A APPENDIX

A.1 THEORETICAL DISCUSSION OF EGCM MODEL

In this section, we conduct in-depth analysis for our multi-behavior contrastive learning paradigm. Specifically, we discuss the benefits brought by the augmented behavior-aware self-supervised learning tasks from two dimensions: i) Cross-behavior mutual information maximization; ii) User interest representation discrimination. The detailed theoretical discussion is presented as follows:

A.1.1 CROSS-BEHAVIOR MUTUAL INFORMATION MAXIMIZATION

In our multi-behavior contrastive learning framework, we propose to capture the multi-behavior commonality, by maximizing the mutual information between type-specific behavior embedding ($\mathbf{e}_i^b \in \mathbb{R}^{d\times 1}$) and multi-behavior representation ($\mathbf{e}_i \in \mathbb{R}^{d\times 1}$). In particular, we define our information maximization function as $I(\mathbf{z}_i, \mathbf{z}_p)$, where $\mathbf{z}_i, \mathbf{z}_p$ represents the anchor and positive instance in the same hypersphere, respectively. Without loss of generality, the embedding $\mathbf{z}$ is formally defined as: $\mathbf{z} = \mathbf{e}/\parallel \mathbf{e} \parallel$. In our multi-behavior contrasting scenario, the mutual information estimation is performed between the fused representation $\overline{\mathbf{E}}_u$ and behavior-specific embdding in $\{\mathbf{E}_u^1, \mathbf{E}_u^2, ..., \mathbf{E}_u^b\}$. Such contrastive-based self-supervision signals are integrated into the BPR loss function to enhance the robustness of user representation paradigm.

Motivated by the research in Van den Oord et al. (2018); Hjelm et al. (2018); Bachman et al. (2019); Tian et al. (2020a), our augmented contrastive loss can form the lower bound of the information maximization function $I(\cdot)$ as:

$$I(\mathbf{z}_i, \mathbf{z}_p) \geq log(k) - \mathcal{L}_{cl} \tag{11}$$

where k is the number of negative samples. The inequality suggests that smaller $\mathcal{L}_{cl}$ results in larger $I(\cdot)$. In other words, minimizing $\mathcal{L}_{cl}$ is equivalent to maximizing the lower bound of mutual information in $I(\cdot)$.

The above situation can be extended to the multi-behavior modeling process. In addition to the equivalence between the contrastive objective and lower bound of information maximization, the self-supervised $\mathcal{L}_{cl}$ is closely related to function $I(\cdot)$. It can provide a theoretical basis for our multi-behavior recommender system as follows:

$$h_\theta^*(\mathbf{z}_i, \mathbf{z}_p) \propto \frac{p(\mathbf{z}_i, \mathbf{z}_p)}{p(\mathbf{z}_i)p(\mathbf{z}_p)} \propto \frac{p(\mathbf{z}_i|\mathbf{z}_p)}{p(\mathbf{z}_i)} \tag{12}$$

where h^* is the optimal point of $h_\theta = \exp(s(\mathbf{e}_u^b, \overline{\mathbf{e}}_u)/\tau)$ that is proportional to the density ratio between the joint distribution $p(\mathbf{z}_i, \mathbf{z}_p)$ and the product of marginals $p(\mathbf{z}_i)p(\mathbf{z}_p)$. This quantity is the point-wise mutual information, and the extended multi-behavior form can be implemented by optimizing the sum of a set of pair-wise objectives Tian et al. (2020a).

A.1.2 USER INTEREST REPRESENTATION DISCRIMINATION

Contrastive loss is a hardness-aware loss function Wang & Liu (2021); Wu et al. (2021). It can push away the hard negative samples a lot from the anchor by giving them greater gradients under contrastive training framework. This property is beneficial to our multi-behavior graph neural architecture. One of the most important challenges in existing GNN architecture lies in how to achieve a nice trade-off between high-order connectivity modeling and over-smoothing issue Li et al. (2018). Stacking more graph-based information propagation layers is more like to involve over-smoothing issue for encoding collaborative effects. Hence, enhance the discrimination ability of user interest representation paradigm is necessary and challenging for recommender system. To tackle the above challenge, our multi-behavior contrastive learning framework will assign larger gradients to hard negative samples so as to enhance the discrimination of user representations.

