Multiplex Behavioral Relation Learning for Recommendation via Memory Augmented Transformer Network

Lianghao Xia*
South China University of Technology
cslianghao.xia@mail.scut.edu.cn

Chao Huang*
JD Finance America Corporation
chaohuang75@gmail.com

Yong Xu†
South China University of Technology
Peng Cheng Laboratory
yxu@scut.edu.cn

Peng Dai
JD Finance America Corporation
peng.dai@jd.com

Bo Zhang
Boshi Qiangzhi Science and
Technology Co., Ltd
13922820911@139.com

Liefeng Bo
JD Finance America Corporation
liefeng.bo@jd.com

ABSTRACT

Capturing users’ precise preferences is of great importance in various recommender systems (*e.g.*, e-commerce platforms and online advertising sites), which is the basis of how to present personalized interesting product lists to individual users. In spite of significant progress has been made to consider relations between users and items, most of existing recommendation techniques solely focus on singular type of user-item interactions. However, user-item interactive behavior is often exhibited with multi-type (*e.g.*, page view, add-to-favorite and purchase) and inter-dependent in nature. The overlook of multiplex behavior relations can hardly recognize the multi-modal contextual signals across different types of interactions, which limit the feasibility of current recommendation methods. To tackle the above challenge, this work proposes a **M**emory-**A**ugmented **T**ransformer **N**etworks (MATN), to enable the recommendation with multiplex behavioral relational information, and joint modeling of type-specific behavioral context and type-wise behavior inter-dependencies, in a fully automatic manner. In our MATN framework, we first develop a transformer-based multi-behavior relation encoder, to make the learned interaction representations be reflective of the cross-type behavior relations. Furthermore, a memory attention network is proposed to supercharge MATN capturing the contextual signals of different types of behavior into the category-specific latent embedding space. Finally, a cross-behavior aggregation component is introduced to promote the comprehensive collaboration across type-aware interaction behavior representations, and discriminate their inherent contributions in assisting recommendations. Extensive experiments on two benchmark datasets and a real-world e-commence user behavior data demonstrate significant improvements obtained by MATN over baselines. Codes are available at: https://github.com/akaxlh/MATN.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Collaborative Filtering; Recommendation; Multi-Behavior Learning; Transformer Network; Deep Neural Networks

ACM Reference Format:
Lianghao Xia, Chao Huang, Yong Xu, Peng Dai, Bo Zhang, and Liefeng Bo. 2020. Multiplex Behavioral Relation Learning for Recommendation via Memory Augmented Transformer Network. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ’20), July 25–30, 2020, Virtual Event, China.* ACM, Xi’an, China, 10 pages. https://doi.org/10.1145/3397271.3401445

1 INTRODUCTION

Recommender system, which facilitates the information-seeking process of users and meet their personalized interests, have played a critical role in various online services, such as e-commerce systems [13, 38], online review platforms [1, 44] and advertising [39]. At its core is to learn low-dimensional representations of user-item interaction while capturing the user preference and the underlying intrinsic characteristics [19]. Early methods towards this goal, have made significant efforts on transforming user-item interactions through vectorized representations based on the conventional Collaborative Filtering (CF) techniques (*e.g.*, matrix factorization scheme [15, 22] and its variations [12, 26]).

Inspired by the advancement of deep learning techniques, various neural network-based collaborative filtering frameworks have been developed to model the relationships between users and items. These methods aim to map sparse input interaction features into low-dimensional user/item embedding vectors and then project them into fixed-length representations in a group-wise manner [16, 54]. For example, neural collaborative filtering models replace the inner product function in the matrix factorization consider non-linearities with multilayer perceptron [11] and metric learning scheme [33]. In addition, auto-encoder architecture has served as an effective solution to learn a mapping function between the explicit

*These authors contributed equally to this work.
†Corresponding author

SIGIR ’20, July 25–30, 2020, Virtual Event, China
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401445

interaction and latent representation through the reconstruction-based encoder-decoder framework. To capture the rich graph-based neighborhood contextual signals, various graph neural encoders have been proposed to aggregate information over the user-item interaction graph, with the graph convolutional network [49] or message-passing mechanism [41].

Despite the prevalence of the above recommendation solutions, they has thus far focused on user preference representation learning with the consideration of singular type of user-item interactive behavior. However, in many practical recommendation scenarios, user-item interactions are multiplex and exhibited with relationship diversity in nature. Let's consider the e-commerce system as an example, there exist multiple types of behavior (*e.g.*, page view, add-to-favourite, add-to-cart and purchase) between users and items [8], which are mutually inter-dependent. For instance, add-to-cart behavior is more likely to co-occur with purchase than the add-to-favorite behavior. The page view and add-to-favourite behavior can also provide useful signals for making purchase decisions. In such cases, the ignorance of such multi-modal relations across different types of user-item interaction behavior, makes existing recommendation methods insufficient to distill effective collaborative signals from the collective users behavior.

The recommendation framework with multiplex interactive behavior pose two key challenges: *First*, the dependencies across different types of user-item interactions can be arbitrary since any pair of type-specific behavior could potentially be correlated due to various factors [42]. For example, users often have correlated online behaviors and exhibit different dependencies in choosing items of different categories due to his/her specialty. Such inter-dependencies between different types of interaction behavior may vary by users and items. While a handful of studies attempt to learn user preference from multi-behavior [7, 8], they merely consider the singular dimensional cascading correlations between multi-type interactions, and cannot comprehensively capture the arbitrary dependencies between different types of interaction over different items. Hence, to build effective recommendation model with the complex behavior dependencies remains a significant challenge.

Second, when modeling the relationships across different types of behavior, it is also important to capture the context and semantics of individual type of user-item interactions, *e.g.*, users' page view are more frequent than their purchases, and add-to-favorite behavior is more likely to happen over users' interested items but may postpone their buying decision. In addition, type-specific behavioral patterns interweave with each other in complex way (*e.g.*, support or mutually exclusive relations) and are difficult to be captured. During the behavioral pattern integration, as the importance of various types of behavior can be different, their relevance in assisting the forecasting task on the target behavior need to be carefully decided.

Motivated by the aforementioned challenges, this work proposes a general and flexible multi-behavior relation learning framework–**M**emory-**A**ugmented **T**ransformer **N**etworks (MATN). Specifically, this work first proposes a multi-behavior transformer network to learn type-specific behavioral representations with the incorporation of inter-dependencies across different types of user-item interactions. By integrating the transformer network with a memory-augmented attention mechanism, we endow the MATN framework with the capability of incorporating type-specific behavior contextual signals. to collectively model the implicit relevance across multi-type behavioral patterns and perform comprehensive learning for making recommendations, we design a behavior type-wise gating mechanism which promotes the collaboration of different types of interactions. In the pattern aggregation layer, MATN could learn cross-type representations in the latent feature spaces by automatically adjusting the contribution of each behavior view point in the behavior predictive model.

The contributions of this paper are highlighted as follows:

- We propose MATN, a new recommendation framework with multiplex behavioral relation learning. MATN explicitly encodes multi-behavior relational structures by preserving both the cross-type behavior collaborative signals and type-specific behavior contextual information.
- We first develop a multi-behavior dependency encoder with a transformer architecture, to inject collaborative signals across different types of user-item interactions into the embedding process. Furthermore, we augment the multi-behavior transformer network with a memory attention mechanism, which is capable of uncovering type-specific behavior semantics during the customized representation recalibration phase.
- Finally, a type-wise pattern aggregation layer with gating mechanism is developed to promote the collaboration of different behavior views for robust representations on user preferences.
- Our extensive experiments on two benchmark datasets and a user behavior data from a major e-commence platform, demonstrate that MATN outperforms 12 baselines from various research lines in yielding better recommendation performance. We further perform case studies with qualitative examples to better understand the interpretation ability of MATN framework, and study the model efficiency under different recommendation scenarios.

