Multi-Behavior Enhanced Recommendation with Cross-Interaction Collaborative Relation Modeling

Lianghao Xia[1], Chao Huang[2], Yong Xu[1*], Peng Dai[2], Mengyin Lu[2], Liefeng Bo[2]

South China University of Technology[1], JD Finance America Corporation[2]

cslianghao.xia@mail.scut.edu.cn, yxu@scut.edu.cn, chaohuang75@gmail.com, {peng.dai,mengyin.lu,liefeng.bo}@jd.com

Abstract—**Many previous studies aim to augment collaborative filtering with deep neural network techniques, so as to achieve better recommendation performance. However, most existing deep learning-based recommender systems are designed for modeling singular type of user-item interaction behavior, which can hardly distill the heterogeneous relations between user and item. In practical recommendation scenarios, there exist multi-typed user behaviors, such as browse and purchase. Due to the overlook of user's multi-behavioral patterns over different items, existing recommendation methods are insufficient to capture heterogeneous collaborative signals from user multi-behavior data. Inspired by the strength of graph neural networks for structured data modeling, this work proposes a G̲raph N̲eural M̲ulti-Behavior Enhanced R̲ecommendation (GNMR) framework which explicitly models the dependencies between different types of user-item interactions under a graph-based message passing architecture. GNMR devises a relation aggregation network to model interaction heterogeneity, and recursively performs embedding propagation between neighboring nodes over the user-item interaction graph. Experiments on real-world recommendation datasets show that our GNMR consistently outperforms state-of-the-art methods. The source code is available at https://github.com/akaxlh/GNMR.**

Index Terms—**Recommender Systems, Multi-Behavior Recommendation, Graph Neural Networks**

I. INTRODUCTION

Recommender systems become the essential part of online platforms, to alleviate the information overload problem and make recommendation for users [7], [10]. The key objective of recommendation frameworks is to accurately capture user's preference over different items based on their observed interactions [11], [17]. As effective feature learning paradigms, deep learning has attracted a lot of attention in recommendation, which results in various neural network-based methods being proposed for user-item interaction modeling [1], [5], [12], [21]. These methods transform users and items to vectorized representations based on different neural network structures. For example, autoencoder has been used for latent representation projection in collaborative filtering [14]. To endow the collaborative filtering architecture with the capability of non-linear feature interaction modeling, NCF [5] integrates the matrix factorization and multi-layer perceptron network.

However, these recommendation solutions mostly focus on singular type of user-item interaction behavior. In real-world online applications, users' behaviors are multi-typed in nature,

which involves heterogeneous relations (*e.g.*, browse, rating, purchase) between user and item [3], [16]. Each type of user behavior may exhibit different semantic information for characterizing diversified interactive patterns between user and item. Hence, the current encoding functions of user-item interactions are insufficient to comprehensively learn the complex user's preference. While having realized the importance of leveraging different types of user behaviors, encoding multi-typed behavioral patterns present unique challenges which cannot be easily handled by recommendation methods designed for single type of user-item interactions. In particular, it is non-trivial to effectively capture implicit relationships among various types of user behaviors. Such different types of interaction behaviors may be correlated in complex ways to provide complementary information for learning user interests. Additionally, although there exist some recent developed multi-behavior user modeling techniques for recommendation [2], [16], they fail to capture the high-order collaborative effects with the awareness of different user-item relations. Taking the inspiration from the effectiveness by employing graph neural networks in recommendation [11], [13], it is beneficial to consider high-order relations between user-item interaction into the embedding space.

Contribution. In this work, we propose a G̲raph N̲eural M̲ulti-Behavior Enhanced R̲ecommendation framework (short for GNMR), to capture users' preference on different items via a multi-behavior modeling architecture. Specifically, the designed graph neural multi-behavior learning framework explores high-order user-item interaction subgraph, characterizing complex relations between different types of user behaviors in an automatic manner. In our graph neural network, we design a relation dependency encoder to capture the implicit dependencies among different types of user behaviors under a message passing architecture. With the aim of modeling the graph-structured interactions between users and items, our developed GNMR performs the embedding propagation over the multi-behavior interaction graph in a recursive way, with the injection of type-specific behavior relationships. We evaluate our framework on real-world datasets from MovieLens, Yelp and Taobao. Evaluation results show the effectiveness of our GNMR model compared with state-of-the-art baselines.

The main contributions of this work are summarized as:

- This work focuses on capturing behavior type-aware collaborative signals with the awareness of high-order relations

*Corresponding author: Yong Xu.

over user-item interaction graph in the embedding paradigm for recommendation.

- We propose a new graph neural network framework GNMR for multi-behavior enhanced recommendation, with the exploration of dependencies between different types of behaviors under a message passing architecture. GNMR performs the embedding propagation between users and items on their graph-structured connections, and aggregates the underlying cross-type behavior relations with a high-order scenario.

