequential recommender systems (SRSs) have been proposed to effectively model the sequential behaviors of users and thus capture their timely preferences for more accurate recommendations. By modeling the sequential user-item interactions generated in a continues time period as a sequence, an SRS is able to capture a user's dynamic and most recent preference over items and accordingly recommend precise items which can well match the user's preference.

In practice, in a sequence of user-item interactions, a user's choice on items not only depends on her main preference over items' function, but also depends on her *auxiliary visual preference [6], [7]*. In this paper, a user's *visual preference* refers to her/his preference regarding *the appearances and styles* of items and it can be revealed by the appearance images of those

The associate editor coordinating the review of this manuscript and approving it for publication was Fabrizio Messina.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

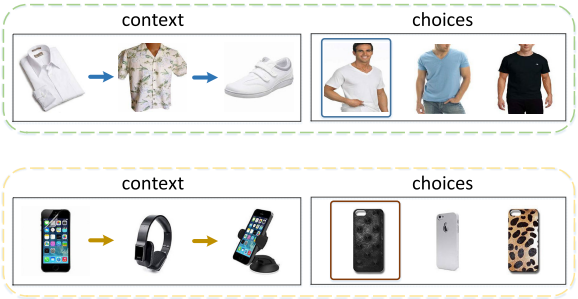


FIGURE 1. The images of items sampled from two anonymous users' purchased history from the Amazon product dataset. Each row contains a given context followed by three potential items for the next choice, while the first one in square was chosen by the user. It is clear that similar style of appearance is observed in each user's contextual items, while the item conforms to such style (the one in square) tends to be chosen out of multiple potential items.

items chosen by the user, such as the appearance and style of a clothes [8]. Although auxiliary and less significant compared with the main preference, a visual preference can still play an important role when the user makes choices on items. The reasons are two folds. On the one hand, a user often prefers to choose the item whose appearance can match her/his visual preference well from multiple candidate items with the same function. For example, a lady may choose one clothes whose style suits her best from multiple clothes with the same function. On the other hand, some particular visual-sensitive users (e.g., young girls) are usually very concerned about the appearances of some visual-sensitive items (e.g., clothes, jewelry). For example, when a girl purchases a necklace, she may not only focus on the function of the necklace, but also cares about the its appearance and style. Moreover, the appearance of an item, especially for those visual-sensitive items, determines a user's first impression of it and thus could influence whether to purchase it or not. Given the significance of visual preferences in affecting users' choices on items, in this paper, we particularly focus on this auxiliary preference and explore its influence on recommendations. It should be noted that the visual preference usually works on those types of items which are visual-sensitive rather than all types of items.

Accordingly, in sequential recommendations, at each time step, the user-item interaction is driven by both her main preference and her visual preference. More importantly, a user's visual preferences over items' appearance evolve along with time and visual preferences at different time steps are usually sequentially dependent. In addition, at each time step, the user's current visual preference usually affects her choice on items. Taking Figure 1 as an example, each row is a sequence of items which are purchased sequentially by an anonymous user on amazon.com. For each sequence, if we take the first three items as the context to predict the following item, two observations can be drawn: (1) although the sequentially purchased items are different from each other, but the visual style revealed by their appearance are quite similar, (2) when facing a variety of similar items with a different appearance for the next choice, the user tends to choose the item with appearance similar to that of those recently chosen items in the context. Both observations indicate that a user's visual preference over item appearance is not only sequentially dependent, but also plays an important role in her choice on items.

However, most of the existing studies on SRSs mainly model the user's main preference on items' main aspects revealed by the IDs of items, e.g., function, while overlooking her visual preference towards items' appearance, reducing the recommendation accuracy greatly. Therefore, a majority of SRSs only take the ID of each item within a sequence as the input to model the sequential dependencies among items for recommendations. For example, Markov chain based SRSs simply factorize the known first-order transition matrix over items in a sequence into latent factors of them and then use the obtained latent factors for the estimation of unknown transitions for the prediction of next item [9]. Recurrent neural network (RNN) based SRSs usually take the embedding of each item in a sequence as the input of each time step and then model the sequential dependencies among items for next-item prediction and recommendation [10]. Some other works introduce attention mechanisms to highlight those items which are more important and relevant to the next-item prediction [11]. A minority of works in SRSs incorporate additional information like item attributes for better next-item recommendation. For instance, both item ID and each of item attributes are mapped into a latent vector which is then taken as the input of a shallow network for next-item recommendation [12]. Although much progress has been achieved, all of these works ignored the impact of items' appearance on users' choice and failed to model users' visual preference effectively. This may limit the performance of sequential recommendations to some degree.

In this paper, we aim to address this gap by jointly modeling users' main preference on items and visual preference on items' appearance. Accordingly, a coupled Double-chain Preference learning Network (DPN) has been designed to model both preferences for recommendations. DPN mainly contains two chains which are connected together: item ID chain for modeling users' main preference revealed by the IDs of items and item image chain for modeling users' visual preference embedded in item appearance. Each chain is equipped with an RNN to model the corresponding dynamic preference which may change along with time. In addition, a bridge has been built between the two chains at each time step to incorporate the hidden state of the item image chain into the item ID chain to model the visual impact of items on users' choices on items. The final hidden state from the item ID chain and item image chain embeds the user's current main preference and visual preference respectively. Finally, the hidden states from both chains are attentively integrated together to form a compound representation of both a user's main preference and visual preference for the subsequent recommendations. Those candidate items showing a high possibility to match the user's compound preference are put into the recommendation list.

Thanks to the particularly designed coupled double-chain network structure in DPN, both a user's main preference on items' main aspects, e.g., function, and her visual preference on items' appearance are modelled respectively. And also, the impact of a user's visual preference on her choice on items (revealed by item IDs) has been well modelled with the bridge between the two chains. The contributions of our work can be summarized below:

- We propose *visual preference learning* to particularly model a user's visual preference towards items' appearance, which may greatly affect the user's choices on items in sequential recommendations.
- We design the coupled Double-chain Preference learning Network (DPN) to jointly learn a user's main preference and visual preference as well as the impact from visual preference on the user's choice on items.
- An item transition unit (ITU) is particularly designed to equip the DPN to model the transitions between item IDs by considering the influence from item appearance.

Extensive experimental results on two real-world datasets show the superiority of our approach over the representative and state-of-the-art approaches and verify the significance of learning a user's visual preference. The rest of this paper will be organized as follows: we review the related work on sequential recommendations in Section 2, and then formalize the problem in Section 3. We present our proposed approach in Section 4, followed by the experiment part in Section 5. Finally, we conclude this work in Section 6.

II. RELATED WORK

In this section, we review the existing works in the area of sequential recommendations. Generally speaking, according to the utilized techniques and approaches, sequential recommendations can be divided into conventional sequential recommendations and deep learning based sequential recommendations. In conventional sequential recommendations, conventional sequence modeling approaches, e.g., sequential pattern mining, Markove chain models, are mainly utilized to model the sequential dependencies among a sequence of user-item interactions for recommendations. In deep learning based sequential recommendations, deep neural networks, including Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), are mainly employed to learn the sequential dependencies for recommendations. In addition to these two types of sequential recommendations, we also review the attribute- augmented sequential recommendations which incorporate item attribute information and thus they are closely related to our work.