2 PRELIMINARY

In the recommendation scenario, we first define the behavior (*e.g.*, purchase) which we aim to predict as *target behavior*, other relevant user-item interactive behavior (*e.g.*, click, add-to-cart and add-to-favorite) is termed as *source behavior*. In this work, we aim to explore the latent relational structures between different types of user behavior (*e.g.*, purchases and click) for making predictions on the target behavior of users in recommender systems.

Definition 1. ***Multi-Behavior Tensor*** $\mathcal{X}$. *We define a three-dimensional multi-behavior tensor* $\mathcal{X} \in \mathbb{R}^{I\times J\times L}$ *to represent the* L *(indexed by* l*) types of behavior from* I *(indexed by* i*) users over* J *(indexed by* j*) items. Without loss of generality, we focus on the implicit user feedback which is more common in practical recommendation scenarios [25, 40]. Particularly, in tensor* $\mathcal{X}$*, each entry* $x_{i,j,l} = 1$ *if user* u_i *is interacted with item* t_j *given the* l*-th behavior type. For example, if user* u_i *purchases item* t_j*, the corresponding element* $x_{i,j,l}$ *will be set as 1 in the purchase behavior matrix* $\mathcal{X}_l$*.*

Problem Statement. Based on the aforementioned definitions, the recommendation task with multiplex behavior learning is formulated as follows: **Input**: the user-item interaction data represented with multi-behavior tensor $\mathcal{X}$ (including both the target and source behavior). **Output**: A predictive model to effectively infer the unknown user-item interactions in $\mathcal{X}$ with the target behavior l.

$$\Pr(x_{i,j,l} = 1) = f(\mathcal{X} \in \mathbb{R}^{I \times J \times L}) \quad i \in [1, ..., I]; j \in [1, ..., J] \qquad (1)$$

3 METHODOLOGY

In this section, we present the technical details of MATN framework, the architecture of which is illustrated in Figure 1. MATN is a hierarchical neural architecture with three key modules in MATN: (i) cross-behavior embedding layers that learn the representations by exploring the inter-dependencies across different types of interactions; (ii) a customized representation recalibration network that refines the latent embeddings, with the preservation of individual behavioral contextual information; (iii) a forecasting layer that aggregates the refined behavior type-specific embeddings and outputs a predicted likelihood of a user-item interaction pair.

3.1 Multi-Behavior Dependency Modeling

As discussed before, different types of user behaviors are correlated with each other, which brings in new challenges to the recommendation framework. To model the inter-dependencies across different types of behavior, we design a multi-behavior transformer network to promote the collaboration of different behavioral views. To achieve this goal, we learns a robust representation for user-item interactive patterns of each individual categorical behavior l, which integrates the relevant information from other behavior views $l' \in [1, ..., L] \& l' \neq l$.

3.1.1 **Initialized Embedding Layer.** Firstly, a projection layer is introduced to map the original multi-behavior user-item interaction data into initial latent representations. We denote the interaction vector of l-th behavior type and i-th user (u_i) over all items ($t_j, 1 \leq j \leq J$) as $\mathcal{X}_{i,l} \in \mathbb{R}^J$. The projection operation for $\mathcal{X}_{i,l}$ is formally defined as $\tilde{\mathcal{X}}_{i,l} = \mathbf{V} \cdot \mathcal{X}_{i,l}$, where $\mathbf{V} \in \mathbb{R}^{d \times J}$ and d denotes learned projection matrix and hidden state dimensionality, respectively. Note that $\mathbf{V}$ is shared across behavior categories to model the common semantics of different interactions. The projected $\tilde{\mathcal{X}}_{i,l}$ serves as an initial parameterized state for user-item interactions $\mathcal{X}_{i,l}$, to be optimized with the following modules.

3.1.2 **Multi-Head Self-Attentive Mechanism.** Inspired by the promising potential of self-attention mechanism in data correlation learning [50], we build our multi-behavior dependency learning module upon the architecture of multi-head self-attention network, which allows the learned behavior type-specific representations to interact with each other and identify the most informative correlated signals across different types of interaction behavior. Furthermore, considering the fact that different types of interaction behavior (*e.g.*, add-to-cart and purchase) can be mutually correlated in a complex way (due to personalized factors) [2], the multi-head learning strategy enable our behavior dependency encoder with the capability of jointly attending to information from different representation subspaces [48]. In our transformer network, we adopts the scaled dot-product attention for each h-th head with the definitions of query, key and value transformation matrices $\mathbf{Q}^h \in \mathbb{R}^{\frac{d}{H} \times d}$, $\mathbf{K}^h \in \mathbb{R}^{\frac{d}{H} \times d}$ and $\mathbf{V}^h \in \mathbb{R}^{\frac{d}{H} \times d}$. Then, the weight $\hat{\alpha}^h_{l,l'}$ assigned to each input value is determined by the dot-product of the query with all the keys as follows:

$$\alpha^h_{l,l'} = \frac{(\mathbf{Q}^h \cdot \tilde{\mathcal{X}}_{i,l})^\top (\mathbf{K}^h \cdot \tilde{\mathcal{X}}_{i,l'})}{\sqrt{\frac{d}{H}}}; \quad \hat{\alpha}^h_{l,l'} = \frac{\exp \alpha^h_{l,l'}}{\sum_{l'=1}^{L} \exp \hat{\alpha}^h_{l,l'}} \qquad (2)$$

where $\alpha^h_{l,l'}$ is the intermediate variable fed into the softmax operation to generate the final relevance score $\hat{\alpha}^h_{l,l'}$ between the l-th and l'-th type of behavior. Based on the learned head-specific attention weights, our dependency encoding module aims to learns a cross-head relevance score for each behavior type-specific representation $\tilde{\mathcal{X}}_{i,l}$ with the following multi-head learning operations:

$$\mathbf{Y}_{i,l} = \text{MH-Att}(\tilde{\mathcal{X}}_{i,l}) = \Big\Vert_{h=1}^{H} \sum_{l'=1}^{L} \alpha^h_{l,l'} \mathbf{V}^h \cdot \tilde{\mathcal{X}}_{i,l'} \qquad (3)$$

To alleviate the gradient vanishing issue, the residual connection [9] is employed in the deep neural network structures. Additionally, we element-wisely add the learned dependency-aware behavior type-specific interaction representation $\mathbf{Y}_{i,l}$ with the projected feature embedding $\tilde{\mathcal{X}}_{i,l}$ of l-th behavior, so as to jointly preserve the behavior type-specific interaction features and the underlying inter-dependent signals across various types of user behavior. Formally, such operation is given as: $\tilde{\mathbf{Y}}_{i,l} = \tilde{\mathcal{X}}_{i,l} + \mathbf{Y}_{i,l}$.

3.2 Customized Behavioral Context Learning

In addition to the implicit multi-behavior dependency encoded by the above introduced transformer network, each type of behavior may have its own characteristics. For instance, users' page view behavior are more frequent than their purchases and add-to-cart behavior is more likely to be followed by a purchase than the add-to-favorite behavior. While the cross-behavior inter-correlation structure can be modeled by our transformer module, the behavior type-specific semantic diversity has been overlooked. Hence, we propose to augment our MATN framework with the capability of capturing the semantic signals of each individual type of interaction behavior. Motivated by the recent advancements of augmented neural architecture and attention mechanism [21, 33], we perform a customized representation recalibration process on behavior type-specific context with a memory-augmented attention network. In our memory-based behavior context learning module, we provide a customized transformations for each type of user behavior representation $\tilde{\mathbf{Y}}_{i,l}$ by stacking a set of memory blocks. By doing so, we endow MATN with the power of distilling the underlying semantics from the specific contextual user-item interaction scenario (*e.g.*, page view, interested in, want to buy, or purchase).

In specific, our customized embedding recalibration module aims to learn M (indexed by m) transformation matrices (individual is referred as $\mathbf{U}_m \in \mathbb{R}^{d \times d}$) as the corresponding augmented memory, in order to project the general behavior embedding $\tilde{\mathbf{Y}}_{i,l}$ into a type-aware latent learning space. By applying different memory transformations over different types of behavior, each type of behavioral features are refined with respect to its own contexts with the designed memory, and a customized behavioral representations are generated through this type-specific transformation procedure.