Embedding Normalization. To map the embeddings with arbitrary value distributions into the same hyperspace, we perform the embedding normalization as $\mathbf{z}_i = \mathbf{e}_i/\parallel \mathbf{e}_i \parallel$, where $\mathbf{e}_i$ denote the output *prior* to normalization. The gradient of the loss with respect to $\mathbf{e}_i$ is related to that with respect to $\mathbf{z}_i$ via the chain rule presented as follows:

$$\frac{\partial \mathcal{L}_i(\mathbf{z}_i)}{\partial \mathbf{e}_i} = \frac{\partial \mathcal{L}_i(\mathbf{z}_i)}{\partial \mathbf{z}_i}\frac{\partial \mathbf{z}_i}{\partial \mathbf{e}_i} \tag{13}$$

$$\frac{\partial \mathbf{z}_i}{\partial \mathbf{e}_i} = \frac{\partial}{\partial \mathbf{e}_i}\left(\frac{\mathbf{e}_i}{\parallel \mathbf{e}_i \parallel}\right) = \frac{1}{\parallel \mathbf{e}_i \parallel} I - \mathbf{e}_i \left(\frac{\partial(1/\parallel \mathbf{e}_i \parallel)}{\partial \mathbf{e}_i}\right)^T$$
$$= \frac{1}{\parallel \mathbf{e}_i \parallel}\left(I - \frac{\mathbf{e}_i \mathbf{e}_i^T}{\parallel \mathbf{e}_i \parallel^2}\right) = \frac{1}{\parallel \mathbf{e}_i \parallel}\left(I - \mathbf{z}_i \mathbf{z}_i^T\right) \tag{14}$$

Gradients of Negative Pairs. We use the loss function $\mathcal{L}_i$ to calculate the partial derivative of the anchor point to analyze the influence of different samples over the gradients:

$$\frac{\partial L_i}{\partial \mathbf{z}_i} = \frac{\partial}{\partial \mathbf{z}_i}\left(-log\frac{\exp\left(\mathbf{z}_i \cdot \mathbf{z}_P/\tau\right)}{\sum_{\mathcal{V}_A}\exp\left(\mathbf{z}_i \cdot \mathbf{z}_A/\tau\right)}\right) = -\frac{1}{\tau}\cdot \mathbf{z}_P + \frac{1}{\tau}\frac{\sum_{\mathcal{V}_A}\mathbf{z}_A \cdot \exp\left(\mathbf{z}_i \cdot \mathbf{z}_A/\tau\right)}{\sum_{\mathcal{V}_A}\exp\left(\mathbf{z}_i \cdot \mathbf{z}_A/\tau\right)}$$
$$= \frac{1}{\tau}\left(\frac{\sum_{\mathcal{V}_N}\mathbf{z}_N \cdot \exp\left(\mathbf{z}_i \cdot \mathbf{z}_N/\tau\right)}{\sum_{\mathcal{V}_A}\exp\left(\mathbf{z}_i \cdot \mathbf{z}_A/\tau\right)} + \frac{\mathbf{z}_P \cdot \left(\exp\left(\mathbf{z}_i \cdot \mathbf{z}_P/\tau\right) - \sum_{\mathcal{V}_A}\exp\left(\mathbf{z}_i \cdot \mathbf{z}_A/\tau\right)\right)}{\sum_{\mathcal{V}_A}\exp\left(\mathbf{z}_i \cdot \mathbf{z}_A/\tau\right)}\right) \tag{15}$$

The two terms of the formula represent the gradient of positive samples and negative samples respectively. Here, $\mathbf{z}_N$ represents the instance from the set of negative samples and $\mathbf{z}_P, \mathbf{z}_A$ are from the positive sample set and the entire set. We then mainly focus on the negative part, which can be :

$$\frac{1}{\tau}\cdot\frac{\sum_{\mathcal{V}_N}\mathbf{z}_N \cdot \exp\left(\mathbf{z}_i \cdot \mathbf{z}_N/\tau\right)}{\sum_{\mathcal{V}_A}\exp\left(\mathbf{z}_i \cdot \mathbf{z}_A/\tau\right)} = \frac{1}{\tau\cdot \parallel \mathbf{e}_i \parallel}\cdot\frac{\sum_{\mathcal{V}_N}\left(\mathbf{z}_N - \left(\mathbf{z}_i \cdot \mathbf{z}_N\right)\cdot \mathbf{z}_i\right)\cdot \exp\left(\mathbf{z}_i \cdot \mathbf{z}_N/\tau\right)}{\sum_{\mathcal{V}_A}\exp\left(\mathbf{z}_i \cdot \mathbf{z}_A/\tau\right)} \tag{16}$$