A. CONVENTIONAL SEQUENTIAL RECOMMENDATIONS

Early studies on sequential recommendations mainly focus on the development of conventional SRSs for sequential recommendations. There are mainly four classes of conventional SRSs: sequential pattern based SRSs, Markov chain based SRSs, latent factor model based SRSs and neighborhood model based SRSs. Specifically, Yap *et al.* [13] introduced a personalized sequential pattern mining based recommendation framework by first mining personalized sequential patterns from users' transaction behavior data and then utilizing the mined patterns to guide the subsequent recommendations. Although simple and sometimes effective, such class of approaches are easy to lose those infrequent but still important items and patterns due to the commonly used frequency constraint in the pattern mining process, and thus reduce the recommendation performance. Markov chain (MC) models are another intuitive solution to model the transitions within sequence data and thus are naturally employed for sequential recommendations. Since Markov chain based SRSs do not involve the aforementioned frequency constraint, and thus they can effectively alleviate the above drawbacks in sequential pattern based SRSs. A typical Markov chain based SRS is the Personalized Ranking Metric Embedding (PRME) model proposed by Feng *et al.* [14]. PRME was proposed to precisely model personalized check-in sequences for next POI recommendation by utilizing a Markov chain framework. However, Markov chain based SRSs usually only model the first-order dependencies in a sequence of interactions while ignoring the high-order ones. This often decreases the corresponding recommendation performance. Rendle *et al.* [15] proposed a classic latent factor model called Factorized Personalized Markov Chains (FPMC) model to learn the latent factors of items by factorizing the transition matrix over underlying Markov chains over items from adjacent baskets. The learned latent factors are then utilized to represent items for next-basket recommendation. Similar to Markov chain based approaches, latent factor models can model the first-order dependencies only while overlooking high-order ones. Moreover, they are easy to suffer from the data sparsity issue. Recently, session-KNN (SKNN) was proposed for next-item recommendation, which utilizes the similarity between sessions to calculate the score of the candidate items to be the next item [16], [17]. Based on SKNN, Garg *et al.* [18] introduced sequence and time aware neighborhood model for session-based recommendation, which additionally takes into account the readily available position information of items within sessions/sequences for more accurate recommendations.

In summary, most of conventional SRSs only model users' main preferences for sequential recommendations. They often overlook the users' visual preferences towards items, which may greatly reduce the recommendation performance.

B. DEEP LEARNING BASED SEQUENTIAL RECOMMENDATIONS

In recent years, deep learning models including RNN, CNN have shown great potential to capture the complex relations in a variety of areas, including Computer Vision (CV) and Natural Language Processing (NLP). Inspired by this, researchers have introduced deep learning models into sequential recommendations. Consequently, a series of deep learning based SRSs have been proposed and achieved great

success. Among various deep neural network architectures, RNN is the prominent one for sequential recommendations, because of its powerful capability to model sequence data. Therefore, a lot of RNN-based SRSs have been developed. Hidasi *et al.* [10] proposed an RNN-based model called GRU4Rec for predicting the next item by equipping the RNN with Gated Recurrent Units (GRU) to model the long-term sequential dependencies embedded in a sequence of interactions. This model was further improved by introducing novel ranking loss functions tailored to RNNs [19]. Some other similar works replaced the GRU with Long Short Term Memory (LSTM) units to build LSTM-based SRSs [20]. To better model users' preferences across multiple sequences of interactions, a hierarchical-RNN was proposed in [21] to model both the short-term intra-sequence dependencies and long-term inter-sequence dependencies for next item recommendations. However, RNN is easy to generate false sequential dependencies due to the employed rigid order assumption over any two adjacent interactions, which may not be true in the real-world cases since the order over some items may not make much sense [22].

In addition to RNN, CNN are also applied in sequential recommendations to build CNN-based SRSs. Benefiting from the strong capability of CNN to capture the dependencies between different local areas, CNN-based SRSs have strength in capturing the collective or local dependencies among interactions in a sequence [23], [24]. Specifically, CNN-based SRSs first learn the embedding of each interaction and then utilize the learned embeddings to generate recommendations. Tang and Wang [23] proposed a convolutional sequence embedding recommendation model called Caser by utilizing both horizontal and vertical convolutional filters to learn the item-level and feature-level dependencies respectively for better recommendations. Further, a 3D CNN model was proposed to jointly model the sequential patterns in sequence data and the item characteristics from item content features for next-item recommendation [25]. However, one obvious drawback of CNN-based SRSs is that they may not be able to effectively capture the long-range dependencies in those long sequences because of the limited perceptive filed of CNN.

Except for the aforementioned two basic deep neural network architectures, i.e., RNN and CNN, some advanced mechanisms are also applied to sequential recommendations to address some specific issues. Out of them, attention is one of the most representative one. To be specific, attention mechanism is often incorporated into some basic models, e.g., embedding model, RNN, to emphasize those more important and relevant interactions in a sequence for making more accurate recommendations. Wang *et al.* [22] incorporated attention into embedding model to learn attentive item and session embeddings for next item recommendations. Later, self attention mechanism was introduced into SRSs to emphasize those more relevant actions from users' historical click or purchase actions for accurate sequential recommendations [11], [26], [27].

Cui *et al.* [28] combined attention and a hierarchical RNN model to emphasize both important items within each session and important sessions for more accurate sequential recommendations. Liu *et al.* [29] used attention model to attentively read out the relevant and important interaction information from a memory module for better next item prediction. Benefiting the powerful representation capacity of a natural language processing model called Bidirectional Encoder Representations from Transformers (BERT), multi-head self attention model was introduced into SRSs to further improve the performance of sequential recommendation [30]. Further, Cho *et al.* [31] proposed a novel multiple self-attention head model to simultaneously extract various patterns from the users' sequential behaviors for accurate sequential recommendation. Although effective, attention mechanism tends to assign larger weights on a few significant interactions while downplaying others, easily leading to bias.

To summarize, all these different types of deep learning based sequential recommendations have achieved great success. However, most of they often only model users' main preferences by taking the item IDs in a sequence as the input (some also consider user IDs as well) while ignoring other aspects, especially item appearance, which may also greatly influence users' choices on items.

C. ADDITIONAL INFORMATION-AUGMENTED SEQUENTIAL RECOMMENDATIONS

In addition to the aforementioned generally utilized item ID information, some works on SRS take additional information such as item attributes, item position information in a sequence, time information, etc. to provide more contextual information for improving sequential recommendation. For example, Wang *et al.* [12] takes both the item ID and its attributes as the input of a shallow neural networks to first learn a compound item presentation and then input the presentation to the prediction layer for the next item prediction. Tuan and Phuong [25] proposed a 3D convolutional neural networks to learn item presentations from both ID information and content feature information for the next item prediction. Garg *et al.* [18] introduced a neighborhood model to incorporate the readily available item position information within sequences for more accurate sequential recommendations. Li *et al.* [32] exploit timestamps of interactions within a sequence to explore the influence of different time intervals between interaction on next-item recommendation. Ye *et al.* [33] utilized both the ''absolute time patterns'' and ''relative time patterns'' embedded in a user interaction sequence to improve sequential recommendation. Although these works have taken a step forward to incorporate more contextual information to enrich the input for sequential recommendations, they still failed to consider users' visual preference towards item appearance.