- Experimental results on three real-world datasets from online platforms show the effectiveness of our GNMR model compared with state-of-the-art baselines.

II. PRELIMINARIES

In a typical recommender system, there exist two types of entities, *i.e.*, set of users U ($u_i \in U$) and items V ($v_j \in V$), where $|U| = I$ (indexed by i) and $|V| = J$ (indexed by j). In our multi-behavior enhanced recommendation scenario, users could interact with items under multiple types of interactions (*e.g.*, different ratings, browse, purchase). We define a three-dimensional tensor $\mathbf{X} \in \mathbb{R}^{I \times J \times K}$ to represent the multi-typed user-item interactions, where K denotes the number of behavior types (indexed by k). In tensor $\mathbf{X}$, each element $x_{i,j}^k = 1$ if there exist interactions between user u_i and item v_j given the k-th behavior type and $x_{i,j}^k = 0$ otherwise. In this work, we aim to improve the recommendation performance on the target behavior of users, by exploring the influences of other types of behaviors. We formally define the problem as:

Problem Statement. Given the multi-behavior interaction $\mathbf{X} \in \mathbb{R}^{I \times J \times K}$ under K behavior types, the goal is to recommended a ranked list of items in terms of probabilities that user u_i will interact with them under the target behavior type.

Connections to Existing Work. Recent years have witnessed the promising results of graph neural networks in learning dependence from graph-structured data [15]. In general, the core of graph neural networks is to aggregate feature information from adjacent vertices over a graph under a message passing architecture [19]. Inspired by the effectiveness of graph convolutional network, recent studies, such as NGCF [11] and GraphSage [4], explore user-item interaction graph to aggregate the embeddings from neighboring nodes by employing graph convolutional network in collaborative filtering. These work performs information propagation between vertices to exploit relations among users and items. This work extends the exploration of designing graph neural network models for recommendation by tackling the challenges in dealing with multi-typed user-item interactions for recommendation.

III. METHODOLOGY

This section presents the details of our proposed neural framework GNMR for multi-behavior enhanced recommendation. Based on the preliminaries in Section II, we generate the user-item interaction graph $G = \{U, V, E\}$, where nodes are constructed by user set U and item set V. E represents the edges connecting users and items based on their multi-typed
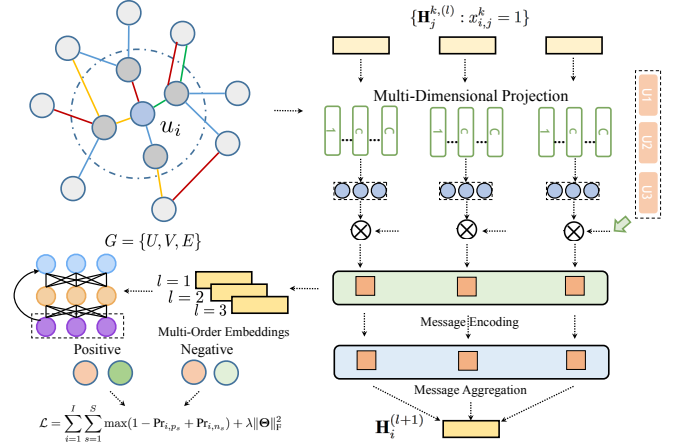

Figure 1. Model Architecture of GNMR.

interactions. Specifically, in graph G, there exists an edge $e_{i,j}^k$ between user $u_i \in U$ and item $v_j \in V$ under the k-th behavior type if $x_{i,j}^k = 1$. The general graph neural network utilizes G as the computation graph for information diffusion, during which the neighborhood messages are aggregated to obtain contextual representations [6], [20]. In our GNMR framework, we update user and item latent embeddings by propagating them on the user-item multi-behavior interaction graph G, to capture the type-aware behavior collaborative signals for recommendation. We show the model architecture of GNMR in Figure 1.

A. Type-specific Behavior Embedding Layer

We suppose the output embedding of the l-th GNMR graph layer as $\mathbf{H}^l$ which is then fed into the $(l+1)$-th layer as input representation. Given edges between users and items under the behavior type of k, we construct the passed message as below:

$$\mathbf{H}_{i \leftarrow}^{k,(l)} = \eta(\{\mathbf{H}_j^{k,(l)} : x_{i,j}^k = 1\})$$
$$\mathbf{H}_{j \leftarrow}^{k,(l)} = \eta(\{\mathbf{H}_i^{k,(l)} : x_{i,j}^k = 1\}) \quad (1)$$

where $\mathbf{H}_{i \leftarrow}^{k,(l+1)} \in \mathbb{R}^d$ and $\mathbf{H}_{j \leftarrow}^{k,(l+1)} \in \mathbb{R}^d$ denotes the embeddings passed to u_i and v_j, respectively. $\eta(\cdot)$ represents the embedding layer which preserves the unique characteristics of each type (*i.e.*, k) of user behaviors. During the message propagation process, we initialize the embeddings $\mathbf{H}_i^0$ and $\mathbf{H}_j^0$ for user u_i and item v_j by leveraging Autoencoder-based pre-training scheme [9] for generating low-dimensional representations based on multi-behavior interaction tensor $\mathbf{X}$.