Temperature and Gradient. The proportional term of the norm of the gradient of each term in the sum formula is as follow:

$$\parallel \mathbf{z}_N - \left(\mathbf{z}_i \cdot \mathbf{z}_N\right)\cdot \mathbf{z}_i \parallel \left|\frac{\exp\left(\mathbf{z}_i \cdot \mathbf{z}_N/\tau\right)}{\sum_{\mathcal{V}_A}\exp\left(\mathbf{z}_i \cdot \mathbf{z}_A/\tau\right)}\right|$$
$$\propto \sqrt{1-\left(\mathbf{z}_i \cdot \mathbf{z}_N\right)^2}\cdot \exp\left(\mathbf{z}_i \cdot \mathbf{z}_N/\tau\right) \tag{17}$$

As z_i and z_N are both unit vectors, we introduce another variable x with the definition of $x = \mathbf{z}_i \cdot \mathbf{z}_N \in [-1, 1]$ to abbreviate the final result of Eq. 17:

$$c(x) \propto \sqrt{1-(x)^2}\cdot \exp\left(x/\tau\right) \tag{18}$$

where $c(x)$ is the relationship function of the gradient from the negative samples. We plot the function in Equation 18 in Figure A.1.2. The independent variable is similarity x and the dependent variable is proportional to the negative sample gradient. With the increase of x, the gradient of negative samples will increase. Moreover, as the temperature coefficient τ (tau) decreases, the gradient of negative samples given by contrastive learning will also increase significantly.

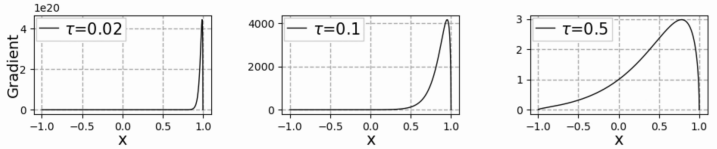


Figure 3: Gradient function $c(x)$ in Eq. 18 when τ = 0.02, τ = 0.1 and τ = 0.5. x is the similarity between positive and negative instances. This demonstrates that the gradient increases with decreasing temperature coefficient τ.

In our recommended scenario, hard negative samples $\mathbf{z}_{N-hard}$ represents the users other than the anchor user. If $\mathbf{z}_{N-hard}$ is very close to the anchor point $\mathbf{z}_i$, the value of x of the hard negative

samples approaches 1, which results in more indistinguishable user representations. The contrastive loss gives a larger gradient, and as relative negative samples, the pairs will be pushed farther away from each other. In this way, EGCM enhance the user representations with multi-behavior diversity. Therefore, the experiments in Sec. 4.4 shows that as the temperature τ decreases which, the more distinguishable the user representation is, which brings the better effect. However, when the gradient is over large, gradient explosion will be observed.

A.2 Alleviating the Data Sparsity Issue.

Our above theoretical discussion analyzes the benefits of our multi-behavior contrastive learning paradigm in capturing the multi-behavior commonality and diversity. To be specific, the mutual information maximization between positive samples is helpful to preserve the common characteristics among different types of behaviors. Additionally, contrastive learning with negative samples can push the hard negative samples away to get distinguishable user embedding, so as to encode the behavior diversity of different users and alleviate the over-smoothing problem of our graph neural model.

Therefore, contrastive learning can improve the quality of representation and alleviate the problem caused by the scarcity of data. To this end, we selected users whose interaction number on the IJCAI dataset in {<5, <15, <35, <60} for training and testing. In order to eliminate the influence of training data and simulate a real sparse data scenario, we carry out the evaluation on our model under the setting of w/o-JBL, and compare it with two best-performed baselines (HyperRec and KHGT) from the lines of sequence-based models and multi-behavior recommender systems, respectively.

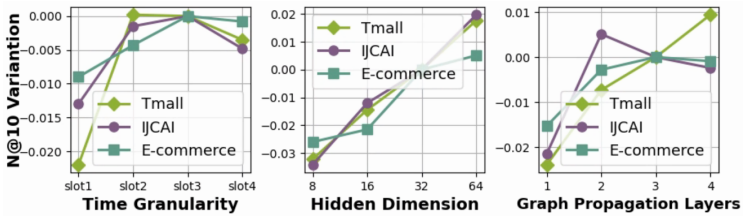
Table 4: Performance with different number of interactions in terms of HR@5@20 & NDCG@5@20.

