Diffusion Multi-Behavior Recommender Model

Anchen Li[1,2,5](✉)[0000-0001-9828-6964], Jinglong Ji[2,3][0009-0009-6971-7636], Riting Xia[4][0000-0003-2184-4360], and Bo Yang[1,2](✉)[0000-0003-1927-8419]

[1] College of Computer Science and Technology, Jilin University, Changchun, China
{liac,ybo}@jlu.edu.cn
[2] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, China
[3] College of Artificial Intelligence, Jilin university, China
jijl22@mails.jlu.edu.cn
[4] College of Computer Science, Inner Mongolia University, Hohhot, China
xiart19@mails.jlu.edu.cn
[5] Department of Computer Science, Aalto University, Espoo, Finland

Abstract. Multi-behavior recommender systems, which utilize auxiliary behaviors (e.g., page view, add-to-favorite, and add-to-cart) to assist in predicting user target behaviors (e.g., purchase), are regarded as an effective way to enhance recommendation accuracy and improve user experience. However, noisy and irrelevant information often exists in auxiliary behaviors, which can mislead target behavior predictions and worsen the semantic gap between target and auxiliary behaviors. To address the above challenges, we propose a denoising Diffusion Multi-Behavior Recommender model (DMBR). First, our method employs a graph diffusion paradigm to mitigate the noisy effects in the auxiliary behavior interaction graph. Specifically, we introduce a customized denoising module and a semantic injection mechanism, both leveraging collaborative relationship semantics from target behaviors to guide our graph diffusion process. Then, we predict user target behaviors by leveraging a dual graph learning encoder to model both the target behavior graph and the denoised auxiliary behavior graph. Moreover, our graph learning encoder is equipped with a semantic transfer unit to bridge the semantic gap between behaviors. Experimental results demonstrate the superiority of our DMBR over various state-of-the-art baselines.

Keywords: Multi-behavior Recommender · Denoising · Diffusion Model.

1 Introduction

Recommender systems have become essential tools for delivering personalized information across online platforms (e.g., e-commerce websites [14] and location-based services [47]). The core of a recommender system is to capture user preferences. As a traditional and widely-used recommender algorithm, collaborative filtering (CF) [32,19] models user preferences based on historical behavior data.

However, CF-based methods [8,36,17] usually focus only on a single type of user-item interaction behavior, whereas real-world applications involve multiple behaviors. For example, on e-commerce platforms, in addition to the purchase behavior, users can interact with items in various ways (e.g., page view, add-to-favorite, and add-to-cart). Thus, CF may struggle to effectively capture multi-dimensional collaborative relations, leading to insufficient recommendation.

To this end, multi-behavior recommender systems have emerged. Specifically, these recommendation methods use knowledge learned from auxiliary behaviors (e.g., page view, add-to-favorite, add-to-cart) to assist in predicting users' target behavior (e.g., purchase). Early multi-behavior recommenders primarily use matrix factorization frameworks to model different behavior interaction matrices [23,31]. With the development of graph representation learning, recent mainstream approaches transform the user multi-behavior data into user-item interaction graphs and then apply graph neural networks (GNNs) on these graphs to distill collaborative signals related to different behaviors of users [11,45,44].

Despite effectiveness, the performance of most multi-behavior recommenders can be limited by the presence of noisy and irrelevant information in user auxiliary behaviors. This leads to two challenges: (1) Mitigating noise in auxiliary behaviors: auxiliary behaviors (e.g., page view) often contain noise (e.g., inadvertent or accidental views) that can mislead recommendations under target behaviors (e.g., purchase). Recent studies [49,50,38] attempt to address this issue using contrastive learning, but they either rely on simplistic random augmentation (e.g., behavior dropout), which can drop essential information, or intuitive cross-behavior contrastive operations, which still overlook the noise present in auxiliary behaviors. Thus, an effective algorithm is needed to denoise auxiliary behaviors and characterize user preferences. (2) Transferring informative semantics between behaviors: user auxiliary and target behaviors can be viewed as two domains with different semantic features [55]. For instance, a user's page view behavior (i.e., auxiliary behavior) does not necessarily indicate a subsequent purchase (i.e., target behavior) [39,48]. Knowledge signals from auxiliary behaviors might be incorrectly transferred to the target behavior semantic space, and vice versa. This issue is further exacerbated by the noisy and irrelevant information in auxiliary behaviors. Hence, bridging the semantic gap between the two behavior domains and elegantly transferring semantics is the second challenge.

To this end, we propose a denoising Diffusion Multi-Behavior Recommender model (DMBR). Our method involves two steps: denoising auxiliary behaviors and predicting target behaviors. In the first step, we transform user-item auxiliary behaviors into a bipartite graph and then, inspired by the denoising capability of the diffusion model (DM) [9,29,24], we design a graph diffusion model (GDM) to denoise this graph. Specifically, the auxiliary behavior bipartite graph is initially corrupted by our GDM through the continuous introduction of Gaussian noise. After multiple noise accumulations, the noise is iteratively removed by a customized denoising module that considers the semantic features of the target behavior and the types of auxiliary behavior as guidance, resulting in a denoised auxiliary behavior graph. Additionally, a semantic injection mechanism is developed to integrate the collaborative relation semantics from target behaviors into the GDM, providing better guidance for the graph diffusion process. In the second step, we apply a dual graph learning encoder on the target behavior graph and denoised auxiliary behavior graph to predict user target behaviors. To bridge the semantic gap between target and auxiliary behaviors, we design a semantic transfer unit in our dual graph learning. This unit facilitates the sharing of semantic signals and filters irrelevant information between behaviors.

In a nutshell, our **contributions** are as:

- We propose a multi-behavior recommender to handle the noise problem in auxiliary behaviors and semantic gaps between target and auxiliary behaviors.
- We design a graph diffusion paradigm and a semantic injection mechanism to alleviate the noise in the auxiliary behavior graph.
- We develop a dual graph encoder equipped with semantic transfer units to model the target and auxiliary behavior graphs and bridge semantic gaps.
- Experimental results show the efficacy of our DMBR on the three datasets.

2 Notation and Problem Formulation

In recommender systems, we let $\mathcal{U}$ and $\mathcal{V}$ represent user set and item set.

