Contrastive Meta Learning with Behavior Multiplicity for Recommendation

Wei Wei[1,3], Chao Huang[1,2*], Lianghao Xia[1], Yong Xu[3], Jiashu Zhao[4], Dawei Yin[5]

[1]Department of Computer Science, [2]Musketeers Foundation Institute of Data Science, University of Hong Kong
[3]South China University of Technology, [4]Wilfrid Laurier University, [5]Baidu Inc
weiwei1206cs@gmail.com, chaohuang75@gmail.com, aka_xia@foxmail.com,
yxu@scut.edu.cn, jzhao@wlu.ca, yindawei@acm.org

ABSTRACT

A well-informed recommendation framework could not only help users identify their interested items, but also benefit the revenue of various online platforms (*e.g.*, e-commerce, social media). Traditional recommendation models usually assume that only a single type of interaction exists between user and item, and fail to model the multiplex user-item relationships from multi-typed user behavior data, such as page view, add-to-favourite and purchase. While some recent studies propose to capture the dependencies across different types of behaviors, two important challenges have been less explored: i) Dealing with the sparse supervision signal under target behaviors (*e.g.*, purchase). ii) Capturing the personalized multi-behavior patterns with customized dependency modeling. To tackle the above challenges, we devise a new model CML, **C**ontrastive **M**eta **L**earning (CML), to maintain dedicated cross-type behavior dependency for different users. In particular, we propose a multi-behavior contrastive learning framework to distill transferable knowledge across different types of behaviors via the constructed contrastive loss. In addition, to capture the diverse multi-behavior patterns, we design a contrastive meta network to encode the customized behavior heterogeneity for different users. Extensive experiments on three real-world datasets indicate that our method consistently outperforms various state-of-the-art recommendation methods. Our empirical studies further suggest that the contrastive meta learning paradigm offers great potential for capturing the behavior multiplicity in recommendation. We release our model implementation at: https://github.com/weiwei1206/CML.git.

CCS CONCEPTS

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

KEYWORDS

Collaborative filtering, Self-Supervised Learning, Multi-Behavior Recommendation, Meta Learning, Graph Neural Network

---

---

1 INTRODUCTION

Recommender systems have emerged as critical components to alleviate information overloading for users in various online applications, *e.g.*, e-commerce [40], online video platform [46] and social media [30]. The goal is to learn user preference and forecast the items that he or she will consume based on observed user behaviors.

Among various recommendation techniques, collaborative filtering (CF) has become the most promising recommendation architecture to model historical user interactions over items [4, 57]. Commonly, the core of existing CF paradigm is to project users and items into latent representation space such that their interaction structural information is preserved. For example, Autoencoder has been employed as the effective embedding function for the representation projection in AutoRec [33] and CDAE [49]. To inject the high-order connection signals in CF, another promising research line model user-item interactions as a graph and generate the user/item feature representations with the graph structural information preserved. These models perform the message passing over the interaction graph to generate node-level embeddings layer by layer, such as PinSage [53], NGCF [43] and LightGCN [15].

However, the majority of existing recommendation models assume that only a single type of interaction exists between user and item, whereas in practical recommendation scenarios are multiplex in nature [12, 41]. Taking the online retail platform as an example, users can interact with items in multiple manners, including page view, add-to-favourite and purchase. Different types of behaviors may characterize user preference from different intention dimensions and complement with each other for better user preference learning [37]. Therefore, it is challenging but valuable to capture behavior multiplicity and the underlying dependencies in recommendation. To address this challenge, existing work models the behavior dependency by introducing different aggregation schemes to integrate type-specific behavior embeddings, to enhance the representation on target user behaviors (*e.g.*, customer purchase) [23, 50, 51]. For example, MATN [50] adopts the self-attention to encode the pairwise correlations between different types of behaviors, and make predictions on the target behaviors. A relation-aware embedding propagation layer is developed to learn

the behavior multiplicity in MBGCN [23], to gather multi-behavior interaction information from high-order neighbors.

Despite the effectiveness of existing methods, these studies share two common limitations: *First*, Sparse Supervision Signal under Target Behaviors: the most of current multi-behavior recommender systems are trained with supervised information in an end-to-end manner. That is to say, for making forecasting on the target user behaviors, it is required to have sufficient labeled data corresponding to the target behaviors (*e.g.*, user purchase data). Unfortunately, the observed interactions under the target behavior type, are often sparse as compared with other types of user-item interactions. For example, purchase prediction task in online retail system still faces the challenge of lacking of ground-truth labels [20]. Hence, directly integrating type-specific behavior embeddings will sacrifice the performance due to lacking supervision signals of target behaviors. *Second*, Personalized Multi-Behavior Patterns: multi-behaviour patterns may vary by users. Semantics of multi-typed user-item interactions and their mutual relationships are diverse, depending on the personalized characteristics of users [27]. Without considering diverse user intents which motivates different types of user behaviors, previous modeling of multiplex user-item relationships leads to suboptimal representations.

Contributions. Having realized the above challenges for recommendation with behavior multiplicity, we focus on exploring diverse multi-behavior patterns under a contrastive self-supervised learning prototype. Towards this end, this work proposes a new model-**C**ontrastive **M**eta **L**earning (CML) for multi-behavior recommendation. In CML, we design a multi-behavior contrastive learning framework to capture the cross-type interaction dependency from different behavior views. This endows our developed recommender system to effectively distill additional supervision signal from different types of user behaviors, which augments the model optimization process with sparse supervision labels. Inspired by the recent success achieved by self-supervised representation learning, we leverage the idea of contrastive learning to design cross-type behavior dependency modeling task with the user self-discrimination. The goal of our multi-behavior contrastive learning is to reach the agreement between user's type-specific behavior representations via the constructed contrastive loss. In addition, to handle the preference diversity of users and capture the personalized multi-behavior patterns, we design contrastive meta network to characterize the customized behavior heterogeneity, empowering CML to maintain dedicated representations for different users. Our meta contrastive encoder first extracts the personalized meta-knowledge from users, and then feeds it into our weighting function for customized multi-behavior dependency modeling.

In a nutshell, this work makes the following contributions:

- We propose a new multi-behavior learning paradigm CML for recommendation by emphasizing the importance of diverse and multiplex user-item relationships, as well as tackling the label scarcity problem for target behaviors.

- In our CML framework, we design a multi-behavior contrastive learning paradigm to capture the transferable user-item relationships from multi-typed user behavior data, which incorporates auxiliary supervision signals into the sparse target behavior modeling. Furthermore, our proposed meta contrastive encoding

scheme allows CML to preserve the personalized multi-behavior characteristics, so as to be reflective of the diverse behavior-aware user preference under a customized self-supervised framework.

- We perform extensive experiments on three real-world recommendation datasets to justify the rationality of our assumptions and the effectiveness of our proposed framework. By comparing CML with 12 baselines, we show that CML is able to consistently improve the performance of different techniques under various settings. Further analysis demonstrates the effectiveness of the designed sub-modules with ablation study.