Only few works have considered the appearance of items when making sequential recommendations. Hidasi *et al.* [34] devised a parallel RNN architecture to incorporate both the visual and textual features of items to build informative item

representation for next item recommendations. In this work, each RNN serves as one channel to independently model one type of item features or item ID, and finally the outputs from all RNNs are combined together to predict the next item. Although such kind of approaches have improved the recommendation performance to some degree, they simply treat item appearance as a kind of feature without particularly learning users' visual preferences. Moreover, they have not well utilized the item appearance well. The reasons come from three folds: first, they often utilize the image feature data pre-trained from item appearance in other tasks, which are not related to recommendations. Consequently, the pre-trained image feature data may lose some important information which is sensitive to sequential recommendations [35]. Second, they treat all parts of an item appearance equally important without any discrimination, consequently, the important information may be overwhelmed by those irrelevant and noisy information embedded in the item appearance. Third, they model each type of features and the item ID independently without interaction at each time step, which may violate the reality that the items' attribute information actually influences users' choice at each step rather than just at the final step. In summary, existing works on SRSs either totally ignore users' visual preference and item appearance, or simply treat item appearance as a kind of attributes while neglecting the users' particular visual preference towards item appearance. To the extent of our knowledge, there is no existing work on SRSs which particularly model users' visual preference for better recommendations. In practice, in some visual-sensitive domains, e.g., clothes, except for users' main preference, e.g., preference towards items' functions, the visual preference also plays an important role on users' choice on items. Inspired by this fact, we propose learning visual preference towards item appearance to generate more reliable and accurate sequential recommendations.

D. ITEM VISUAL APPEARANCE LEARNING IN RECOMMENDATION

Some researchers investigated the effect of visual information of items in recommendations, especially in fashion recommendation area, but most of them focus on non-sequential recommendations. For instance, Liu *et al.* [36] proposed a DeepStyle model to learn style features of items and sensing preferences of users from non-sequential data for personalized recommendation. Lee *et al.* [37] proposed Style2Vec to learn and encode the items' visual style into vector representations for the downstream fashion recommendation, i.e., to recommend an outfit of items with good compatibility. Similarly, Yin *et al.* [7] and Li *et al.* [38] learn and utilize visual compatibility relationship between items to enhance fashion recommendation or complementary item recommendation. Hidayati *et al.* [8] try to match clothing style and personal body shape for effective fashion recommendation.

In summary, learning item visual appearance has been widely researched in the fashion recommendation area and has achieved great success. However, fashion recommendation aims at recommending an outfit of items (usually clothes) with good compatibility among them without modeling the sequential relations between users' historical actions [39]. This is totally different from sequential recommendation which mainly relies on the sequential dependencies over users' sequential behaviors for predicting and recommending the next item [40]. Therefore, the aforementioned fashion recommendation methods cannot be utilized for sequential recommendation due to its different work mechanism and aim from sequential recommendation.

III. PROBLEM STATEMENT

Given a sequence set $S = \{s_1, \ldots, s_{|S|}\}$, each sequence $s = \{v_1, \ldots, v_i, \ldots, v_{|s|}\}$ consists of a list of items which are sequentially interacted by a certain user in one transaction event. All the items together form the item universe set V. $|S|$ and $|s|$ denotes the number of sequences in S and the number of items in s respectively. v_i denotes the i^{th} item in sequence s. Each item is indicated by a unique item ID d and is associated with one image m to show its appearance. For example, the ID and image of item v_i is denoted as d_i and m_i respectively. Given one item v_t in s, all the items occurring before v_t form the sequence context c_t of it and each item in c_t is called the contextual item of v_t. Specifically, c_t contains two sequences, one is the sequence of IDs of all contextual items and the other is the sequence of images of all contextual items, namely, $c_t =< c_t^d, c_t^m >$, where $c_t^d = \{d_1, \ldots, d_{t-1}\}$ and $c_t^m = \{m_1, \ldots, m_{t-1}\}$.

Generally, given a sequence context c consisting of $(t-1)$ precedent items, including the ID and image for each item, an SRS is built to predict the subsequent t^{th} item from all the candidate item set V. Accordingly, our model can be trained as a classifier to learn the conditional probability $p(v_t|c)$ over each candidate item. Once the model is trained, it can be used to recommend the next item by ranking all the candidate items in terms of their conditional probability over the given context. Those items with the top-k conditional probabilities are taken to constitute the recommendation list.

IV. COUPLED DOUBLE-CHAIN PREFERENCE LEARNING NETWORK

The architecture of our proposed coupled Double-chain Preference learning Network (DPN) is presented in Figure 2. DPN consists of two main modules: (1) the preference learning module to collaboratively learn a user's main preference and visual preference towards items with an item ID chain and an item image chain respectively, and (2) the prediction module to predict the next item by taking the learned preferences as the input. To be specific, in preference learning module, for each item in a sequence context, its ID d and image m are taken as the initial input in the input layer. d and m are then projected into a latent embedding vector $\mathbf{d}$ and matrix $\mathbf{m}$ respectively, which are then input into the corresponding item transition unit (ITU) and gated recurrent unit (GRU) in item ID chain and item image chain respectively. Item ID chain and item image chain model a user's main preference and

visual preference respectively, while the influence of visual preference on main preference is considered at each time step with. Namely, a user's choice on one item not only depends on the IDs of previous chosen items but also depends on their appearance. Consequently, the last hidden state at time step $(t-1)$ from item ID chain and item image chain has encoded the user's main preference and visual preference and are combined together with an element-wise attention layer to predict the next item at step t. We present the technical details of our proposed DPN in the following subsections.

A. ITEM REPRESENTATION LEARNING

Given an item from a sequence, we build its ID embedding and visual embedding respectively.

1) ITEM ID EMBEDDING

To be specific, for the i^{th} item in a sequence context c, its ID embedding $\boldsymbol{d}_i \in \boldsymbol{R}^K$ is obtained by projecting its ID d_i into a latent space with a embedding layer. Formally,

$$\boldsymbol{d}_i = \boldsymbol{W}^e_{:,i}, \boldsymbol{W}^e \in \boldsymbol{R}^{K\times|V|} \tag{1}$$

where $\boldsymbol{W}^e$ is the embedding matrix, and each column of $\boldsymbol{W}^e$ corresponds to the embedding of one item.

2) ITEM VISUAL EMBEDDING

For the visual embedding of item v_i, we first introduce a pre-trained convolution neural network (CNN) to extract annotation vectors from the item appearance image m_i and then utilize an attention to integrate these vectors together to form the final visual embedding $\boldsymbol{m}_i$. Note that in order to keep more initial information in the image which may be more sensitive to the final task, we use the low-level latent features extracted by CNN rather than the high-level abstract ones from the final layer of the pre-trained CNN. More specifically, a 16-layer CNN named VGGNet (shortened as VGG-16) [41] which was pre-trained on ImageNet [35] without fine-tuning is applied to extract the initial features from item appearance images. The VGG-16 network contains 13 convolutional layers and 3 fully-connected layers and it finally outputs a 1000-dimensional feature vector for each image from the final layer. Instead of using the final feature vector directly, we first extract the output, i.e., a feature map with a size of $14\times14\times512$, of the fifth convolutional layer (conv5-3) of VGG-16 and then split it into a set of l annotation vectors ($l = 14\times14 = 196$) where each vector is with the dimension of 512:

$$A_i = \left\{\mathbf{a}_{i1}, \ldots, \mathbf{a}_{ij}, \ldots, \mathbf{a}_{il}\right\}, \quad \mathbf{a}_{ij} \in \boldsymbol{R}^{512} \tag{2}$$

Each annotation vector $\mathbf{a}_j$ can be seen as the representation of a local, small and uniform-sized region of the image of an item. All the annotation vectors are then attentively aggregated into the visual representation $\mathbf{m}$ of an item. Noted that any other variants of CNN are also able to achieve similar results as VGG-16 does here.