Furthermore, in order to alleviate the overfitting phenomenon of type-specific memory augmented neural network architecture [53],

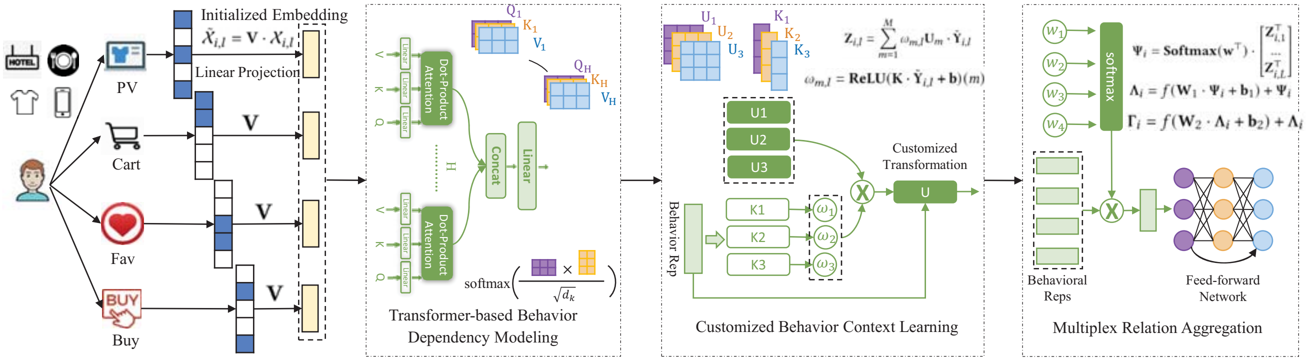

PV
Cart
Fav
Buy
Initialized Embedding
$\tilde{X}_{i,l} = \mathbf{V} \cdot X_{i,l}$
Linear Projection
V
V
V
V K Q
Linear
Dot-Product Attention
H
Concat
Linear
Q_1 K_1 V_1
Q_H K_H V_H
softmax$\left(\frac{\times}{\sqrt{d_k}}\right)$
Transformer-based Behavior Dependency Modeling
U_1 U_2 U_3
K_1 K_2 K_3
$\mathbf{Z}_{i,l} = \sum_{m=1}^{M} \omega_{m,l}\mathbf{U}_m \cdot \tilde{\mathbf{Y}}_{i,l}$
$\omega_{m,l} = \mathbf{ReLU}(\mathbf{K} \cdot \tilde{\mathbf{Y}}_{i,l} + \mathbf{b})(m)$
U1
U2
U3
Behavior Rep
K1
K2
K3
ω_1 ω_2 ω_3
Customized Transformation
U
Customized Behavior Context Learning
w_1 w_2 w_3 w_4
softmax
$\Psi_i = \mathbf{Softmax}(\mathbf{w}^\top) \cdot \begin{bmatrix} \mathbf{Z}_{i,1}^\top \\ \cdots \\ \mathbf{Z}_{i,L}^\top \end{bmatrix}$
$\Lambda_i = f(\mathbf{W}_1 \cdot \Psi_i + \mathbf{b}_1) + \Psi_i$
$\Gamma_i = f(\mathbf{W}_2 \cdot \Lambda_i + \mathbf{b}_2) + \Lambda_i$
Behavioral Reps
Feed-forward Network
Multiplex Relation Aggregation

Figure 1: The model architecture of the proposed MATN framework. The initialized embedding layer shares parameters across different behavior types. The transformer-based behavior dependency encoder takes all kinds of behavioral interaction data for dependency modeling. Different types of behaviors are individually transformed by the customized context learning with shared key and memory slots. $\otimes$ is the dot-product between the embeddings and transformation weight matrix.

we employ an attention network to learn the relations between L behavior types and M memory matrices in an explicit way, and generate a behavior type-specific transformations with weighted summation. Formally, the refined representation with customized behavioral context for the l-th interaction type is

$$\mathbf{Z}_{i,l} = \sum_{m=1}^{M} \omega_{m,l}\mathbf{U}_m \cdot \tilde{\mathbf{Y}}_{i,l}; \quad \omega_{m,l} = \mathbf{ReLU}(\mathbf{K} \cdot \tilde{\mathbf{Y}}_{i,l} + \mathbf{b})(m) \qquad (4)$$

where $\mathbf{K} \in \mathbb{R}^{M \times d}, \mathbf{b} \in \mathbb{R}^{M}$ are the transformation and bias for calculating attention weights. Instead of using Softmax, we use ReLU to relieve the gradient vanishing issue and mke it easier to train the attentive weight calculating. The memory transformation matrices $\mathbf{U}_m$ and calculating attention weights $\omega_{m,l}$ are jointly trained with other components of MATN.

3.3 Multiplex Relation Aggregation Layer

Next, we build upon a behavior type-wise gating mechanism to aggregate the learned latent representations from the memory-augmented transformer network, with the exploration of their contributions in capturing user's preference and assisting making predictions on the target behavior. Considering the distinct effects of different types of behavior in characterizing user's interest, *e.g.* user's historical purchases may be more relevant to his future purchases as compared to his page view activities, the type-specific importance is learned in our gated mechanism in an adaptive manner. Formally, the applied weighted aggregation gate outputs a d-dimensional unified representation for u_i as follows:

$$\Psi_i = \mathbf{Softmax}(\mathbf{w}^\top) \cdot \begin{bmatrix} \mathbf{Z}_{i,1}^\top \\ \cdots \\ \mathbf{Z}_{i,L}^\top \end{bmatrix} \qquad (5)$$

where $\mathbf{w} \in \mathbb{R}^d$ is the parametric weights for aggregation, the Softmax activation function is used to normalize the weights. By applying the weighted aggregation gate, MATN learns the contributions of different behavior types and thus can enable the adaptive aggregation in modeling the cross-type behavior relations.

After obtaining the aggregated user behavior representation, the MATN adopts a two-layer feed-forward network with non-linear activation, to capture the complex feature interactions in the latent embeddings. Formally the deeply-extracted user representations are learned with the following operation:

$$\Lambda_i = f(\mathbf{W}_1 \cdot \Psi_i + \mathbf{b}_1) + \Psi_i; \quad \Gamma_i = f(\mathbf{W}_2 \cdot \Lambda_i + \mathbf{b}_2) + \Lambda_i \qquad (6)$$

where $\mathbf{W}_* \in \mathbb{R}^{d \times d}$ and $\mathbf{b}_* \in \mathbb{R}^d$ are transformation and bias vectors of the neural network, f is element-wisely applying non-linear activation functions, and residual connections are also employed. $\Gamma_i \in \mathbb{R}^d$ is the final user representation.

