Global and Personalized Graphs for Heterogeneous Sequential Recommendation by Learning Behavior Transitions and User Intentions

Weixin Chen
Shenzhen University
Shenzhen, China
2016154094@email.szu.edu.cn

Mingkai He
Shenzhen University
Shenzhen, China
mingkai_he@163.com

Yongxin Ni
National University of Singapore
Queenstown, Singapore
niyongxin@u.nus.edu

Weike Pan*
Shenzhen University
Shenzhen, China
panweike@szu.edu.cn

Zhong Ming*
Shenzhen University
Shenzhen, China
mingz@szu.edu.cn

Li Chen
Hong Kong Baptist University
Hong Kong, China
lichen@comp.hkbu.edu.hk

ABSTRACT

Heterogeneous sequential recommendation (HSR) is a very important recommendation problem, which aims to predict a user's next interacted item under a target behavior type (e.g., purchase in e-commerce sites) based on his/her historical interactions with different behaviors. Though existing sequential methods have achieved advanced performance by considering the varied impacts of interactions with sequential information, a large body of them still have two major shortcomings. Firstly, they usually model different behaviors separately without considering the correlations between them. The transitions from item to item under diverse behaviors indicate some users' potential behavior manner. Secondly, though the behavior information contains a user's fine-grained interests, the insufficient consideration of the local context information limits them from well understanding user intentions. Utilizing the adjacent interactions to better understand a user's behavior could improve the certainty of prediction. To address these two issues, we propose a novel solution utilizing global and personalized graphs for HSR (GPG4HSR) to learn behavior transitions and user intentions. Specifically, our GPG4HSR consists of two graphs, i.e., a global graph to capture the transitions between different behaviors, and a personalized graph to model items with behaviors by further considering the distinct user intentions of the adjacent contextually relevant nodes. Extensive experiments on four public datasets with the state-of-the-art baselines demonstrate the effectiveness and general applicability of our method GPG4HSR.

KEYWORDS

Sequential Recommendation, Graph Neural Network, Behavior Transition, User Intention

ACM Reference Format:
Weixin Chen, Mingkai He, Yongxin Ni, Weike Pan, Zhong Ming, and Li Chen. 2022. Global and Personalized Graphs for Heterogeneous Sequential Recommendation by Learning Behavior Transitions and User Intentions. In *Sixteenth ACM Conference on Recommender Systems (RecSys '22), September 18–23, 2022, Seattle, W A, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3523227.3546761

1 INTRODUCTION

In lots of scenes including e-commerce sites, recommender systems have played a crucial booster role. Through learning from the historical (user, item) interactions, it is able to generate the digital scores or a ranked list for the uninteracted items, and recommend some appropriate items for each user to alleviate the information overload problem. As a user's preferences widely exist in the form of sequence and the way he/she interacted with items, how to model the user's sequential and heterogeneous behaviors becomes a research hotspot recently.

Although existing methods model the sequential and behavior information in an effective way, they usually suffer from the following two issues. Firstly, they often ignore the correlations of different behaviors. Specifically, the different transitions according to their behavior types indicate importance to a different extent, but modeling behaviors separately, i.e., splitting an interaction sequence into multiple subsequences according to the behavior types, cannot learn the potential correlations well as the connection between different types of behaviors is broken. Secondly, they usually model behaviors without utilizing the local context information. Even though the behavior information could capture different levels of user intentions, the context information is able to further enhance the certainty and decrease the randomness. For example, if a user Alice performed a sequence of browsing item A, adding item A to the cart, and purchasing item A, and another user Bob also performed an interaction sequence but with a slight difference, i.e., browsing item A, adding item A to the cart, and browsing item B. The behavior of adding item A to the cart exists in both sequences, which indicates a high probability of purchasing item A. However, the subsequent interactions are quite different, and are also very useful to obtain an enhanced understanding of "adding item A to the cart" for each user, i.e., Alice would purchase items in cart with higher certainty, but there is more randomness for Bob to do so.

[*] co-corresponding authors

RecSys '22, September 18–23, 2022, Seattle, WA, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9278-5/22/09…$15.00
https://doi.org/10.1145/3523227.3546761

Most of traditional recommendation algorithms [9, 18] fill a rating matrix of (user, item) pairs to deal with general recommendation, without utilizing the sequential and behavior information. Compared with general recommendation, early works for sequential recommendation [8, 10, 21, 29] obtain better performance since they capture more accurate preferences by learning the sequential patterns. For instance, GRU4Rec [8] utilizes gated recurrent units (GRUs), a variant of recurrent neural networks (RNNs), to transmit the previous information to subsequent units for learning the sequential patterns inherently. With the increasing amount of e-commerce users and items, the information overload problem becomes crucial, while the increasing behavior data offers more detailed information to gain a deeper understanding of a user's intentions. This actually raises an emerging and valuable topic, i.e., heterogeneous sequential recommendation (HSR). For example, TransRec++ [31] adapts TransRec [5] to strengthen its ability of capturing the behavior transitions of users, which considers the correlations between different behaviors but lacks consideration of distinguishing those behaviors. RIB [32] concatenates the item and behavior embeddings as the input embedding to consider the influences of different behaviors, and adopts an RNN layer to learn the sequential patterns. However, concatenation of two embeddings could only learn the distinct importance of behaviors but not the relations between them. MGNN-SPred [23] constructs two item-to-item graphs by incorporating the context information between items under the same behavior in order to learn the transitions, but ignores the transitions under different behaviors.

To overcome these issues, we propose a novel solution called global and personalized graphs for heterogeneous sequential recommendation (GPG4HSR). Firstly, we introduce a global graph layer, equipped with a global graph (GG) of items and behavior transitions to capture the transitions between all behaviors, which addresses the aforementioned first issue. Secondly, to learn user intentions by distinguishing the varied effects of the same interaction with different context information and address the second issue, we utilize a personalized graph layer with a personalized graph (PG) of interaction sequence to model heterogeneous feedback with the influence of adjacent interactions. In addition, to balance these two modules (i.e., the global and personalized graph layers) and take a further step in learning the sequential patterns, we apply an effective fusion layer and an attention layer. Finally, we introduce a prediction layer in order to predict the next likely-to-purchase item. Note that we group all behaviors into examinations and purchases as their two representative behavior types, and the exploratory study in Section 6.2 indicates that our GPG4HSR is applicable to scenes with more types of behaviors.

We summarize our main contributions as follows: (i) We propose a novel model called GPG4HSR to address two fundamental issues in HSR by considering the correlations of all behaviors via a global graph (GG) and enhancing the interaction representation with the context information via a personalized graph (PG). (ii) We develop a global graph layer and a personalized graph layer to model GG and PG, respectively, in which the former weighs the transitions between different behaviors and the latter incorporates the representations of a local behavior and its adjacent contextually relevant interactions. (iii) We study the recommendation performance of our GPG4HSR and eight competitive baselines on four datasets, and find that our GPG4HSR performs the best in most cases. Moreover, further studies, including exploratory studies with more than two different types of behaviors, ablation studies of the major components, framework studies with different backbone models and sensitivity analysis of the hyperparameter, show the effectiveness of our solution.