		<5		<15		<35		<60	
		@5	@20	@5	@20	@5	@20	@5	@20
HyperRec	HR	0.0934	0.3052	0.1309	0.3465	0.1616	0.4017	0.3182	0.5909
	NDCG	0.0728	0.1108	0.0804	0.1401	0.0930	0.1655	0.2073	0.2872
KHGT	HR	0.1136	0.3371	0.1763	0.4174	0.2198	0.4702	0.4091	0.6364
	NDCG	0.0753	0.1367	0.1161	0.1715	0.1463	0.2101	0.2374	0.3171
w/o-cl	HR	0.1326	0.3674	0.1856	0.4153	0.2271	0.4920	0.4091	0.7727
	NDCG	0.0824	0.1482	0.1238	0.1883	0.1514	0.2256	0.2911	0.3895
w/o-JBL	HR	0.1806	0.4268	0.2168	0.4579	0.2547	0.5560	0.3636	0.7727
	NDCG	0.1174	0.1865	0.1460	0.2138	0.1786	0.26210	0.2010	0.3150

As shown in Table 4, it can be observed that under different sparsity degrees of user interaction data, w/o-JBL with contrastive learning task will get better results than w/o-CL. Moreover, the performance gap between our contrastive learning method and other baselines become larger with the higher sparsity degrees of interaction data, which again justfies the effectiveness of our EGCM in addressing the data scarcity for recommender system.

A.3 Hyperparameter Sensitivity Analysis

To explore the sensitivity of our proposed EGCM framework, we perform experiments to evaluate the influence of key hyperparameters, including short-term time granularity, # of graph propagation layers L, the regularization strength of contrastive learning. Figure A.3 presents the evaluation results.



Short-Term Time Granularity. To construct the short-term multi-behavior graphs, we tune the parameter of the time granularity from different time ranges due to the time span of different

experimental datasets. In particular, the time granularity of Tmall, IJCAI, E-commerce dataset is selected from {1,3,5,9} days, {1,3,6,12} months, {1,2,4,12} weeks, respectively. From the evaluation results in Figure A.3, we can observe that shorter time period (with the higher time granularity) will leads to the overfitting problem in modeling the short-term behavior-aware user preference.

Representation Hidden Dimensionality. To investigate the effects of hidden dimensionality in our representation performance, the embedding size is chosen from the range of {8, 16, 32, 64} for parameter sensitivity evaluation. It can be seen that the results on Tmall and IJCAI datasets improves significantly with the increase of dimension size, due to the stronger representation power of larger embedding size.

# of Graph Propagation Layers. To capture the short-term multi-behavior user interests, we design the behavior-aware message passing scheme to refine user/item embeddings with the injection of multi-typed behavior context. We select the number of graph propagation layers from the range of {1, 2, 3, 4} to explore the effect of model depth. We can observe that deeper graph models may bring benefits to model the high-order collaborative effects on Tmall and E-commerce datasets. By stacking more embedding propagation layers may involve noise in representation refinement on E-commerce dataset.

A.4 Model Complexity Analysis

The main time consumption in our EGCM framework are from several key components: i) Short-term multi-behavior graph encoder: the computational cost of our graph neural architecture for item representation is $O(|\mathcal{B}| \times \Lambda \times L \times |\mathcal{E}^b_{t,M^TM}| \times d)$ for performing message passing across graph layers. Then for user, it is $O(|\mathcal{B}| \times \Lambda \times |\mathcal{E}^b_{t,M}| \times d)$. $|\mathcal{E}^b_{t,M^TM}|$, $|\mathcal{E}^b_{t,M}|$ respectively represents the number of non-zero elements in the incidence matrix $\Gamma(\mathbf{M}_t^{bT})$ and $\Gamma(\mathbf{M}_t^{bT}\mathbf{M}_t^b)$, under the behavior type of b during the time slot t. Here, L denotes the number of graph propagation layers of item. The operations of concatenation and linear transformations for layer aggregation takes $O(|\mathcal{B}| \times \Lambda \times L \times |\mathcal{V}^b_i| \times d)$. ii) Dynamic cross-relational memory network: The most computational cost for the cross-relational memory component comes from the self-attention operation with the time complexity of $O(\Lambda \times |\mathcal{V}^b| \times |\mathcal{B}|^2 \times d)$ quadratic with the behavior number $|\mathcal{B}|$. iii) Multi-behavior contrastive learning: The cost of InfoNCE-based mutual information calculation is $O(d)$ and $O(batch \times d)$ for the numerator and denominator (in Eq.11), respectively. Thus, our multi-behavior contrastive learning paradigm takes $O(\Lambda \times |\mathcal{B}| \times |\mathcal{E}^b| \times d)$ per epoch. Given the smaller values of L, $|\mathcal{B}|$ and Λ, our EGCM model can achieve comparable time complexity as compared to state-of-the-art GNN-based recommendation techniques.