2 PRELIMINARY

We first define $\mathcal{U}$ and $\mathcal{I}$ to represent the set of users and items, respectively. In our multi-behavior recommendation scenario, let $\mathcal{X}^{(k)}$ denote the user-item interaction matrix under the k-th behavior type (*e.g.*, page view, add-to-favorite, purchase). Hence, multi-behavior interaction data is represented as $\{\mathcal{X}^{(1)}, ..., \mathcal{X}^{(k)}, ..., \mathcal{X}^{(K)}\}$, where K is the number of behavior types. In particular, the element $x_{u,i}^k = 1$ indicates that user u has interacted with item i under the behavior type of k before, and $x_{u,i}^k = 0$ otherwise. Generally, there exist *target behavior* as the prediction objective. Other types of user behaviors serve as the *auxiliary behaviors*. For example, purchases are directly related to Gross Merchandise Value (GMV) in E-commerce services, and are usually considered as the target behaviors in various user modeling applications. Auxiliary behaviors could be the interactions of page view and add-to-favorite/cart.

Problem Statement. The studied task is formally stated as: **Input**: observed user-item interactions with multiplex K types of behaviors $\{\mathcal{X}^{(1)}, ..., \mathcal{X}^{(k)}, ..., \mathcal{X}^{(K)}\}$ among users $\mathcal{U}$ and items $\mathcal{I}$. **Output**: a predictive function which estimates the likelihood of user u will interact with item i under the target type (k) of behaviors.

Multi-Behavior Interaction Graph. Inspired by the representation paradigm of graph collaborative filtering methods [43, 45], we explore the user-item graph structure for our multi-behavior recommendation scenario. Specifically, given K types of user-item interaction matrices $\{\mathcal{X}^{(1)}, ..., \mathcal{X}^{(k)}, ..., \mathcal{X}^{(K)}\}$, we generate the multi-behavior interaction graph, in which the set of nodes $\mathcal{V} = \mathcal{U} \cup \mathcal{I}$ involves the user and item set. We further define the set of multiplex edges $\mathcal{E}$ to represent observed interactions with K types of behaviors. In $\mathcal{E}$, edge $e_{u,i}^k$ between u and i indicates that $x_{u,i}^k = 1$.

3 METHODOLOGY

We present our Contrastive Meta Learning (CML) framework in this section, which encapsulates the customized meta learning into a self-supervised neural architecture, for personalized multi-behavior dependency modeling. The overall model flow is shown in Figure 1. Key components will be elaborated in following subsections.

3.1 Behavior-aware Graph Neural Network

To inject the high-order connectivity into the multiplex relation learning across users/items, we first develop a graph-based message passing framework with the awareness of behavior context. Motivated by graph-based information propagation neural architecture [55] and the findings in the state-of-the-art model Light-GCN [15, 22], our behavior-aware message passing scheme is built
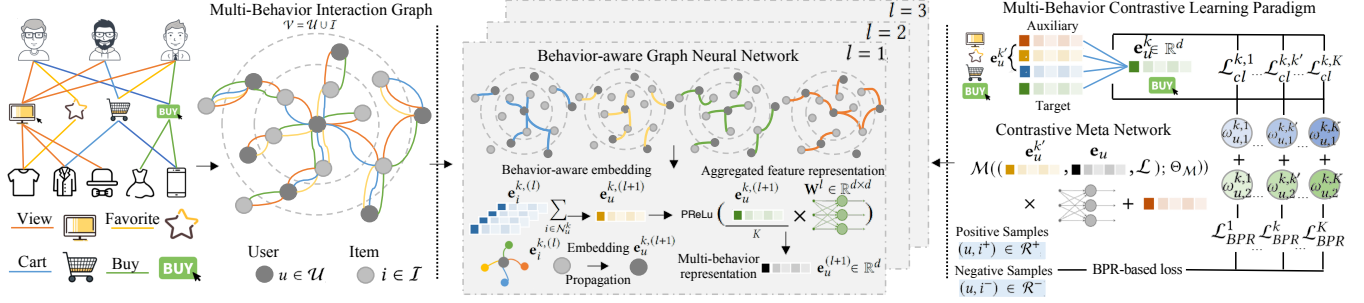
Figure 1: The model flow of CML framework. i) The designed graph neural network $\mathcal{G}(\mathcal{A}; \Theta_{\mathcal{G}})$ performs the behavior-aware message passing over the multi-behavior interaction graph $G = \{\mathcal{V}, \mathcal{E}\}$. ii) The contrastive views are constructed between auxiliary and target behavior embeddings $e_u^k, e_u^{k'}$. iii) Our proposed meta contrastive encoder captures the customized cross-type behavior dependency with the meta weight network $\mathcal{M}((\mathcal{L}, E, E^k); \Theta_{\mathcal{M}})$. $\omega_u^{k,k'}$ is the personalized contrastive loss weight.

over a lightweight graph architecture, which can be represented:

$$\mathbf{e}_u^{k,(l+1)} = \sum_{i \in \mathcal{N}_u^k} \mathbf{e}_i^{k,(l)}; \quad \mathbf{e}_i^{k,(l+1)} = \sum_{u \in \mathcal{N}_i^k} \mathbf{e}_u^{k,(l)} \tag{1}$$

where $\mathbf{e}_v^{k,(l+1)} \in \mathbb{R}^d$ is defined as the obtained representation of node v ($v \in \{u, i\}$) under the l-th graph neural layer. $\mathcal{N}_u^k$ and $\mathcal{N}_i^k$ denotes the neighboring nodes of item i and user u, respectively. After encoding the behavior-specific interaction patterns of users, we propose to perform the embedding aggregation across different types of behaviour patterns with the following operation for user representations (similar aggregation is applied for item side):

$$\mathbf{e}_u^{(l+1)} = \text{PReLu}(\mathbf{W}^l \cdot \frac{\sum_{k \in K} \mathbf{e}_u^{k,(l+1)}}{K}) \tag{2}$$

The aggregated feature representation $\mathbf{e}_u^{(l+1)}$ could preserve multi-behavior contextual information. $\mathbf{W}^l \in \mathbb{R}^{d \times d}$ represents the transformation matrix corresponding to l-th graph propagation layer.

3.2 Multi-Behavior Contrastive learning

In our CML framework, we propose a multi-behavior contrastive learning paradigm to capture the complex dependencies across different types of user interactions via a self-supervised principle. Conceptually, we utilize the idea of contrastive learning strategy for instance discrimination by contrasting positive and negative samples [31, 54]. Our contrastive learning architecture endows our main supervised task (*i.e.*, target behavior prediction) with the auxiliary supervision signals from the auxiliary behaviors.