In our message passing architecture, $\eta(\cdot)$ is designed to obtain representations for each behavior type of k, by considering type-specific behavior contextual signals (*e.g.*, behavior frequency). We formally represent $\eta(\cdot)$ as follows:

$$\alpha_{c,k} = \delta(\sum_{j \in N(i,k)} \mathbf{W}_1 \cdot \mathbf{H}_j^{k,(l)} + \mathbf{b}_1)(c)$$

$$\eta(\{\mathbf{H}_j^{k,(l)} : x_{i,j}^k = 1\}) = \sum_{c=1}^{C} \alpha_{c,k} \mathbf{W}_{2,c} \cdot \sum_{j \in N(i,k)} \mathbf{H}_j^{k,(l)} \quad (2)$$

1932

where $\alpha_{c,k}$ represents the learned weight of the k-th type of user behavior from the projected c-th latent dimension. $N(i,k)$ denotes the neighboring item nodes connected with user u_i under behavior type of k in the interaction graph G. $\mathbf{W}_1 \in \mathbb{R}^{C \times d}$ and $\mathbf{b}_1 \in \mathbb{R}^C$ are learn hyperparameters. We define $\delta(\cdot)$ as ReLU activation function. In our embedding process, we aggregate embeddings from different latent dimensions with weight $\alpha_{c,k}$ and transformation parameter $\mathbf{W}_{2,c} \in \mathbb{R}^{d \times d}$. The message passing procedure for the target item node v_j from its adjacent user nodes $N(j,k)$ under the behavior type of k is conducted in an analogous way in Eq 2.

B. Message Aggregation Layer

After performing the propagation of type-specific behavior embeddings between user and item, we propose to aggregate representations from different behavior types, by exploiting the underlying dependencies. In GNMR framework, our message aggregation layer is built upon the attentional neural mechanism. In particular, we first define two transformation weight matrices $\mathbf{Q}$ and $\mathbf{K}$ for embedding projection between different behavior types k and k'. The explicit relevance score between type-specific behavior embeddings is represented as $\beta_{k,k'}$, which is formally calculated as follows:

$$\beta_{k,k'}^s = [(\mathbf{Q}^s \mathbf{H}_{i\leftarrow}^{k,(l)})^\top \cdot (\mathbf{K}^s \mathbf{H}_{i\leftarrow}^{k',(l)})]/\sqrt{d/S} \qquad (3)$$

We perform the embedding projection process with multiple latent spaces ($s \in S$), to enable the behavior dependency modeling from different hidden dimensions. $\mathbf{Q}^s \in \mathbb{R}^{\frac{d}{S} \times d}$ and $\mathbf{K}^s \in \mathbb{R}^{\frac{d}{S} \times d}$ correspond to the transformation matrices of s-th projection space. We further apply the softmax function on $\beta_{k,k'}^s$. Then, we recalibrate the type-specific behavior embedding by concatenating representations from different learning subspaces with the following operation:

$$\tilde{\mathbf{H}}_{i\leftarrow}^{k,(l)} = \xi(\mathbf{H}_{i\leftarrow}^{k,(l)}) = \Big(\Big\|_{s=1}^S \sum_{k'=1}^K \beta_{k,k'}^s \mathbf{V}^s \cdot \mathbf{H}_{i\leftarrow}^{k',(l)} \Big) \oplus \mathbf{H}_{i\leftarrow}^{k,(l)}$$

where $\big\|$ represents the concatenation operation and $\mathbf{V}_s \in \mathbb{R}^{\frac{d}{S} \times d}$ denotes the transformation matrix. We define the propagated recalibrated embedding as $\tilde{\mathbf{H}}_i^{k,(l)}$. To preserve the original type-specific behavioral patterns, the element-wise addition is utilized between the original embedding $\mathbf{H}_{i\leftarrow}^{k,(l)}$ and recalibrated representation $\tilde{\mathbf{H}}_{i\leftarrow}^{k,(l)}$, *i.e.*, $\hat{\mathbf{H}}_{i\leftarrow}^{k,(l)} = \tilde{\mathbf{H}}_{i\leftarrow}^{k,(l)} \oplus \mathbf{H}_{i\leftarrow}^{k,(l)}$, where $\hat{\mathbf{H}}_{i\leftarrow}^{k,(l)}$ is the updated embedding propagated through the connection edge of k-th behavior type.