Multi-Behavior Graph Data. We transform the user multi-behavior data into a target behavior bipartite graph $\mathcal{G}_{\mathcal{T}} = (\mathcal{U} \cup \mathcal{V}, \mathbf{Y}_{\mathcal{T}})$ and an auxiliary behavior bipartite graph $\mathcal{G}_{\mathcal{A}} = (\mathcal{U} \cup \mathcal{V}, \mathbf{Y}_{\mathcal{A}}, \mathbf{C}_{\mathcal{A}})$. In graph $\mathcal{G}_{\mathcal{T}}$ and $\mathcal{G}_{\mathcal{A}}$, both $\mathbf{Y}_{\mathcal{T}} \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{V}|}$ and $\mathbf{Y}_{\mathcal{A}} \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{V}|}$ are user-item interaction matrices, which can be viewed as edges in the graph. The elements $y_{\mathcal{T},uv} \in \mathbf{Y}_{\mathcal{T}}$ and $y_{\mathcal{A},uv} \in \mathbf{Y}_{\mathcal{A}}$ $= 0$ or 1 represent whether user $u \in \mathcal{U}$ interacted with item $v \in \mathcal{V}$ under the target behavior and the auxiliary behavior, respectively. To distinguish the types of auxiliary behaviors between users and items, we introduce a relation matrix $\mathbf{C}_{\mathcal{A}} \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{V}|}$ in $\mathcal{G}_{\mathcal{A}}$, where each element $c_{\mathcal{A},uv}$ is set to 0 (indicating no relationship) or the index of the behavior type between user u and item v.

Task Description. Given the target behavior graph $\mathcal{G}_{\mathcal{T}}$ and original auxiliary behavior graph $\mathcal{G}_{\mathcal{A}}$, our method DMBR first generates a denoised auxiliary behavior graph $\mathcal{G}_{\mathcal{A}^*}$, and then learns a user target behavior prediction function $\mathcal{F}$: $\hat{y}_{uv} = \mathcal{F}(u, v|\Theta, \mathcal{G}_{\mathcal{T}}, \mathcal{G}_{\mathcal{A}^*})$, where $\hat{y}_{uv}$ is the prediction score that u will engage with v under the target behavior and Θ is the parameter of function $\mathcal{F}$.

3 Methodology

Our DMBR includes a graph denoising process and a recommendation framework, as shown in Figure 1. In graph denoising, a graph diffusion model (GDM) is introduced to denoise the original auxiliary behavior graph. In recommendation framework, we use a dual graph learning structure to model both the target behavior graph and denoised auxiliary behavior graph for recommendation.

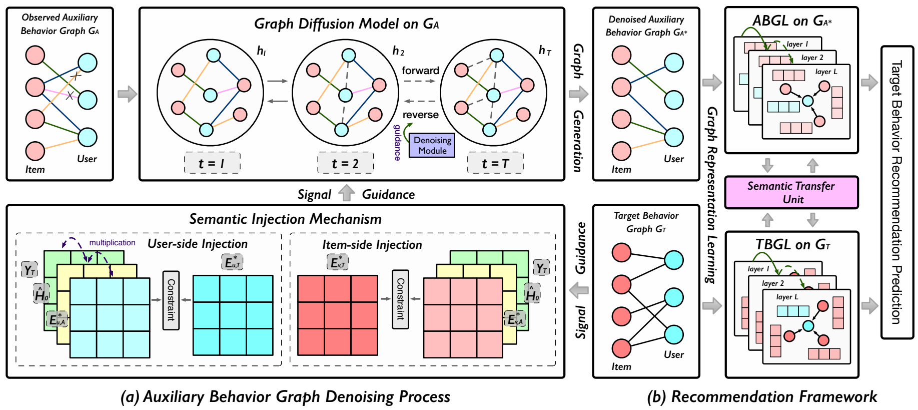

Fig. 1: The structure of our DMBR. Best view in color.

3.1 Graph Denoising

We introduce our graph diffusion model and semantic injection mechanism.

Graph Diffusion Model. Drawing inspiration from the success of diffusion models (DMs) in data denoising [9,28,35,16], we introduce a graph diffusion model (GDM) to alleviate the adverse effects of irrelevant or noisy information in user auxiliary behaviors. Specifically, our GDM denoises observed auxiliary behavior graph $\mathcal{G}_{\mathcal{A}}$, producing denoised graph $\mathcal{G}_{\mathcal{A}^*}$. GDM implements a forward process that incrementally adds noise to the initial graph structure, followed by a reverse process that iteratively removes the noise through customized denoising operations, thereby minimizing the impact of noisy signals.

• **Forward process.** We represent user neighbors in auxiliary behavior graph $\mathcal{G}_{\mathcal{A}}$ as $\mathcal{H}_0 = \{\boldsymbol{h}_{u,0}\}_{u=1}^{|\mathcal{U}|}$, where $\boldsymbol{h}_{u,0} = [h_u^1, \cdots, h_u^{|\mathcal{V}|}]$ is user u's neighbors over item set $\mathcal{V}$ and $h_u^i = 1$ or 0 implies whether u interacts with item i or not. Based on $\mathcal{H}_0$, the forward transition is defined as follows:

$$q(\mathcal{H}_t|\mathcal{H}_{t-1}) = \prod\nolimits_{u=1}^{|\mathcal{U}|} \mathcal{N}(\boldsymbol{h}_{u,t}; \sqrt{1-\beta_t}\boldsymbol{h}_{u,t-1}, \beta_t \boldsymbol{I}), \tag{1}$$

where $t \in \{1, 2, \cdots T\}$ is the diffusion step, $\beta_t \in (0, 1)$ is the noise scale, and $\mathcal{N}$ is the Gaussian distribution.

• **Reverse process.** After T steps, we denoise $\mathcal{H}_T$ in the reverse process through the sequence $\mathcal{H}_T \to \mathcal{H}_{T-1} \cdots \to \mathcal{H}_0$. Specifically, DMs learn the denoising process $\mathcal{H}_t \to \mathcal{H}_{t-1}$, as follows:

$$p_\theta(\mathcal{H}_{t-1}|\mathcal{H}_t) = \prod\nolimits_{u=1}^{|\mathcal{U}|} \mathcal{N}(\boldsymbol{h}_{u,t-1}; \boldsymbol{\mu}_\theta(\mathcal{H}_t, t), \boldsymbol{\Sigma}_\theta(\mathcal{H}_t, t)), \tag{2}$$

in which $\boldsymbol{\mu}_\theta$ and $\boldsymbol{\Sigma}_\theta$ are Gaussian parameters modeled by the denoising module and θ is the parameter of the module. The learning of $\boldsymbol{\Sigma}_\theta$ is commonly ignored to maintain training stability [9,16].