2 RELATED WORK

2.1 General Recommendation

Bayesian personalized ranking (BPR) [18] factorizes an implicit feedback matrix with a pairwise preference assumption that users prefer interacted items to uninteracted ones. Fusing item similarity model (FISM) [9] integrates an item similarity model into matrix factorization by replacing a user-specific latent vector with an aggregated latent vector of his/her interacted items. Recently, due to the superior performance of deep learning (DL)-based methods, more works are developed by taking advantage of their merits. Among them, a large amount of graph neural network (GNN)-based recommendation algorithms [1, 7, 15, 24, 30] make use of (user, item) bipartite graphs to capture the collaborative information. For example, PinSage [30] combines random walk and GraphSage [3] to successfully apply GNN to commercial recommender systems. GHCF [1] captures heterogeneous relations with high-hop structure of (user, item) interactions based on graph convolutional network (GCN). NGCF [24] explicitly learns node embeddings with the collaborative filtering signals through the graph structure. By removing the nonlinear part in NGCF, LightGCN [7] obtains a simple but effective model. UltraGCN [15] derives a compact model from the ultimate limit form of LightGCN, which thus avoids some critical problems such as over-smoothing caused by stacked graph networks. Most recently, some studies [26, 28] learn multi-behavior information by meta learning. For instance, CML [26] distills transferable knowledge across different behavior types through a multi-behavior contrastive learning framework. However, all these methods do not utilize the sequential information, which is very important in learning users' preferences.

2.2 Sequential Recommendation

Compared with the general recommendation methods, sequential recommendation methods predict the next likely-to-interact item through exploiting the extra position information of the interactions that provides more accurate real-time interests and intentions of the users. We classify sequential recommendation methods into two branches according to the number of behavior types exploited.

2.2.1 Homogeneous Sequential Recommendation. Early works on sequential recommendation only tackle one single type of behavior on items such as one-class or homogeneous implicit feedback, and learn the sequential patterns relying on Markov chains or deep neural networks (DNNs). Methods based on Markov chains [6] model user preferences with transitions of sequential items where the current state of an individual taste is only affected by the last state and the current interacted item. FPMC [19] extends matrix factorization with Markov chains to learn both the long-term and short-term preferences with consecutive items for each user. Fossil [6] further adopts higher-order Markov chains to address the sparsity issue. Translation-based recommendation (TransRec) [5] applies a translation-based structure in which it regards the items as space and users as transition vectors. DNN-based methods learn the sequential patterns through different structures of networks such as RNN [8], CNN [21], attention [10] and GNN [23, 25, 27, 29]. The RNN-based method GRU4Rec [8] inherently utilizes the sequential information by updating the state recurrently but facing the notorious gradient vanishing problem. CNN-based methods [21] slide elaborately designed convolution filters over an embedding matrix to extract low dimensional features. Attention-based methods [10] add position embeddings and item embeddings to learn features of items with the position information and pay the calculated attention according to some designated rules to them. GNN-based methods [25, 27, 29] usually construct graphs in diverse forms to capture deep relations between users and/or items, especially some item-to-item graphs that capture complicated transitions of items efficiently. Based on SR-GNN [27], GC-SAN [29] uses a self-attention mechanism to better capture the contextual information between the historical items. GCE-GNN [25] constructs a session graph and a global graph for learning session-level item representations and global-level item representations, respectively, which is similar to our work. However, it lacks consideration of the effects caused by different user behaviors. All these works only model homogeneous behaviors and ignore that the heterogeneous feedback can better reflect the preferences of users in different degrees.

2.2.2 Heterogeneous Sequential Recommendation. There are few works modeling both sequential and multi-behaviors information, and most of them are DL-based methods. As one of the few matrix factorization (MF)-based methods, TransRec++[31] extends TransRec [5] with behavior transition vectors to learn the dynamics of users' heterogeneous behaviors. Compared with the MF-based methods, DL-based methods are more powerful in processing sequences and heterogeneous behaviors. RIB [32] concatenates a behavior embedding with an item embedding to introduce behavior-specific influences, along with an RNN layer and an attention layer to capture the sequential patterns and distinguish the varied effects of interactions, respectively. BINN [11] uses w-item2vec to obtain a unified representation of each item and takes the behavior information as an auxiliary input of the bidirectional RNN modules. MKM-SR [16] learns an item embedding and a behavior embedding via gated GNN (GGNN) and GRU, respectively, and concatenates them as an interaction embedding. MGNN-SPred [23] is a closely related work to ours that captures the sequential information and distinguishes the impact of different behaviors via GNN. Similarly, MBGNN [17] and MGCNet [20] learn the transition patterns considering only the same types of behaviors. They split one sequence into multiple subsequences according to the behavior types on items, assuming that each type of behavior is independent of each other and ignoring the correlations between different behaviors.

We can see that recommendation with users' sequential and heterogeneous feedback or behaviors has the potential to learn the users' preferences more accurately, but the existing few works can not exploit the global correlations of the users' behaviors and the local context in each user's sequence well. This motivates us to design a novel solution for modeling users' sequential and heterogeneous feedback in recommender systems.

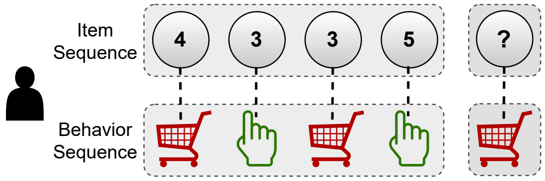

Figure 1: Illustration of the heterogeneous sequential recommendation (HSR) problem, where each number denotes an item ID.

3 PRELIMINARIES

3.1 Problem Definition

In this paper, we aim to address the problem of heterogeneous sequential recommendation (HSR) in which the prediction of the next likely-to-purchase item is based on an interaction sequence of items with two or more different types of behaviors. Note that different from session-based recommendation, users are not anonymous in sequential recommendation. With the goal to model the behaviors in a more general and common way, we categorize all the behaviors into two major types, i.e., examination e and purchase p, where the former denotes all relatively positive non-target behaviors such as clicks, adds-to-cart and favorites in e-commerce sites, and the latter represents the target behavior.

To be specific, we have an interaction sequence $S_u = \{(i^1, b^1), (i^2, b^2), \dots, (i^t, b^t), \dots, (i^{|S_u|}, b^{|S_u|})\}$ for each user u, where (i^t, b^t) denotes the (item, behavior) pair of the t-th interaction in sequence S_u. Depending on the historical data above, our goal is to recommend the next likely-to-purchase item for each user u more accurately. We illustrate the studied HSR problem in Figure 1.

3.2 Challenges

The heterogeneous feedback provides more detailed information of a user's interactions but also brings the following challenges to be addressed. Firstly, how to model heterogeneous behaviors in sequential recommendation. Different behaviors which represent different levels of users' interests should be distinguished when modeling the interactions in a sequence. There are rife solutions for sequential recommendation, e.g., Caser [21], GRU4Rec [8] and SASRec [10], which could model the item sequences adequately from different classic deep-learning perspectives, but they do not consider the heterogeneity of the users' feedback. Secondly, how to capture the global relationship between items with different types of behaviors more accurately when the user behavior data is sparse. Since the interactions between a certain user and the items are often very rare, some GNN-based methods like MKM-SR [16] and MGNN-SPred [23] turn to capture the relationship between items via some global graphs. However, they both underrate the correlations of different behaviors because they split an interaction sequence as separated ones w.r.t. the behavior types and ignore the correlations between different behaviors.

3.3 Overview of Our Solution

To address the aforementioned challenges, we propose a novel solution called GPG4HSR shown in Figure 2, which mainly consists of two types of graphs, i.e., a global graph (GG) $\mathcal{G}_g$ of items and behavior transitions for all users and a personalized graph (PG) $\mathcal{G}_p^u$ of interaction sequence for each user u.