A.5 Asymmetric normalization

To alleviate the large value effects of embeddings during the recursive propagation Wang et al. (2020b), we applied normalization into the message passing on user-item heterogeneous bipartite graph, which is different from the symmetrical graph Laplacian of eigenvectors Kipf & Welling (2016). Specifically, the two diagonal degree matrices $\mathbf{D}_t^b \in \mathbb{R}^{N\times M}$ and $\mathbf{B}_t^b \in \mathbb{R}^{M\times N}$ based on the interaction matrix $\mathbf{M}_t^b$ are generated as follows:

$$D^b_{t,(n,n)} = \sum_{n=1}^{|\mathcal{V}^b_{t,i}|} \mathbf{M}^b_{t,(n)}; \qquad B^b_{t,(m,m)} = \sum_{m=1}^{|\mathcal{V}^b_{t,u}|} (\mathbf{M}^b_{t,(m)})^T \tag{19}$$

$\mathbf{D}_t^b$ is used for messaging from item to user, while $\mathbf{B}_t^b$ is the opposite. And $D^b_{t,(n,n)}$, $B^b_{t,(m,m)}$ denotes the elements of n-th, m-th row of these diagonal matrices. The specific normalization operation will be introduced later in the message pass in the form of $\Gamma(\cdot)$.

A.6 NOTATION AND FRAMEWORK ALGORITHM

Notation	Description
$\mathcal{V}_u, \mathcal{V}_i, \mathcal{B}, \mathcal{E},$	user set, item set, behavior set, interactive edge set
$u \in \mathcal{V}_u$, $i \in \mathcal{V}_i$, $b \in \mathcal{B}$, t	a specific user/item/behavior/time slot
Λ, L, d, h	number of time slot/layer, hidden dim, head num
$\mathbf{M}_t^b, \mathbf{D}_t^b, \mathbf{B}_t^b$	incidence matrix/diagonal matrix of u/i during t under b
$\mathbf{E}_{t,i}^{b,0}, \mathbf{E}_{t,i}^{b,(l)}, \mathbf{E}_{t,i}^b, \mathbf{E}_{t,u}^b$	embeddings of GNNs layer $0/l$, short term u/i during t under b
$\widehat{\mathbf{E}}_t, \widetilde{\mathbf{E}}_t$	short/long term high dimensional multi-behavior embeddings
$\overline{\mathbf{E}}_{t,u}, \overline{\mathbf{E}}_u$	short/long term aggregated embeddings
$\sigma(\cdot), \Gamma(\cdot), \gamma(\cdot)$	PReLU/Laplacian normalized/cross-relational memory function
$\alpha, \beta, \zeta, \tau$	short/long contrastive strength, combined weight, temperature

Algorithm 1: The Learning Process of EGCM Framework

Input: Behavior-aware interaction sequence $S_u = \{(i_1, b_1), (i_2, b_2), ..., (i_{|S_u|}, b_{|S_u|})\}$.
Short-term multi-behavior graph $\mathcal{G}_t^b = (\mathcal{V}_t^b, \mathcal{E}_t^b, \mathbf{M}_t^b)$.

Output: Aggregated long term user/item representations $\overline{\mathbf{E}}_i$, $\overline{\mathbf{E}}_u$. The probability of the most likely next item $i_{|S_u|+1}$.

Initialize: Xavier initialized behavior-specific short term item embeddings $\mathbf{E}_{t,i}^{b,0}$. Parameters: i)Short-term multi-behavior graph encoder $\{\mathbf{W}_{t,i}^{b,(l)}, \mathbf{W}_{t,u}^b, \mathbf{W}_\zeta^b, \mathbf{W}_{t,cat}^b, \zeta\}$. ii)Dynamic cross-relational memory network $\{\mathbf{W}_t^Q, \mathbf{W}_t^K\}$. iii)Memory cross behavior self-attention and behavior fusion attention $\{\mathbf{W}_t^Q, \mathbf{W}_t^K\}$, $\{\mathbf{W}_f\}$.