3.2.1 Contrastive View Generation.
In contrastive learning paradigm, it is important to generate appropriate views for constructing diverse representations for the method to contrast with [5]. In our recommendation scenario with behavior multiplicity, we propose to consider each type of behaviors as individual view, which performs the contrastive learning between user embeddings in different behavior views. Different from current multi-behavior recommender systems (*e.g.*, MATN [50], MBGCN [23]) which merely rely on behavior-wise embedding combination for target behavior prediction, we conduct the data augmentation by incorporating auxiliary behavior contextual information as supervision signals. This design not only encodes the cross-type behavior dependency,

but also alleviates the skewed data distribution across different types of user interaction data.

3.2.2 Behavior-Wise Contrastive Learning Paradigm.
After establishing contrastive views from multi-behavior context, we further devise a behavior-wise contrastive learning paradigm between the target behaviors and auxiliary behaviors. In particular, different behavior views of the same user are considered as positive pairs, and the views of different users are sampled as negative pairs. Given the encoded target behavior representation $\mathbf{e}_u^k$ from our graph neural architecture, the generated positive and negative pairs are $\{\mathbf{e}_u^k, \mathbf{e}_u^{k'} | u \in \mathcal{U}\}$ and $\{\mathbf{e}_u^k, \mathbf{e}_{u'}^{k'} | u, u' \in \mathcal{U}, u \neq u'\}$. The incorporated auxiliary supervision enables our model to still recognize user u from different behavior views (*i.e.*, k and k'; $k, k' \in K$) and captures the latent relationships between the auxiliary behaviors and target behaviors. Meanwhile, for different users u and u', the contrastive loss aims to discriminate their behavior embeddings after data augmentation.

Following works [48, 58], we utilize the InfoNCE [29] loss in our multi-view contrastive learning framework, to measure the distance between embeddings. We define our self-supervised learning loss with the objective of maximizing the Mutual Information (MI) between user representations through contrasting positive pairs with the sampled negative pair counterparts. The InfoNCE-based contrastive loss is calculated as below:

$$\mathcal{L}_{cl}^{k,k'} = \sum_{u \in \mathcal{U}} -log \frac{\exp(\varphi(\mathbf{e}_u^k, \mathbf{e}_u^{k'})/\tau)}{\sum_{u' \in \mathcal{U}} \exp(\varphi(\mathbf{e}_u^k, \mathbf{e}_{u'}^{k'})/\tau)} \tag{3}$$

Here, we define $\varphi(\cdot)$ as the similarity function (*e.g.*, inner-product or cosine similarity) between two embeddings. τ represents the temperature hyperparameter for the softmax function. To sum up, we perform the contrastive learning via maximizing the agreement between two behavior views based on the above defined contrastive loss, and enforcing the divergence among different users. We obtain the contrastive loss $\mathcal{L}_{cl}^{k,k'}$ for each pair of target behavior (k) and auxiliary behavior (k'). Therefore, we generate a list of contrastive loss functions as: $\mathcal{L}_{cl} = \mathcal{L}_{cl}^{k,1} + ... + \mathcal{L}_{cl}^{k,k'} + ... + \mathcal{L}_{cl}^{k,K}$.

3.3 Meta Contrastive Encoding

In our recommendation scenario, different users have various behaviour patterns and item interaction preferences. For example, some users are likely to pick up most of products from their favorite item list to purchase, while others may only buy sporadic products given that they add a lot of items with less interest into their list [27]. The diversity of multi-behavior patterns from different users, results in different item interactions. Hence, effectively modeling the personalized dependencies across different types of behaviors, is also important in making accurate recommendations. To achieve this goal, we propose a meta contrastive encoding scheme to learn an explicit weighting function for the integration of multi-behavior contrastive loss. This module customizes our self-supervised learning paradigm with the diverse constrastive loss integration. Our meta contrastive encoding schema is a two-phase learning paradigm: i) We propose a meta-knowledge encoder to capture the personalized multi-behavior characteristics, so as to reflect the diverse behavior-aware user preferences. ii) Then, the extracted meta-knowledge will be incorporated into our developed meta weight network, to generate customized contrastive loss weight for cross-type behavior dependency modeling.

3.3.1 Meta-Knowledge Encoder.
In our meta contrastive encoding framework, we firstly extract the meta-knowledge to preserve user-specific behavior dependencies. Inspired by feature interaction mechanisms in [16, 56], we design two types of meta-knowledge encoder with different integration techniques based on learned user behavior representations: $\mathbf{e}_u$ and $\mathbf{e}_u^{k'}$ (auxiliary behavior of k'):

$$\mathbf{Z}_{u,1}^{k,k'} = (d(\mathcal{L}_{cl}^{k,k'}) \cdot \gamma) \parallel \mathbf{e}_u^{k'} \parallel \mathbf{e}_u; \ \mathbf{Z}_{u,2}^{k,k'} = \mathcal{L}_{cl}^{k,k'} \cdot (\mathbf{e}_u^{k'} \parallel \mathbf{e}_u) \quad (4)$$

where the encoded meta-knowledge is represented by $\mathbf{Z}_{u,1}^{k,k'}$ and $\mathbf{Z}_{u,2}^{k,k'}$. We define $d(\cdot)$ as the duplicate function to generate a value vector corresponding to the embedding dimensionality. $\parallel$ denotes the concatenation operation. γ is a scale factor for the enlarge value. With this design for learning the personalized characteristics, both the auxiliary-target behavior dependency and user-specific interaction context are preserved in the extracted meta-knowledge.

3.3.2 Meta Weight Network.
After encoding the meta knowledge with user-specific multi-behavior patterns, we design a weighting function $\xi(\cdot)$ mapping from meta-knowledge to contrastive loss weights. This module endows our recommendation framework with the capability of learning the multi-behavior relationships in a customized manner, to be reflective of personalized user preference under various types of behavior intentions. Formally, we define our weighting function as the following transformation layer:

$$\xi(\mathbf{Z}_u^{k,k'}) = \text{PReLU}(\mathbf{Z}_u^{k,k'} \cdot \mathbf{W}_\xi + \mathbf{b}_\xi) \quad (5)$$

where $\mathbf{W}_\xi \in \mathbb{R}^{d \times d}$ and $\mathbf{b}_\xi \in \mathbb{R}^d$ represent the projection layer and bias term, respectively. Here, we utilize the PReLU activation function to incorporate non-linearity. On the basis of our meta weight network, we can obtain our personalized contrastive loss weight as follows:

$$\omega_u^{k,k'} = \omega_{u,1}^{k,k'} + \omega_{u,2}^{k,k'} = \xi(\mathbf{Z}_{u,1}^{k,k'}) + \xi(\mathbf{Z}_{u,2}^{k,k'}) \quad (6)$$

For each user u, $\omega_u^{k,k'}$ weight represents the customized explicit dependence between the target behavior type of k and auxiliary behavior type of k'. Accordingly, with our meta contrastive encoding scheme, we can generate two lists of loss weights for InfoNCE-based self-supervised loss and Bayesian personalized ranking (BPR)-based recommendation objective loss.