In order to capture the relationship between items with heterogeneous behaviors, we construct a global item graph $\mathcal{G}_g = (\mathcal{V}_g, \mathcal{E}_g)$, where $\mathcal{G}_g$ takes all the items as nodes $\mathcal{V}_g$ and defines four basic edge types in the global edge set $\mathcal{E}_g$, i.e., from examination to examination (*e2e*), from purchase to purchase (*p2p*), from examination to purchase (*e2p*) and from purchase to examination (*p2e*). To better leverage the importance of heterogeneous behaviors, we assign a learnable weight to each basic edge type by an edge gating network. For further distinguishing the order of two items on behavior transitions *e2p* and *p2e* and identifying the behavior on an item, we extend each one of them to two sub directed edge types with suffixes '+' and '−', where '+' and '−' mean an inward direction and an outward direction, respectively. For example, a directed edge of type *e2p−* on node pair $(\boldsymbol{v}, \boldsymbol{v}')$ (or a corresponding edge of type *e2p+* on node pair $(\boldsymbol{v}', \boldsymbol{v})$) denotes that a user examines the item of node $\boldsymbol{v}$ and subsequently purchases the item of node $\boldsymbol{v}'$. These two sub edge types will share half the assigned weight of their parent edge type, i.e., *e2p*.

Though the structure information of one single sequence may not be sufficient to improve the representations of the involved interactions, there is still some potentially valuable information from the contextually relevant interactions worth discovering since they could contribute to capturing the corresponding intentions of a target user. Hence, we also construct a user-specific graph $\mathcal{G}_p^u = (\mathcal{V}_p^u, \mathcal{E}_p^u)$ to enhance the understanding of each user u's personalized interests by taking the items with specific behaviors as nodes. Different from fusing the behavior information into the edges on $\mathcal{G}_g$, we combine the items and their associated behavior information into the nodes in the personalized graph $\mathcal{G}_p^u$, and also consider two edge types + and − which are identical to that in $\mathcal{G}_g$. For the goal of distinguishing the importance of different transitions, we also consider the frequency f of each edge in both the global and personalized graphs. The construction of $\mathcal{G}_g$ and $\mathcal{G}_p^u$ is very efficient. Specifically, we just need to traverse all the interactions in the form of each user's sequence which indicates the algorithmic complexity of our graph construction is $O(|\mathcal{R}|)$, where $\mathcal{R}$ is the whole set of observed (user, item, behavior) tuples.

We also introduce a global personalized fusion (GPF) to balance and combine the global and personalized graph representations in order to acquire the final graph representation of a sequence. Subsequently, we apply an attention layer from SASRec [10] which stacks multiple self-attention blocks followed by GPF to weigh the sequential items within the final graph representation efficiently. Then a prediction layer is placed at the end to score and rank all the candidate items for top-N recommendation.

4 OUR SOLUTION

4.1 The Input and Embedding Layer

For each sequence S_u, we truncate the latest L (item, behavior) pairs, i.e, $S_u = \{(i^1, b^1), (i^2, b^2), \ldots, (i^L, b^L)\}$, for which our goal is to predict the next item i^{L+1} under the behavior type b^{L+1}. For each interaction pair (i^t, b^t), we have three one-hot indicator vectors $\mathbf{x}_i \in \mathbb{R}^{m\times 1}$, $\mathbf{x}_b \in \mathbb{R}^{2\times 1}$ and $\mathbf{x}_t \in \mathbb{R}^{L\times 1}$ corresponding to the item i^t, the behavior type b^t and the position t, respectively. And we obtain the corresponding latent feature vectors $\mathbf{v}_{i^t} \in \mathbb{R}^{d\times 1}$, $\mathbf{b}_{b^t} \in \mathbb{R}^{d\times 1}$ and $\mathbf{p}_t \in \mathbb{R}^{d\times 1}$ by multiplying their embedding matrices $\mathbf{W}^I \in \mathbb{R}^{d\times m}$, $\mathbf{W}^B \in \mathbb{R}^{d\times 2}$ and $\mathbf{W}^P \in \mathbb{R}^{d\times L}$:

$$\mathbf{v}_{i^t} = \mathbf{W}^I \mathbf{x}_{i^t},\ \mathbf{b}_{b^t} = \mathbf{W}^B \mathbf{x}_{b^t},\ \mathbf{p}_t = \mathbf{W}^P \mathbf{x}_t \tag{1}$$

4.2 The Global Graph Layer

In order to study the behavior transitions, we utilize the proposed global graph ($\mathcal{G}_g$) which captures the transitions between the items with behaviors to learn the diversified effects of each specific transition type. For each item i^t in a sequence S_u, we first feed its global node $\boldsymbol{v}_{i^t}$ and all the edges linked to it in $\mathcal{G}_g$ into a global graph layer. For brevity, we remove the subscript symbol i^t of the node $\boldsymbol{v}_{i^t}$ and denote it as $\boldsymbol{v}$. We classify the neighbour groups w.r.t. the node $\boldsymbol{v}$ by six edge types, i.e., *e2e*, *p2p*, *e2p+*, *e2p−*, *p2e+* and *p2e−*, to distinguish a specific transition of behaviors. Taking *e2p+* as an example, we generate its neighbour groups and the corresponding update vectors as follows:

$$\mathcal{A}_{e2p+}(\boldsymbol{v}) \quad = \quad \{(\boldsymbol{v}', f) \mid (\boldsymbol{v}, \boldsymbol{v}', f, e2p+) \in \mathcal{E}_g\} \tag{2}$$

$$\mathbf{h}_{\boldsymbol{v}}^{e2p+} \quad = \quad \frac{\sum_{(\boldsymbol{v}', f)\in\mathcal{A}_{e2p+}(\boldsymbol{v})} f \times \mathbf{v}_{\boldsymbol{v}'}}{\sum_{(\boldsymbol{v}', f)\in\mathcal{A}_{e2p+}(\boldsymbol{v})} f} \tag{3}$$

where $\boldsymbol{v}'$ denotes a node that links to v, $f \in \mathbb{R}$ denotes the frequency of the corresponding edge, and $\mathbf{v}_{\boldsymbol{v}'} \in \mathbb{R}^{d\times 1}$ denotes the feature vector of item i' w.r.t. the node $\boldsymbol{v}'$. For example, $(\boldsymbol{v}, \boldsymbol{v}', f, e2p+)$ represents an edge from an examined item $\boldsymbol{v}'$ to a purchased item $\boldsymbol{v}$ with a frequency f. In particular, we take the mean of the sub update vectors $\mathbf{h}_{\boldsymbol{v}}^{e2p+}$ and $\mathbf{h}_{\boldsymbol{v}}^{e2p-}$ as their parent update vector $\mathbf{h}_{\boldsymbol{v}}^{e2p}$, and the mean of $\mathbf{h}_{\boldsymbol{v}}^{p2e+}$ and $\mathbf{h}_{\boldsymbol{v}}^{p2e-}$ as $\mathbf{h}_{\boldsymbol{v}}^{p2e}$ to balance the importance of the inward and outward relations:

$$\mathbf{h}_{\boldsymbol{v}}^{e2p} = \frac{1}{2}(\mathbf{h}_{\boldsymbol{v}}^{e2p+} + \mathbf{h}_{\boldsymbol{v}}^{e2p-}), \quad \mathbf{h}_{\boldsymbol{v}}^{p2e} = \frac{1}{2}(\mathbf{h}_{\boldsymbol{v}}^{p2e+} + \mathbf{h}_{\boldsymbol{v}}^{p2e-}) \tag{4}$$

We also design a gating network inside the global layer to weigh different item-to-item relationship:

$$\mathbf{w}_g = \mathbf{W}_g[\mathbf{h}_{\boldsymbol{v}}^{e2e}; \mathbf{h}_{\boldsymbol{v}}^{p2p}; \mathbf{h}_{\boldsymbol{v}}^{e2p+}; \mathbf{h}_{\boldsymbol{v}}^{e2p-}; \mathbf{h}_{\boldsymbol{v}}^{p2e+}; \mathbf{h}_{\boldsymbol{v}}^{p2e-}] \tag{5}$$

where $\mathbf{W}_g \in \mathbb{R}^{4\times 6d}$ is a transition matrix used to calculate the weights of the edge types, and $\mathbf{w}_g = [w_{e2e}, w_{p2p}, w_{e2p}, w_{p2e}] \in \mathbb{R}^{4\times 1}$ denotes the learned weights of the four basic transition types. Finally, we can obtain the node $\boldsymbol{v}$'s global graph representation $\mathbf{h}_{\boldsymbol{v}}^g \in \mathbb{R}^{d\times 1}$ in $\mathcal{G}_g$:

$$\mathbf{h}_v^g \quad = \mathbf{v}_{\boldsymbol{v}} + w_{e2e}\mathbf{h}_{\boldsymbol{v}}^{e2e} + w_{p2p}\mathbf{h}_{\boldsymbol{v}}^{p2p} + w_{e2p}\mathbf{h}_{\boldsymbol{v}}^{e2p} + w_{p2e}\mathbf{h}_{\boldsymbol{v}}^{p2e} \tag{6}$$

where $\mathbf{v}_{\boldsymbol{v}} \in \mathbb{R}^{d\times 1}$ denotes the feature vector of the item w.r.t. the node $\boldsymbol{v}$.