To fuse the behavior type-specific representations during the embedding propagation process, we develop our message aggregation layer with the following functions as:

$$\mathbf{H}_{i\leftarrow}^{(l)} = \psi(\{\hat{\mathbf{H}}_{i\leftarrow}^{k,(l)} : k = [1,2,...,K]\})$$
$$\mathbf{H}_{j\leftarrow}^{(l)} = \psi(\{\hat{\mathbf{H}}_{j\leftarrow}^{k,(l)} : k = [1,2,...,K]\}) \qquad (4)$$

where $\psi(\cdot)$ represents the message aggregation function. To fuse user/item embeddings from different behavior types, we aim to identify the importance score of individual behavior

type-specific representation in assisting the recommendation phase between users and items. To achieve this goal, we feed $\hat{\mathbf{H}}_{i\leftarrow}^{k,(l)}$ into a feed-forward neural network to calculate the importance weights as follows (take the user side as example):

$$\gamma_k = \mathbf{w}_2^\top \cdot \delta(\mathbf{W}_3 \hat{\mathbf{H}}_{i\leftarrow}^{k,(l)} + \mathbf{b}_2) + b_3$$
$$\hat{\gamma}_k = \frac{\exp \gamma_k}{\sum_{k'=1}^K \exp \gamma_{k'}} \qquad (5)$$

where γ_k represents the intermediate values which are fed into a softmax function to generate the importance weight $\hat{\gamma}_k$. In addition, $\delta(\cdot)$ denotes the ReLU activation function. $\mathbf{W}_1 \in \mathbb{R}^{d' \times d}$ and $\mathbf{w}_2 \in \mathbb{R}^{d'}$ represents trainable transformation matrices. $\mathbf{b}_1 \in \mathbb{R}^{d'}$ and $b_2 \in \mathbb{R}$ are bias terms. d' denotes the embedding dimension of hidden layer. After obtaining the weight $\hat{\gamma}_k$ corresponding to behavior type of k, the embedding aggregation process is performed as: $\mathbf{H}_i^{(l+1)} = \sum_{k=1}^K \hat{\gamma}_k \hat{\mathbf{H}}_{i\leftarrow}^{k,(l+1)}$ and $\mathbf{H}_j^{(l+1)} = \sum_{k=1}^K \hat{\gamma}_k \hat{\mathbf{H}}_{j\leftarrow}^{k,(l+1)}$, where $\mathbf{H}_i^{(l+1)}$ and $\mathbf{H}_j^{(l+1)}$ serve as the input user/item embedding for $(l+1)$-th graph layer.

Given the generated graph structure of user-item interactions, we learn the high-order multi-behavioral relations over $G = \{U, V, E\}$ by stacking multiple information propagation layers. The embedding propagation between the (l)-th and $(l+1)$-th graph layers can be formally represented as below:

$$\mathbf{H}_i^{(l+1)} = \psi(\xi(\eta(\{\mathbf{H}_j^{(l)} : x_{i,j}^k = 1\})))$$
$$\mathbf{H}_U^{(l+1)} = \sum_{k=1}^K \hat{\gamma}_k \cdot \text{MH-Att}(\mathbf{X}^k \sum_{c=1}^C \alpha_{c,k} \cdot \mathbf{H}_V^{k,(l)} \cdot \mathbf{W}_{2,c}) \quad (6)$$

where $\mathbf{H}_{(i)}^{(l+1)} \in \mathbb{R}^{I \times d}$ denotes the embeddings of all users for the $(l+1)$-th graph layer. Given the behavior type of k, the adjacent relations are represented as $\mathbf{X}^k \in \mathbb{R}^{I \times J}$. Similarly, embeddings of items ($v_j \in V$) can be generated based on the above propagation and aggregation operations.

C. Model Optimization of GNMR

To optimize our GNMR model and infer the hyperparameters, we perform the learning process with the pairwise loss which has been widely used in item recommendation task. Specifically, for each user u_i in the mini-batch training, we define the positive interacted items (*i.e.*, $(v_{p_1}, v_{p_2}, ..., v_{p_S})$) of user u_i as S. For generating negative instances, we randomly sample S non-interacted items $(v_{n_1}, v_{n_2}, ...v_{n_S})$ of user u_i. Given the sampled positive and negative instances, we define our loss function as follows:

$$\mathcal{L} = \sum_{i=1}^I \sum_{s=1}^S \max(0, 1 - \text{Pr}_{i,p_s} + \text{Pr}_{i,n_s}) + \lambda \|\mathbf{\Theta}\|_\text{F}^2 \quad (7)$$

we incorporate the regularization term $\|\mathbf{\Theta}\|_\text{F}^2$ with the parameter λ. The learnable parameters are denoted as $\mathbf{\Theta}$. The training process of our model is elaborated in Algorithm 1.