3.4 The Learning Process of *MATN*

Given the user behavior representation aggregated from different views (*i.e.*, behavior type-specific semantics and cross-type behavior dependencies), MATN could make predictions on user's preference over items for the target l-th type of behavior. In particular, the prediction process is performed through a dot product operation $\Pr(X_{i,j,l}) = \mathbf{P}_j^\top \cdot \Gamma_i$, where $\mathbf{P}_j \in \mathbb{R}^d$ is from a parametric embedding table for all items, and the result $\Pr(X_{i,j,l})$ is a scalar score representing u_i's tendency of interacting with t_j under behavior l. Inspired by the settings of learning process on top-N recommendation tasks [23, 51], we leverage the pair-wise loss to model the relative position in ranking-based recommendation scenarios. For each training step for user u_i, we sample a positive interaction set $\{t_{p_1}, t_{p_2}, ...t_{p_s}\}$ composed of interacted items with u_i for the target l-th type of behavior. Here. s is defined as the number of the positive samples. Correspondingly, the same number of items that have no interactions with u_i in the training set are randomly sampled to form the negative interaction set $\{t_{n_1}, t_{n_2}, ..., t_{n_s}\}$. Based on the above descriptions, we formally define our loss function over all the samples of all users as below:

$$\text{Loss} = \sum_{i=1}^{I} \sum_{k=1}^{s} \mathbf{max}(0, 1 - \Pr(X_{i,p_k,l}) + \Pr(X_{i,n_k,l})) + \lambda \|\Theta\|_{\text{F}}^2 \qquad (7)$$

where the first term is the pair-wise loss for a positive-negative pair. It expands the signed difference between two predictions, until it

Algorithm 1: Learning Process of MATN Framework

```
Input: user-item interaction tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times L}$, target
       behavior $l$, sample number $s$, maximum epoch
       number $E$, regularization weight $\lambda$, learning rate $\eta$
Output: trained parameters in $\mathbf{\Theta}$
Initialize all parameters in $\mathbf{\Theta}$
for $e = 1$ to $E$ do
    Draw a mini-batch $\mathbf{U}$ from all users $\{1, 2, ..., I\}$
    Loss $= \lambda \cdot \|\mathbf{\Theta}\|_{\mathrm{F}}^2$
    for each $u_i \in U$ do
        Sample $s$ positive items $\{t_{p_1}, ..., t_{p_s}\}$ from $\mathcal{X}_{i,l}$
        Sample $s$ negative items $\{t_{n_1}, ..., t_{n_s}\}$ from $\mathcal{X}_i$
        Compute $\mathbf{\Gamma}_i$ according to Eq 2 to Eq 6
        for $k = 1$ to $s$ do
            $\Pr(\mathcal{X}_{i,p_k,l}) = \mathbf{P}_{p_k}^{\top} \cdot \mathbf{\Gamma}_i$
            $\Pr(\mathcal{X}_{i,n_k,l}) = \mathbf{P}_{n_k}^{\top} \cdot \mathbf{\Gamma}_i$
            Loss+ $= \mathbf{max}(0, 1 - (\Pr(\mathcal{X}_{i,p_k,l}) - \Pr(\mathcal{X}_{i,n_k,l})))$
        end
    end
    for each parameter $\theta \in \mathbf{\Theta}$ do
        $\theta = \theta - \eta \cdot \partial\mathrm{Loss}/\partial\theta$
    end
end
return all parameters $\mathbf{\Theta}$
```

reaches a big enough scale. The latter term is a weight decay regularization term to prevent over-fitting, and λ is the regularization weight. The learning process is elaborated in Algorithm 1.

4 EVALUATION

In this section, we perform experiments on different datasets to demonstrate the effectiveness of our *MATN*. We aim to answer the following research questions:

- **RQ1**: Compared to state-of-the-art models, does *MATN* achieve better performance in various recommendation applications?
- **RQ2**: What is the impact of the designed modules in *MATN*? Are the proposed cross-behavior transformer network and attention memory module necessary for improving performance?
- **RQ3**: How is the *MATN*'s recommendation accuracy *w.r.t* the integration of different types of behavior?
- **RQ4**: What is the influence of hyperparameter settings in *MATN* for the recommendation performance?
- **RQ5**: What behavior relational patterns does the proposed *MATN* model capture for the final recommendation decision?
- **RQ6**: How is the scalability of the *MATN* framework?

4.1 Experiment Settings

4.1.1 **Data Description.** We evaluate the model performance on three different types of datasets: (i) MovieLens: a benchmark dataset for movie recommendations; (ii) Yelp: another benchmark dataset for location-based venue recommendations from the online review platform Yelp; (iii) E-Commerce: an user behavior data from a real-world e-commence platform. Table 1 summarizes the data statistics and we present the data details as below:

Table 1: Statistics of 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}
E-Commerce	805506	584050	6.4×10^7	{Page View, Favorite, Cart, Purchase}

MovieLens Data[1]. It is a widely-used dataset for performance validation of various recommendation methods. Following the partition strategy in [18, 24], we differentiate the explicit user-item interactive behavior into three types in terms of user rating scores (*i.e.*, ranging from 1 (worst) to 5 (best) stars with 0.5 star as increment): the original rating score ≤ 2, > 2 and < 4, ≥ 4 corresponds to the *dislike*, *neutral* and *like* user behavior, respectively. In the MovieLens dataset, we regard the *like* interaction as the target behavior and other interactions (*dislike* and *neutral*) as source behavior, because the positive interactions between users and items may be more useful for capturing user's preferences in recommendations [20].

Yelp Data[2]. This is another recommendation benchmark dataset collected from Yelp. We use the same multi-behavior differentiation strategy as the MovieLens data and partition the 5-star range rating behavior into *dislike*, *neutral* and *like* user behavior. In addition to the user rating behavior, this data includes an additional tip behavior to indicate that user writes a tip on his/her visited venues. Similar to the MovieLens data, the target behavior in Yelp data is also set as the *like* interaction and others are set as source behavior.

E-Commerce Data. Besides the two benchmark datasets for movie and location-based venue recommendations, we also evaluate our MATN framework in a real-world recommendation scenario with explicit multiple user behavior data from a major online retailing platform. Specifically, this data contains four types of interaction behavior, *i.e.*, *page view*, *add-to-favorite*, *add-to-cart* and *purchase*. We consider the *purchase* behavior as the target one, since the purchase is directly related with the conversion rate of recommendation in real-life E-commerce sites [8].

4.1.2 **Evaluation Settings and Metrics.** In our experiments, we utilize the leave-one-out evaluation strategy which has been widely utilized in recommendation literature [11, 12]. Following their evaluation settings, we regard the latest interaction of each user as the test set and use the rest of data for training. For efficient and fair evaluation, we follow the common strategy in [14, 32] to associate each ground truth item with 99 randomly sampled negative instances which have not interacted with the corresponding user.

We leverage two widely-used ranking metrics: *Hit Ratio (HR@k)* and *Normalized Discounted Cumulative Gain (NDCG@k)* [4, 41], to investigate the ranking performance (top-k ranked recommended items) of all compared methods. Note that higher HR and NDCG scores reflect better recommendation results. In our experiments, we also evaluate the model performance by varying the k value.

[1] https://grouplens.org/datasets/movielens/10m/

[2] https://www.yelp.com/dataset/download

4.1.3 Competitive Baselines.

To perform a comprehensive performance validation, we compare our *MATN* with 12 baselines from six research lines, which are elaborated as follows:

Conventional Matrix Factorization-based Recommendation:

- **BiasMF** [15]: This method is built upon the matrix factorization architecture with the incorporation of user and item biases.

Neural Collaborative Filtering Models for Recommendation:

- **DMF** [47]: It is a deep matrix factorization model which takes both the explicit and implicit feedback as the input.
- **NCF** [10]: NCF aims to supercharge collaborative filtering with non-linear neural networks. We consider three variants of NCF *w.r.t* user-item interaction encoders: *i.e.*, Multilayer perceptron (*i.e.*, NCF-M), concatenated element-wise-product branch (*i.e.*, NCF-N) and the fixed element-wise product (*i.e.*, NCF-G).

Collaborative Filtering with Auto-Encoder:

- **AutoRec** [27]: It leverages a three-layer autoencoder to map user-item interactions into latent representations.
- **CDAE** [45]: In this autoencoder CF, an adaptive loss is incorporated into the embedding projection process for users/items.

Neural Auto-regressive Recommendation Models:

- **CF-NADE** [53]: It enhances the autoregressive collaborative filtering with the parameter sharing between different ratings.
- **CF-UIcA** [5]: It is a neural co-autoregressive framework to consider the structural correlation for both users and items.

Graph Neural Network Recommendation Models:

- **ST-GCN** [49]: It stacks encoder-decoder blocks using graph convolutional networks to learn embeddings of users and items.
- **NGCF** [41]: This approach explore the structural knowledge with the message-passing mechanism to capture the high-order connections in the user-item interaction graph.