4.3 The Personalized Graph Layer

The structure information of one sequence is normally not as significant as that in $\mathcal{G}_g$ since the interaction data of one user is only a tiny part of the whole data. We thus fuse the behavior information into the nodes in $\mathcal{G}_p^u$ instead of into the edges in $\mathcal{G}_g$ to differentiate the varied importance of multiple behaviors. After that, we further learn the local context information by considering the node linked by inward and outward edges to learn the intentions of a user's

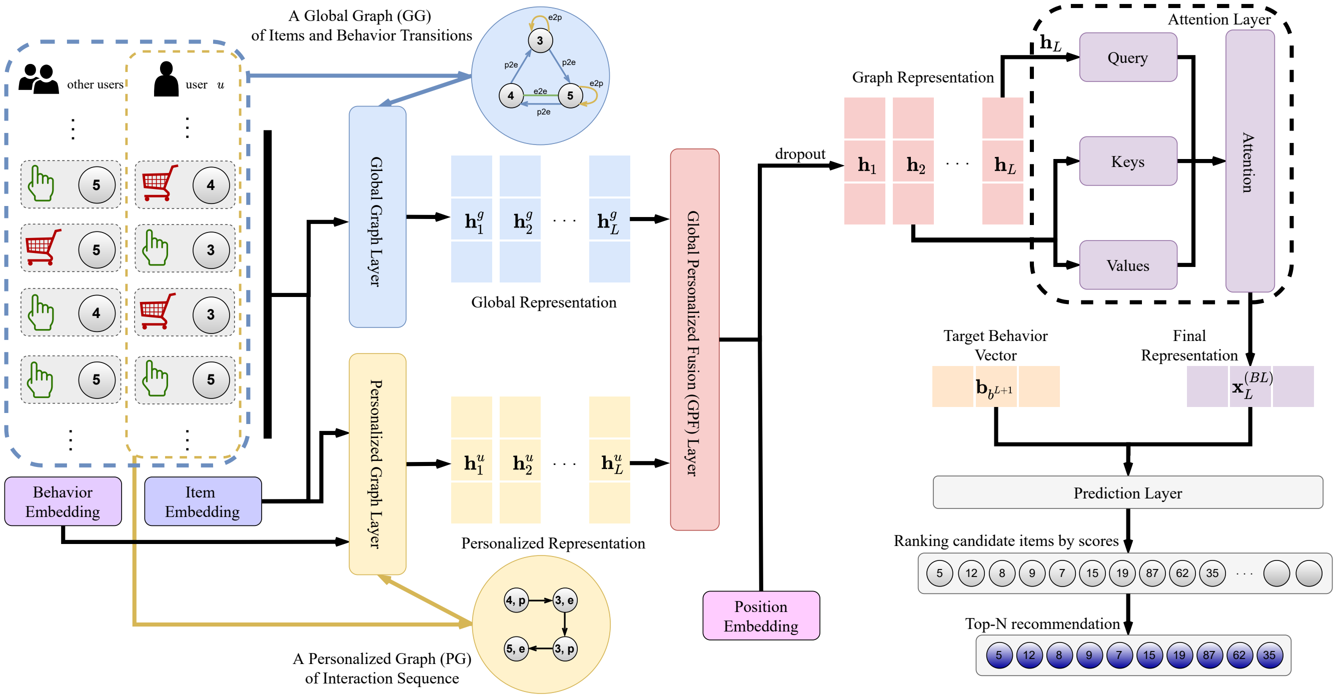

Figure 2: Illustration of our Global and Personalized Graphs for Heterogeneous Sequential Recommendation (GPG4HSR).

interaction. Different from the global graph layer, we take the summation of the item and behavior feature vectors, i.e., $(\mathbf{v}_{\boldsymbol{v}} + \mathbf{b}_{\boldsymbol{v}})$, as the representation of the node v, and feed it into a personalized graph layer. The neighbour groups of the node $\boldsymbol{v}$ classified by the local edge type in $\mathcal{E}_p^u$, i.e., classified by identifying whether it is an inward (+) or an outward (-) edge, are also fed simultaneously. For each node $\boldsymbol{v}$, we have its neighbour groups $\mathcal{A}_+^u(\boldsymbol{v})$ and $\mathcal{A}_-^u(\boldsymbol{v})$ as below:

$$\begin{aligned} \mathcal{A}_+^u(\boldsymbol{v}) &= \{(\boldsymbol{v\prime}, f) \mid (\boldsymbol{v}, \boldsymbol{v\prime}, f, +) \in \mathcal{E}_p^u\} \\ \mathcal{A}_-^u(\boldsymbol{v}) &= \{(\boldsymbol{v\prime}, f) \mid (\boldsymbol{v}, \boldsymbol{v\prime}, f, -) \in \mathcal{E}_p^u\} \end{aligned} \tag{7}$$

Then we can further obtain the inward update vector $\mathbf{h}_{\boldsymbol{v}}^+ \in \mathbb{R}^{d\times1}$ and the outward update vector $\mathbf{h}_{\boldsymbol{v}}^- \in \mathbb{R}^{d\times1}$:

$$\begin{aligned} \mathbf{h}_{\boldsymbol{v}}^+ &= \frac{\sum_{(\boldsymbol{v\prime}, f)\in\mathcal{A}_+^u(\boldsymbol{v})} f(\mathbf{v}_{\boldsymbol{v\prime}} + \mathbf{b}_{\boldsymbol{v\prime}})}{\sum_{(v', f)\in\mathcal{A}_+^u(\boldsymbol{v})} f} \\ \mathbf{h}_{\boldsymbol{v}}^- &= \frac{\sum_{(\boldsymbol{v\prime}, f)\in\mathcal{A}_-^u(\boldsymbol{v})} f(\mathbf{v}_{\boldsymbol{v\prime}} + \mathbf{b}_{\boldsymbol{v\prime}})}{\sum_{(v', f)\in\mathcal{A}_-^u(\boldsymbol{v})} f} \end{aligned} \tag{8}$$

where $\mathbf{v}_{\boldsymbol{v\prime}}$ and $\mathbf{b}_{\boldsymbol{v\prime}}$ denote the item feature vector and the behavior feature vector w.r.t. the node $\boldsymbol{v\prime}$, respectively. With these representations, the output of the personalized graph layer w.r.t. the node $\boldsymbol{v}$ can then be calculated as below:

$$\mathbf{h}_{\boldsymbol{v}}^u = \mathbf{v}_{\boldsymbol{v}} + \mathbf{b}_{\boldsymbol{v}} + \mathbf{h}_{\boldsymbol{v}}^+ + \mathbf{h}_{\boldsymbol{v}}^- \tag{9}$$

where $\mathbf{v}_{\boldsymbol{v}}$ and $\mathbf{b}_{\boldsymbol{v}}$ again denote the item embedding and the behavior embedding of the node $\boldsymbol{v}$, respectively.