IV. EVALUATION

In this section, we conduct experiments to evaluate our proposed *GNMR* method by comparing it with state-of-the-art baselines on real-world dataests.

1933

Algorithm 1: Training Process of GNMR

Input: adjacent tensor $\mathbf{X} \in \mathbb{R}^{I \times J \times K}$, 0-order node features $\bar{\mathbf{E}}^{(0)}$, maximum GNN layer L, training sampling number S, maximum epoch number N, regularization weight λ

Output: trained model parameters Θ

1   Initialize all parameters Θ
2   **for** $n = 1$ *to* N **do**
3   Randomly sample seed nodes $\mathbb{U}$, $\mathbb{V}$
4   Get $\mathbf{H}^{(0)}$ from $\bar{\mathbf{H}}^{(0)}$ for u_i in $\mathbb{U}$ and v_j in $\mathbb{V}$
5   **for** $l = 1$ *to* L **do**
6   **for** *each* u_i *in* $\hat{\mathbb{U}}$, v_j *in* $\hat{\mathbb{V}}$ *and* $k = 1$ *to* K **do**
7   Construct type-specific behavior message $\mathbf{H}^k$
8   Recalibrate the message and get $\hat{\mathbf{H}}^k$
9   Acquire the aggregated embedding $\mathbf{H}^{(l)}$
10   **end**
11   **end**
12     $\mathcal{L} = \lambda \|\Theta_{\mathrm{F}}^2\|$
13   **for** *each* u_i *in* $\hat{\mathbb{U}}$ **do**
14   Sample S positive and S negative items v_{p_s} and v_{n_s} from $\hat{\mathbb{V}}$
15   **for** *each* v_{p_s} *and* v_{n_s} **do**
16   Calculate $\mathrm{Pr}_{i,j}$ with multi-order matching
17         $\mathcal{L} += \max(0, 1 - \mathrm{Pr}_{i,p_s} + \mathrm{Pr}_{i,n_s})$
18   **end**
19   **end**
20   Optimize Θ using Adam with loss $\mathcal{L}$
21   **end**
22   return Θ

A. Experimental Settings

1) **Data Description:** We evaluate our *GNMR* on three datasets collected from MovieLens, Yelp and Taobao platforms. The statistical information of them is shown in Table I.

- **MovieLens Data**[1]. It is a benchmark dataset for performance evaluation of recommender systems. In this data, we differentiate users' interaction behaviors over items in terms of the rating scores, *i.e.*, $r_{i,j} \leq 2$: *dislike* behavior. (2) $2 < r_{i,j} < 4$. *neutral* behavior. (3) $r_{i,j} > 4$: *like* behavior.
- **Yelp Data**[2]. This dataset is collected from the public data repository from Yelp platform. Besides the users' rating data, this data also contains the tip behavior if user gives a tip on his/her visited venues. Ratings are also mapped into three interactions types with the same partition strategy in MovieLens. Similarly, like behavior is the our target and other auxiliary behaviors are {tip, neutral, dislike}.
- **Taobao Data**[3]. This dataset contains different types of user behaviors from Taobao platform, *i.e.*, page view, add-to-cart, add-to-favorite and purchase.

2) **Evaluation Metrics:** We utilize two metrics *Hit Ratio (HR@N)* and *Normalized Discounted Cumulative Gain (NDCG@N)* for evaluation. The higher HR and NDCG scores represent more accurate recommendation results. We sample

[1] https://grouplens.org/datasets/movielens/10m/
[2] https://www.yelp.com/dataset/download
[3] https://tianchi.aliyun.com/dataset/dataDetail?dataId=649&userId=1

TABLE I
STATISTICS OF THE EXPERIMENTED DATASETS

Dataset	User #	Item #	Interaction #	Interactive Behavior Type
Yelp	19800	22734	1.4×10^6	{Tip, Dislike, Neutral, Like}
ML10M	67788	8704	9.9×10^6	{Dislike, Neutral, Like}
Taobao	147894	99037	7.6×10^6	{Page View, Favorite, Cart, Purchase}

each positive instance with 99 negative instances from users' interacted and non-interacted items, respectively.