for $epoch \leftarrow 0, 1, ...$ **do**
  Update learning rate scheduler.
  for $step \leftarrow 0, 1, ...$ **do**
    `// Short-Term Multi-Behavior Graph:`
    for $t \leftarrow 0, 1, ..., |\Lambda|$ **do**
      Get short term embeddings: $\mathbf{E}_{t,u}^b, \mathbf{E}_{t,i}^b \longleftarrow \mathcal{G}_t^b$, Eq.1, Eq.2, Eq.2
      Prepare behavior aggregated embeddings for $\mathcal{L}_{cl}^{short}$: $\overline{\mathbf{E}}_{t,u} \longleftarrow$ Eq.6
    end
    `// Dynamic Cross-Relational Memory:`
    for $t \leftarrow 1, ..., |\Lambda|$ **do**
      Modeling time-evolving cross-type dependencies: $\widetilde{\mathbf{E}}_{t,u}^b, \widetilde{\mathbf{E}}_{t,i}^b \longleftarrow$ Eq.4, Eq.5
      Aggregate $\widetilde{\mathbf{E}}_{t,u}^b, \widetilde{\mathbf{E}}_{t,i}^b$ convey across t and b for $\mathcal{L}_{cl}^{long}$, $\mathcal{L}_{BPR}$: $\overline{\mathbf{E}}_u, \overline{\mathbf{E}}_i \longleftarrow$ Eq.6
    end
    `// Cross-Behavior Contrastive Task & Recommendation Task:`
    Get final multi-task objective $\mathcal{L} \longleftarrow \mathcal{L}_{cl}^b \leftarrow$ Eq.7 + $\mathcal{L}_{BPR} \leftarrow$ Eq.8
    Gradient descent back propagation.
  end
end

A.7 DIMENSIONAL TRANSFORMATION OF THE MEMORY MODULE

Table 5: Dimensional Details of Eq. 4 for Self-attention of Multi-behavior Relation

Dimensional Transformation of the Memory Module	
Parameters	Dimensionality
Input	$(\|\mathcal{B}\| \times \mathrm{N} \times \mathrm{d})$
Q,K,V Transformation	$(\|\mathcal{B}\| \times \mathrm{N} \times \mathrm{d}) \cdot (\mathrm{d} \times \mathrm{d}) \longrightarrow (\|\mathcal{B}\| \times \mathrm{N} \times \mathrm{d})$
Q Extension	$(\|\mathcal{B}\| \times \mathrm{N} \times \mathrm{d}) \longrightarrow (\|\mathcal{B}\| \times 1 \times \mathrm{N} \times \mathrm{d})$
K Extension	$(\|\mathcal{B}\| \times \mathrm{N} \times \mathrm{d}) \longrightarrow (1 \times \|\mathcal{B}\| \times \mathrm{N} \times \mathrm{d})$
V Extension	$(\|\mathcal{B}\| \times \mathrm{N} \times \mathrm{d}) \longrightarrow (1 \times \|\mathcal{B}\| \times \mathrm{N} \times \mathrm{d})$
Self-attention	$(\|\mathcal{B}\| \times 1 \times \mathrm{N} \times \mathrm{d}) \cdot (1 \times \|\mathcal{B}\| \times \mathrm{N} \times \mathrm{d}) \longrightarrow (\|\mathcal{B}\| \times \|\mathcal{B}\| \times \mathrm{N} \times \mathrm{d})$
Reduce Sum	$(\|\mathcal{B}\| \times \|\mathcal{B}\| \times \mathrm{N} \times \mathrm{d}) \longrightarrow (\|\mathcal{B}\| \times \|\mathcal{B}\| \times \mathrm{N} \times 1)$
Softmax	$(\|\mathcal{B}\| \times \|\mathcal{B}\|(\mathrm{softmax}) \times \mathrm{N} \times 1)$
Attention Matrix*V	$(\|\mathcal{B}\| \times \|\mathcal{B}\| \times \mathrm{N} \times \mathrm{d}) \cdot (1 \times \|\mathcal{B}\| \times \mathrm{N} \times \mathrm{d}) \longrightarrow (\|\mathcal{B}\| \times \|\mathcal{B}\| \times \mathrm{N} \times \mathrm{d})$
Output	$(\|\mathcal{B}\| \times \|\mathcal{B}\|(\text{reduce sum+squeeze}) \times \mathrm{N} \times \mathrm{d}) \longrightarrow (\|\mathcal{B}\| \times \mathrm{N} \times \mathrm{d})$

* 'N' denotes the dimension of user or item.

A.8 BEHAVIORAL ABLATION EXPERIMENTS

	H@10	N@10	H@10	N@10	H@10	N@10	H@10	N@10
Tmall	w/o-View		w/o-Favorite		w/o-Cart		Purchase	
	0.4625	0.2641	0.5469	0.3265	0.5338	0.3186	0.3696	0.2295
IJCAI_15	w/o-View		w/o-Favorite		w/o-Cart		Purchase	
	0.3546	0.1973	0.4171	0.2341	0.4634	0.2693	0.3046	0.1773
E-commerce	w/o-Review		w/o-Browse		Purchase		-	
	0.7323	0.4456	0.7109	0.4412	0.6768	0.4108	-	-