4.4 Global Personalized Fusion Layer

To integrate the outputs of the global and personalized layers, i.e., $\mathbf{h}_{\boldsymbol{v}}^g$ in Eq.(6) and $\mathbf{h}_{\boldsymbol{v}}^u$ in Eq.(9), and learn the final graph representation $\mathbf{h}_v$, we apply this global personalized fusion layer $GPF(\cdot,\cdot)$ to assign weights to the global and personalized representations $\mathbf{h}_v^g$ and $\mathbf{h}_v^u$. We form $GPF(\cdot,\cdot)$ as follows:

$$\gamma_u = GPF(\mathbf{h}_{\boldsymbol{v}}^g, \mathbf{h}_{\boldsymbol{v}}^u) = \sigma(\mathbf{W}_{gp}[\mathbf{h}_{\boldsymbol{v}}^g; \mathbf{h}_{\boldsymbol{v}}^u]) \tag{10}$$

where $\mathbf{W}_{gp} \in \mathbb{R}^{1\times2d}$ is a transition vector to calculate the weight of the global graph representation, i.e., $\gamma_u \in \mathbb{R}$, and σ is the sigmoid function $\sigma(\xi) = 1/(1 + e^{-\xi})$. Based on the learned gating factor, we further linearly combine the outputs of the global and personalized layers:

$$\mathbf{h}_v = \gamma_u \mathbf{h}_{\boldsymbol{v}}^g + (1 - \gamma_u)\mathbf{h}_{\boldsymbol{v}}^u \tag{11}$$

where $\mathbf{h}_{\boldsymbol{v}}$ is the final graph representation w.r.t. the node $\boldsymbol{v}$, i.e., item i^t in $\mathcal{G}_g$ and interaction pair (i^t, b^t) in $\mathcal{G}_p^u$.

By projecting each interaction pair (i^t, b^t) in a sequence S_u into the graph layer, we can obtain the final graph representation of the sequence S_u, i.e., $\mathbf{H}_u = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_L\}$.

4.5 The Attention Layer

At the beginning of this attention layer, a dropout layer is placed after the global personalized fusion layer. The dropout technique is typically useful for successful recommendation, and its ratio is especially highly dependent on a specific dataset which will also be investigated in Section 6.5.

To learn the inner sequential patterns, we apply stacking self-attention blocks as the attention layer to weigh and summarize each interaction in a sequence following [10]. Specifically, we first inject a learnable position embedding $\mathbf{p}_t \in \mathbb{R}^{1\times d}$ into the graph representation $\mathbf{h}_t$ for each interaction (i^t, b^t) in the sequence S_u to capture the influence of the relative position t and obtain the input matrix $\mathbf{X}^{(0)} = [\mathbf{x}_1; \mathbf{x}_2; \ldots; \mathbf{x}_L] \in \mathbb{R}^{L\times d}$ of the first block:

$$\mathbf{X}^{(0)} = \left[\ \mathbf{h}_1 + \mathbf{p}_1; \mathbf{h}_2 + \mathbf{p}_2; \ldots; \mathbf{h}_L + \mathbf{p}_L\ \right] \tag{12}$$

To distinguish the different importance of each input, we then define the self-attention layer $SA(\cdot)$ and the pointwise feed-forward network $FFN(\cdot)$ as below [10]:

$$SA(\mathbf{X}) = \text{softmax}(\frac{QK^T}{\sqrt{d}})V \tag{13}$$

$$FFN(\mathbf{X}) = \text{ReLU}(\mathbf{X}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 \tag{14}$$

where $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{d\times d}$ and $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^{d\times 1}$ are the weight matrices and bias vectors. Same as [22] and [10], we define the self-attention block by combining $SA(\cdot)$ and $FFN(\cdot)$, and stack the self-attention blocks to capture deeper relationship:

$$SAB(\mathbf{X}) = FFN(SA(\mathbf{X})) \tag{15}$$

$$\mathbf{X}^{(bl)} = SAB^{(bl)}(\mathbf{X}^{(bl-1)}), \forall bl \in \{1, 2, \ldots, BL\} \tag{16}$$

where bl denotes the current block and BL is the number of stacking blocks. We take the last output vector $\mathbf{x}_L^{(BL)} \in \mathbb{R}^{d\times 1}$ as the final representation of the interaction sequence.

4.6 The Prediction Layer

To learn how likely each candidate item would be the next interacted item under a specific behavior, we apply a prediction layer and calculate the predicted probabilities of the candidate items as below:

$$\mathbf{y} = \text{softmax}(\mathbf{W}_o[\mathbf{x}_L^{(BL)}; \mathbf{b}_{b^{L+1}}] + \mathbf{bias}_o) \tag{17}$$

where $\mathbf{W}_o \in \mathbb{R}^{m\times 2d}$ and $\mathbf{bias}_o \in \mathbb{R}^{m\times 1}$ are the weight and bias for the output, and $\mathbf{y} = [y_1, y_2, \ldots, y_m] \in \mathbb{R}^{m\times 1}$ represents the predicted scores of all the candidate items. We then adopt the cross-entropy loss function for model training:

$$\mathcal{L} = -\sum_{i=1}^{m} Y_i \log(y_i) + (1 - Y_i)\log(1 - y_i) \tag{18}$$

where $Y_i = 1$ if i is the next item and $Y_i = 0$ otherwise, and y_i is the ranking score for item i at $(L + 1)$-th position.

5 EXPERIMENTAL SETTING

5.1 Datasets

We conduct experiments on four public datasets including MovieLens 1M (ML1M), Rec15, User Behavior (UB) and Tmall, and define the examinations and purchases based on their contained behaviors legitimately following [4].

ML1M[1]. MovieLens 1M (ML1M) is a popular dataset collected and provided by GroupLens. It contains about 1 million ratings of (user, movie) pairs where the rating values are from 1 to 5 w.r.t. 6040 users and 3952 movies. A rating value equals to 5 is regarded as a purchase behavior and the rest as an examination behavior to simulate heterogeneous feedback.

Rec15[2]. Rec15 is a collection of anonymous sessions in a short period like minutes or hours and contains 36,917 users, 52,739 items, and 34,151,697 interactions with 33,000,944 examinations and 1,150,753 purchases.

UB[3]. User Behavior (UB) is a dataset of user behaviors collected from Taobao and offered by Alibaba. There are 987,994 users performing about 1 million interactions over 4,162,024 items from November 25 to December 3, 2017.

Tmall[4]. Tmall contains 48,550,713 examinations and 3,292,144 purchases with shopping logs on the "Double 11" day (i.e., November 11) and in the past six months before. To make the comparison fairer, we remove the data on the "Double 11" day to avoid the various promotion activities' influence.

We split each dataset into three parts, i.e., the last interactions in sequences as the test data, the penultimate ones as the validation data and the rest as the training data [4]. The statistics of the preprocessed datasets are shown in Table 1.

Table 1: Statistics of the preprocessed data used in the experiments.

Dataset	#Users	#Items	#Examinations	#Purchases	Density
ML1M	5,645	2,357	628,892	223,305	6.41%
Rec15	36,917	9,621	446,442	233,263	0.45%
UB	20,858	30,793	470,731	136,250	0.09%
Tmall	17,209	16,176	831,117	240,901	0.38%

5.2 Evaluation Metrics

We evaluate the top-N recommendation performance via two common metrics, i.e., hit ratio (HR@N) and normalized discounted cumulative gain (NDCG@N), where N is the number of recommended items, HR@N represents the ratio of the number of top-N accurately predicted items, and NDCG@N further considers the ranking orders. We set N=10 in the experiments, and recommend items from all the uninteracted ones (instead of some randomly sampled ones) of each user to avoid the situation that the metrics collapse to AUC [2].