3) **Baselines:** We consider the following baselines:

- **BiasMF** [8]: This method enhances the matrix factorization framework with the consideration of user and item bias.
- **DMF** [18]: It integrates the matrix factorization and neural network to project users and items into embeddings.
- **NCF** [5]: It augments the collaborative filtering with deep neural networks. We consider three variants with different feature modeling methods, *i.e.*, i) NCF-N: fusing the matrix factorization and multi-layer perceptron; ii) NCF-G: performing fixed element-wise product on user and item embeddings; iii) NCF-M: using multi-layer perceptron to model the interaction between user's and item's features.
- **AutoRec** [9]: It is based on the autoencoder paradigm for embedding generation in collaborative filtering with the reconstruction objective in the output space.
- **CDAE** [14]: In CDAE, a denoising autoencoder model is developed to learn latent representations of users and items with the incorporation of non-linearities.
- **NADE** [21]: It is a feed-forward neural autoregressive framework with parameter sharing strategy between different ratings of users, to improve collaborative filtering.
- **CF-UIcA** [1]: It performs autoregression to capture correlations between users and items for collaborative filtering.
- **NGCF** [11]: It is a graph neural collaborative filtering model to project users and items into latent representations over the structure of user-item interaction graph. The embedding propagation is performed across graph layers.
- **NMTR** [2]: It is a multi-task learning framework to correlate the prediction of different types of user behaviors. In NMTR framework, a shared embedding layer is designed for different behavior types of interactions.
- **DIPN** [3]: This method is on the basis of attention mechanism and recurrent neural network to aggregate signals from users' browse and purchase behaviors.

4) **Parameter Settings:** We implement our *GNMR* model with TensorFlow and the model is optimized using the Adam optimizer during the training phase. We set the dimension of embeddings as 16 and the number of latent dimensions in our memory neural module as 8. In addition, the batch size and learning rate in our model is set as 32 and $1e^{-3}$. The decay rate of 0.96 is applied during the learning process.

B. Performance Comparison

In this section, we evaluate all approaches in predicting like behaviors of user over different items on different real-world

1934

Model	MovieLens Data		Yelp Data		Taobao Data	
	HR	NDCG	HR	NDCG	HR	NDCG
BiasMF	0.767	0.490	0.755	0.481	0.262	0.153
DMF	0.779	0.485	0.756	0.485	0.305	0.189
NCF-M	0.757	0.471	0.714	0.429	0.319	0.191
NCF-G	0.787	0.502	0.755	0.487	0.290	0.167
NCF-N	0.801	0.518	0.771	0.500	0.325	0.201
AutoRec	0.658	0.392	0.765	0.472	0.313	0.190
CDAE	0.659	0.392	0.750	0.462	0.329	0.196
NADE	0.761	0.486	0.792	0.499	0.317	0.191
CF-UIcA	0.778	0.491	0.750	0.469	0.332	0.198
NGCF	0.790	0.508	0.789	0.500	0.302	0.185
NMTR	0.808	0.531	0.790	0.478	0.332	0.179
DIPN	0.791	0.500	0.811	0.540	0.317	0.178
GNMR	**0.857**	**0.575**	**0.848**	**0.559**	**0.424**	**0.249**

Model	HR					NDCG				
	@1	@3	@5	@7	@9	@1	@3	@5	@7	@9
BiasMF	0.287	0.474	0.626	0.714	0.741	0.287	0.378	0.432	0.461	0.474
NCF-N	0.260	0.481	0.604	0.695	0.742	0.260	0.396	0.444	0.477	0.492
AutoRec	0.228	0.455	0.586	0.684	0.732	0.228	0.362	0.410	0.449	0.462
NADE	0.265	0.508	0.642	0.720	0.784	0.265	0.402	0.454	0.478	0.497
CF-UIcA	0.235	0.449	0.576	0.659	0.731	0.235	0.360	0.412	0.440	0.463
NMTR	0.214	0.466	0.610	0.700	0.762	0.214	0.360	0.419	0.450	0.469
GNMR	**0.320**	**0.590**	**0.700**	**0.784**	**0.831**	**0.320**	**0.473**	**0.519**	**0.542**	**0.558**

(a) ML-HR   (b) ML-NDCG   (c) Yelp-HR   (d) Yelp-NDCG

Figure 2. Ablation study of *GNMR* framework on Yelp and MovieLens data in terms of *HR*@*10* and *NDCG*@*10*.

datasets. We report HR@10 and NDCG@10 in Table II. We can observe that *GNMR* outperforms all baselines on various datasets in terms of both HR@10 and NDCG@10. For example, *GNMR* achieves relatively 6.06% and 8.34% improvements in terms of HR, and 8.28% and 15.0% improvements in terms of NDCG, over NMTR and DIPN, respectively on MovieLens dataset. Such significant performance gain could be attributed to the explicitly modeling of inter-dependencies among different types of user-item interactions. The best performance is followed by NMTR which defines the correlations between different types of user behaviors in a cascaded way. In contrast, our *GNMR* model automatically captures the implicit behavior dependencies based on our developed graph neural networks. Furthermore, the performance gap between *GNMR* and the state-of-the-art graph neural collaborative filtering model–NGCF, justifies the advantage of incorporating the cross-interaction collaborative relations into the graph neural architecture for user's preference encoding.