Once annotation vectors are ready, a feature-level attention model is devised to build an attentive visual representation for each item. Specifically, by assigning larger weights to those more important and relevant annotation vectors and smaller weights to other ones, the feature-level attention model is able to emphasize those local regions in an image which are more sensitive and relevant to a user's preference and her choice on items, while downgrading the noisy information from the irrelevant regions. Note that the weights in attention model are jointly trained with the final prediction in an end-to-end manner. In addition, to incorporate the influence of the last item, the last hidden state $\mathbf{h}^m_{i-1}$ of the item image chain is used to guide the calculation of attention weights. Accordingly, the attention weight α_{ij} of the j^{th} annotation vector of the i^{th} item is calculated below:

$$\alpha_{ij} = \frac{\exp\left(\mathbf{W}^\alpha \mathbf{e}_{ij}\right)}{\sum_{k=1}^{l}\exp\left(\mathbf{W}^\alpha \mathbf{e}_{ik}\right)} \tag{3}$$

$$\mathbf{e}_{ij} = \sigma\left(\mathbf{W}^a \mathbf{a}_{ij} + \mathbf{W}^h \mathbf{h}^m_{i-1}\right) \tag{4}$$

where $\mathbf{W}^\alpha$, $\mathbf{W}^a$, $\mathbf{W}^h$ are the corresponding weighting matrices for $\mathbf{e}$, $\mathbf{a}$ and $\mathbf{h}$ respectively.

Once the attention weights are ready, the attentive visual embedding of the i^{th} item can be calculated as:

$$\mathbf{m}_i = \sum_{j=1}^{l}\alpha_{ij}\mathbf{a}_{ij} \tag{5}$$

B. PREFERENCE LEARNING MODULE

After the ID embedding and visual embedding for each item are obtained, they are input into the ITU cell and GRU cell in item ID chain and item image chain for main preference learning and visual preference learning respectively.

1) VISUAL PREFERENCE LEARNING VIA ITEM IMAGE CHAIN

An item image chain built on a recurrent structure is particularly designed to model a user's visual preference over items. At each time step, the visual embedding is input into the corresponding GRU cell to model the sequential dependencies among the appearance of a sequence of items. Specifically, in time step i, the hidden state $\mathbf{h}^m_i$ is calculated based on the current input $\mathbf{m}_i$ and the last hidden state $\mathbf{h}^m_{i-1}$. First, the reset gate $\mathbf{r}^m_i$ and forget gate $\mathbf{z}^m_i$ are calculated:

$$\mathbf{r}^m_i = \sigma_s(\mathbf{W}^m_r[\mathbf{h}^m_{i-1}, \mathbf{m}_i]) \tag{6}$$

$$\mathbf{z}^m_i = \sigma_s(\mathbf{W}^m_z[\mathbf{h}^m_{i-1}, \mathbf{m}_i]) \tag{7}$$

where $\mathbf{W}^m_r$ and $\mathbf{W}^m_z$ are the corresponding weight matrices to be learned during the training process. σ_s and σ_t are activation functions and are specified as sigmoid and tanh respectively.

Then the candidate state $\widetilde{\mathbf{h}}^m_i$ and the hidden state $\mathbf{h}^m_i$ of the current step can be computed as:

$$\widetilde{\mathbf{h}}^m_i = \sigma_t(\mathbf{W}^m_h[\mathbf{r}^m_i * \mathbf{h}^m_{i-1}, \mathbf{m}_i]) \tag{8}$$

$$\mathbf{h}^m_i = (1-\mathbf{z}^m_i) * \mathbf{h}^m_{i-1} + \mathbf{z}^m_i * \widetilde{\mathbf{h}}^m_i \tag{9}$$

where $\mathbf{W}^m_h$ and σ_t are the weight matrix and activation function respectively, and σ_t is specified to tanh.

Consequently, a user's visual preference at step i is encoded into the hidden state $\mathbf{h}^m_i$.

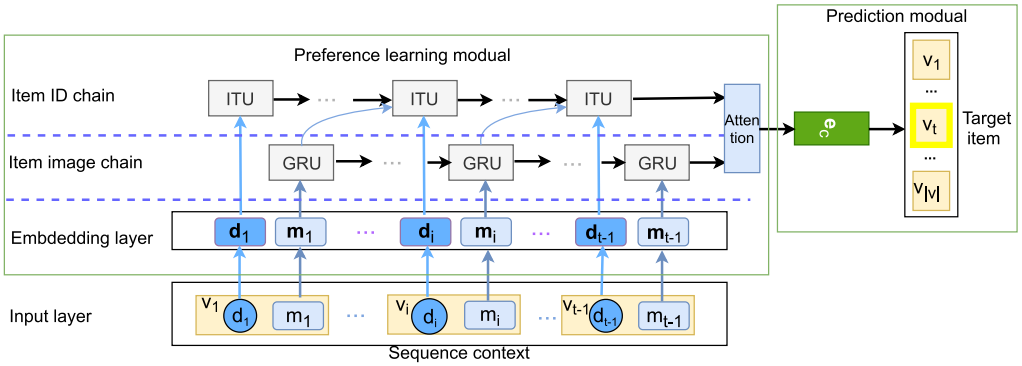


FIGURE 2. The coupled double-chain preference learning network (DPN) model mainly consists of two modules: the preference learning module to collaboratively learn a user's main preference and visual preference with an item ID chain and an item image chain respectively, and the prediction module to predict the next item by taking the learned preference as the input.

2) MAIN PREFERENCE LEARNING VIA ITEM ID CHAIN

We learn a user's main preference towards items with a particularly designed item ID chain. In the item ID chain, the cell at each time step is equipped with a carefully deigned item transition unit (ITU) to model the transitions between item IDs by considering the influence from item appearance. Specifically, the hidden state of last time step from the item image chain is carefully taken as an input of the ITU at the current time step. To be specific, in time step i, the hidden state $\mathbf{h}_i^d$ is calculated based on both the current item ID embedding $\mathbf{d}_i$ and the last hidden state $\mathbf{h}_{i-1}^m$ in the item image chain, as well as the last hidden state $\mathbf{h}_{i-1}^d$ in the item ID chain. First, for the i^{th} ITU, the reset gate $\mathbf{r}_i^d$ and forget gate $\mathbf{z}_i^d$ are inherited from a universal GRU:

$$\mathbf{r}_i^d = \sigma_s(\mathbf{W}_r^d[\mathbf{h}_{i-1}^d, \mathbf{d}_i]) \tag{10}$$

$$\mathbf{z}_i^d = \sigma_s(\mathbf{W}_z^d[\mathbf{h}_{i-1}^d, \mathbf{d}_i]) \tag{11}$$

where $\mathbf{W}_r^d$, $\mathbf{W}_z^d$ and σ_s are the analogical weight matrices and activation functions in Eqs.(6)(7).

To incorporate the influence of a user's visual preference on her main preference, an extra gate $\mathbf{g}_i^d$, called preference coupling gate is devised. The calculation of $\mathbf{g}_i^d$ is presented as below:

$$\mathbf{g}_i^d = \sigma_s(\mathbf{W}_g^d[\mathbf{h}_{i-1}^d, \mathbf{h}_{i-1}^m]) \tag{12}$$

$\mathbf{g}_i^d$ is then used to determine how much information from the visual preference modelled by item image chain should be incorporated into the learning of main preference modelled by item ID chain. Specially, in item ID chain, the candidate hidden sate and hidden sate at the current time step can be computed as:

$$\widetilde{\mathbf{h}}_i^d = \sigma_t(\mathbf{W}_h^d[\mathbf{r}_i^d * \mathbf{h}_{i-1}^d, \mathbf{g}_i^d * \mathbf{h}_{i-1}^m, \mathbf{d}_i]) \tag{13}$$

$$\mathbf{h}_i^d = (1 - \mathbf{z}_i^d) * \mathbf{h}_{i-1}^d + \mathbf{z}_i^d * \widetilde{\mathbf{h}}_i^d \tag{14}$$

where $\mathbf{W}_g^d$, $\mathbf{W}_h^m$, σ_s and σ_t in Eq.(12)(13) are the weight matrices and activation functions respectively. As a result, the last hidden state $\mathbf{h}_i^d$ of the item ID chain has encoded the main preference of the user at the moment.