5.3 Baselines

To demonstrate the superiority of our GPG4HSR, we choose the following state-of-the-art methods.

Fossil [6]. A well-known method combining FISM [9] and FMPC [19], where the former is designed to learn the global preferences and the latter intends to capture the dynamic interests. It also balances these two components by some global and personalized factors.

TransRec++ [31]. A novel MF-based model for HSR as an extended version of TransRec [5]. It applies some behavior transition vectors in an item transition space to learn the dynamics of users' heterogeneous behaviors.

GRU4Rec [8]. An RNN-based model which first introduces RNN from NLP to sequential recommendation. It applies GRU on items

[1] https://grouplens.org/datasets/movielens/1m

[2] https://recsys.acm.org/recsys15/challenge

[3] https://tianchi.aliyun.com/dataset/dataDetail?dataId=649

[4] https://tianchi.aliyun.com/dataset/dataDetail?dataId=42

in sequences to recurrently learn a user's preferences at each time step.

Caser [21]. A CNN-based model which applies some horizontal and vertical convolution filters to a sequence of recent item embeddings to capture a user's short-term preferences.

SASRec [10]. An efficient sequential model with an advanced self-attention mechanism. It stacks self-attention networks to capture deep relationship of the input and predicts the next interacted item. We use its core component, i.e., the hierarchical self-attention network, as our attention layer to learn the sequential patterns. Following [10], we predict the target item without using the target behavior, which is thus different from that of the proposed GPG4HSR.

RIB [32]. An RNN-based model for HSR which incorporates the behavior embedding and the item embedding learned by a modified version of item2vec. It then adopts an RNN layer to model the sequence and an attention layer to distinguish the interactions' varied effects.

SR-GNN [27]. A GNN-based sequential method which models sequences as a graph structured data to capture the items' transitions. It also learns the long-term and short-term preferences of the users using an attention network.

MGNN-SPred [23]. A GNN-based model for HSR which constructs a multi-relational item graph (MRIG), splits a sequence w.r.t. different types of behaviors, and then learns the correlations of the items with a same behavior. It balances the effects of different behaviors via an efficient gating network, which is also adopted by our model to leverage the global and user-specific graph sequence representations.

5.4 Implementation Details

Following [10], we set the sequence length L to 50, the hidden size d to 64, the batch size to 128, and the learning rate to 0.001 for all methods. We construct a training set by a sliding window, where one single training sample contains a sequence of (item, behavior) pairs and a target behavior used to predict the target item. We also pad the items at the left if the training sequence is shorter than 50 [13]. In particular, we set the sequence length L to 10 for MGNN-SPred on ML1M since it performs poorly using the default setting. The dropout ratio is set to 0.2 for ML1M and 0.5 for the other datasets [10]. We use a single-head attention layer and set the number of blocks to 2 [13]. We set the recurrent unit size to 64 for GRU4Rec and RIB to align with the hidden size. The numbers of the vertical and horizontal filters are 4 and 16, respectively, and the height of the horizontal filters is chosen from $\{2, 3, 4\}$ for Caser following [13]. For the other hyperparameters, we reference the original papers or tune them on the validation data of each dataset. The related code and scripts of GPG4HSR[5] used in our experiments are made publicly available for reproducibility.

6 RESULTS

6.1 Main results

We report the main results in Table 2. We re-implemented most methods in [4] and achieved very similar results, and for consistency, we used the results of MF-based methods (i.e., Fossil and TransRec++) from [31] and the results of some DL-based methods (i.e., GRU4Rec, Caser, SASRec, RIB and SR-GNN) from [4] for direct comparison. The best result in each column of each dataset is marked in bold, and the second-best one is marked with an underline.

We have the following observations from Table 2: i) Our GPG4HSR outperforms two MF-based methods, four non-GNN DL-based methods and one GNN-based method (i.e., MGNN-SPred) on all the four datasets. Besides, our GPG4HSR beats the other GNN-based method (i.e., SR-GNN) on three of the four datasets (i.e., ML1M, UB and Tmall), and achieves the second-best performance on HR@10 on Rec15 in which case its indicator score is very close to the best. The reason why GPG4HSR performs slightly worse than SR-GNN may be because of insufficient correlations between different behaviors as there are more continuous interactions with a same behavior type in Rec15 than in other datasets.[6] Hence, the results clearly demonstrate the superior performance of our GPG4HSR. ii) Compared with SASRec, our GPG4HSR surpasses it on all the four datasets, which means that the input embedding learned by our graph layers captures more correlations of the behaviors and is capable of distinguishing the user intentions of a same behavior with different context information. iii) SASRec achieves the second best on average, which demonstrates the effectiveness and superiority of the attention mechanism for sequential recommendation. Such a result indicates that the current GNN-based methods are more suitable for anonymous and short sessions but may not be very competitive for sequential recommendation with multiple behaviors probably because they ignore the correlations of different behaviors. iv) The results of the DL-based methods exceed the MF-based methods in almost all cases, which shows the advantage of modeling sequences using deep neural networks. v) TransRec++ performs better than Fossil on three of the four datasets, which indicates the potential information of the correlations (i.e., transitions) of different behaviors is useful for the next likely-to-purchase item prediction. vi) For the DL-based methods, the two GNN-based methods only defeat the non-GNN methods on Rec15, which means that the existing GNN-based models learn the sequential patterns insufficiently and their performances are better when the average length of sequences is shorter. To be specific, the average length of Rec15 is 18.41, which is much shorter than 150.96 of ML1M, 29.10 of UB and 62.29 of Tmall.

6.2 Exploration with More Types of Behaviors

To explore the capability of our GPG4HSR dealing with more types of behaviors, we conduct an exploration study and report the results in Figure 3. Note that the label on the horizontal axis with suffix '_click' means that we only take the clicking behaviors as the examinations, and the label with suffix '_mul' means that we

[5] https://csse.szu.edu.cn/staff/panwk/publications/GPG4HSR

[6] 91.6% sequences in Rec15 exist a continuous sub-sequence of clicks (i.e., examinations) which takes 60% length of its parent sequence.

Table 2: Recommendation performance of two MF-based methods, four non-GNN DL-based methods, two GNN-based methods and our GPG4HSR on four datasets. The results except MGNN-SPred and our GPG4HSR are copied from [31] and [4] for direct comparison.

Methods	ML1M		Rec15		UB		Tmall	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
Fossil	0.1114	0.0528	0.3736	0.2029	0.0508	0.0270	0.0437	0.0241
TransRec++	0.1088	0.0508	0.4064	0.2209	0.0661	0.0413	0.0593	0.0377
GRU4Rec	0.1249	0.0600	0.3588	0.1921	0.0571	0.0320	0.0744	0.0431
Caser	0.1215	0.0608	0.3985	0.2042	0.0562	0.0289	0.0390	0.0207
SASRec	0.1350	0.0651	0.3615	0.1889	0.0744	0.0412	0.0862	0.0521
RIB	0.1302	0.0646	0.3668	0.1921	0.0660	0.0370	0.0776	0.0454
SR-GNN	0.1267	0.0614	**0.4211**	**0.2291**	0.0575	0.0307	0.0707	0.0419
MGNN-SPred	0.1134	0.0548	0.4164	0.2108	0.0727	0.0386	0.0449	0.0236
GPG4HSR	**0.1460**	**0.0737**	0.4198	0.2160	**0.0830**	**0.0462**	**0.0944**	**0.0548**