• **Condition Encoders.** Designing meaningful guidance in the denoising process of our GDM is crucial. We consider three key conditions as guidance in our denoising module: (1) Relation types of auxiliary behaviors. Since the auxiliary behavior graph is multi-relational, using relation information can better guide the denoising process. (2) Collaborative filtering signals from target behaviors. Since target behaviors reflect users' true personalized preferences, injecting collaborative semantics from target behaviors into the denoising process allows the module to receive additional guidance. (3) Step information in the diffusion process. Step is another important factor in denoising, as it provides insight into the current noise level of the diffusion process. In light of the above discussion, we define the reverse process, as follows:

$$p_\theta(\mathcal{H}_{t-1}|\mathcal{H}_t) = \prod\nolimits_{u=1}^{|\mathcal{U}|} p_\theta(\boldsymbol{h}_{u,t-1}|\boldsymbol{h}_{u,t}, t, \mathbf{c}_u, \mathbf{u}^*_{\mathcal{T}}), \tag{3}$$

where $\mathbf{c}_u$ is the user adjacency relationship representation (i.e., the u-th row in auxiliary behavior edge matrix $\mathbf{C}_{\mathcal{A}}$) and $\mathbf{u}^*_{\mathcal{T}}$ is the user feature in the target behavior space, which is learned in our dual graph learning (i.e., Equation (22)).
• **Training.** To optimize our GDM, we define its loss function as follows:

$$\mathcal{L}_{\text{GDM}} = \frac{1}{|\mathcal{U}|} \sum\nolimits_{u=1}^{|\mathcal{U}|} \mathbb{E}_{\boldsymbol{h}_{u,t}, \boldsymbol{h}_{u,0}, t}[||f_\theta(\boldsymbol{h}_{u,t}, t, \mathbf{c}_u, \mathbf{u}^*_{\mathcal{T}}) - \boldsymbol{h}_{u,0}||_2^2], \tag{4}$$

where f_θ is the denoising module for GDM. We define it using a two-layer neural network, and its input is the concatenation of the four inputs.

Denoised Graph Generation. After training, a denoised user auxiliary behavior graph $\mathcal{G}_{\mathcal{A}^*}$ can be generated. In the forward process, GDM corrupts $\boldsymbol{h}_{u,0}$ as $\boldsymbol{h}_{u,0} \to \boldsymbol{h}_{u,1} \to \cdots \to \boldsymbol{h}_{u,T'}$ over T' steps. Then, in the reverse process, GDM sets $\bar{\boldsymbol{h}}_{u,T} = \boldsymbol{h}_{u,T'}$ and executes denoising as $\bar{\boldsymbol{h}}_{u,T} \to \bar{\boldsymbol{h}}_{u,T-1} \to \cdots \to \bar{\boldsymbol{h}}_{u,0}$ for T steps. Next, for each user u, we have $\bar{\boldsymbol{h}}_{u,0} = [\bar{h}_u^1, \cdots, \bar{h}_u^{|\mathcal{V}|}]$. We first select top k_u value $\bar{h}_u^i$ from $\bar{\boldsymbol{h}}_{u,0}$. Then, we take the items corresponding to these k_u values as the candidate neighbor set and intersect it with the original item neighbor set in $\mathcal{G}_{\mathcal{A}}$ to obtain the user neighborhood of denoised auxiliary behavior graph $\mathcal{G}_{\mathcal{A}^*}$.

Semantic Injection Mechanism. We propose a semantic injection (SI) mechanism to further guide our GDM in generating a suitable denoised auxiliary behavior graph for recommendation. We consider injecting the collaborative relations from the user and item sides of the target behaviors into the GDM.

We first use the interaction matrix $\mathbf{Y}_{\mathcal{T}}$ in the target behavior graph $\mathcal{G}_{\mathcal{T}}$ to update the denoised matrix $\hat{\boldsymbol{H}}_0$ (by combining $\hat{\boldsymbol{h}}_{u,0}$ predicted by f_θ), resulting in a user-user relation matrix $\mathbf{M}_u$ and an item-item relation matrix $\mathbf{M}_v$:

$$\mathbf{M}_u = \mathbf{Y}_{\mathcal{T}} \hat{\boldsymbol{H}}_0^\top, \tag{5}$$

$$\mathbf{M}_v = \mathbf{Y}_{\mathcal{T}}^\top \hat{\boldsymbol{H}}_0, \tag{6}$$

Next, we combine them with the user/item auxiliary behavior embedding $\mathbf{E}^*_{u,\mathcal{A}}/\mathbf{E}^*_{v,\mathcal{A}}$ based on graph convolution ideas [10,16] and use mean squared error to align it with the user/item target behavior embedding $\mathbf{E}^*_{u,\mathcal{T}}/\mathbf{E}^*_{v,\mathcal{T}}$:

$$\mathcal{L}_{\text{SI}} = \underbrace{\left\|\mathbf{M}_u\mathbf{E}^*_{u,\mathcal{A}} - \mathbf{E}^*_{u,\mathcal{T}}\right\|_2^2}_{\text{User-side Injection}} + \underbrace{\left\|\mathbf{M}_v\mathbf{E}^*_{v,\mathcal{A}} - \mathbf{E}^*_{v,\mathcal{T}}\right\|_2^2}_{\text{Item-side Injection}}, \tag{7}$$

where embedding matrices $\mathbf{E}^*_{u,\mathcal{A}}$, $\mathbf{E}^*_{u,\mathcal{T}}$, $\mathbf{E}^*_{v,\mathcal{A}}$, and $\mathbf{E}^*_{v,\mathcal{T}}$ are learned from our dual graph modeling (*cf.* Equations (20) - (23) in Section 3.2).

In practice, to improve efficiency, we perform the above operations using batch data rather than the entire matrix. We believe that our SI mechanism not only integrates collaborative signals from the target behavior but also bridges the semantic gap between target and auxiliary behaviors.

Graph Denoising Training. To train our graph denoising module, we integrate the losses from GDM and semantic injection mechanism, as follows:

$$\mathcal{L}_{\text{De}} = \mathcal{L}_{\text{GDM}} + \mathcal{L}_{\text{SI}}. \tag{8}$$

3.2 Recommendation Framework

In this section, we introduce our recommendation framework, which includes the dual graph learning and semantic transfer unit.

Dual Graph Learning. Based on the target behavior graph $\mathcal{G}_{\mathcal{T}}$ and generated denoised auxiliary behavior graph $\mathcal{G}_{\mathcal{A}^*}$ (from in Section 3.1), we introduce a dual graph learning structure to model representations of user and item nodes.