3) COMPOUND PREFERENCE LEARNING VIA ATTENTION MODEL

Finally, to learn a user's final compound preference, an element-wise attention model is devised to attentively integrate the user's main preference and visual preference. Specifically, a user's compound preference $\mathbf{h}_{t-1}$ at the last time step $(t-1)$ is computed with element-wise product between the attentive weight β_j, and the last step's hidden state $\mathbf{h}_{t-1}^d$ from the item ID chain:

$$\mathbf{h}_{t-1,k} = \beta_k \mathbf{h}_{t-1,k}^d, \quad k \in \{1, \ldots, K\} \tag{15}$$

where K is the dimension of $\mathbf{h}_{t-1}^d$. For each element (dimension) $\mathbf{h}_{t-1,k}^d$ in $\mathbf{h}_{t-1}^d$, its attentive weight β_k is computed from the corresponding dimension $\mathbf{h}_{t-1,k}^m$ of the last hidden state $\mathbf{h}_{t-1}^m$ in the item image chain with a softmax layer:

$$\beta_k = \frac{\exp\left(\mathbf{W}^\beta \mathbf{h}_{t-1,k}^m\right)}{\sum_{n=1}^K \exp\left(\mathbf{W}^\beta \mathbf{h}_{t-1,n}^m\right)} \tag{16}$$

where $\mathbf{W}^\beta$ is a weight matrix for $\mathbf{h}_{t-1}^m$. Consequently, the user's main preference $\mathbf{h}_{t-1}^d$ is fine-tuned guided by the attentive weights calculated from the visual preference $\mathbf{h}_{t-1}^m$. In this way, those dimensions in $\mathbf{h}_{t-1}^d$ which are more relevant to the next choice as well as the user's visual preference are emphasized.

This compound preference $\mathbf{h}_{t-1}$ is then taken as the representation $\mathbf{e}_c$ of the sequence context c, i.e., $\mathbf{e}_c = \mathbf{h}_{t-1}$, which will be input into the prediction module for subsequent next item prediction.

C. PREDICTION MODULE

Once the sequence context embedding, i.e., the user's compound preference, is learned, a fully connected layer is

employed to connect it to the output layer. Specifically, a softmax layer is employed as prediction module to predict the conditional probability distribution $p(v|c) \in \boldsymbol{R}^{|V|}$ over the candidate items, with c as the input context. Particularly, the conditional probability of the true target item v_t is computed as:

$$P_\theta(v_t|c) = \frac{\exp\left(\mathbf{W}^o_{t,:}\boldsymbol{e}_c\right)}{\sum_{v_g \in |V|} \exp\left(\mathbf{W}^o_{g,:}\boldsymbol{e}_c\right)} \tag{17}$$

where $\mathbf{W}^o$ is the weight matrix for the aforementioned fully connected layer. θ denotes all the parameters in our model to be learned.

Given the conditional probability calculated by Eq.(17), our goal is to find the optimal parameters θ for maximizing the conditional log-likelihood. Consequently, given a training dataset $D =< c, v_t >$, the conditional log-likelihood is obtained by:

$$L_\theta = \sum_{d \in D} \log P_\theta(v_t|c) \tag{18}$$

with which, DPN is able to be trained with Adagrad optimizer [42] w.r.t all the learn-able parameters in DPN. Specifically, the batch size for mini-batch gradient descent is set to 8, and an extra L-2 regularization term is involved in Eq.(18) where regularization parameter λ is set to 0.001. We used adagrad optimizer with a initial learning rate of 0.1. A learning rate decay mechanism is designed to adjust the learning rate along with the training process.

Once the model parameters θ have been learned, DPN is ready to generate predictions and further, next-item recommendations. Given a user's transactional context which contains both item ID and its corresponding appearance image, the probabilities of choosing next candidate items are calculated according to Eq.(17), and a recommendation list reflecting the rank of the candidate items is achieved.

V. EXPERIMENTS AND ANALYSIS

A. DATA PREPARATION

Two real-world product review datasets are used for experiments, namely two categories of Amazon Product Data[1] [43]: (1) "Clothing, Shoes and Jewelry" (shorten as "Amazon Clothing" below); and (2) "Cell Phones and Accessories" (shorten as "Amazon Phones" below). Users' visual preferences are supposed to play an important role in their shopping choices on these two categories of products since they are more likely to be visual-sensitive. Both datasets record the reviewed items of each user on *amazon.com* from May 1996 to July 2014. To avoid data sparsity issue, reviews before 2012 are very sparse and thus are excluded in the experimental datasets.

We follow a common practice in SRS studies to prepare the training-test instances for training and testing our proposed model. First, a set of sequences is extracted from each original dataset by putting all items in one user's review history together to form a sequence, as most of studies on SRSs did [21]. In practice, a user's review on an item is regraded as a kind of her implicit feedback on it [34], [44]. Second, those sequences containing less than five items, and items reviewed by less than five times are removed, since those sparse users and items heavily reduce recommendation accuracy. Finally, to build training-test instances of format $\langle c_t; v_t \rangle$, for each sequence s, the last item v_t in s was picked up as the target item and all those items occurred before v_t in s constituted its corresponding context c_t. For each item v_i, in addition to its ID v_i, an appearance image m_i was obtained from its metadata. Therefore, the context $c_t =< c^d_t, c^m_t >$ is built on both the sequence $c^d_t = \{v_1, v_2, \ldots, v_{t-1}\}$ of item IDs and the sequence $c^m_t = \{m_1, m_2, \ldots, m_{t-1}\}$ of item images. The statistics of the experimental datasets are shown in Table 1. Once all the training and test instances are built, we randomly select 70%, 20% and 10% to form the training set, test set and validation set respectively. We conducted 10-fold cross-validation to achieve stable results and reported the average results from them.

[1] http://jmcauley.ucsd.edu/data/amazon/

TABLE 1. Statistics of experimental datasets.

Statistics	Amazon Clothing	Amazon Phones
#Users	35,579	24,260
#Items	22,931	10,199
#Feedback	250,155	167,371
Avg. Sequence Length	7.03	6.90

B. EXPERIMENTAL SETTINGS

1) EVALUATION METRICS

Four commonly-used metrics: Recall, Precision, Mean Reciprocal Rank (MRR) and normalized Discounted Cumulative Gain (NDCG) are employed to evaluate the recommendation accuracy [22] of all compared methods. All four metrics are evaluated on a top-K recommendation list ($K \in \{10, 20\}$) for evaluating the rank efficacy of candidate items.

2) COMPARISON METHODS

The following representative and state-of-the-art SRSs built on various frameworks including Markov chain, RNN, CNN, attention model are selected to be the baselines. They are commonly used as baselines in most of SRS studies. In addition, two simplified versions of our proposed model are also implemented for ablation analysis. Next we briefly introduce the baselines one by one.

- RAND. A method generates random recommendation list from the item set to each user.
- POP. A method recommends top-K popular items (by their frequencies of occurrence in the whole review history) to each user.
- FPMC. A Markov Chain-based model which constructs a personalized interaction matrix computed from item transition probability for next-basket recommendation [15]. The latent factor of FPMC was set to 50.
- LSTM. A RNN-based model which utilized recurrent unit to model sequential dependencies for

TABLE 2. Recommendation accuracy on Amazon clothing dataset (all values are in percentages).