Recommendation with Multi-Behavior Learning:

- **NMTR** [7]: It is a multi-task recommendation model which considers the behavior correlations in a cascaded manner.
- **DIPN** [8]: This model utilizes bi-directional recurrent network and attention mechanism to consider the correlations between the buying or browsing activities.

4.1.4 Parameter Settings.

In the latent learning space of *MATN* framework, we set the hidden state dimensionality d as 16. In the multi-behavior transformer module, we set the number of attention heads for multi-dimensional learning as 2. Furthermore, the number of memory transformations is set as 8 in our customized behavior-specific context encoding. We implement our *MATN* with TensorFlow and use Adam optimizer for model optimization with the learning rate and batch size of $1e^{-3}$ and 32, respectively. The decay rate of 0.96 is applied for each epoch during the training phase. To reduce the overfitting effect, we adopt set weight decay as the regularization strategy with the selection from {0.05, 0.01, 0.005, 0.001, 0}. The depth of our feature extraction module is set as 3. For the baselines (*i.e.*, NCF and NMTR) which employ the point-wise loss, we set the sampling ratio for positive and negative instances from the range of $1:1$ to $1:4$.

Table 2: Prediction performance on Yelp (like behavior), MovieLens (like behavior) and E-Commerce (purchase behavior) data, in terms of *HR@k* and *NDCG@k* ($k = 10$).

Model	Yelp Data				MovieLens				E-Commerce			
	HR	Imp	NDCG	Imp	HR	Imp	NDCG	Imp	HR	Imp	NDCG	Imp
BiasMF	0.755	9.4%	0.481	10.2%	0.767	10.4%	0.490	16.1%	0.262	35.1%	0.153	36.6%
DMF	0.756	9.3%	0.485	9.3%	0.779	8.7%	0.485	17.3%	0.305	16.1%	0.189	10.6%
NCF-M	0.714	15.7%	0.429	23.5%	0.757	11.9%	0.471	20.8%	0.319	11.0%	0.191	9.4%
NCF-G	0.755	9.4%	0.487	8.8%	0.787	7.6%	0.502	13.3%	0.290	22.1%	0.167	15.1%
NCF-N	0.771	7.1%	0.500	6.0%	0.801	5.7%	0.518	9.8%	0.325	8.9%	0.201	4.0%
AutoRec	0.765	8.0%	0.472	12.3%	0.658	28.7%	0.392	45.2%	0.313	13.1%	0.190	10.0%
CDAE	0.750	10.1%	0.462	14.7%	0.659	28.5%	0.392	45.2%	0.329	7.6%	0.196	6.6%
CF-NADE	0.792	4.3%	0.499	6.2%	0.761	11.3%	0.486	17.1%	0.317	11.7%	0.191	9.4%
CF-UIcA	0.750	10.1%	0.469	13.0%	0.778	8.9%	0.491	15.9%	0.332	6.6%	0.198	5.6%
ST-GCN	0.775	6.6%	0.465	14.0%	0.738	14.8%	0.444	28.2%	0.347	2.0%	0.206	1.5%
NGCF	0.789	4.7%	0.500	6.0%	0.790	7.2%	0.508	12.0%	0.302	17.2%	0.185	13.0%
NMTR	0.790	4.6%	0.478	10.9%	0.808	4.8%	0.531	7.2%	0.332	6.6%	0.179	16.8%
DIPN	0.791	4.4%	0.500	6.0%	0.811	4.4%	0.540	5.4%	0.317	11.7%	0.178	17.4%
MATN	**0.826**	–	**0.530**	–	**0.847**	–	**0.569**	–	**0.354**	–	**0.209**	–

4.2 Performance Comparison (RQ1)

4.2.1 Performance on Target Behavior.

In the evaluation, we first perform experiments to separately make recommendations on venue, movie and online retailing products with three types of datasets and the results are shown in Table 2 ("Imp" indicates the relatively improvement ratio). We observe the remarkable performance improvement achieved by our *MATN* in predicting different types of behaviors. We attribute such improvements to exploration of the cross-type behavior dependencies which are neglected by most existing methods, although they attempt to model complex user-item interactive relations with various deep neural encoders (*e.g.*, autoencoder, graph neural network, attention mechanism).

Additionally, by jointly analyzing the results among the three datasets, we find that the improvement of *MATN* on the E-Commerce data is the most significant with the largest data scale. This may be caused by the behavior diversity: the multiple behaviors from the E-Commerce site are constructed with four different types of behavior which may show strong ordinal relations between the target (purchase) and source behaviors (*e.g.*, page view → add-to-cart → purchase) in the real-world online retailing systems. The consistent improvement across datasets with different user-item interaction densities, suggests the robustness of *MATN* in accurately learning user preference under different sparsity degrees.

Lastly, it is worth mentioning that although the correlations between behavior has been considered in recent recommendation solutions (*i.e.*, NMTR and DIPN), they merely model the singular dimensional cascading correlations between multi-type interactions, and cannot comprehensively capture the arbitrary dependencies between different types of interaction with different items. Therefore, such oversimplification on the behavior dependency leads to suboptimal recommendation results.

4.2.2 Overall Prediction Click Behavior.

We also conduct experiments to evaluate the recommendation performance of all compared methods by forecasting the overall user-item interaction (*i.e.*, click behavior), since the accurate predictions on overall interactive behavior (*e.g.*, including all page view, add-to-cart and purchase behavior) could also provide useful insights for recommendation scenarios which focus on optimizing the click rate. As shown in Table 3, our *MATN* still achieves the best performance on all datasets as compared to various types of baselines. This validation shows

Table 3: Overall recommendation performance in forecasting click behavior in terms of *HR@k* and *NDCG@k* ($k = 10$).

Data	Metric	BiasMF	DMF	NCF-M	NCF-G	NCF-N	AutoRec	CDAE	CF-NADE	CF-UIcA	ST-GCN	NGCF	NMTR	DIPN	*MATN*
Yelp	HR	0.809	0.801	0.770	0.808	0.812	0.745	0.753	0.771	0.808	0.796	0.810	0.794	0.816	**0.848**
	NDCG	0.513	0.503	0.464	0.519	0.523	0.456	0.456	0.478	0.512	0.483	0.521	0.473	0.514	**0.548**
MovieLens	HR	0.727	0.730	0.693	0.745	0.748	0.612	0.613	0.677	0.718	0.688	0.735	0.773	0.776	**0.808**
	NDCG	0.456	0.461	0.427	0.470	0.473	0.361	0.360	0.421	0.442	0.425	0.468	0.497	0.499	**0.535**
E-Commerce	HR	0.383	0.399	0.401	0.400	0.409	0.423	0.427	0.486	0.428	0.452	0.470	0.409	0.405	**0.535**
	NDCG	0.228	0.245	0.239	0.240	0.244	0.257	0.262	0.430	0.257	0.257	0.281	0.236	0.237	**0.326**

the potential of the overall prediction performance of *MATN* by jointly considering multi-type behavior of users.

4.2.3 **Ranking Performance v.s. Top-*K* Value.** We also evaluate the model ranking performance by varying the K value in terms of HR@K and NDCG@K. We compare *MATN* with the best performed method of each baseline categories (see Section 4.1.3 for baseline description), and report the results of predicting the click and like behavior on Yelp data in Table 4. We can observe that *MATN* consistently outperforms other representative baselines with different settings of K.