Given a node u and its initial embedding $\mathbf{u} \in \mathbb{R}^d$, we use target behavior graph learning (TBGL) and auxiliary behavior graph learning (ABGL) to model the representation of node u's in target and auxiliary behavior graphs:

$$\mathbf{u}^l_{\mathcal{T}} = \sum\nolimits_{v\in\mathcal{N}^{\mathcal{T}}_u} p_{uv}\mathbf{v}^{l-1}_{\mathcal{T}}, \tag{9}$$

$$\mathbf{u}^l_{\mathcal{A}} = \sum\nolimits_{v\in\mathcal{N}^{\mathcal{A}}_u} q_{uv}\mathbf{v}^{l-1}_{\mathcal{A}}, \tag{10}$$

where $\mathbf{u}^0_{\mathcal{T}} = \mathbf{u}^0_{\mathcal{A}} = \mathbf{u}$ is the initial embedding, and $\mathcal{N}^{\mathcal{T}}_u$ and $\mathcal{N}^{\mathcal{A}}_u$ are neighborhoods for node u in graphs $\mathcal{G}_{\mathcal{T}}$ and $\mathcal{G}_{\mathcal{A}^*}$, respectively.

Following [7], in TBGL, we set p_{uv} as $1/\sqrt{|\mathcal{N}^{\mathcal{T}}_u|}\sqrt{|\mathcal{N}^{\mathcal{T}}_v|}$. In addition, unlike graph $\mathcal{G}_{\mathcal{T}}$, graph $\mathcal{G}_{\mathcal{A}^*}$ is a multi-relational graph. We employ a relation-aware attention mechanism to design ABGL. The attention value q_{uv} is defined as:

$$q^l_{uv} = \frac{\exp\left(\sigma\left(\mathbf{r}^\top_{uv}\mathbf{w}^l(\mathbf{u}^{l-1}_{\mathcal{A}} \odot \mathbf{v}^{l-1}_{\mathcal{A}})\right)\right)}{\sum_{\tilde{v}\in\mathcal{N}^{\mathcal{A}}_u}\exp\left(\sigma\left(\mathbf{r}^\top_{u\tilde{v}}\mathbf{w}^l(\mathbf{u}^{l-1}_{\mathcal{A}} \odot \tilde{\mathbf{v}}^{l-1}_{\mathcal{A}})\right)\right)}, \tag{11}$$

where σ denotes the activation function, $\mathbf{w}$ is the weight matrix, $\mathbf{r}_{uv}$ is the relation embedding between node u and v, and $\odot$ is the element-wise multiplication.

Semantic Transfer Unit. Since target behaviors and auxiliary behaviors can be viewed as two domains with a semantic gap between them [55], bridging this gap is crucial for modeling user preferences. To achieve this, inspired by cross-domain and transfer learning [33,27,57,16,15], we design a semantic transfer (ST) unit to bridge the gap between the TBGL and ABGL, thereby establishing semantic communication and mutual learning between two domains.

Specifically, the ST unit first takes node representations $\mathbf{u}_{\mathcal{T}}$ from TBGL and $\mathbf{u}_{\mathcal{A}}$ from ABGL at each layer as inputs, and then the two inputs extract semantic information from each other to update themselves, as $[\mathbf{u}^{\star}_{\mathcal{T}}, \mathbf{u}^{\star}_{\mathcal{A}}] = \mathrm{ST}(\mathbf{u}_{\mathcal{T}}, \mathbf{u}_{\mathcal{A}})$. We use a suffix $[\mathcal{T}]$ or $[\mathcal{A}]$ to distinguish the two outputs of our ST unit:

$$\mathbf{u}^{\star}_{\mathcal{T}} = \mathrm{ST}(\mathbf{u}_{\mathcal{T}}, \mathbf{u}_{\mathcal{A}})[\mathcal{T}] = \mathbf{u}_{\mathcal{T}} + \mathrm{G}_{\mathcal{T}} \odot f_{\psi}(\mathbf{u}_{\mathcal{T}}, \mathbf{u}_{\mathcal{A}}), \tag{12}$$

$$\mathbf{u}^{\star}_{\mathcal{A}} = \mathrm{ST}(\mathbf{u}_{\mathcal{T}}, \mathbf{u}_{\mathcal{A}})[\mathcal{A}] = \mathbf{u}_{\mathcal{A}} + \mathrm{G}_{\mathcal{A}} \odot f_{\varphi}(\mathbf{u}_{\mathcal{A}}, \mathbf{u}_{\mathcal{T}}). \tag{13}$$

In the ST unit, we design functions f_{ψ} and f_{φ} for sufficient feature interaction and gating mechanisms $\mathrm{G}_{\mathcal{T}}$ and $\mathrm{G}_{\mathcal{A}}$ for filtering information, as follows:

$$f_{\psi}(\mathbf{u}_{\mathcal{T}}, \mathbf{u}_{\mathcal{A}}) = \mathbf{u}_{\mathcal{T}} \mathbf{u}_{\mathcal{A}}^{\top} \mathbf{w}_{\psi} + \mathbf{b}_{\psi}, \tag{14}$$

$$\mathrm{G}_{\mathcal{T}} = \tanh\big(\mathbf{w}_{g_{\mathcal{T}}}(\mathbf{u}_{\mathcal{T}} \odot \mathbf{u}_{\mathcal{A}})\big), \tag{15}$$

$$f_{\varphi}(\mathbf{u}_{\mathcal{A}}, \mathbf{u}_{\mathcal{T}}) = \mathbf{u}_{\mathcal{A}} \mathbf{u}_{\mathcal{T}}^{\top} \mathbf{w}_{\varphi} + \mathbf{b}_{\varphi}, \tag{16}$$

$$\mathrm{G}_{\mathcal{A}} = \tanh\big(\mathbf{w}_{g_{\mathcal{A}}}(\mathbf{u}_{\mathcal{A}} \odot \mathbf{u}_{\mathcal{T}})\big), \tag{17}$$

where $\mathbf{w}_{\psi}, \mathbf{w}_{\varphi}, \mathbf{w}_{g_{\mathcal{T}}}, \mathbf{w}_{g_{\mathcal{A}}}$, are weights, $\mathbf{b}_{\psi}$ and $\mathbf{b}_{\varphi}$ are bias vectors.

Function f first uses the outer product operation and a linear transformation to model the interaction between any elements of two input vectors. Then, a gating mechanism G is applied to further filter the information. Finally, the refined vector is integrated into the target vector representation.