3.4 The Learning Process of CML Framework

In this section, we first introduce our optimization objective and then present the training strategy for our CML framework. Finally, the analysis on the time complexity of our model is provided.

3.4.1 Optimization Objective.
In the model inference of CML, we leverage the Bayesian Personalized Ranking (BPR) loss to learn parameters, which encourages the probability estimation of user's observed interaction to be higher than his/her unobserved counterparts. Formally, the behavior-specific BPR loss is defined as:

$$\mathcal{L}_{BPR}^k = \sum_{(u,i^+,i^-) \in O_k} -\text{In}(\text{sigmoid}(\hat{x}_{u,i^+}^k - \hat{x}_{u,i^-}^k)) + \lambda \|\Theta\|^2 \quad (7)$$

O_k represents the pairwise training samples of k-th behavior type, i.e., $O_k = \{(u,i^+,i^-)|(u,i^+) \in \mathcal{R}^+, (u,i^-) \in \mathcal{R}^-\}$. Here, $\mathcal{R}^+$ and $\mathcal{R}^-$ denotes the corresponding observed and unobserved interaction of user u. Θ represents the learnable parameters and the L_2 regularization is applied for alleviating overfitting issue.

3.4.2 Model Training.
In this work, we follow the training strategy of meta-learning methods in previous work [9, 34], by updating the parameters of our graph neural architecture (represented as $\mathcal{G}(\mathcal{A}; \Theta_\mathcal{G})$) and multi-behavior contrastive meta network (represented as $\mathcal{M}((\mathcal{L}, \mathbf{E}, \mathbf{E}^k); \Theta_\mathcal{M})$) in an alternative way. Here, $\mathcal{A}$ denotes the input adjacent matrix of behavior-aware user-item interaction graph. $\mathbf{E}$ and $\mathbf{E}^k$ represents the learned cross-type and behavior-specific embedding matrix of all users, respectively. The model training consists of three phases in an optimization loop to improve the training efficiency of our models. In particular: i) In the first stage, we integrate the behavior-aware graph neural network (with cloned state) and contrastive meta network, to learn initial parameter space of our multi-behavior contrastive encoder over the entire training data. ii) In the second stage, we refine the model parameters $\Theta_\mathcal{M}$ of our contrastive meta network based on the meta data. iii) After generating the personalized contrastive loss weights, we leverage the updated Θ_M to ameliorate the parameter $\Theta_\mathcal{G}$ of our graph neural network. We formally present the nested optimization process as follows (B denote the size of training batch):

$$\Theta_\mathcal{G}^* = \arg\min_\theta \triangleq \sum_{k=1}^K \sum_{b=1}^B \Big(\mathcal{M}((\mathcal{L}_{cl,k}^{train \cup meta}, \mathbf{E}, \mathbf{E}^k); \Theta_\mathcal{M}) \cdot \mathcal{L}_{cl,k}^{train}$$
$$+ \mathcal{M}((\mathcal{L}_{bpr,k}^{train \cup meta}, \mathbf{E}, \mathbf{E}^k); \Theta_\mathcal{M}) \cdot \mathcal{L}_{bpr,k}^{train} \Big) \quad (8)$$

3.4.3 Model Complexity Analysis.
We analyze the complexity of our CML framework from several key components: i) the computational cost of our lightweight graph neural architecture is $O(L \times K \times |\mathcal{R}^{k+}| \times d)$ for performing message passing across graph layers. $|\mathcal{R}^{k+}|$ represents the number of non-zero elements in the adjacent matrix under the behavior of k, and L denotes the number of information propagation layers. The operations of linear

Table 1: Statistics of experimented datasets

Dataset	User #	Item #	Interaction #	Interactive Behavior Type
Tmall	31,882	31,232	1,451,219	{Page View, Favorite, Cart, Purchase}
IJCAI-Contest	17,435	35,920	799,368	{Page View, Favorite, Cart, Purchase}
Retail Rocket	2,174	30,113	97,381	{Page View, Cart, Transaction}

transformations and mean-pooling for multi-behavior aggregation takes $O(L \times (N + M) \times d \times (K + d))$ time. ii) Our meta contrastive encoder takes $O(K \times |\mathcal{R}^{k+}| \times d^2)$ time overhead. iii) The cost of InfoNCE-based mutual information calculation is $O(B \times d)$ and $O(B \times S \times d)$ for the numerator and denominator (in Equation 3), respectively. Here, S is the sampling size of contrastive learning for reducing the time complexity and increasing the randomness to achieve model robustness [44]. Therefore, our multi-behavior contrastive learning paradigm takes $O(K \times |\mathcal{R}^{k+}| \times S \times d)$ time per epoch. In conclusion, our model could achieve comparable time complexity with state-of-the-art multi-behavior recommendation techniques (*e.g.*, MBGCN, EHCF).

4 EVALUATION

To evaluate CML's performance, we conduct experiments on several real-world datasets by answering the following research questions:

- **RQ1**: How effective is the developed CML framework to tackle the behavior multiplicity in recommendation?

- **RQ2**: How do different modules contribute to the performance of CML, such as the multi-behavior contrastive learning paradigm and meta contrastive encoder?

- **RQ3**: How does CML perform to alleviate interaction data sparsity, when competing with state-of-the-art methods?

- **RQ4**: How do different hyperparameter settings affect CML?

- **RQ5**: How is the model interpretation ability of our CML?

4.1 Experimental Settings

4.1.1 **Datasets.** We evaluate the effectiveness of our proposed CML on three publicly available recommendation datasets. We present the statistical information in Table 1. **Tmall**: This dataset is collected from Tmall site–one of the largest E-commerce platform in China. The user behavior data contains various interactions: *Page View*, *Add-to-Favorite*, *Add-to-Cart* and *Purchase*. Following the setting in [50], we keep users with at least three purchases for training and test. **IJCAI-Contest**: This data was adopted in IJCAI15 Challenge from a business-to-customer retail system. It shares the same behavior types with the Tmall data, which are reflective of various user intention over items. **Retailrocket**: It is another benchmark dataset collected from Retailrocket recommender system. In this dataset, user interactions are consisted of *Page View*, *Add-to-Cart* and *Transaction*. Following previous works for recommendation with multi-behaviors [23, 50], purchase behaviors are set as the target behaviors and other types of interactions are considered as the auxiliary behaviors.

4.1.2 **Baselines.** We compare our CML with the following state-of-the-art methods from two groups: *Single-Behavior* and *Multi-Behavior* recommender systems. These methods leverage various techniques to improve the recommendation performance:
Single-Behavior Recommendation Methods:

- **BPR** [32]: It is a widely adopted matrix factorization model with the optimization criterion of Bayesian personalized ranking.
- **PinSage** [53]: This method defines the importance-based neighboring nodes to perform the graph convolution. In PinSage, the message passing paths are constructed through the random walk.
- **NGCF** [43]: it is a representative graph neural framework which captures the collaborative effects in the embedding function of users based on the convolutional message passing scheme.
- **LightGCN** [15]: it simplifies the graph convolution network-based recommendation architecture by removing the feature transformation and nonlinear activation operations.
- **SGL** [48]: this method performs the self-supervised learning over the user-item interaction graph with data augmentation from different views (*e.g.*, node and edge dropout). The integrated auxiliary task is on the basis of node self-discrimination.