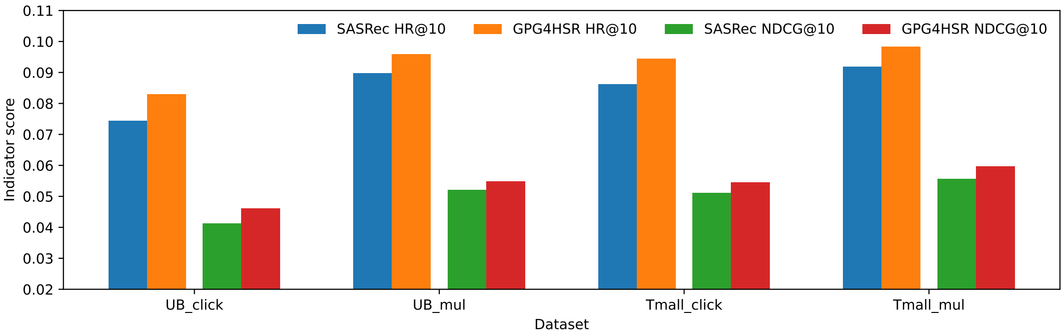

Figure 3: Recommendation performance in scenes with two (marked with '_click') and more than two (marked with '_mul') types of behaviors on UB and Tmall. Note that there are only two types of behaviors in Rec15, which is thus not included in this study.

take all non-purchase positive feedback, i.e., clicks, adds-to-cart and favorites as the examinations. We also show the results of SASRec because it performs the second best in most cases as shown in Table 2 and is also an important module of our GPG4HSR. From Figure 3 we can see that SASRec and our GPG4HSR both perform better when using more types of behaviors and our GPG4HSR beats SASRec in both scenes showing the general applicability of our GPG4HSR.

Table 3: Recommendation performance (HR@10) of our GPG4HSR with different architectures for ablation study.

Architectures	ML1M	Rec15	UB	Tmall
w/o GPF	0.1454	0.4122	**0.0840**	0.0901
w/o GPF, GG	0.1456	0.4104	0.0815	0.0899
w/o GPF, PG	0.1422	0.3785	0.0769	0.0870
w/o GPF, GG, PG	0.1350	0.3615	0.0744	0.0862
GPG4HSR	**0.1460**	**0.4198**	0.0830	**0.0944**

6.3 Ablation Study

In order to figure out the effectiveness of different components of our GPG4HSR, we conduct an ablation study and report the results in Table 3. Note that we only report the results on HR@10 because of the limited space and its consistent tendency with NDCG@10. We study the performance of our GPG4HSR without the global personalized fusion layer (i.e., $\mathbf{h}_v = \mathbf{h}_{\boldsymbol{v}}^g + \mathbf{h}_{\boldsymbol{v}}^u$, denoted as 'w/o GPF'), without the global graph GG (i.e., $\mathbf{h}_{\boldsymbol{v}} = \mathbf{h}_{\boldsymbol{v}}^u$, denoted as 'w/o GPF, GG'), without the personalized graph PG (i.e., $\mathbf{h}_{\boldsymbol{v}} = \mathbf{h}_{\boldsymbol{v}}^g$, denoted as 'w/o GPF, PG'), and without GG, PG and target behavior information in the prediction layer, (i.e., our GPG4HSR reduces to SASRec, denoted as 'w/o GPF, GG, PG'). The best and the second-best results are marked in bold and underlined, respectively. We could have the following observations from Table 3:

- **GPG4HSR vs. others.** Our GPG4HSR with all the components achieves the best performance on average, which shows that each component in GPG4HSR contributes to learning the graph representation because the performance drops gradually as more components are removed in most cases.

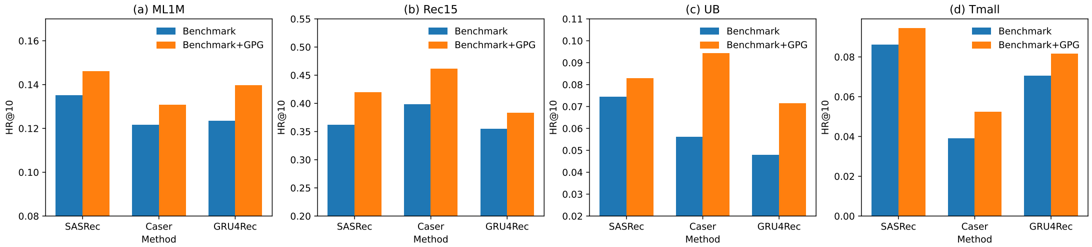

Figure 4: Recommendation performance (HR@10) of each sequential recommendation method (i.e., SASRec, Caser or GRU4Rec) and that of adding our GPG as a generic behavior-aware framework to it, denoted as "Benchmark" and "Benchmark+GPG", respectively.

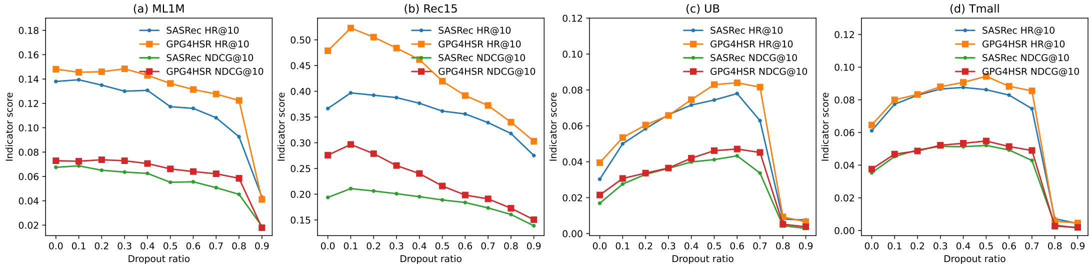

Figure 5: Recommendation performance of SASRec and our GPG4HSR with different values of dropout ratios.

- **GPG4HSR vs. 'w/o GPF'.** The results show that our GPG4HSR with complete structure outperforms GPG4HSR without the fusion layer on ML1M, Rec15 and Tmall, and there is a very small gap between them on Rec15. These results show that the fusion component is effective in balancing the two different graph representations.
- **'w/o GPF, GG', 'w/o GPF, PG' vs. 'w/o GPF, GG, PG' .** The former two outperform the last one on all the four datasets, demonstrating the effectiveness of PG and GG as each single component for improving the recommendation performance. And the first one (i.e., without the global graph GG) slightly exceeds the second one (i.e., without the personalized graph PG), indicating the distinct user intentions of the behaviors with the local context information which offers more learnable information than their correlations in GG.

According to Table 3 and the corresponding observations, we can see that our GPG4HSR again achieves the best on average, which indicates that each component is useful. In particular, the personalized graph PG captures the distinct user intentions of different behaviors and the global graph GG learns the correlations of behaviors from their transitions, and the fusion layer combines these two complementary components.

6.4 Framework Study

As described in Section 4.5, we adopt the advanced attention technique to learn the sequential patterns in our GPG4HSR. In this subsection, we explore whether our global and personalized graphs (GPG) would work well with other sequential models. For this purpose, we conduct exploratory experiments with some other sequential recommendation methods.

Specifically, we replace the attention layer in our GPG4HSR shown in Figure 2 with the convolution component in Caser [21] and GRU in GRU4Rec [8], which are denoted as 'Caser+GPG' and 'GRU+GPG', respectively. Note that our GPG4HSR can be considered as 'SASRec+GPG' since its attention layer is designed using the same technology with that in SASRec [10]. With GPG being integrated, we investigate whether the three effective sequential models will be boosted and report the results of this framework study in Figure 4, where the blue bars represent the backbone models and the orange bars represent the ones with GPG as the behavior-aware framework. Similarly, due to space limitation and similar observations on two metrics, we also report the results on HR@10 only.