We equip the aforementioned ST unit into our dual graph encoder (i.e., Equations.(9)-(11)). The matrix form of the layer-wise propagation rules is as:

$$\mathbf{E}^{l}_{\mathcal{T}} = \big(\mathbf{D}_{\mathcal{T}}^{-\frac{1}{2}} \mathbf{A}_{\mathcal{T}} \mathbf{D}_{\mathcal{T}}^{-\frac{1}{2}}\big) \cdot \mathrm{ST}\big(\mathbf{E}^{l-1}_{u,\mathcal{T}}, \mathbf{E}^{l-1}_{u,\mathcal{A}}\big)[\mathcal{T}], \tag{18}$$

$$\mathbf{E}^{l}_{\mathcal{A}} = \mathrm{ATT}(\mathbf{E}^{l-1}_{\mathcal{A}}, \mathcal{G}_{\mathcal{A}^*}) \cdot \mathrm{ST}\big(\mathbf{E}^{l-1}_{u,\mathcal{T}}, \mathbf{E}^{l-1}_{u,\mathcal{A}}\big)[\mathcal{A}], \tag{19}$$

where $\mathbf{D}_{\mathcal{T}}$ and $\mathbf{A}_{\mathcal{T}}$ are diagonal and adjacency matrices for graph $\mathcal{G}_{\mathcal{T}}$; $\mathbf{E}_{\mathcal{T}}$ and $\mathbf{E}_{\mathcal{A}}$ are node embedding matrices from TBGL and ABGL; ATT is ABGL's attention function (Equation (11)), which calculates the attention value matrix for the current layer based on the embeddings of previous layer and graph $\mathcal{G}_{\mathcal{A}^*}$.

Recommendation Framework Training. After modeling with TBGL and ABGL, we combine the embeddings from each layer to obtain the final node representations in the target behavior space and the auxiliary behavior space:

$$\mathbf{E}^{*}_{\mathcal{T}} = \frac{1}{L+1} \sum\nolimits_{l=0}^{L} \mathbf{E}^{l}_{\mathcal{T}}, \tag{20}$$

$$\mathbf{E}^{*}_{\mathcal{A}} = \frac{1}{L+1} \sum\nolimits_{l=0}^{L} \mathbf{E}^{l}_{\mathcal{A}}. \tag{21}$$

Next, given a user u and an item v, we use the lookup operation $[\cdot]_u,\ [\cdot]_v$ to obtain their embeddings, as follows:

$$\mathbf{u}^*_{\mathcal{T}} = \left[\mathbf{E}^*_{\mathcal{T}}\right]_u, \quad \mathbf{u}^*_{\mathcal{A}} = \left[\mathbf{E}^*_{\mathcal{A}}\right]_u, \tag{22}$$

$$\mathbf{v}^*_{\mathcal{T}} = \left[\mathbf{E}^*_{\mathcal{T}}\right]_v, \quad \mathbf{v}^*_{\mathcal{A}} = \left[\mathbf{E}^*_{\mathcal{A}}\right]_v. \tag{23}$$

To model user u's preference for item v under the target behavior, we consider both target and auxiliary behavior features, as follows:

$$\hat{y}_{uv} = \left(\mathbf{u}^*_{\mathcal{T}} + \mathbf{u}^*_{\mathcal{A}}\right)^{\top} \left(\mathbf{v}^*_{\mathcal{T}} + \mathbf{v}^*_{\mathcal{A}}\right). \tag{24}$$

To train the framework, we define the loss function as:

$$\mathcal{L}_{\text{Rec}} = \sum\nolimits_{(u,v,\tilde{v})\in\mathcal{D}} - \ln \sigma(\hat{y}_{uv} - \hat{y}_{u\tilde{v}}) + \lambda||\Theta||_2^2, \tag{25}$$

where Θ is the parameter and $\mathcal{D} = \left\{(u, v, \tilde{v}) \mid u \in \mathcal{U} \wedge v \in \mathcal{N}_u^{\mathcal{T}} \wedge \tilde{v} \in \mathcal{V}\backslash\mathcal{N}_u^{\mathcal{T}}\right\}$.

3.3 Model Optimization

To optimize our DMBR, we combine the denoising loss (i.e., Equation (8)) and the recommendation loss (i.e., Equation (25)), as follows:

$$\mathcal{L}_{\text{DMBR}} = \mathcal{L}_{\text{De}} + \mathcal{L}_{\text{Rec}}, \tag{26}$$

where two loss terms are optimized alternately.

4 Experiment

In this section, we first introduce the experimental setup and then conduct experiments to analyze the effectiveness of our method DMBR.

4.1 Experimental Setup

Datasets. We use three datasets: IJCAI, Tmall, and Yelp. Table 1 shows the statistics, where the bolded behavior is the target behavior.

Table 1: Statistical details of the three datasets.

Datasets	Users#	Items#	Interaction#	Interactive Behavior Type
IJCAI	17,435	15,626	554,661	{View, Favorite, Cart, **Purchase**}
Tmall	30,407	21,381	1,299,441	{View, Favorite, Cart, **Purchase**}
Yelp	15,578	14,943	1,156,788	{Dislike, Neutral, Tip, **Like**}

Baselines. We compare our approach DMBR with two types of baselines: (1) single-behavior recommenders (NeuCF [8] and DiffRec [35]); (2) multi-behavior recommenders (MBGCN [11], MB-GMN [45], MBSSL [49], and KMCLR [50]). The characteristics of these baselines are introduced as follows:

(1) Single-Behavior Recommendation Baselines:

- NeuCF integrates matrix factorization model with feedforward neural networks to model user-item interaction relations [8].
- DiffRec utilizes the diffusion model to reconstruct user historical interactions and infer user preferences for items [35].

(2) Multi-Behavior Recommendation Baselines:

- MBGCN propagates behavior-aware embeddings on a unified user multi-behavior graph to model user multiple behaviors [11].
- MB-GMN enhances multi-behavior modeling by using a graph meta network that incorporates the meta-learning paradigm [45].
- MBSSL leverages a behavior-aware graph neural network and a self-supervised learning paradigm at inter-behavior and intra-behavior levels to mitigate data noise and model user preferences [49].
- KMCLR leverages item relationships and introduces the multi-behavior contrastive learning mechanism to alleviate noise and enhance the learning of user multi-behavior patterns [50].

Hyper-parameter Setting. We utilizes Adam [12] as the optimizing strategy for all methods. For our model DMBR, we implemented it using Pytorch. Some settings are as follows: diffusion step T is turned in $[2, 100]$, embedding size is set to 32, and GNN layer is set to 2. For baseline methods, hyper-parameters are carefully tuned to achieve optimal performance.

Evaluation Metric. Following the related work [50,55,45], we utilize leave-one-out evaluation. Normalized discounted cumulative gain (N@K) and hit ratio (H@K) are employed as the metrics, and K is set as 20 by default.