Multi-Behavior Recommendation Models:

- **NMTR** [11]: it combines the multi-task learning framework and neural collaborative filtering to investigate multi-typed user interaction behaviors based on the predefined cascading relationships.
- **MATN** [50]: it adopts the attention mechanism for multi-behavior recommendation. Specifically, it uses memory-enhanced self-attention to measure the influence between different behaviors. The number of memory units is tuned from the range of [2,8].
- **MBGCN** [23]: this approach is a GCN-based model by capturing the multi-behavioral patterns over the constructed user-item interaction graph. The high-order connectivity is considered during the information propagation.
- **KHGT** [51]: this approach leverages transformer to incorporate the temporal information into the multi-behavior modeling, and differentiates the behaviors with graph attention network.
- **EHCF** [2]: it conducts the knowledge transfer among heterogeneous user feedback to correlate behavior dependency. A new loss is used for model optimization from the positive-only data.

We further compare our CML with two state-of-the-art heterogeneous graph neural networks, by applying them to capture the heterogeneous behavior relations in recommendation.

- **HGT** [17]: This graph transformer models heterogeneous relations in graphs. We adopt the heterogeneous message passing schema to encode the multiplex behaviors with dedicated representations.
- **HeCo** [44]: It is a recently developed heterogeneous graph neural network based on the cross-view supervised learning architecture. We generate the meta-path relation from our multi-behavior interaction graph.

4.1.3 **Hyperparameters and Metrics.** We implement our CML with PyTorch. The embedding initialization is performed with Xavier [14] and the model is optimized by adopting the AdamW optimizer [26] and the Cyclical Learning Rate (CyclicLR) strategy [35]. In specific, the base and max learning rate is searched from $\{0.6e^{-4}, 1e^{-4}, 1e^{-3}\}$ and $\{0.6e^{-3}, 1e^{-3}, 2e^{-3}, 5e^{-3}\}$, respectively. For all graph-based baselines, the number of graph-based message propagation layers is tuned from {1,2,3,4}. We apply the L2 regularization for the learned embeddings with the weight tuned from $\{1e^{-3}, 5e^{-3}, 1e^{-2}\}$. Additionally, to alleviate the overfitting issue, the dropout is used in our designed meta network.

Table 2: Performance comparison of all compared methods on different datasets in terms of *NDCG@*10 and *HR@*10

Dataset	Metric	BPR	PinSage	NGCF	LightGCN	SGL	HGT	HeCo	NMTR	MBGCN	MATN	KHGT	EHCF	**CML**	Imprv.	*p*-val.
Tmall	HR	0.243	0.274	0.322	0.342	0.350	0.357	0.358	0.362	0.381	0.406	0.391	<u>0.433</u>	**0.543**	25.4%	$3e^{-5}$
	NDCG	0.143	0.151	0.184	0.205	0.210	0.210	0.199	0.215	0.213	0.225	0.232	<u>0.260</u>	**0.327**	25.8%	$2e^{-4}$
IJCAI-Contest	HR	0.163	0.176	0.256	0.257	0.249	0.250	0.262	0.294	0.304	0.369	0.317	<u>0.409</u>	**0.477**	16.6%	$9e-5$
	NDCG	0.085	0.091	0.124	0.122	0.123	0.119	0.121	0.161	0.160	0.209	0.182	<u>0.237</u>	**0.283**	19.4%	$6e-3$
Retail Rocket	HR	0.235	0.247	0.260	0.261	0.263	0.305	0.297	0.314	0.308	0.301	<u>0.324</u>	0.321	**0.356**	9.9%	$1e^{-3}$
	NDCG	0.146	0.139	0.140	0.152	0.165	0.176	0.178	0.201	0.181	0.181	0.202	<u>0.207</u>	**0.222**	7.3%	$1e^{-2}$

We adopt the widely used *leave-one-out* strategy by generating the test set from users' last interacted items under the target behavior type (*i.e.*, purchase/transaction). Two representative evaluation metrics are used for performance comparison: NDCG (Normalized Discounted Cumulative Gain) and HR (Hit Ratio). We also run our CML model and the best-performed baseline method for 10 times to calculate p-values for significance analysis.

4.2 Performance Comparison (RQ1)

We present the detailed evaluation results of all methods on different datasets in Table 2 where the results of our CML and the best performed baselines are highlighted with bold and underlined, respectively. Key observations are as follows:

- CML consistently outperforms all types of baselines on three datasets. The p-values are much less than 0.05, which indicates statistically significant improvements between our method and baselines. We attribute the significant performance improvements to the following two reasons: 1) Through the meta contrastive network, CML captures the multi-behavior dependencies in a customized manner; 2) The designed contrastive learning paradigm incorporates auxiliary self-supervised signals from different types of behavior dimensions, which offers informative gradients to the graph-based collaborative filtering architecture.

- Multi-behavior recommendation approaches (*e.g.*, MBGCN, EHCF, KHGT) yield better performance than single-behavior recommendation methods (*e.g.*, NGCF, LightGCN, PinSage), which reveals the helpfulness of exploring multi-behavioral information into the user preference modeling. Among various multi-behavior recommendation models, EHCF is the best baseline in most cases. This observation indicates that incorporating the different behavior semantics with supervision labels is able to guide the model optimization. Additionally, different from the topology-based self-supervised method-SGL, our CML designs new contrastive learning paradigm to fit the multi-behavior recommendation.

- CML outperforms heterogeneous graph neural networks (*i.e.*, HGT and HeCo) by a large margin in all cases, verifying that our designed meta contrastive network endows the heterogeneous collaborative filtering with the capability of effectively encoding the relation heterogeneity.

4.3 Ablation and Effectiveness Analyses (RQ2)

To shed light on the performance improvement, we further conduct the ablation study for our CML, to justify the rationality of the designed key components. Analysis details are summarized as:

- **Effect of multi-behavior contrastive learning framework**. We first aim to answer the question: is it beneficial to integrate behavior-wise dependency under a contrastive learning prototype for CML. Towards this end, we generate a model variant CML(w/o)-CLF by disabling the contrastive learning between the target and auxiliary user behaviors. Instead, we only rely on the behavior-aware graph neural network to capture the behavior relationships. We present the evaluation results in Table 3 with the following key summaries: 1) CML always outperforms CML(w/o)-CLF. This suggests the effectiveness of our contrasive learning paradigm, by capturing the complex dependent relations across different types of behaviors. 2) This design also mitigates the effect of skewed data distribution in the multi-behavior data, and effectively transfers knowledge from different behavior views.