Table 4: Ranking performance evaluation on Yelp dataset with varying Top-*K* value in terms of *HR@K* and *NDCG@K*

Model	Metric	Click					Like				
		@1	@3	@5	@7	@9	@1	@3	@5	@7	@9
BiasMF	HR	0.261	0.519	0.644	0.728	0.799	0.287	0.474	0.626	0.714	0.741
	NDCG	0.261	0.409	0.460	0.490	0.508	0.287	0.378	0.432	0.461	0.474
NCF-N	HR	0.278	0.535	0.661	0.747	0.800	0.260	0.481	0.604	0.695	0.742
	NDCG	0.278	0.426	0.474	0.505	0.519	0.260	0.396	0.444	0.477	0.492
AutoRec	HR	0.227	0.443	0.568	0.650	0.715	0.228	0.455	0.586	0.684	0.732
	NDCG	0.227	0.343	0.398	0.426	0.440	0.228	0.362	0.410	0.449	0.462
CF-NADE	HR	0.236	0.466	0.597	0.682	0.744	0.265	0.508	0.642	0.720	0.784
	NDCG	0.236	0.368	0.423	0.452	0.468	0.265	0.402	0.454	0.478	0.497
CF-UIcA	HR	0.261	0.501	0.640	0.723	0.784	0.235	0.449	0.576	0.659	0.731
	NDCG	0.261	0.390	0.447	0.478	0.500	0.235	0.360	0.412	0.440	0.463
ST-GCN	HR	0.231	0.474	0.614	0.704	0.766	0.216	0.445	0.580	0.669	0.744
	NDCG	0.231	0.369	0.426	0.458	0.478	0.216	0.347	0.400	0.430	0.454
NMTR	HR	0.203	0.459	0.608	0.700	0.767	0.214	0.466	0.610	0.700	0.762
	NDCG	0.203	0.352	0.412	0.445	0.465	0.214	0.360	0.419	0.450	0.469
MATN	HR	**0.296**	**0.560**	**0.693**	**0.771**	**0.828**	**0.279**	**0.529**	**0.659**	**0.741**	**0.798**
	NDCG	**0.296**	**0.447**	**0.500**	**0.529**	**0.545**	**0.279**	**0.423**	**0.477**	**0.507**	**0.524**

4.3 Model Ablation Study (RQ2)

Furthermore, we conduct ablation experiments over a several key components of *MATN* to better understand the component-specific effects. Particularly, we introduce the following model variants:

- **Effect of Multi-Behavior Transformer Network**: *MATN*-T. We do not utilize the multi-behavior transformer network to capture mutual relations between different types of behavior.
- **Effect of Memory Attention Mechanism**: *MATN*-M. We remove the memory-augmented attention network in the joint *MATN* model to encode behavior type-specific semantics.
- **Effect of Gating Mechanism**: *MATN*-G. We replace the designed gating mechanism with the simplified average pooling operation over all behavior type-specific representations in the behavioral pattern aggregation layer.

Figure 2 presents the model ablation study results. We summary the following findings (*MATN* is the default model version).

(1) The incorporation of mutual dependencies between different types of interaction behavior over all items, is capable of boosting the performance substantially. It demonstrates the rationality of our multi-head self-attention architecture in learning explicit pair-wise relations between different behavior types.

(2) *MATN* is consistently superior to *MATN*-M, which hence illustrates the importance of considering context and semantics of individual type of behavior in profiling user preferences.

(3) The replacement of our gating mechanism (*MATN*) with average pooling operation (*MATN*-G), degrades the model's performance. It make sense since *MATN*-G fails to model the different importance across different types of behavior in making final recommendations.

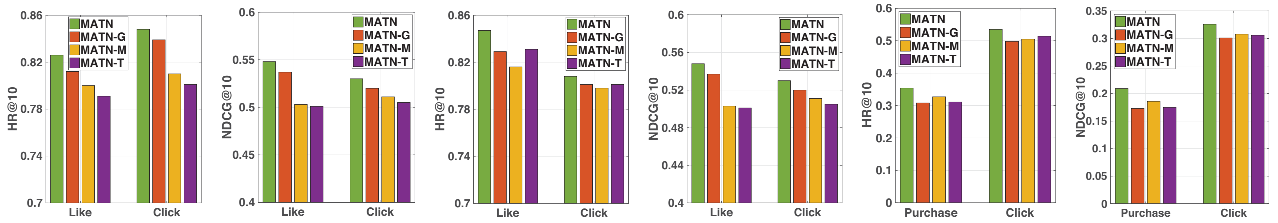
4.4 Impact Studies of Multi-Behavior Relation Integration (RQ3)

To investigate whether exploiting multi-type interaction behavior helps to achieve better performance, we further perform ablation experiments for the purchase prediction task on E-Commerce data, to show the effect of incorporating different types of user-item interactions in our *MATN* with four model variants: $MATN_F$−without the *add-to-favorite* behavior; $MATN_C$−without the *add-to-cart* behavior; and $MATN_P$−without the *page view* behavior. Furthermore, we design another variant by removing all other types of interactions and only contain the purchase behavior $MATN_B$.

Figure 3 shows the evaluation results of different variants under varying top-k settings. We summarize the following findings:
(1) *MATN* using all types of interaction behaviors consistently outperforms other variants with varying top-k settings, except for one exception on top-1 prediction with minor performance defect. The results validate that our *MATN* improve purchase forecasting through integrating multi-behavior relations.
(2) $MATN_B$ using only purchase data yields worst performance, which shows the positive contribution of the three additional behavior types (*i.e.* page view, add-to-favorite and add-to-cart) in helping with user modeling in the e-commerce scenario.
(3) Among the three variants that utilize two additional behavior types (*i.e.* $MATN_F$, $MATN_C$, $MATN_P$), $MATN_P$ clearly shows more severe performance degradation compared to the other two. This sheds light on the higher importance and effectiveness of utilizing page view data in *MATN* and online shopping recommendation.

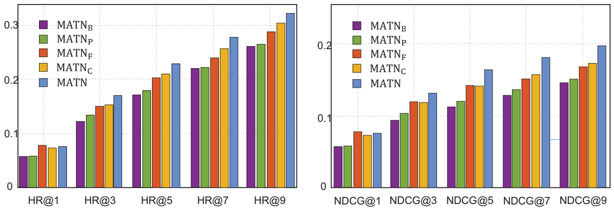
4.5 Hyperparameter Study of *MATN* (RQ4)

In our experiments, we also investigate the impact of different hyperparameter settings in our developed *MATN* framework. Specifically, we evaluate the model recommendation performance by varying the values of several key hyperparameters, including the hidden state dimensionality d, the memory dimension M in our memory attention network, and the number of neural network layers N in our deep feature extraction module. The evaluation



(a) Yelp-HR@10 (b) Yelp-NDCG@10 (c) MovieLens-HR@10 (d) MovieLens-NDCG@10 (e) E-Commerce-HR@10 (f) E-Commerce-NDCG@10

Figure 2: Model ablation study of *MATN* on Yelp, MovieLens and E-Commerce data in terms of HR@K and NDCG@K.



(a) HR@k (b) NDCG@k

Figure 3: Impact study of multi-behavior relation integration on purchase prediction of E-Commerce dataset.

results on the Yelp data in predicting both click and like behavior are shown in Figure 4. The major findings are summarized as below:

- **Hidden State Dimensionality** d. We can observe that when we increase d from 4 to 16, the recommendation performance becomes better, but the further increase the d value (≥ 32) may not be helpful for the model prediction accuracy. The potential reason for this observation is that the large number of latent units could bring a stronger representation capability.
- **Memory Dimension** M. Our memory attention network enables the behavior type-specific semantics learning could be performed from M different dimensions. The parameter study results on memory dimension M indicates that performing the transformation with more latent learning sub-spaces will benefit the recommendation at the early stage, but the continuous increase of M will lead to the overfitting issue.
- **Feature Extraction Network Depth** N. We further examine whether designing a deep feature extraction network is beneficial to the recommendation task. As we can see, stacking two hidden layers is beneficial to the performance, which is attributed to the high non-linearities brought by more non-linear layers. However, the overfitting phenomenon can be observe when we perform more transformation-based feature interaction operation with more hidden layers (≥ 3).