	Tmall		IJCAI		E-commerce	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
EHCF	0.011751	0.005270	0.024629	0.012748	0.135780	0.065764
CML	0.013989	0.006279	0.029593	0.014911	0.139536	0.067804
EGCM	0.015703	0.006932	0.035461	0.018640	0.151299	0.075318

A.9 Additional Details of Experiments

A.9.1 The Embedding t-SNE Visualization Experiment

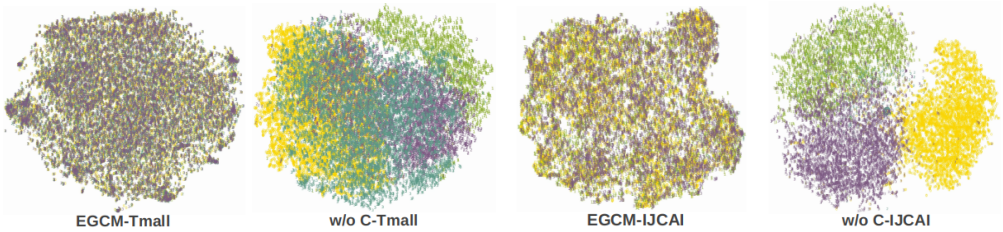
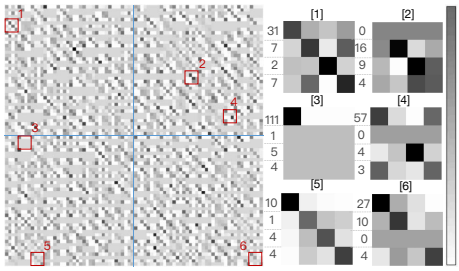


Figure 4: t-SNE Visualization

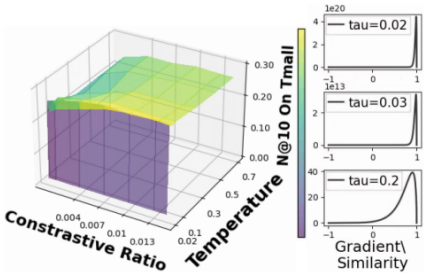
Visualizing behavior-specific embeddings in Fig. 4 aims to show the rationality of bringing the contrastive task into the multi-behavior recommendation, so as to maximize mutual information across behaviors. Technically, we utilize t-SNE initialized with PCA Dunteman (1989) to reduced the behavior-specific embedding to two dimensions. And we conduct the experiment on datasets(Tmall, IJCAI) contain four types of behaviors(*page view/click, favorite, cart, purchase*), and the behaviors in Fig. 4 were represented by different colors. We can observe that the behaviors of EGCM are closer in the reduced dimension space because EGCM integrates cross behavior information by maximizing mutual information. In other words, for user u, embedding of other behaviors with the same index become closer, while users with different indexes $u \neq u'$ are pulled away.

A.9.2 The Self-attention Visualization Experiment



We visualized the learned self-attention weight matrices Φ in Eq. 4 for multi-relational memory networks in Sec. 3.2 which retains the customized long-term behavior preference of the user sequence between timeslots on Tmall dataset. The dataset has four behaviors $PageView, Favorite, Cart, Purchase$, therefore, the self-attention learns the relationship between these four behaviors. We sampled some users and then visualized their memory self-attention matrices. Then, we select six cases from the sampled data and combine the real statistics of multi-behavior data to analyze the significance of the learned weight matrices. First of all, it can be observed in left part that most matrices have the darkest diagonal color, which is the characteristic of self attention and also shows that the behavior itself has the greatest correlation with itself. Right part shows that the weight of each behavior is related to the number of interactions of the behavior itself. For example, for user 22186([3]) the *super node* with 111 interactions in *page view* behavior, the number of interactions in other behaviors are {1, 5, 4} which are too small relatively. Thus it is difficult to learn differentiated values in the other three rows relative to the first row of the weight matrix. Analogously, some behaviors without interaction cannot learn effective weights. On the contrary, if a type behavior has more interactions, the values of its corresponding row in the weight matrix will be larger.

A.9.3 THE CONTRASTIVE HYPER-PARAMETER EXPERIMENT



A.1.1 has been proved that the similarity between the difficult negative sample and the anchor point is proportional to its gradient. Moreover, the strength of this relationship can be adjusted by temperature coefficient. And the influence of temperature coefficient on gradient is shown in diagram. This suggests that contrastive learning will push different users away in the representation space, and the more similar representations will be pushed away. In this way, the representation inditinguishable problem may be alleviated which due to the oversmoothing of GNNs or the sparse data. In figure, we can see that, firstly, in the multi-task framework, the contrastive loss will have effect within a reasonable range. Then, the smaller the temperature sparsity, that is, the greater the negative sample gradient, the better the result will be. However, too small a temperature coefficient results in gradient explosion, shown in purple part.

