Data Science and Engineering (2026) 11:100–115
https://doi.org/10.1007/s41019-025-00319-5

RESEARCH PAPERS

Sign-aware Recommendation Based on Virtual Semantic Knowledge Graph

Shenggen Ju[1] · Yuxin Zhang[1] · Tianyu Cai[1] · Rongmei Zhao[1] · Jieping Sun[1]

Received: 17 April 2025 / Revised: 30 July 2025 / Accepted: 9 September 2025 / Published online: 17 November 2025

Abstract

Knowledge graph-based recommendation systems have strong capabilities in deep association mining and structured reasoning, which effectively alleviate data sparsity and cold-start problems in recommendation. However, traditional knowledge graph-based recommendation considers each graph relation in isolation, failing to capture the potential semantic correlations between different relation types, which leads to incomplete semantic representations. In addition, existing methods generally ignore negative feedback signals in users' historical interactions, resulting in biased preference modeling. To overcome the above challenges, we introduce **S**ign-aware **R**ecommendation based on **V**irtual **S**emantic **K**nowledge **G**raph(SRVSKG), which enhances recommendation performance by combining virtual semantic collaborative representations with sign-aware learning techniques. Firstly, we propose a virtual semantic subgraph collaborative representation module to learn user and item embeddings. In this module, we build virtual semantic subgraphs through relation clustering based on latent semantic similarity measurement. Then a hierarchical feature extraction mechanism based on graph attention network is applied to virtual semantic subgraphs, which captures semantic associations across relations and enriches item embedding. At the same time, we emphasize user preference for attributes to enrich user embedding. Secondly, we design a sign-aware learning module, which constructs a user-item signed graph, applies Laplacian matrix factorization to simultaneously model the topological features of positive and negative feedback, and introduces the transformer architecture to dynamically fuse signed information, effectively utilizing negative feedback information to eliminate bias in user preference modeling. Finally, we establish a multi-feature fusion mechanism that deeply combines features from the virtual semantic subgraph collaborative representation module and the sign-aware learning module to enable recommendation. Experiment results on three public datasets demonstrate that SRVSKG outperforms state-of-the-art recommendation baselines.

Keywords Recommendation system · Knowledge graph · Sign-aware recommendation · Graph neural network · Transformer

1 Introduction

With the rapid development of the internet and the widespread adoption of modern technologies, modern society has entered the era of big data, where the amount of information we need to process in daily life is growing exponentially. Since human's capacity to process data is far lower than the speed of information dissemination, we are now facing a serious problem of information overload. Recommendation system [1], which can filter massive amounts of products and services to identify content users may be interested in, has become an indispensable tool for information filtering and a popular area of research. Traditional recommendation methods [2–4] extract user and item features based