Methods	Amazon Clothing					
	@10			@20		
	Recall	MRR	NDCG	Recall	MRR	NDCG
RAND	0.0260	0.0310	0.0399	0.0607	0.0345	0.0530
POP	0.5121	0.1335	0.2213	0.9375	0.1628	0.3286
FPMC	0.7378	0.3115	0.4119	0.9693	0.3271	0.4699
LSTM	1.0995	0.3318	0.5048	1.7361	0.3744	0.6638
GRU	1.0874	0.3294	0.5011	1.7231	0.3677	0.6580
Caser	0.6910	0.2355	0.3390	1.1201	0.2634	0.4451
ATEM	0.4949	0.2049	0.2618	0.6510	0.2171	0.3064
SASRec	0.1089	0.3304	0.5011	0.1720	0.3674	0.6568
p-RNN	1.1574	0.3781	0.5555	1.8807	0.4260	0.7355
DPN	**1.2152**	**0.3861**	**0.5766**	**2.0109**	**0.4389**	**0.7747**
improvement	4.99%	2.12%	3.80%	6.92%	3.03%	5.33%

TABLE 3. Recommendation accuracy on Amazon phones dataset (all values are in percentages).

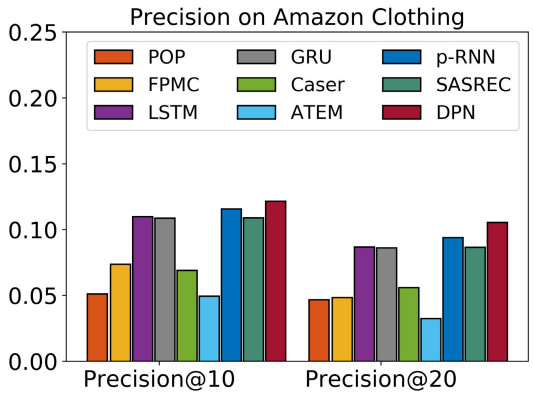
Methods	Amazon Phones					
	@10			@20		
	Recall	MRR	NDCG	Recall	MRR	NDCG
RAND	0.0371	0.0212	0.0364	0.0989	0.0291	0.0647
POP	1.8674	0.6613	0.9374	3.2525	0.7561	1.2861
FPMC	2.5762	1.1388	1.4754	4.2044	1.2523	1.8166
LSTM	3.4418	1.2873	1.7830	5.5853	1.4283	2.3154
GRU	3.4076	1.2745	1.7782	5.5321	1.4124	2.3056
Caser	2.8194	1.0514	1.4576	4.5342	1.1669	1.8667
ATEM	2.0174	0.9008	1.1644	4.9267	1.1021	1.8992
SASRec	3.4112	1.2742	1.7810	5.5212	1.4096	2.3068
p-RNN	3.2976	1.1871	1.6701	5.5230	1.3390	2.2296
DPN	**3.5243**	**1.3638**	**1.8628**	**5.9150**	**1.5292**	**2.4667**
improvement	2.40%	5.94%	4.48%	5.90%	7.06%	6.53%

next-basket recommendation [45]. Rather than basic RNN, we adopted LSTM for better performance. The dimension of the hidden state in LSTM is set to 30.

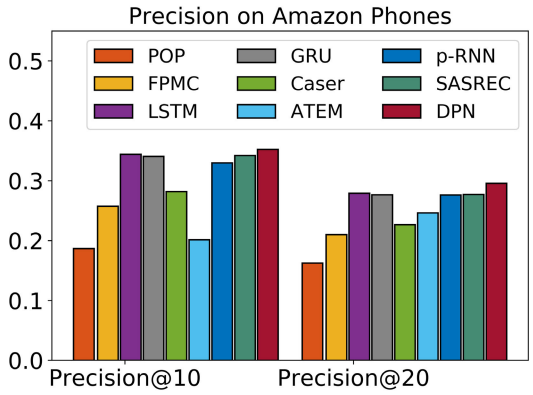
- GRU. A RNN-based model which adopted GRU to capture sequential dependencies [10]. The dimension of the hidden state of GRU is set to 30.
- Caser. A convolutional sequence embedding model which embeds a sequence of items into a matrix, then learns sequential preferences as local features using convolutional filters [23]. The dimension of the embedding size is set to 50.
- ATEM. An attentive embedding model with a soft attention mechanism to emphasize those items in a sequence context which are more important and relevant to users' next choice [22]. The dimension of the embedding is set to 50.
- SASRec. A sequential recommendation model with a self-attention mechanism [11]. The dimension of item embedding is set to 50. Other parameters are set as suggested by the authors of the method.
- p-RNN. A feature-rich SRSs model which brings textual and visual factors as the additional input [34]. It adopts a parallel GRU-based network architecture where item ID, textual features and visual features are modelled in three sub-nets independently and the hidden states of these sub-nets are fused at the last step for the next item prediction. In our work, we do not involve textual features of items, and thus we adapted p-RNN into our case by implementing it into a parallel GRU-based network containing two independent sub-nets, one for modeling item IDs and the other for modeling item visual features. The dimension of the hidden state is set to 30.

C. RECOMMENDATION ACCURACY COMPARISON

We compare the recommendation accuracy of our proposed model DPN and that of compared SRS methods. Tables 2 and 3 and Figure 3 report the Recall, MRR, nDCG and Precision respectively. RAND cannot model any users' preferences on items, and thus performs worst among all these methods. POP always recommends identical items to each user and thus could not model users' personalized preference,



(a) Precision on Amazon Clothing

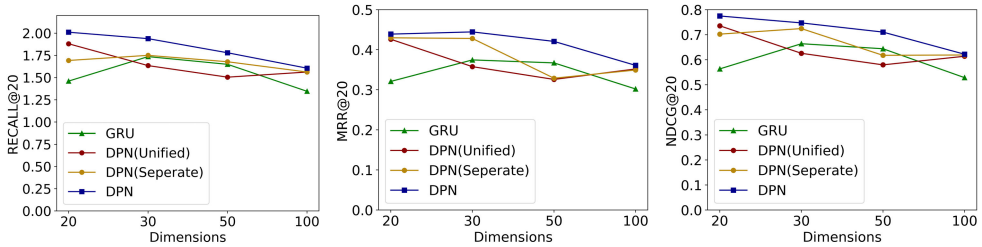


(b) Precision on Amazon Phone

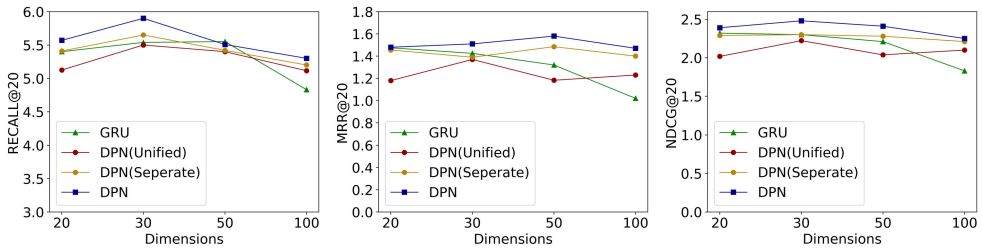
FIGURE 3. The precision of DPN and all other compared models. DPN leads all other methods with a clear margin on both datasets.

not to say users' preferences embedded in their sequences of interactions with items.