- **Effect of meta contrastive network**. To investigate whether the meta contrastive network benefit the multi-behavior dependency modeling, we propose another variant CML(w/o)-MCN which only conducts the contrastive learning between type-specific behavior embeddings based on the estimated mutual information. In other words, cross-behavior contrastive loss functions are integrated with the BPR-based loss using the equal weights, *i.e.*, without explicitly differentiating the influence degrees under the augmented self-supervised learning tasks. Clearly, CML obtains better performance than CML(w/o)-MCN. It suggests that by employing the meta contrastive network, we can automatically discriminate the influence between different target-auxiliary behavior pairs. The cross-view behavior dependency can mutually complement with each other.

- **Effect of meta knowledge encoder**. To verify the impact of meta knowledge encoder in our contrastive learning framework, we do an ablation study (with variant CML(w/o)-MKE) by disabling the meta contrastive weight network $\mathcal{M}(\cdot)$. Instead, we use a weighted gating mechanism to aggregate the behavior-specific contrastive loss in a uniform manner. Removing the incorporation of our meta knowledge degrades the performance, suggesting the necessity of our customized contrastive learning for different types of target-auxiliary behavior dependency.

4.4 Model Performance on Alleviating Interaction Data Sparsity (RQ3)

In this section, we aim to show the rationality of bringing the contrastive learning into the multi-behavior recommendation, so as to alleviate the data sparsity issue. In Figure 2, we show the evaluation result comparison with respect to different interaction sparsity degrees on Tmall data. Due to space limit, we select several

Table 3: Ablation study on key components of CML

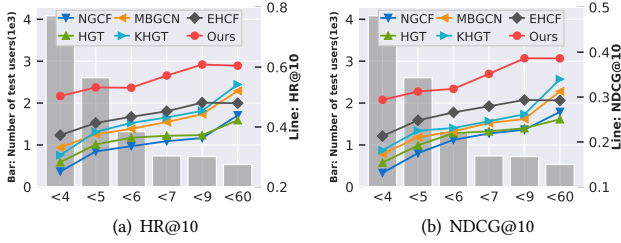| Data	Tmall		IJCAI-Contest		Retailrocket	
Metrics	HR	NDCG	HR	NDCG	HR	NDCG
w/o-CLF	0.4665	0.2752	0.3636	0.1978	0.3032	0.1864
w/o-MCN	0.5211	0.3097	0.4527	0.2703	0.3523	0.2185
w/o-MKE	0.5237	0.2988	0.4601	0.2715	0.3506	0.2079
CML	**0.5431**	**0.3266**	**0.4769**	**0.2829**	**0.3560**	**0.2219**

(a) HR@10 (b) NDCG@10

Figure 2: Performance comparison *w.r.t* different interaction sparsity degrees on Tmall data.

representative baselines to make comparison. Specifically, we split users into six groups in terms of the number of their interactions (*e.g.*, "<7" and "<60"). The reported model performance measured by HR and NDCG (as shown in the right side of y-axis in Figure 2) is averaged over all users in each group. The total number of users belonging to each group is shown in the left side of Figure 2.

We have the following findings: i) The recommendation accuracy improves for all compared methods as the number of user interactions increases. It is reasonable since the quality behavior embeddings are more likely to be learned with sufficient user behaviors. ii) As compared to the vanilla collaborative filtering model (NGCF), multi-behavior recommender systems (*e.g.*, KHGT, MBGCN) achieve better performance, suggesting the effectiveness of incorporating multi-typed behavior context for data sparsity alleviation. iii) CML consistently outperforms other multi-behavior recommendation methods under different interaction degrees. This observation indicates that CML solves the data sparsity issue better, by embracing the self-supervised contrastive learning paradigm for preserving the behavior heterogeneity in recommendation.

4.5 Hyperparameter Analysis on CML (RQ4)

This section examines the impact of different settings of several key hyperparameters in our proposed CML framework, including # graph propagation layers L, representation dimensionality d, batch size in training process. Figure 3 reports the evaluation results. For each time, we investigate the effect of one hyperparameter at a time and keep other parameters with their default settings.

 # graph propagation layers L. From Figure 3, we can observe that more graph propagation layers results in better performance when $L \leq 3$. This suggests that more message passing layers will capture latent dependency from high-order neighbors. When further stacking more graph layers might introduce noise to the user representations, which leads to the oversmoothing issue [3, 28].
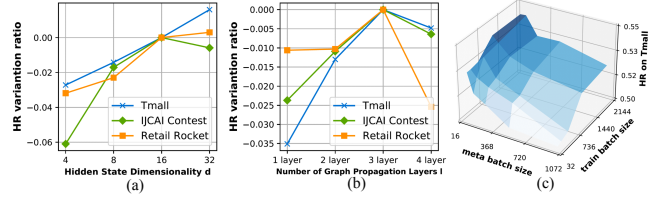

Figure 3: Hyperparameter analysis of CML.

Representation dimensionality d. Our model can achieve good performance with the embedding dimensionality $16 \leq d \leq 32$. It indicates that our CML can boost the performance with small hidden state dimensionality, This can be attributed to effectively enhancing the user-item interaction learning with multiplex relationships.

Batch size in learning process. We search the batch size for our meta contrastive network (meta batch) and the graph neural architecture (train batch) from the range of {128, 256, 512, 1024, 2048} and {256, 512, 1024, 2048, 4096}, respectively. Darker color signals better performance in Figure 3 (c). When the sampled batch size of meta network is smaller than that of base graph network, the model performance becomes better. This configuration will improve the cooperation between our augmented self-supervised learning task and BPR-based ranking objective.

## 4.6 Qualitative Evaluation (RQ5)

In this section, we perform the qualitative evaluation to show the model interpretation with the learned meta contrastive weights across different behavior types. We also visualize the projected behavior embeddings to have a better understanding of our achieved agreement between type-specific behavior embeddings.

Meta Contrastive Weight Visualization. We visualize the learned meta contrastive weights $\omega_u^{k,k'}$ for each auxiliary behavior pairs $(k - k')$ from several sampled users. The customized contrastive weights can be observed in Figure 4 (a), which reflect the personalized multi-behavior interaction patterns of different users. Each $\omega_u^{k,k'}$ value indicates the weight of individual contrastive loss between the target and auxiliary behavior views. For example, for user with id: 27310, the learned weights for the constructed view-buy and favorite-buy contrastive loss is 0.243 and 0.595, respectively. This suggests that this user is more likely to place the order after he or she adds the products into the favorite list, as compared with his/her page view behaviors.