A.9.4 IMPLEMENTATION DETAILS

Our experiments are conducted on a machine equipped with a 24 GB Nvidia RTX 3090 GPU. We use pytorch Paszke et al. (2019) to implement our model and initialize model parameters with Xavier initializer Glorot & Bengio (2010), and adopt AdamW Loshchilov & Hutter (2017) optimizer. Instead of using fixed learning rate, we adopt the Cyclical Learning Rate (CyclicLR) strategy Smith (2017) with the boundary $[1e^{-4}, 1e^{-3}]$, $[1e^{-3}, 2e^{-3}]$, $[1e^{-3}, 5e^{-3}]$ for the three datasets. The weights of L_2 regularization are $\{1.45e^{-2}, 1.4e^{-2}, 1e^{-2}\}$ for the datasets, respectively. To ensure fair comparison, the embedding dimensionality of our model and all baselines are set as 16. The depth of our graph neural layers in our multi-relational encoder is selected from 1 to 4. The value of the temperature coefficient in contrastive learning is selected from {0.02, 0.035, 0.05, 0.07, 0.1, 0.3, 0.5, 0.7}. The impact of model hyperparameters is explored in Section A.3. The performance of most baselines is based on their source code, and part of them comes from the public sequence recommendation framework ReChorus Wang et al. (2020a).

A.9.5 BASELINES

To explore how EGCM boosts recommendation performance. We compare our EGCM with the following state-of-the-art baselines from different research lines:

Firstly, we choose one classical model, three traditional sequential models and four self-attendation-based models.

- **HGN** Ma et al. (2019): This method uses hierarchical gating network with an item-item product module which can decide the item features passes to downstream layers to capture users' long-term and short-term preferences.
- **Caser** Tang & Wang (2018): Caser conducted top-N sequential recommendation with CNN. The method models recent interactions as an "image" among time and latent dimensions.
- **Chorus** Wang et al. (2020a): It models time-aware item knowledge by mining the relationship information changes over time, and injects the information into the representation of item.
- **SASRec** Kang & McAuley (2018): It uses self-attention instead of any recurrent or convolutional operations to capture long-term data semantics. And in each time step, it adaptively looks for which item is related to the user's history.
- **TiSASRec** Li et al. (2020): It is a time interval aware self-attention method for next-item recommendation based on SASRec Kang & McAuley (2018) which uses the absolute position and relative time interval of items in the sequence for modeling.

- **AttRec** Zhang et al. (2018): It models users' short-term and long-term preferences at the same time, in which short-term preferences are learned through self attention, and uses metric learning methodology which has produced good results.
- **BERT4Rec** Sun et al. (2019): This model based on deep bidirectional self-attention architecture Devlin et al. (2018). In the process of training the recommendation model, the method introduce Close task to mask and predict the left and right items, and this equivalent to data augmentation for sequential data.

Then, GNN-based models illustrate the effectiveness of Graph Neural Network in sequential recommendation.

- **HyperRec** Wang et al. (2020b): This is a next-item recommendation framework which considers the dynamic semantics in the real situation. It uses hypergraph to model the correlation of short-term items, while residual gating and the fusion layer can model user preferences more accurately.
- **MA-GNN** Ma et al. (2020):The model uses graph neural network to model the short-term information of item, and uses memory mechanism to model the long-term dependent information. It also uses bilinear function to learn the correlation between item features.

We further compare our EGCM with behavior state-of-the-art multi-behavior recommendation model to explore the effectiveness of multiple behaviors. These models and our models take the purchase behavior as target and other behaviors as auxiliary information.

- **NMTR** Gao et al. (2019): The method combines NCF He et al. (2017) and multi-task learning for recommendation. And it modeling the cascading relationship among different behaviors exploit multiple types of users.
- **DIPN** Guo et al. (2019): This is a attention-based multi-task framework for recommendation which uses the data of browse behavior buy behavior to predict users' purchase intention.
- **MBGCN** Jin et al. (2020b): It is a graph-based recommendation method that reconstructs the relationship matrix of multiple behaviors into a unified matrix to fully model the preference intensity of different behaviors through the changes and semantics of different behaviors.
- **MBGMN** Xia et al. (2021b): This work uses meta network to learn the heterogeneity and diversity of interaction. It directly correlates a variety of user behaviors and integrates them into the collaborative filtering framework.