FPMC only captures first-order dependencies in modeling sequential behaviors while ignoring the high-order ones, resulting in worse performance compared with sequential models like LSTM. Although, LSTM and GRU can model the high-order and long-term sequential dependencies, it can easily generate false dependencies due to the employed over-strong order assumption between any two adjacent items



(a) Amazon Clothing



(b) Amazon Phones

FIGURE 4. Recall@20, MRR@20, NDCG@20 of DPN and its three simplified versions, i.e., DPN(Unified), DPN(Seperate) and GRU, when they are with varied hidden state dimensions d = [20;30;50;100]. DPN exceeds all the other three models under almost all the different dimension settings.

in a sequence. In addition, it may be also easy to bias to the most recent items in a sequence. Caser does not perform very well either as it lacks the ability to model long-range dependencies in a sequence due to the limited perceptive fields in the pooling layers in CNN. ATEM dismisses the order of items in a sequence and usually biases to the minor main preference while overlooking other. Hence it could not perform very well. Further more, all of the above compared SRS models only take item IDs as their input to model users' main preferences towards items' main aspects, e.g., function. They are not able to model users' additional preference, e.g., visual preference, on items' other aspects, e.g., appearance. By utilizing the powerful self-attention mechanism to better capture the complex dependencies embedded in sequential context, SASRec clearly improves the performance compared with ATEM. Similarly, it only focus on the users' main preferences towards items while ignoring their visual preference.

By incorporating additional visual features of items into SRSs, p-RNN achieved higher recommendation accuracy than the above methods. However, in p-RNN, item IDs and item features are modelled in two separate sub-nets independently. In this way, the interactions between these two different factors are lost, particularly, at each step, the impact of users' visual preference reflected by item visual features on their main preference and choice on items is ignored.

In our proposed DPN, the dimension of item ID embedding is set to 30 on both datasets while the dimension of hidden state is empirically set to 20 and 30 on Amazon Clothing and Amazon Phones respectively for the best performance. By modeling users' main preference embedded in item IDs and their visual preference embedded in item visual features, as well as the interactions between them, with a particularly designed coupled double-chain network, our DPN model is able to comprehensively model users' preferences towards both the main aspect and visual aspects of items. As a result, DPN achieves the best performance on all datasets. Particularly, DPN demonstrates 5% improvement over the best-performing compared method in terms of Recall@20, nDCG@20 on both datasets (c.f. Tables 2 and 3). The precision (c.f. Figure 3) on two datasets also shows DPN leads the baselines with a clear margin.

D. ABLATION AND DIMENSION ANALYSIS

To demonstrate the effectiveness of our proposed DPN model for user preference modeling, we implemented the full model of DPN plus three simplified versions of it to show the contribution of each key module. In addition, to test the sensitivity of key hyper parameter, i.e., the dimension of hidden state, we test the recommendation performance of these four models with different dimension settings. To be specific, the four models include: (1) the full model of DPN; (2) a single chain network called DPN (Unified) with only one chain is adopted to model users' main and visual preference together. Specifically, at each time step, the embeddings of ID and visual feature of each item are concatenated into one unified embedding vector of the item, which is then input into the single chain network; (3) a separate-chains network called DPN (Separate), where the bridge at each time step in DPN is removed and thus users' main preference and visual preference are modelled in two separate chains independently. The hidden states from both chains are fused at the last time step for the next item prediction; (4) GRU-based RNN, denoted as GRU, one of the baseline models, which can also be seen as the most simplified version of DPN. It is a single chain RNN by taking GRU as its cell while only the item ID is taken as

its input. The dimension of the hidden state in each of these models was set to $d = [20; 30; 50; 100]$.

Figure 4 presents the recommendation performance in terms of Recall@20, MRR@20, and nDCG@20 of the above four models when the dimensions of their hidden states are set to 20, 30, 50 and 100 respectively. It is clear that in most of the cases, the full model DPN clearly outperforms all the simplified versions, which demonstrates the rationality of the design of our model DPN. To be specific, the superiority of DPN over the single chain network, i.e., DPN-Unified, justified the importance to model users' main and visual preference in different chains respectively. The superiority of DPN over the separate-chains network, i.e., DPN(Separate), demonstrates the significance to consider the interaction between users' main preference and visual preference.

It is also worth noting that on Amazon Phones dataset, the GRU model which only takes item ID as the input achieves similar or sometimes better performance as the other two simplified versions of DPN which takes both item ID and visual features into account, i.e., DPN(Unified) and DPN(Separate). One possible reason is that the visual features of phones are not as sensitive as those of clothing, and thus they may easily bring noisy information for the preference modeling. In practice, to fix this issue, we have devised an attention module in the full model of DPN to carefully extract those more important and relevant information from item appearance images while reducing the interference of noisy information. The superiority of DPN over all other three models proves the efficacy of our designed attention module in DPN. For the dimensions, it seems all the models achieved the best performance when the dimension is set to 20 on Amazon Clothing dataset and 30 on Amazon Phones dataset. When the dimension increase, the performance decrease. This may be caused by the overfilling.

VI. CONCLUSION AND FUTURE WORK

In this paper, in Section 1, we introduced the research problem of how to learn users' visual preference in sequential recommendations and justified the significance of this problem. In Section 2, we extensively reviewed the related work in the sequential recommendation area. Then in Section 3, we formalized the problem, followed by our approach proposed in Section 4. Then, we introduced our experiments in Section 5.

To be specific, we have proposed DPN for visual preference learning to better learn users' visual preference towards items' appearance for generating more accurate sequential recommendations, which cannot be well addressed by existing studies on SRSs. In DPN, an item image chain and an item ID chain are devised to learn users' visual preference on item images and main preference on item IDs respectively. In addition, the interaction between these two kinds of preferences at each time step is carefully modelled. Finally, the two kinds of preferences are integrated to form the compound preference for the next item recommendations. Extensive experiments conducted on two real-world datasets have shown the superiority of our proposed DPN over the representative and state-of-the-art SRSs. In addition, the ablation analysis has justified the rationality of our design and the significance of the incorporation of item visual features when making recommendations.

Compared with the state-of-the-art sequential recommendation approaches, our proposed DPN can well capture users' visual preferences towards items. As a result, it is able to recommend items which can not only well satisfy users' demands w.r.t. item functions but also well match users' personalized preference on items' appearance and styles. DPN is particularly useful in the real-world scenarios where the items are visually-sensitive, such as clothes and jewellery.

In the future, on the one hand, we will explore the utilization of more advanced attention mechanisms such as cross-attention [46] to better model the influence of item images on users' choice. On the other hand, we will explore the application of DPN to the problem of explainable recommendations, i.e., to better explain the recommendation results by modeling users' visual preference.

REFERENCES

[1] A. M. Rashid, I. Albert, D. Cosley, S. K. Lam, S. M. McNee, J. A. Konstan, and J. Riedl, "Getting to know you: Learning new user preferences in recommender systems," in *Proc. 7th Int. Conf. Intell. user Interfaces (IUI)*, 2002, pp. 127–134.

[2] F. Ricci, L. Rokach, and B. Shapira, "Recommender systems: Introduction and challenges," in *Recommender Systems Handbook*. New York, NY, USA: Springer, 2015, pp. 1–34.

[3] A. Said and A. Bellogín, "Coherence and inconsistencies in rating behavior: Estimating the magic barrier of recommender systems," *User Model. User-Adapted Interact.*, vol. 28, no. 2, pp. 97–125, 2018.

[4] D. Rafailidis and F. Crestani, "Learning to rank with trust and distrust in recommender systems," in *Proc. 11th ACM Conf. Recommender Syst.*, Aug. 2017, pp. 5–13.

[5] R. Saia, L. Boratto, and S. Carta, "A semantic approach to remove incoherent items from a user profile and improve the accuracy of a recommender system," *J. Intell. Inf. Syst.*, vol. 47, no. 1, pp. 111–134, 2016.