Embedding Visualization. We further show the visualization (2-D projection with t-SNE [38]) of user behavior embeddings encoded from CML and w/o-CLF on IJCAI-Contest data, respectively. In particular, we use different colors to represent different types of behaviors, *i.e.*, red: page view, blue: add-to-favorite, black: add-to-cart, green: purchase. From Figure 4 (b), we observe the embedding agreement achieved by our CML. This again justifies the effectiveness of our CML in alleviating data scarcity issue with the knowledge transfer across different types of behaviors, under our contrastive self-supervised learning architecture.
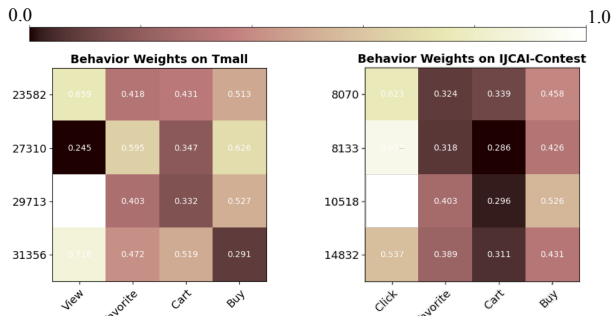
(a) Learned meta contrastive weights across multi-typed behaviors

IJCAI Data: w/o-CLF　　　　Tmall Data: w/o-CLF

IJCAI Data: CML　　　　　Tmall Data: CML
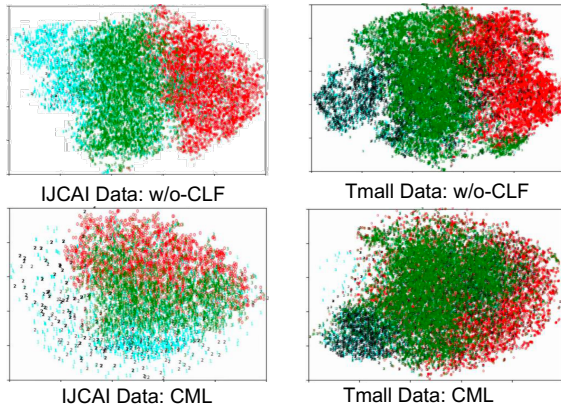
(b) Behavior embedding visualization

Figure 4: Model interpretation study with (a) case studies of personalized contrastive weights from sampled different users; and (b) behavior embedding visualization, *i.e.*, red: page view, blue: add-to-favorite, black: add-to-cart, green: purchase. Best viewed in colors.

5 RELATED WORK

5.1 Graph-based Recommendation Models

Recent studies have demonstrated the promising results offered by GNN-based recommendation models, by using different information propagation functions to aggregate embeddings over neighbors [1, 7, 15, 18, 19, 36]. For example, by stacking multiple embedding propagation layers, NGCF [43] can gather information from neighboring nodes with high-order connectivity. To address the burdensome design of GCN-based message passing in NGCF, LightGCN [15] omits the weight matrix and utilizes the sum-based pooling operation to obtain better recommendation performance. Additionally, to differentiate relations in recommendation, attention-based aggregation functions have been designed for fusing various information in recommender systems, such as social influence [8, 21, 36], knowledge graph embedding [24, 42], textual information [47]. Specifically, GraphRec [8] discriminates influence between users using graph-based attention mechanism. Wu *et al.* [47] develops an attentional graph neural paradigm to enhance the user and item representations with textural information. Motivated by the above research works, our contrastive meta learning framework is built over the graph neural network to capture the behavior-aware collaborative effects between users and items.

5.2 Multi-Behavior Recommender Systems

Under the multi-typed user-item interactions, there exist some recent works attempting to designing effective approaches for handling behavior multiplicity [2, 23, 50–52]. In particular, the behavior-wise relationships are characterized by attention mechanism in [50, 51]. MBGCN [23] learns discriminative behavior representations using graph convolutional network. MATN [50] considers the influences among different types of interactions with attentive weights for pattern aggregation. However, most of them are not designed with the sparse behavior data in mind. To fill this gap, we propose a new model with contrastive learning at behavior semantic levels, which provides auxiliary informative supervision signals for knowledge transferring between behavior types.

5.3 Contrastive Representation Learning

Self-supervised learning techniques have been demonstrated to be effective in learning representations from both image data [6] and textual data [10]. It aims to learn quality discriminative representations by contrasting positive and negative samples from different views. For visual data, different data augmentation strategies (*e.g.*, rotation [13], color distortion [5]) are used to generate negative instances. To better represent the graph topological structures, Deep Graph InfoMax (DGI) [39] aims to maximize the mutual information between node embedding and graph representations based on the original and corrupted graphs. In addition, a model-agnostic recommendation model SGL [48] has been proposed to augment the supervised task of recommendation with auxiliary tasks. It performs dropout operations over the graph connection structures with different strategies, *i.e.*, node dropout, edge dropout and random walk. SMIN [25] is a social-aware recommendation method with generative self-supervision. Inspired by the existing contrastive learning paradigms, this work proposes a new graph contrastive representation framework with the adaptive multi-behavior modeling, by exploring various semantic aspects of user-item interactions.

6 CONCLUSION

In this paper, we develop a novel multi-behavior contrastive meta learning framework for recommendation. Our model learns user representations by preserving behavior heterogeneous context with the agreement between behaviors views constructed from our contrastive learning paradigm. The behavior-aware graph neural architecture with multi-behavior self-supervision bring benefits to the heterogeneous relational learning for recommendation. We perform comprehensive experiments using several real-world datasets to demonstrate the effectiveness of our proposed CML method, by comparing it with various state-of-the-arts.

In this paper, we take the initial step to capture the diverse multi-behavior patterns of users for recommendation under the self-supervised learning paradigm. In the future, it would be interesting to explore the pre-train model strategy of our CML for online user modeling applications (*e.g.*, user profiling). Additionally, another meaningful future research direction can be extending our framework to learn disentangled representations of users, which could reflect the multi-dimensional user interests.

REFERENCES

[1] Chong Chen, Weizhi Ma, Min Zhang, Zhaowei Wang, Xiuqiang He, Chenyang Wang, Yiqun Liu, and Shaoping Ma. 2021. Graph Heterogeneous Multi-Relational Recommendation. In *AAAI*, Vol. 35. 3958–3966.

[2] Chong Chen, Min Zhang, Yongfeng Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2020. Efficient heterogeneous collaborative filtering without negative sampling for recommendation. In *AAAI*, Vol. 34. 19–26.

[3] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. 2020. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *AAAI*, Vol. 34. 3438–3445.

[4] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *SIGIR*. 335–344.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*. PMLR, 1597–1607.

[6] Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. 2020. Disentangled and controllable face image generation via 3d imitative-contrastive learning. In *CVPR*. 5154–5163.

[7] Shaohua Fan, Junxiong Zhu, Xiaotian Han, Chuan Shi, Linmei Hu, Biyu Ma, and Yongliang Li. 2019. Metapath-guided heterogeneous graph neural network for intent recommendation. In *KDD*. 2478–2486.

[8] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *WWW*. 417–426.

[9] Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. 2018. Bilevel programming for hyperparameter optimization and meta-learning. In *ICML*. PMLR, 1568–1577.

[10] Hao Fu, Shaojun Zhou, Qihong Yang, Junjie Tang, Guiquan Liu, Kaikui Liu, and Xiaolong Li. 2021. LRC-BERT: Latent-representation Contrastive Knowledge Distillation for Natural Language Understanding. In *AAAI*, Vol. 35. 12830–12838.