4.2 Performance Comparison

We present the comparison results in Table 2 and find that:

- Generally, single-behavior recommender models perform worse than multi-behavior ones because the latter consider more user-item interaction data. However, this is not always the case. For instance, DiffRec sometimes outperforms MBGCN and MB-GMN, suggesting that user auxiliary behaviors may contain noise or irrelevant information.
- Among the baseline models, multi-behavior recommender methods with noise reduction capabilities (i.e., MBSSL and KMCLR) generally achieve better performance. This further emphasizes the importance of filtering information from user auxiliary behaviors.

- Our DMBR achieves the best performance. We attribute these improvements to DMBR's ability to filter noise through the graph denoising and the effectiveness of its recommendation framework in bridging the behavioral semantic gap. In Section 4.3, we validate the effectiveness of key designs.

Table 2: Model performance results on the three datasets.

Method	IJCAI		Tmall		Yelp	
	N@20	H@20	N@20	H@20	N@20	H@20
NeuCF	0.0161	0.0330	0.0242	0.0503	0.0214	0.0547
DiffRec	0.0215	0.0477	0.0304	0.0629	0.0336	0.0782
MBGCN	0.0242	0.0505	0.0287	0.0585	0.0343	0.0829
MB-GMN	0.0264	0.0533	0.0355	0.0771	0.0320	0.0753
MBSSL	0.0287	0.0599	0.0378	0.0793	0.0374	0.0918
KMCLR	0.0323	0.0660	0.0376	0.0804	0.0402	0.0937
DMBR	**0.0352**	**0.0695**	**0.0396**	**0.0860**	**0.0445**	**0.1010**

4.3 Ablation Study

We conduct ablation experiments to thoroughly analyze our DMBR.

Effect of Model Design. To analyze the components in our method DMBR, we design the following four operations:

- w/o GDM: We remove the auxiliary behavior Graph Diffusion Model (GDM) from the denoising process.
- w/o R&C: We remove the Relation and Collaborative signal guidance from the denoising module in the auxiliary behavior graph denoising process.
- w/o SI: We remove the Semantic Injection (SI) mechanism from the auxiliary behavior graph denoising process.
- w/o ST: We remove the Semantic Transfer (ST) unit from dual graph learning of the recommendation framework.

Table 3 shows the results, and we observe that:

- Removing the GDM results in a significant decline in the model performance, indicating that GDM plays a crucial role in mitigating noise in the user auxiliary behavior graph.
- The operations w/o R&C and w/o SI both degrade the model's performance, highlighting the importance of guidance during the denoising process.
- Excluding the ST unit makes it difficult for the model to effectively bridge the behavioral semantic gap, leading to reduced recommendation quality.
- Overall, removing any component from our DMBR leads to a decrease in performance, showing the soundness of our model's design.

Table 3: Ablation study of key designs on the three datasets.

Operation	IJCAI		Tmall		Yelp	
	N@20	H@20	N@20	H@20	N@20	H@20
w/o GDM	0.0315	0.0643	0.0367	0.0794	0.0408	0.0942
w/o R&C	0.0340	0.0688	0.0383	0.0831	0.0427	0.0965
w/o SI	0.0331	0.0666	0.0380	0.0820	0.0436	0.0982
w/o ST	0.0327	0.0671	0.0385	0.0844	0.0416	0.0950
DMBR	**0.0352**	**0.0695**	**0.0396**	**0.0860**	**0.0445**	**0.1010**

Effect of Semantic Transfer Unit. In our dual graph learning, we design a semantic transfer (ST) unit to bridge the semantic gap between behavior domains. To validate the effectiveness of our ST unit, we introduce several commonly used modules (i.e., cross-stitch (CS) [26], cross-compress (CC) unit [33], and feature evolution (FE) unit [41]), which have similar semantic transfer functions, from cross-domain and transfer learning areas to replace our ST unit in dual graph learning. The results on two datasets are shown in Figure 2. We observe that replacing our ST unit with these other units led to a decrease in performance. We attribute the performance improvement brought by the ST unit to its interaction function f_ψ, f_φ and gating mechanism $\mathrm{G}_\mathcal{T}, \mathrm{G}_\mathcal{A}$.

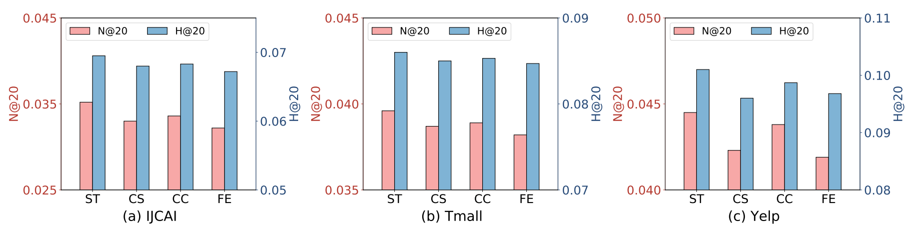

Fig. 2: Effect of the semantic transfer unit on the three datasets.

4.4 Hyper-parameter Sensitivity Analysis

We analyze several important hyperparameters involved in DMBR: diffusion step T, embedding size d, and GNN layer size L. The results on the IJCAI dataset are shown in Figure 3. We observe that:

- Initially, the model performance improves as the diffusion step T increases, but it eventually stabilizes and starts to decline.
- The model's performance improves as the latent dimension increases, but too large a size might lead to overfitting. On the IJCAI dataset, we recommend setting d to 32 or 64.

- Although increasing layer size L can enhance the performance to some extent, it also introduces computational overhead. We set $L = 2$ to balance model performance and costs.

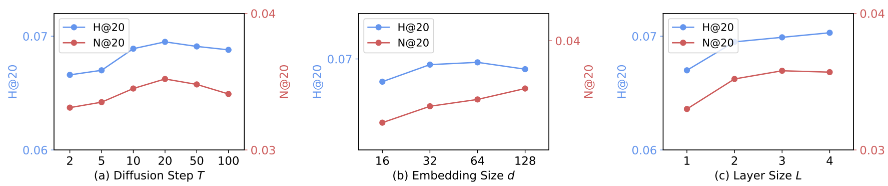

Fig. 3: Hyper-parameter sensitivity analysis on the IJCAI dataset.

5 Related Work

In this section, we review three relevant research areas: multi-behavior recommendation, denoising recommendation, and graph-based recommendation.