4.6 Case Study on Model Interpretation (RQ5)

In this subsection, we perform qualitative analyses to show the model interpretation of *MATN* in comprehending user behavior relationships and generate more convincing recommendation. To be specific, we visualize the learned quantitative weights learned by our multi-head self-attention mechanism, memory-augmented attention network and multiplex relation aggregation layer. Four typical cases (*i.e.*, samp$_1$,...,samp$_4$) are sampled from the prediction of overall click behavior and purchase behavior on the E-Commerce dataset. From the visualization results, we have the following observations:

(1) *page view* and *purchase* behavior could provide more informative signals in predicting the *click* and *purchase*, respectively. This make sense since the same type of behavior may share closer relationships than other behavior types. (2) In the head-specific self-attention layer, the 4 × 4 behavior relevance matrix indicates across four types of user behavior. An interesting observation is that: add-to-favorite activity is more like to be correlated with page view and purchase, than add-to-favorite. Similar results can be observed for add-to-cart. It might indicate that add-to-favorite has a high co-occurrence probability than others in the real-world e-commerce platform. (3) The memory attention could learn weights in an adaptive way which corresponds to the importance score generated by our gating mechanism. The reason lies in the utilization of ReLU activation in the attention calculation instead of the mandatory restriction with Softmax function. Overall, all above observations demonstrate the model interpretation power of *MATN* in capturing complex behavior relations from different perspective.

4.7 Scalability Study of *MATN* (RQ6)

In addition to recommendation accuracy, the model efficiency is also an important factor to investigate. In this subsection, we evaluate the computation time cost of our *MATN* as compared to other baselines. In Table 5, we report the running time of each epoch during the training phase of each compared approach. We can observe that our *MATN* model could achieve comparable performance when competing with most baselines, especially in dealing with the large-scale user-item interaction data.

Although we lose in the cases when comparing with some of the competitive baselines–learning user-item interaction representations with simple relation encoders (*e.g.*, Multilayer Perceptron, vanilla autoencoder), our *MATN* still exhibits competitive model scalability due to the comparable time complexity. However, *MATN* can show obvious performance superiority over these techniques. In addition, the performance gap (measured by running time) between *MATN* and graph neural network recommendation methods (*i.e.*, ST-GCN and NGCF), may stem from the high computational cost of graph convolution operation when performing information aggregation and propagation.

5 RELATED WORK

Deep Collaborative Filtering Techniques. Deep learning have been revolutionizing collaborative filtering techniques and achieve promising results in many recommendation scenarios. For example,

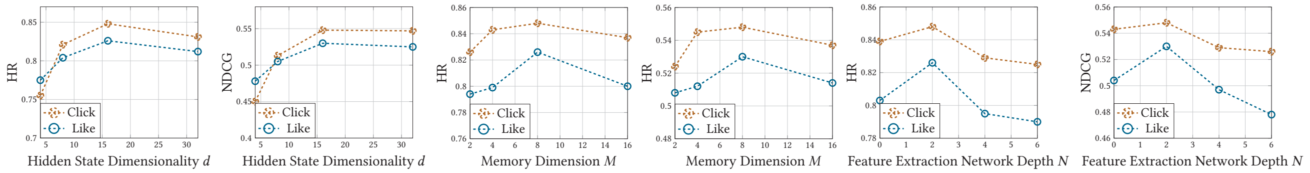


Figure 4: Hyper-parameter study in terms of *HR@10* and *NDCG@10*

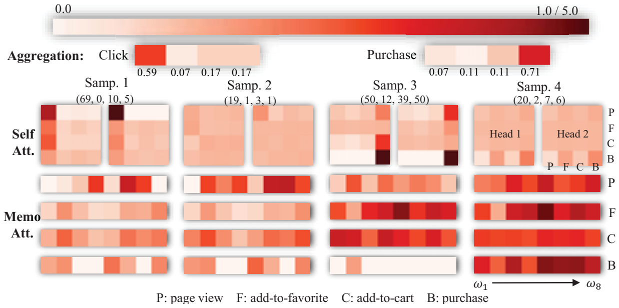


Figure 5: Case study of the learned quantitative weights from key modules in *MATN*. Pair-wise relations between four types of behavior (*e.g.*, P, F, C and B) are represented with a 4×4 weight matrix in the multi-head self-attention layer. ω_1,...,ω_8 indicate the learned weights across 8 memory dimensions in the memory attention network. The four relevance scores encoded by gating mechanism corresponds to four behavior types. Tuples (*e.g.*, <50,12,39,50>) are numbers of behaviors in the order of (P, F, C, B).

Table 5: Computational time cost (seconds) investigation.

Models	Yelp	MovieLens	E-Commerce
BiasMF	34	34	241
DMF	48	59	231
NCF-N	33	35	247
AutoRec	32	37	228
CDAE	33	37	228
CF-NADE	32	37	229
CF-UIcA	51	63	476
ST-GCN	55	65	491
NGCF	61	70	503
NMTR	33	35	268
DIPN	99	134	1062
MATN	40	49	234

Multi-layer Perceptron has been integrated into the collaborative filtering architecture to handle non-linear feature interactions [11, 47]. Several work attempts to utilize encoder-decoder network to map explicit user-item interactions into latent representations, using autoencoder [27] and its variants [29]. In addition, another research line lie in leveraging graph neural network to incorporate user-item graph signals into the recommendation framework, such as NGCF [41], STAR-GCN [49] and Multi-GCCF [31]. The major difference between these methods and ours is that MATN explores the cross-behavior interactive knowledge to assist recommendation.

Relation-aware Recommender Systems. Prior work has made significant advances to develop recommender systems with the consideration of various relations between users and items. For example, the social-aware recommender systems aim to boost recommendation performance by exploring user's social relations based on the information dissemination [3, 6]. Furthermore, knowledge graph has become another information source from item side to help recommendation models capture relationships between items [37, 40]. In addition to the relation of collaborative similarity, there exist work aiming to consider multiple item relationships (*e.g.*, shared director or categories) to learn fine-grained item knowledge [46]. Different from these methods which focus on using the exogenous information from either user or item side, this work explores the multiplicity of pairwise user-item interactions and carefully learns their underlying inter-dependencies.

Attention Network for Recommendations. Attention mechanism has been proven to be effective in differentiating various relations for recommendations [34], such as item transitions [17], user connections [28] and customer group dynamics [36]. To address the limitation of recurrent neural architectures in capturing long-range dependencies without the rigid order assumption, self-attention mechanism has been introduced to model correlations from any pair of positions of input data points [35]. For example, Sun *et al.* [30] proposed a bidirectional self-attention framework for sequential recommendation. Additionally, multi-head self-attentive model is introduced to recommended news to users [43]. Our MATN framework is motivated by the multi-head self-attentive learning architecture in a sense that a memory augmented transformer is designed to model multiplex behavior relation dynamics from different types of user-item interactions.

6 CONCLUSION

In this work, we propose MATN, a novel memory augmented transformer neural architecture which incorporates multiple types of user behavior relationships into a cross-behavior collaborative filtering framework. We argue that these different types of user-item interactions are usually neglected in conventional methods. MATN demonstrates the state-of-the-art performance on two benchmark datasets and a large-scale user behavior data from a major online retailing platform. In addition, via the qualitative analysis of the attentive weights, we discover that the implicit cross-type behavioral dependencies are encoded within the MATN framework.

Notwithstanding the interesting problem and promising results, some directions exist for future work. We will next incorporate rich auxiliary data source (*e.g.*, user review text information or item description [52]) to further enhance the current recommendation framework. Additionally, another time dimension of the problem deserves more investigation. When multi-type user-item interaction data arrives in a timely manner, how to best account for it in the current MATN framework? One possible direction is adapting MATN to a time-sensitive model by analyzing the trade-off between accuracy and complexity.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive feedback. This work was supported by National Nature Science Foundation of China (61672241, U1611461), Major Project of National Social Science Foundation of China (18ZDA062), Natural Science Foundation of Guangdong Province (2016A030308013), Science and Technology Program of Guangdong Province (2019A050510010), and Fundamental Research Funds for the Central Universities (x2js-D2192830).

REFERENCES

[1] Kristen M Altenburger and Daniel E Ho. 2019. Is Yelp Actually Cleaning Up the Restaurant Industry? A Re-Analysis on the Relative Usefulness of Consumer Reviews. In *WWW*. 2543–2550.