[6] H. Zhang, Y. Sun, M. Zhao, T. W. S. Chow, and Q. M. J. Wu, "Bridging user interest to item content for recommender systems: An optimization model," *IEEE Trans. Cybern.*, vol. 50, no. 10, pp. 4268–4280, Oct. 2020.

[7] R. Yin, K. Li, J. Lu, and G. Zhang, "Enhancing fashion recommendation with visual compatibility relationship," in *Proc. World Wide Web Conf. (WWW)*, 2019, pp. 3434–3440.

[8] S. C. Hidayati, C.-C. Hsu, Y.-T. Chang, K.-L. Hua, J. Fu, and W.-H. Cheng, "What dress fits me best?: Fashion recommendation on the clothing style for personal body shape," in *Proc. 26th ACM Int. Conf. Multimedia*, Oct. 2018, pp. 438–446.

[9] 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.

[10] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in *Proc. 4th Int. Conf. Learn. Represent.*, 2016, pp. 1–10.

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

[12] S. Wang, L. Hu, and L. Cao, "Perceiving the next choice with comprehensive transaction embeddings for online recommendation," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2017, pp. 285–302.

[13] G.-E. Yap, X.-L. Li, and S. Y. Philip, "Effective next-items recommendation via personalized sequential pattern mining," in *Proc. 17th Int. Conf. Database Syst. Adv. Appl.*, 2012, pp. 48–64.

[14] S. Feng, X. Li, Y. Zeng, G. Cong, Y. M. Chee, and Q. Yuan, "Personalized ranking metric embedding for next new poi recommendation," in *Proc. 24th Int. Conf. Artif. Intell.*, 2015, pp. 2069–2075.

[15] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized Markov chains for next-basket recommendation," in *Proc. 19th Int. Conf. World Wide Web (WWW)*, 2010, pp. 811–820.

[16] D. Jannach and M. Ludewig, "When recurrent neural networks meet the neighborhood for session-based recommendation," in *Proc. 11th ACM Conf. Recommender Syst.*, Aug. 2017, pp. 306–310.
[17] M. Ludewig, N. Mauro, S. Latifi, and D. Jannach, "Empirical analysis of session-based recommendation algorithms," *User Model. User-Adap.*, vol. 31, no. 1, pp. 149–181, 2021.
[18] D. Garg, P. Gupta, P. Malhotra, L. Vig, and G. Shroff, "Sequence and time aware neighborhood for session-based recommendations: STAN," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2019, pp. 1069–1072.
[19] B. Hidasi and A. Karatzoglou, "Recurrent neural networks with top-k gains for session-based recommendations," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 843–852.
[20] C.-Y. Wu, A. Ahmed, A. Beutel, A. J. Smola, and H. Jing, "Recurrent recommender networks," in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, Feb. 2017, pp. 495–503.
[21] 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.
[22] S. Wang, L. Hu, L. Cao, X. Huang, D. Lian, and W. Liu, "Attention-based transactional context embedding for next-item recommendation," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 2532–2539.
[23] 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.
[24] 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.
[25] T. X. Tuan and T. M. Phuong, "3D convolutional networks for session-based recommendation with content features," in *Proc. 11th ACM Conf. Recommender Syst.*, Aug. 2017, pp. 138–146.
[26] 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.*, Aug. 2019, pp. 4320–4326.
[27] R. Ren, Z. Liu, Y. Li, W. X. Zhao, H. Wang, B. Ding, and J.-R. Wen, "Sequential recommendation with self-attentive multi-adversarial network," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 89–98.
[28] Q. Cui, S. Wu, Y. Huang, and L. Wang, "A hierarchical contextual attention-based network for sequential recommendation," *Neurocomputing*, vol. 358, pp. 141–149, Sep. 2019.
[29] Q. Liu, Y. Zeng, R. Mokhosi, and H. Zhang, "STAMP: Short-term attention/memory priority model for session-based recommendation," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1831–1839.
[30] 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.
[31] S. M. Cho, E. Park, and S. Yoo, "MEANTIME: Mixture of attention mechanisms with multi-temporal embeddings for sequential recommendation," in *Proc. 14th ACM Conf. Recommender Syst.*, Sep. 2020, pp. 515–520.
[32] 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.
[33] W. Ye, S. Wang, X. Chen, X. Wang, Z. Qin, and D. Yin, "Time matters: Sequential recommendation with complex temporal information," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 1459–1468.
[34] B. Hidasi, M. Quadrana, A. Karatzoglou, and D. Tikk, "Parallel recurrent neural network architectures for feature-rich session-based recommendations," in *Proc. 10th ACM Conf. Recommender Syst.*, Sep. 2016, pp. 241–248.
[35] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, and A. C. Berg, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[36] Q. Liu, S. Wu, and L. Wang, "DeepStyle: Learning user preferences for visual recommendation," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2017, pp. 841–844.
[37] H. Lee, J. Seol, and S.-G. Lee, "Style2 Vec: Representation learning for fashion items from style sets," 2017, *arXiv:1708.04014*.
[38] Z. Li, B. Wu, Q. Liu, L. Wu, H. Zhao, and T. Mei, "Learning the compositional visual coherence for complementary recommendations," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 3536–3543.
[39] W.-C. Kang, C. Fang, Z. Wang, and J. McAuley, "Visually-aware fashion recommendation and design with generative image models," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2017, pp. 207–216.
[40] H. Fang, D. Zhang, Y. Shu, and G. Guo, "Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations," *ACM Trans. Inf. Syst. (TOIS)*, vol. 39, no. 1, pp. 1–42, 2020.
[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[42] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Feb. 2011.
[43] 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.
[44] Q. Cui, S. Wu, Q. Liu, W. Zhong, and L. Wang, "MV-RNN: A multi-view recurrent neural network for sequential recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 2, pp. 317–331, Feb. 2020.
[45] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, "A dynamic recurrent model for next basket recommendation," in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2016, pp. 729–732.
[46] Y. Ji, H. Zhang, Z. Jie, L. Ma, and Q. M. Jonathan Wu, "CASNet: A cross-attention Siamese network for video salient object detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 6, pp. 2676–2690, Jun. 2021.

SHOUJIN WANG received the Ph.D. degree in computer science. He is currently a Postdoctoral Researcher with the School of Communication and Information Engineering and Institute of Smart City, Shanghai University. His research interests include computer version and visual content understanding.

CHENLU YANG is currently pursuing the master's degree in computer science with the School of Communication and Information Engineering and Institute of Smart City, Shanghai University. Her current research interests include recommendation systems and text-based semantic understanding of images.

TONG QU received the B.E. degree in electronic science and technology from the School of Communication and Information Engineering and Institute of Smart City, Shanghai University, Shanghai, China, in 2018, where he is currently pursuing the master's degree, under the supervision of Professor Wanggen Wan. His current research interests include recommender systems and data mining.

KAI YANG is currently pursuing the dual Ph.D. degree with the School of Communication and Information Engineering, Shanghai University, China, and the School of Software Engineering, University of Technology Sydney, Australia. He is a member of the Institute of Smart City, Shanghai University, and the Decision Systems and e-Service Intelligence Research Laboratory, Centre for Artificial Intelligence, University of Technology Sydney. His research interests include transfer learning and multi-source domain adaptation.

WANGGEN WAN (Senior Member, IEEE) received the Ph.D. degree in electronic engineering from Xidian University, Xi'an, China, in 1992. He is currently a Professor with the School of Communication and Information Engineering and Institute of Smart City, Shanghai University. His research interests include virtual reality, data mining, and computer vision. He is a fellow of IET.