5.1 Multi-Behavior Recommendation

Due to the presence of various user-item interaction behaviors in recommendation scenarios (e.g., page view, add-to-favorite, add-to-cart, and purchase), multi-behavior recommender systems have emerged. Early methods primarily use matrix factorization strategies to model different behavior interaction matrices [23,31]. Some follow-on studies [5,43] leverage the deep learning components to further capture the nonlinear relationships between user-item interactions. In recent years, mainstream approaches [11,45,44] employ graph neural networks (GNNs) because interaction behaviors can naturally be represented as bipartite graph structures. Despite effectiveness, most of these methods overlook the noise in auxiliary behaviors and the semantic gap between target and auxiliary behaviors. Recent studies [49,50,38] attempts to address this issue using contrastive learning, but they either rely on simplistic random augmentation (e.g., behavior dropout), which can drop essential information, or intuitive cross-behavior contrastive operations, which still overlook the noise present in auxiliary behaviors. To this end, in this paper, we first develop a diffusion model-based graph denoising strategy to mitigate noise in the auxiliary behavior graph. Then, we introduce a dual graph learning structure equipped with the semantic transfer unit to model both target behavior graph and auxiliary behavior graph, bridging the semantic gap between behaviors.

5.2 Denoising Recommendation

In recommendation scenarios, interaction data often contains noise, which will negatively impact the performance of recommenders. To address this issue, two common classes of methods have been proposed [56,16]. The first class introduces data cleaning strategies that usually rely on some heuristic assumptions to remove the noisy information [3,4,53] or assign lower weights to the noisy information [34,37,2]. The second class is the model-based method, which improves the noise resistance of the model [6,40,42]. Recently, with the emergence of diffusion models (DMs) [9], DM-based denoising recommender methods [35,56,16] have gained significant attention. However, these methods are difficult to apply to the multi-behavior recommendation. In this work, we design a DM paradigm tailored for multi-behavior recommendation. Specifically, our graph diffusion model incorporates a customized denoising module and a semantic injection mechanism to mitigate noise in auxiliary behavior graphs.

5.3 Graph-based Recommendation

Graph representation learning is widely used in many fields [54,30,46]. Graph-based recommender systems model input data as a graph structure and design graph-learning models for recommendation. Traditional approaches primarily rely on network embedding techniques (e.g., random walks [1]) to build models. Recently, with the development of graph neural networks (GNNs), GNN-based recommenders have been widely applied in various scenarios, including collaborative filtering [18,21], social recommendation [25,22], and knowledge graph-based recommendation [15,20]. In this paper, we focus on using graph representation learning for multi-behavior recommendation. We design a dual GNN to model the target and auxiliary behavior graphs. To bridge the semantic gap during the modeling of two graphs, we introduce a semantic transfer unit. We believe that this dual graph learning structure can be used to other recommender scenarios.

6 Conclusion

In this paper, we propose a multi-behavior recommender DMBR. In DMBR, we first introduce a graph diffusion paradigm to mitigate noise in the auxiliary behavior graph. To guide the graph diffusion process, we develop a customized denoising module and a semantic injection mechanism. Then, we design a recommendation framework that utilizes a dual graph learning structure to model both the target behavior graph and the denoised auxiliary behavior graph for recommendation. To bridge the semantic gap between behaviors, we equip the dual graph learning structure with a semantic transfer unit. Experimental results show that our DMBR outperforms various state-of-the-art recommender baselines. Ablation studies further validate the rationale and effectiveness of our method DMBR. We believe that the novel perspectives and encouraging results in this work will inspire other research fields, such as bioinformatics [13] and visual relationship detection [52,51].

Acknowledgement

The authors thank all the anonymous reviewers for their helpful comments. This work was supported by the National Natural Science Foundation of China under Grant Nos. 62402197, U22A2098, 62172185, 62206105, and 62202200; the National Science and Technology Major Project under Grant No. 2021ZD0112500.

References

1. Baluja, S., et al: Video suggestion and discovery for youtube: taking random walks through the view graph. In: WWW. pp. 895–904 (2008)
2. Chen, C., Zhang, M., et al: An efficient adaptive transfer neural network for social-aware recommendation. In: SIGIR. pp. 225–234 (2019)
3. Ding, J., Feng, F., He, X., Yu, G., Li, Y., Jin, D.: An improved sampler for bayesian personalized ranking by leveraging view data. In: WWW. pp. 13–14 (2018)
4. Ding, J., Yu, G., He, X., Feng, F., Li, Y., Jin, D.: Sampler design for bayesian personalized ranking by leveraging view data. TKDE **33**(2), 667–681 (2019)
5. Gao, C., He, X., et al: Neural multi-task recommendation from multi-behavior data. In: ICDE. pp. 1554–1557. IEEE (2019)
6. Gao, Y., Du, Y., Hu, Y., Chen, L., Zhu, X., Fang, Z., Zheng, B.: Self-guided learning to denoise for robust recommendation. In: SIGIR. pp. 1412–1422 (2022)
7. He, X., Deng, K., et al: Lightgcn: Simplifying and powering graph convolution network for recommendation. In: SIGIR. pp. 639–648 (2020)
8. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: WWW. pp. 173–182 (2017)
9. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. NIPS **33**, 6840–6851 (2020)
10. Jiang, Y., Yang, Y., Xia, L., Huang, C.: Diffkg: Knowledge graph diffusion model for recommendation. In: WSDM. pp. 313–321 (2024)
11. Jin, B., Gao, C., He, X., Jin, D., Li, Y.: Multi-behavior recommendation with graph convolutional networks. In: SIGIR. pp. 659–668 (2020)
12. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
13. Li, A., Casiraghi, E., Rousu, J.: Chemical reaction enhanced graph learning for molecule representation. Bioinformatics **40**(10), btae558 (2024)
14. Li, A., Liu, X., Yang, B.: Item attribute-aware graph collaborative filtering. Expert Systems with Applications **238**, 122242 (2024)
15. Li, A., Yang, B.: Gsirec: Learning with graph side information for recommendation. World Wide Web **24**(5), 1411–1437 (2021)
16. Li, A., Yang, B.: Dual graph denoising model for social recommendation. In: WWW (2025)
17. Li, A., Yang, B., Huo, H., Chen, H., Xu, G., Wang, Z.: Hyperbolic neural collaborative recommender. TKDE **35**(9), 9114–9127 (2022)
18. Li, A., Yang, B., Huo, H., Hussain, F.: Hypercomplex graph collaborative filtering. In: WWW. pp. 1914–1922 (2022)
19. Li, A., Yang, B., Huo, H., Hussain, F.K.: Leveraging implicit relations for recommender systems. Information Sciences **579**, 55–71 (2021)
20. Li, A., Yang, B., Huo, H., Hussain, F.K., Xu, G.: Structure-and logic-aware heterogeneous graph learning for recommendation. In: ICDE. pp. 544–556. IEEE (2024)