[11] 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 *ICDE*. IEEE, 1554–1557.

[12] Chen Gao, Xiangnan He, Danhua Gan, Xiangning Chen, Fuli Feng, Yong Li, Tat-Seng Chua, Lina Yao, Yang Song, and Depeng Jin. 2019. Learning to recommend with multiple cascading behaviors. *TKDE* (2019).

[13] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. 2018. Unsupervised representation learning by predicting image rotations. In *ICLR*.

[14] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*. JMLR Workshop and Conference Proceedings, 249–256.

[15] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *SIGIR*. ACM.

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

[17] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *WWW*. 2704–2710.

[18] Chao Huang. 2021. Recent Advances in Heterogeneous Relation Learning for Recommendation. In *IJCAI*.

[19] Chao Huang, Jiahui Chen, Lianghao Xia, Yong Xu, Peng Dai, Yanqing Chen, Liefeng Bo, Jiashu Zhao, and Jimmy Xiangji Huang. 2021. Graph-enhanced multi-task learning of multi-level transition dynamics for session-based recommendation. In *AAAI*, Vol. 35. 4123–4130.

[20] 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.

[21] Chao Huang, Huance Xu, Yong Xu, Peng Dai, Lianghao Xia, Mengyin Lu, Liefeng Bo, Hao Xing, Xiaoping Lai, and Yanfang Ye. 2021. Knowledge-aware coupled graph neural network for social recommendation. In *AAAI*.

[22] Houye Ji, Junxiong Zhu, Xiao Wang, Chuan Shi, Bai Wang, Xiaoye Tan, Yanghua Li, and Shaojian He. 2021. Who You Would Like to Share With? A Study of Share Recommendation in Social E-commerce. In *AAAI*, Vol. 35. 232–239.

[23] Bowen Jin, Chen Gao, Xiangnan He, Depeng Jin, et al. 2020. Multi-behavior recommendation with graph convolutional networks. In *SIGIR*. 659–668.

[24] Yong Liu, Susen Yang, Yonghui Xu, Chunyan Miao, Min Wu, and Juyong Zhang. 2021. Contextualized Graph Attention Network for Recommendation with Item Knowledge Graph. *TKDE* (2021).

[25] Xiaoling Long, Chao Huang, Yong Xu, Huance Xu, Peng Dai, Lianghao Xia, and Liefeng Bo. 2021. Social Recommendation with Self-Supervised Metagraph Informax Network. In *CIKM*. 1160–1169.

[26] Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *ICLR*.

[27] Yuanfu Lu, Yuan Fang, and Chuan Shi. 2020. Meta-learning on heterogeneous information networks for cold-start recommendation. In *KDD*. 1563–1573.

[28] Dongsheng Luo, Wei Cheng, Wenchao Yu, Bo Zong, Jingchao Ni, Haifeng Chen, and Xiang Zhang. 2021. Learning to drop: Robust graph neural network via topological denoising. In *WSDM*. 779–787.

[29] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[30] Tao Qi, Fangzhao Wu, Chuhan Wu, and Yongfeng Huang. 2021. Personalized News Recommendation with Knowledge-aware Interactive Matching. In *SIGIR*.

[31] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. Gcc: Graph contrastive coding for graph neural network pre-training. In *KDD*. 1150–1160.

[32] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. *UAI* (2009).

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

[34] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. 2019. Meta-weight-net: Learning an explicit mapping for sample weighting. (2019).

[35] Leslie N Smith. 2017. Cyclical learning rates for training neural networks. In *WACV*. IEEE, 464–472.

[36] 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.

[37] Md Mehrab Tanjim, Congzhe Su, Ethan Benjamin, Diane Hu, Liangjie Hong, and Julian McAuley. 2020. Attentive sequential models of latent intent for next item recommendation. In *WWW*. 2528–2534.

[38] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *JMLR* 9, 11 (2008).

[39] Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2019. Deep Graph Infomax. *ICLR* 2, 3 (2019), 4.

[40] Jizhe Wang, Pipei Huang, Huan Zhao, Zhibo Zhang, Binqiang Zhao, and Dik Lun Lee. 2018. Billion-scale commodity embedding for e-commerce recommendation in alibaba. In *KDD*. 839–848.

[41] Wen Wang, Wei Zhang, Shukai Liu, Qi Liu, Bo Zhang, Leyu Lin, and Hongyuan Zha. 2020. Beyond clicks: Modeling multi-relational item graph for session-based target behavior prediction. In *WWW*. 3056–3062.

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

[43] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *SIGIR*. ACM, 165–174.

[44] Xiao Wang, Nian Liu, Hui Han, and Chuan Shi. 2021. Self-supervised Heterogeneous Graph Neural Network with Co-contrastive Learning. In *KDD*.

[45] Xiao Wang, Ruijia Wang, Chuan Shi, Guojie Song, and Qingyong Li. 2020. Multi-component graph convolutional collaborative filtering. In *AAAI*, Vol. 34. 6267–6274.

[46] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, and Tat-Seng Chua. 2020. Graph-refined convolutional network for multimedia recommendation with implicit feedback. In *MM*. 3541–3549.

[47] Chuhan Wu, Fangzhao Wu, Tao Qi, Suyu Ge, Yongfeng Huang, and Xing Xie. 2019. Reviews meet graphs: Enhancing user and item representations for recommendation with hierarchical attentive graph neural network. In *EMNLP*. 4886–4895.

[48] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *SIGIR*. 726–735.

[49] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In *WSDM*. 153–162.

[50] 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 *SIGIR*. 2397–2406.

[51] Lianghao Xia, Chao Huang, Yong Xu, Peng Dai, Xiyue Zhang, Hongsheng Yang, Jian Pei, and Liefeng Bo. 2021. Knowledge-Enhanced Hierarchical Graph Transformer Network for Multi-Behavior Recommendation. In *AAAI*, Vol. 35. 4486–4493.

[52] Lianghao Xia, Yong Xu, Chao Huang, Peng Dai, and Liefeng Bo. 2021. Graph meta network for multi-behavior recommendation. In *SIGIR*. 757–766.

[53] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *KDD*. ACM, 974–983.

[54] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *NIPS* 33 (2020), 5812–5823.

[55] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. 2019. Heterogeneous graph neural network. In *KDD*. 793–803.

[56] Hanwang Zhang, Yang Yang, Huanbo Luan, Shuicheng Yang, and Tat-Seng Chua. 2014. Start from scratch: Towards automatically identifying, modeling, and naming visual attributes. In *MM*. 187–196.

[57] Lei Zheng, Chun-Ta Lu, Fei Jiang, Jiawei Zhang, and Philip S Yu. 2018. Spectral collaborative filtering. In *Recsys*. 311–319.

[58] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2021. Graph contrastive learning with adaptive augmentation. In *WWW*. 2069–2080.