In addition, we can observe that recommendation techniques (*i.e.*, NMTR and DIPN) with the consideration of multi-behavioral patterns improve the performance as compare to other types of baselines which fail to differentiate interaction relationships between user and item. This points to the positive effect of incorporating multi-typed interactive patterns in the embedding function of recommender systems. Among other baselines, the graph neural network-based model (*i.e.*, NGCF) achieves better performance in most cases, which justifies the rationality of encoding the collaborative signals by conducting the embedding propagation between neighboring nodes based on user-item graph-structural relations.

We further measure the ranking quality of top-*N* recommended items with varying *N* from 1 to 9 with an increment of 2. The evaluation results (measured by HR@*N* and NDCG@*N*) on Yelp data are presented in Table III. We can observe that *GNMR* achieves the best performance under different values of *N*, which suggests that *GNMR* assigns higher score to the user's interested item in the top-ranked list and hit the ground truth at top positions.

C. Component Ablation Evaluation

We further conduct experiments to evaluate the component-wise effect of our proposed *GNMR* framework. Evaluation results of several designed model variants are shown in Figure 2.

Type-specific Behavior Embedding Component. We first design a contrast method *GNMR*-be without the type-specific behavior embedding layer, to generate the propagated feature embedding of individual type of user behavior during the message passing process. The performance gap between *GNMR*-be and *GNMR* indicates the effectiveness of our attention-based projection layer to capture the type-specific behavioral features.

Message Aggregation Component. Another dimension of model ablation study aims to evaluate the efficacy of our message aggregation layer. *GNMR*-ma is the variant of our framework, in which the behavior dependency modeling component is removed in our graph-structured embedding propagation. From Figure 2, we observe that *GNMR* outperforms *GNMR*-ma, which shows that capturing the type-wise behavior dependencies is helpful for learning relation-aware collaborative signals and improving the recommendation performance.

Dataset	Metric	w/o dislike	w/o neutral	w/o like	only like	*GNMR*
MovieLens	HR	0.834	0.816	0.838	0.835	**0.857**
	NDCG	0.549	0.532	0.559	0.559	**0.575**
Dataset	Metric	w/o tip	w/o dislike	w/o neutral	only like	*GNMR*
Yelp	HR	0.837	0.833	0.831	0.821	**0.848**
	NDCG	0.535	0.542	0.532	0.527	**0.559**

D. Performance with Various Behavior Types

We also perform experiments to evaluate the recommendation accuracy with the consideration of different user-item relations from multi-typed user behaviors on both MovieLens and Yelp. Particularly, for MovieLens, we first design three variants: w/o dislike, w/o neutral, w/o like indicate that we do not include the dislike, neutral and like behavior respectively
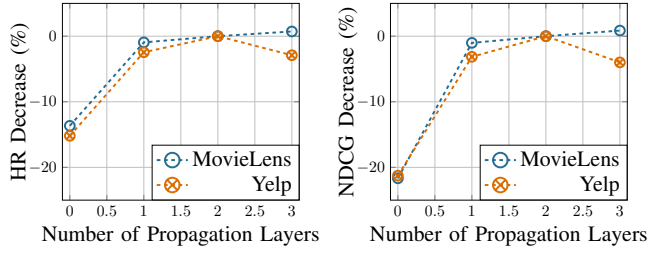
1935

Figure 3. Impact study of model depth in terms of *HR@10* and *NDCG@10*.

into our multi-behavior enhanced graph-based recommendation framework. We further introduce another model variant (only like) which merely relies on the target type of behaviors for making recommendation. We apply similar settings to Yelp data with four variants of our *GNMR* method. Evaluation results in terms of HR@10 and NDCG@10 are listed in Table IV. As we can see, the improvement is quite obvious when we consider more diverse behavior types for multi-behavioral knowledge integration in our graph neural network. In addition, compared with the model variant which only considers the target type of behavior data (*i.e.*, like behavior) for prediction, we find that our multi-behavior enhanced recommender system *GNMR* do help to improve the performance, which indicates the positive effect of incorporating auxiliary knowledge from other types of user-item interaction behaviors into the embedding function of recommendation model.

E. Impact of Model Depth in GNMR Framework

In this subsection, we examine the influence of the depth of *GNMR* to investigate the effectiveness of stacking multiple information propagation layers. Specifically, we vary the depth of our graph neural networks from 0 (without the message passing mechanism) to 3. *GNMR*-1 is defined to represent the method with one embedding propagation layer and similar notations for other variants. In Figure 3, the y-axis represents the performance variation percentage compared with *GNMR*-2, so as to present results of different datasets in the same figure. From evaluation results, we can observe that *GNMR*-2 and *GNMR*-3 outperform *GNMR*-1, which justifies the effective modeling of high-order collaborative effects in our multi-behavior enhanced recommendation scenario. Learning second- or third-order collaborative relations is sufficient to encode the user-item interactive patterns. Further increasing the number of propagation layers is prone to involve noise in our graph-based neural collaborative filtering.