21. Li, A., Yang, B., Huo, H., Hussain, F.K., Xu, G.: Self-supervised dual graph learning for recommendation. Knowledge-Based Systems **310**, 112967 (2025)
22. Li, A., Yang, B., Hussain, F.K., Huo, H.: Hsr: Hyperbolic social recommender. Information Sciences **585**, 275–288 (2022)
23. Loni, B., Pagano, R., Larson, M., Hanjalic, A.: Bayesian personalized ranking with multi-channel user feedback. In: RecSys. pp. 361–364 (2016)
24. Lovelace, J., Kishore, V., Wan, C., Shekhtman, E., Weinberger, K.Q.: Latent diffusion for language generation. NIPS **36** (2024)
25. Miao, H., Li, A., Yang, B.: Meta-path enhanced lightweight graph neural network for social recommendation. In: DASFAA. pp. 134–149. Springer (2022)
26. Misra, I., Shrivastava, A., Gupta, A., Hebert, M.: Cross-stitch networks for multi-task learning. In: CVPR. pp. 3994–4003 (2016)
27. Pan, S.J., Yang, Q.: A survey on transfer learning. TKDE **22**(10), 1345–1359 (2009)
28. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR. pp. 10684–10695 (2022)
29. Saharia, C., et al: Photorealistic text-to-image diffusion models with deep language understanding. NIPS **35**, 36479–36494 (2022)
30. Shi, D., Li, A., Yang, B.: Counterfactual-guided and curiosity-driven multi-hop reasoning over knowledge graph. In: DASFAA. pp. 171–179. Springer (2022)
31. Singh, A.P., Gordon, G.J.: Relational learning via collective matrix factorization. In: KDD. pp. 650–658 (2008)
32. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Advances in artificial intelligence **2009**(1), 421425 (2009)
33. Wang, H., et al: Multi-task feature learning for knowledge graph enhanced recommendation. In: WWW. pp. 2000–2010 (2019)
34. Wang, W., Feng, F., He, X., Nie, L., Chua, T.S.: Denoising implicit feedback for recommendation. In: WSDM. pp. 373–381 (2021)
35. Wang, W., Xu, Y., Feng, F., Lin, X., He, X., Chua, T.S.: Diffusion recommender model. In: SIGIR. pp. 832–841 (2023)
36. Wang, X., He, X., Wang, M., Feng, F., Chua, T.S.: Neural graph collaborative filtering. In: SIGIR. pp. 165–174 (2019)
37. Wang, Y., Xin, X., Meng, Z., Jose, J.M., Feng, F., He, X.: Learning robust recommenders through cross-model agreement. In: WWW (2022)
38. Wei, W., Huang, C., Xia, L., Xu, Y., Zhao, J., Yin, D.: Contrastive meta learning with behavior multiplicity for recommendation. In: WSDM. pp. 1120–1128 (2022)
39. Wen, H., et al: Entire space multi-task modeling via post-click behavior decomposition for conversion rate prediction. In: SIGIR. pp. 2377–2386 (2020)
40. Wu, J., Wang, X., Feng, F., He, X., Chen, L., Lian, J., Xie, X.: Self-supervised graph learning for recommendation. In: SIGIR (2021)
41. Wu, Q., Jiang, L., et al: Feature evolution based multi-task learning for collaborative filtering with social trust. In: IJCAI. pp. 3877–3883 (2019)
42. Wu, Y., DuBois, C., Zheng, A.X., Ester, M.: Collaborative denoising auto-encoders for top-n recommender systems. In: WSDM (2016)
43. Xia, L., et al: Multiplex behavioral relation learning for recommendation via memory augmented transformer network. In: SIGIR. pp. 2397–2406 (2020)
44. Xia, L., Huang, C., Xu, Y., Dai, P., Zhang, X., Yang, H., Pei, J., Bo, L.: Knowledge-enhanced hierarchical graph transformer network for multi-behavior recommendation. In: AAAI. vol. 35, pp. 4486–4493 (2021)
45. Xia, L., Xu, Y., Huang, C., Dai, P., Bo, L.: Graph meta network for multi-behavior recommendation. In: SIGIR. pp. 757–766 (2021)

46. Xia, R., Zhang, C., Li, A., Liu, X., Yang, B.: Attribute imputation autoencoders for attribute-missing graphs. Knowledge-Based Systems **291**, 111583 (2024)
47. Xie, M., Yin, H., Wang, H., Xu, F., Chen, W., Wang, S.: Learning graph-based poi embedding for location-based recommendation. In: CIKM. pp. 15–24 (2016)
48. Xu, C., et al: Privileged features distillation at taobao recommendations. In: KDD. pp. 2590–2598 (2020)
49. Xu, J., et al: Multi-behavior self-supervised learning for recommendation. In: SIGIR. pp. 496–505 (2023)
50. Xuan, H., Liu, Y., Li, B., Yin, H.: Knowledge enhancement for contrastive multi-behavior recommendation. In: WSDM. pp. 195–203 (2023)
51. Yu, D., Liu, X., Pan, S., Li, A., Yang, B.: A novel neural-symbolic system under statistical relational learning. arXiv preprint arXiv:2309.08931 (2023)
52. Yu, D., Yang, B., Wei, Q., Li, A., Pan, S.: A probabilistic graphical model based on neural-symbolic reasoning for visual relationship detection. In: CVPR. pp. 10609–10618 (2022)
53. Yu, W., Qin, Z.: Sampler design for implicit feedback data by noisy-label robust learning. In: SIGIR. pp. 861–870 (2020)
54. Zang, X., Yang, B., Liu, X., Li, A.: Dnea: Dynamic network embedding method for anomaly detection. In: KSEM. pp. 236–248. Springer (2021)
55. Zhang, C., Chen, R., Zhao, X., Han, Q., Li, L.: Denoising and prompt-tuning for multi-behavior recommendation. In: WWW. pp. 1355–1363 (2023)
56. Zhao, J., Wenjie, W., Xu, Y., Sun, T., Feng, F., Chua, T.S.: Denoising diffusion recommender model. In: SIGIR. pp. 1370–1379 (2024)
57. Zhu, F., Wang, Y., Chen, C., Zhou, J., Li, L., Liu, G.: Cross-domain recommendation: challenges, progress, and prospects. arXiv preprint arXiv:2103.01696 (2021)