[2] Yukuo Cen, Xu Zou, Jianwei Zhang, Hongxia Yang, Jingren Zhou, and Jie Tang. 2019. Representation learning for attributed multiplex heterogeneous network. In *KDD*. 1358–1368.

[3] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2019. Social attentional memory network: Modeling aspect-and friend-level differences in recommendation. In *WSDM*. 177–185.

[4] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential recommendation with user memory networks. In *WSDM*. ACM, 108–116.

[5] Chao Du, Chongxuan Li, Yin Zheng, Jun Zhu, and Bo Zhang. 2018. Collaborative filtering with user-item co-autoregressive models. In *AAAI*.

[6] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph Neural Networks for Social Recommendation. In *WWW*. ACM, 417–426.

[7] Chen Gao, Xiangnan He, Dahua Gan, Xiangning Chen, Fuli Feng, Yong Li, Tat-Seng Chua, and Depeng Jin. 2019. Neural multi-task recommendation from multi-behavior data. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 1554–1557.

[8] Long Guo, Lifeng Hua, Rongfei Jia, Binqiang Zhao, Xiaobo Wang, and Bin Cui. 2019. Buying or Browsing?: Predicting Real-time Purchasing Intent using Attention-based Deep Network with Multiple Behavior. In *KDD*. 1984–1992.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.

[10] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *SIGIR*. ACM, 355–364.

[11] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. 173–182.

[12] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *SIGIR*. 549–558.

[13] Chao Huang, Xian Wu, Xuchao Zhang, Chuxu Zhang, Jiashu Zhao, Dawei Yin, and Nitesh V Chawla. 2019. Online Purchase Prediction via Multi-Scale Modeling of Behavior Dynamics. In *KDD*. 2613–2622.

[14] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *ICDM*. IEEE, 197–206.

[15] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.

[16] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *CIKM*. 2615–2623.

[17] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *CIKM*. 1419–1428.

[18] Daryl Lim, Julian McAuley, and Gert Lanckriet. 2015. Top-n recommendation with missing implicit feedback. In *Recsys*. 309–312.

[19] Chenghao Liu, Tao Lu, Xin Wang, Zhiyong Cheng, Jianling Sun, and Steven CH Hoi. 2019. Compositional Coding for Collaborative Filtering. In *SIGIR*. 145–154.

[20] Babak Loni, Roberto Pagano, Martha Larson, and Alan Hanjalic. 2019. Top-n recommendation with multi-channel positive feedback using factorization machines. *Transactions on Information Systems (TOIS)* 37, 2 (2019), 1–23.

[21] Chen Ma, Liheng Ma, Yingxue Zhang, Jianing Sun, Xue Liu, and Mark Coates. 2020. Memory Augmented Graph Neural Networks for Sequential Recommendation. In *AAAI*.

[22] Andriy Mnih and *et al.* 2008. Probabilistic matrix factorization. In *NIPS*. 1257–1264.

[23] Athanasios N Nikolakopoulos and George Karypis. 2019. Recwalk: Nearly uncoupled random walks for top-n recommendation. In *WSDM*. 150–158.

[24] Vito Claudio Ostuni, Tommaso Di Noia, Eugenio Di Sciascio, and Roberto Mirizzi. 2013. Top-n recommendations from implicit feedback leveraging linked open data. In *Recsys*. 85–92.

[25] Jiarui Qin, Kan Ren, Yuchen Fang, Weinan Zhang, and Yong Yu. 2020. Sequential Recommendation with Dual Side Neighbor-based Collaborative Relation Modeling. In *WSDM*. 465–473.

[26] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*.

[27] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. Autorec: Autoencoders meet collaborative filtering. In *WWW*. ACM, 111–112.

[28] Weiping Song, Zhiping Xiao, Yifan Wang, Laurent Charlin, Ming Zhang, and Jian Tang. 2019. Session-based social recommendation via dynamic graph attention networks. In *WSDM*. 555–563.

[29] Florian Strub, Romaric Gaudel, and Jérémie Mary. 2016. Hybrid recommender system based on autoencoders. In *DLRS*. 11–16.

[30] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *CIKM*. 1441–1450.

[31] Jianing Sun, Yingxue Zhang, Chen Ma, Mark Coates, Huifeng Guo, Ruiming Tang, and Xiuqiang He. 2019. Multi-Graph Convolution Collaborative Filtering. In *ICDM*. IEEE, 1306–1311.

[32] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *WSDM*. 565–573.

[33] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Latent relational metric learning via memory-based attention for collaborative ranking. In *WWW*. 729–739.

[34] Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. Multi-pointer co-attention networks for recommendation. In *KDD*. 2309–2318.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*. 5998–6008.

[36] Lucas Vinh Tran, Tuan-Anh Nguyen Pham, Yi Tay, Yiding Liu, Gao Cong, and Xiaoli Li. 2019. Interact and decide: Medley of sub-attention networks for effective group recommendation. In *SIGIR*. 255–264.

[37] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, et al. 2019. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *KDD*. 968–977.

[38] Jianling Wang, Raphael Louca, Diane Hu, Caitlin Cellier, James Caverlee, and Liangjie Hong. 2020. Time to Shop for Valentine's Day: Shopping Occasions and Sequential Recommendation in E-commerce. In *WSDM*. 645–653.

[39] Weixun Wang, Junqi Jin, Jianye Hao, Chunjie Chen, Chuan Yu, Weinan Zhang, Jun Wang, Xiaotian Hao, Yixi Wang, Han Li, et al. 2019. Learning Adaptive Display Exposure for Real-Time Advertising. In *CIKM*. 2595–2603.

[40] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *KDD*. 950–958.

[41] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *SIGIR*.

[42] Hongfa Wen, Xin Liu, Chenggang Yan, Linhua Jiang, Yaoqi Sun, Jiyong Zhang, and Haibing Yin. 2019. Leveraging Multiple Implicit Feedback for Personalized Recommendation with Neural Network. In *AIAM*. 1–6.

[43] Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019. Neural News Recommendation with Multi-Head Self-Attention. In *EMNLP*. 6390–6395.

[44] Xian Wu, Baoxu Shi, Yuxiao Dong, Chao Huang, and Nitesh V Chawla. 2019. Neural tensor factorization for temporal interaction learning. In *WSDM*. 537–545.

[45] Yao Wu, Christopher DuBois, Alice X Zheng, et al. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In *WSDM*. ACM, 153–162.

[46] Xin Xin, Xiangnan He, Yongfeng Zhang, Yongdong Zhang, and Joemon Jose. 2019. Relational Collaborative Filtering: Modeling Multiple Item Relations for Recommendation. In *SIGIR*.

[47] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, et al. 2017. Deep Matrix Factorization Models for Recommender Systems.. In *IJCAI*. 3203–3209.

[48] Seongjun Yun, Raehyun Kim, Miyoung Ko, and Jaewoo Kang. 2019. SAIN: Self-Attentive Integration Network for Recommendation. In *SIGIR*. 1205–1208.

[49] Jiani Zhang, Xingjian Shi, Shenglin Zhao, and Irwin King. 2019. STAR-GCN: Stacked and Reconstructed Graph Convolutional Networks for Recommender Systems. *IJCAI* (2019).

[50] Shuai Zhang, Yi Tay, Lina Yao, Aixin Sun, and Jake An. 2019. Next item recommendation with self-attentive metric learning. In *AAAI*, Vol. 9.

[51] Lei Zheng, Chaozhuo Li, Chun-Ta Lu, Jiawei Zhang, and Philip S Yu. 2019. Deep Distribution Network: Addressing the Data Sparsity Issue for Top-N Recommendation. In *SIGIR*. 1081–1084.

[52] Lei Zheng, Vahid Noroozi, and Philip S Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In *WSDM*. 425–434.

[53] Yin Zheng, Bangsheng Tang, Wenkui Ding, and Hanning Zhou. 2016. A neural autoregressive approach to collaborative filtering. In *ICML*.

[54] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *KDD*. 1059–1068.