V. CONCLUSION

This paper contributes a new framework, named GNMR for multi-behavior enhanced recommendation through modeling inter-type behavior dependencies under a graph-structured message passing architecture. GNMR effectively aggregates the heterogeneous behavioral patterns of users by performing embedding propagation over user-item interaction graph. Our GNMR could capture the heterogeneous relations among different types of user-item interactions for encoding graph-structured collaborative signals for recommendation. Through experiments on real-world datasets, our proposed GNMR is

shown to achieve better performance compared with state-of-the-art baselines. Furthermore, the ablation studies show that GNMR is able to comprehensively leverage the multi-typed behavioral patterns to improve recommendation accuracy. In future, we are interested in exploring the attribute features from user and item side, such as user profile and item textual information, so as to further alleviate the data sparsity problem. More generally, we could extend our GNMR framework to model other heterogeneous relationships (*e.g.*, different social connections or item dependencies) in recommendation.

REFERENCES

[1] C. Du, C. Li, Y. Zheng, J. Zhu, and B. Zhang. Collaborative filtering with user-item co-autoregressive models. In *AAAI*, 2018.
[2] C. Gao, X. He, D. Gan, X. Chen, F. Feng, Y. Li, T.-S. Chua, and D. Jin. Neural multi-task recommendation from multi-behavior data. In *ICDE*, pages 1554–1557. IEEE, 2019.
[3] L. Guo, L. Hua, R. Jia, B. Zhao, X. Wang, and B. Cui. Buying or browsing?: Predicting real-time purchasing intent using attention-based deep network with multiple behavior. In *KDD*, pages 1984–1992, 2019.
[4] W. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *NIPS*, pages 1024–1034, 2017.
[5] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. Neural collaborative filtering. In *WWW*, pages 173–182, 2017.
[6] Z. Hu, Y. Dong, K. Wang, and Y. Sun. Heterogeneous graph transformer. In *WWW*, pages 2704–2710, 2020.
[7] C. Huang, X. Wu, X. Zhang, C. Zhang, J. Zhao, D. Yin, and N. V. Chawla. Online purchase prediction via multi-scale modeling of behavior dynamics. In *KDD*, pages 2613–2622, 2019.
[8] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.
[9] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie. Autorec: Autoencoders meet collaborative filtering. In *WWW*, pages 111–112, 2015.
[10] C. Shi, B. Hu, W. X. Zhao, and S. Y. Philip. Heterogeneous information network embedding for recommendation. *TKDE*, 31(2):357–370, 2018.
[11] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua. Neural graph collaborative filtering. *SIGIR*, 2019.
[12] X. Wang, H. Jin, A. Zhang, X. He, T. Xu, and T.-S. Chua. Disentangled graph collaborative filtering. In *SIGIR*, pages 1001–1010, 2020.
[13] C. Wu, F. Wu, T. Qi, S. Ge, et al. Reviews meet graphs: Enhancing user and item representations for recommendation with hierarchical attentive graph neural network. In *EMNLP*, pages 4886–4895, 2019.
[14] Y. Wu, C. DuBois, et al. Collaborative denoising auto-encoders for top-n recommender systems. In *WSDM*, pages 153–162. ACM, 2016.
[15] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip. A comprehensive survey on graph neural networks. *TNNLS*, 2020.
[16] L. Xia, C. Huang, Y. Xu, P. Dai, B. Zhang, and L. Bo. Multiplex behavioral relation learning for recommendation via memory augmented transformer network. In *SIGIR*, pages 2397–2406, 2020.
[17] H. Xu, C. Huang, Y. Xu, L. Xia, H. Xing, and D. Yin. Global context enhanced social recommendation with hierarchical graph neural networks. In *ICDM*. IEEE, 2020.
[18] H.-J. Xue, X. Dai, J. Zhang, S. Huang, et al. Deep matrix factorization models for recommender systems. In *IJCAI*, pages 3203–3209, 2017.
[19] R. Ying, R. He, K. Chen, et al. Graph convolutional neural networks for web-scale recommender systems. In *KDD*, pages 974–983, 2018.
[20] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla. Heterogeneous graph neural network. In *KDD*, pages 793–803, 2019.
[21] Y. Zheng, B. Tang, W. Ding, and H. Zhou. A neural autoregressive approach to collaborative filtering. In *ICML*, 2016.