Through the results in Figure 4, we witness that each sequential model with GPG as a behavior-aware framework outperforms the corresponding original model on all the four datasets, which demonstrates the ability of our GPG to capture the rich behavior information including their transitions and their representations learned with the varied influences of adjacent nodes.

6.5 Impact of the Dropout Ratio

We adjust the dropout ratio $\in \{0.0, 0.1, 0.2, \ldots, 0.9\}$ for SASRec and our GPG4HSR on four datasets and show the performance in Figure 5. We use HR@10 and NDCG@10 as the labels on the vertical axis for consistency with the main results. We can see that there is a similar tendency of the results with the change of the values of the dropout ratio for both methods, and our GPG4HSR performs better than SASRec in almost all cases. This again indicates the effectiveness of our GBG. Moreover, the dropout ratios to achieve the best performance on different datasets are different and there is a sharp drop at a turning point with a large value of the dropout ratio, e.g., 0.8 to 0.9 on ML1M, and 0.7 to 0.8 on UB and Tmall. The results in Figure 5 also provide us some guidance on configuring the dropout ratio on a certain dataset, i.e., a low value is more suitable for datasets with low sparsity, and a medium value is better for datasets with high sparsity.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we address two limitations of existing works for heterogeneous sequential recommendation (HSR), i.e., modeling different behaviors separately, and learning the user interests beneath behaviors without considering the local context information. To address these two issues, we propose a novel solution called GPG4HSR, which captures the transitions of behaviors via a global graph (GG), and learns the user intentions of different feedback with adjacent interactions via a personalized graph (PG). Extensive empirical studies on four public datasets demonstrate that our GPG4HSR achieves the state-of-the-art performance compared with several competitive baselines. Ablation studies also show that the design of PG, GG and GPF in our GPG4HSR is reasonable.

For future works, we will take the users' privacy [12, 14] into consideration and design some GNN- and federated learning-based methods for the studied problem. Moreover, we are interested in exploiting some textual information such as the users' reviews and the items' attributes to further improve our model.

ACKNOWLEDGMENTS

We thank the support of National Natural Science Foundation of China Nos. 62172283 and 61836005.

REFERENCES

[1] Chong Chen, Weizhi Ma, Min Zhang, Zhaowei Wang, Xiuqiang He, Chenyang Wang, Yiqun Liu, and Shaoping Ma. 2021. Graph Heterogeneous Multi-Relational Recommendation. In *AAAI'21*. 3958–3966.

[2] Alexander Dallmann, Daniel Zoller, and Andreas Hotho. 2021. A Case Study on Sampling Strategies for Evaluating Neural Sequential Item Recommendation Models. In *RecSys'21*. 505–514.

[3] William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *NeurIPS'17*. 1024–1034.

[4] Mingkai He, Weike Pan, and Zhong Ming. 2022. BAR: Behavior-Aware Recommendation for Sequential Heterogeneous One-Class Collaborative Filtering. *Information Sciences* 608 (2022), 881–899.

[5] Ruining He, Wang-Cheng Kang, and Julian McAuley. 2017. Translation-based recommendation. In *RecSys'17*. 161–169.

[6] Ruining He and Julian McAuley. 2016. Fusing similarity models with Markov chains for sparse sequential recommendation. In *ICDM'16*. 191–200.

[7] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *SIGIR'20*. 639–648.

[8] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *ICLR'16*.

[9] Santosh Kabbur, Xia Ning, and George Karypis. 2013. FISM: Factored Item Similarity Models for top-N Recommender Systems. In *KDD'13*. 659–667.

[10] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *ICDM'18*. 197–206.

[11] Zhi Li, Hongke Zhao, Qi Liu, Zhenya Huang, Tao Mei, and Enhong Chen. 2018. Learning from History and Present: Next-Item Recommendation via Discriminatively Exploiting User Behaviors. In *KDD'18*. 1734–1743.

[12] Feng Liang, Enyue Yang, Weike Pan, Qiang Yang, and Zhong Ming. 2022. Survey of Recommender Systems Based on Federated Learning (in Chinese). *SCIENTIA SINICA Informationis* 52(5) (2022), 713–741.

[13] Jing Lin, Weike Pan, and Zhong Ming. 2020. FISSA: Fusing Item Similarity Models with Self-Attention Networks for Sequential Recommendation. In *RecSys'20*. 130–139.

[14] Zhaohao Lin, Weike Pan, Qiang Yang, and Zhong Ming. 2022. Recommendation Framework via Fake Marks and Secret Sharing. *ACM Transactions on Information Systems* (2022).

[15] Kelong Mao, Jieming Zhu, Xi Xiao, Biao Lu, Zhaowei Wang, and Xiuqiang He. 2017. UltraGCN: Ultra Simplification of Graph Convolutional Networks for Recommendation. In *CIKM'21*. 1253–1262.

[16] Wenjing Meng, Deqing Yang, and Yanghua Xiao. 2020. Incorporating User Micro-behaviors and Item Knowledge into Multi-task Learning for Session-based Recommendation. In *SIGIR'20*. 1091–1100.

[17] Wenhao Pan and Kai Yang. 2021. Multi-behavior Graph Neural Networks for Session-based Recommendation *(MLDBBI'21)*. 756–761.

[18] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI'09*. 452–461.

[19] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized Markov chains for next-basket recommendation. In *WWW'10*. 811–820.

[20] Qi Shen, Lingfei Wu, Yitong Pang, Yiming Zhang, Zhihua Wei, Fangli Xu, and Bo Long. 2021. Multi-behavior Graph Contextual Aware Network for Session-based Recommendation. *CoRR* abs/2109.11903 (2021).

[21] Jiaxi Tang and Ke Wang. 2018. Personalized top-N Sequential Recommendation via Convolutional Sequence Embedding. In *WSDM'18*. 565–573.

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *NeurIPS'17*. 6000–6010.

[23] Wen Wang, Wei Zhang, Shukai Liu, Qi Liu, Bo Zhang, Leyu Lin, and Hongyuan Zha. 2020. Beyond Clicks: Modeling Multi-Relational Item Graph for Session-Based Target Behavior Prediction. In *WWW'20*. 3056–3062.

[24] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *SIGIR'19*. 165–174.

[25] Ziyang Wang, Wei Wei, Gao Cong, Xiao-Li Li, Xianling Mao, and Minghui Qiu. 2020. Global Context Enhanced Graph Neural Networks for Session-based Recommendation. In *SIGIR'20*. 169–178.

[26] Wei Wei, Chao Huang, Lianghao Xia, Yong Xu, Jiashu Zhao, and Dawei Yin. 2022. Contrastive Meta Learning with Behavior Multiplicity for Recommendation. In *WSDM'22*. 1120–1128.

[27] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based Recommendation with Graph Neural Networks. In *AAAI'19*. 346–353.

[28] Lianghao Xia, Yong Xu, Chao Huang, Peng Dai, and Liefeng Bo. 2021. Graph Meta Network for Multi-Behavior Recommendation. In *SIGIR'21*. 757–766.

[29] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. 2019. Graph Contextualized Self-Attention Network for Session-based Recommendation. In *IJCAI'19*. 3940–3946.

[30] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. 2018. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In *KDD'18*. 974–983.

[31] Zhuoxin Zhan, Mingkai He, Weike Pan, and Zhong Ming. 16(2):162615, 2022. TransRec++: Translation-based Sequential Recommendation with Heterogeneous Feedback. *FCS* (16(2):162615, 2022). https://doi.org/fcs/EN/10.1007/s11704-022-1184-8

[32] Meizi Zhou, Zhouye Ding, Jiliang Tang, and Dawei Yin. 2018. Micro Behaviors: A New Perspective in E-commerce Recommender Systems. In *WSDM'18*. 727–735.