✉ Tianyu Cai
caity@scu.edu.cn

Shenggen Ju
jsg@scu.edu.cn

Yuxin Zhang
zhangyuxin8@stu.scu.edu.cn

Rongmei Zhao
rmz@stu.scu.edu.cn

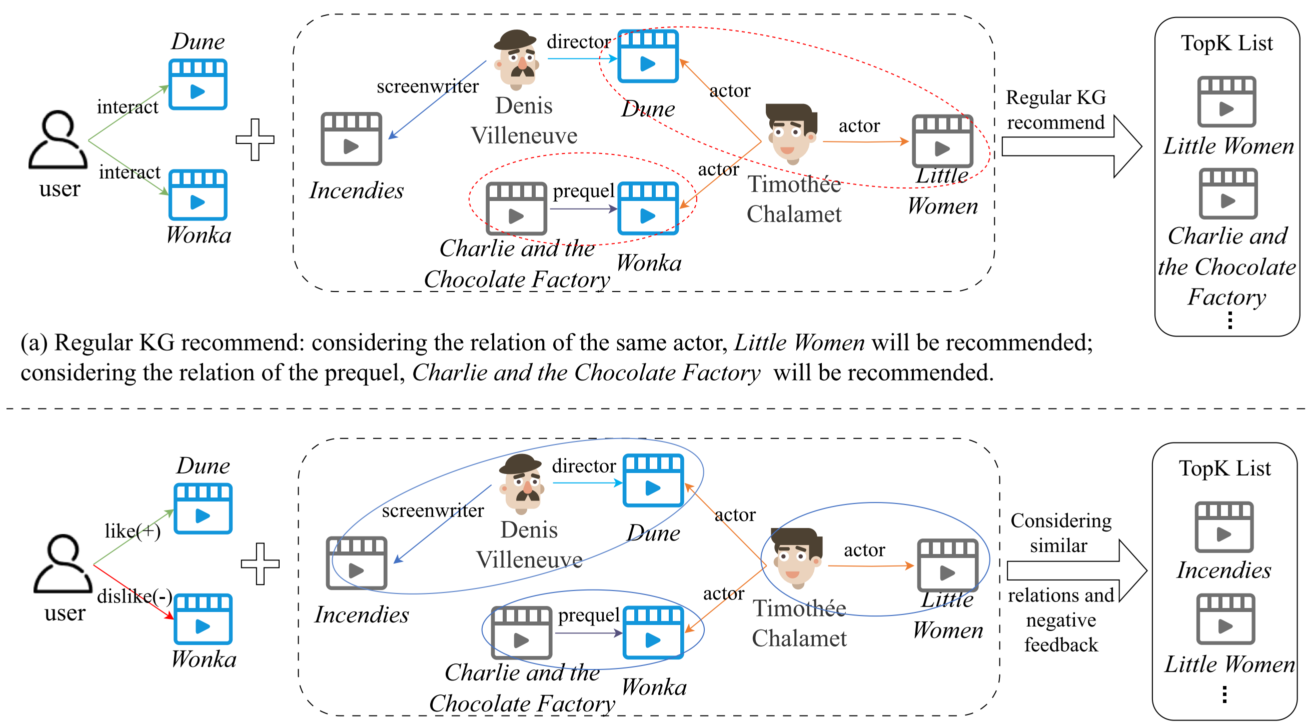
Jieping Sun
sunjieping@scu.edu.cn

[1] School of Computer Science, Sichuan University, 610065 Chengdu, Sichuan, China

on historical interactions and recommend items following the idea of collaborative filtering. However, these methods struggle to handle the sparsity of user-item interactions and the cold-start problem.

Since knowledge graphs (KG) contain rich entities and semantic relations, some researchers have explored incorporating KG as auxiliary information into recommendation systems [5–10] to enhance user and item embeddings, effectively alleviating data sparsity and the cold-start problem. Early researches on knowledge graph-based recommendation systems use knowledge representation learning methods (e.g., the Trans series algorithms [11–14]) to learn KG embedding, and then learn item embedding based on them. Although these methods effectively model KG, they focus only on first-order relations in the graph and ignore high-order semantic information. Some researchers have proposed path-based knowledge graph recommendation methods [15–17], which view the KG as a heterogeneous information network with rich entities and relations, enriching the embeddings of users and items by capturing long-range paths between entities. Although path-based methods can better capture user preference by leveraging high-order semantic information, they often require substantial memory and computational time when applied to large-scale knowledge graphs. With the development of graph neural networks (GNN), researchers have leveraged GNN [18–23] to aggregate information from neighbor nodes in knowledge graphs and enhance item embedding using these neighbors. Stacking multiple GNN layers can capture high-order neighbor messages, which improves recommendation performance and maintains scalability for large-scale datasets. Latest research on SR-HetGNN [24] also validates the efficacy of GNN in message modeling. However, the knowledge graph recommendation methods based on GNN still have the following problems:

(1) Existing methods typically use neighbor aggregation directly on the KG and treat each relation type in isolation. However, analysis on real-world datasets reveals that different relations may convey identical or similar semantic information. Isolated modeling of each relation while ignoring their semantic associations can lead to biased semantic representations. For example, in movie recommendation, as shown in Fig. 1(a), for the movie *Dune* that the user has already interacted, recommendation methods that handle each relation in isolation are easy to recommend the movie *Little Women* based on the same relation (same actor), while *Incendies* will not be recommended because there is no similar association. However, in real-world scenarios, the director and screenwriter have similar semantics, so the user may show interest in *Incendies*. As illustrated in Fig. 1(b),



(a) Regular KG recommend: considering the relation of the same actor, *Little Women* will be recommended; considering the relation of the prequel, *Charlie and the Chocolate Factory* will be recommended.

(b) Considering similar relations and negative feedback: the director and screenwriter have a similar relation and *Incendies* will be recommended; due to disliking *Wonka*, the ranking of *Little Women* with the same actor will decrease, and *Charlie and the Chocolate Factory* won't be recommended.

Fig. 1 In movie recommendation scenarios, (**a**) regular KG recommendation and (**b**) recommendation considering similarity relations and negative feedback when the latent semantic similarity between director and screenwriter is considered, *Incendies* can also be correctly recommended. **Therefore, for KG with a large number of entities and relations, handling each relation independently while ignoring the potential semantic similarities between them is flawed.**

(2) Existing knowledge graph-based recommendation methods primarily focus on the positive feedback in interactions and their connections to entities in the KG, while overlooking the user preference information embedded in negative feedback (such as low ratings in movie data). In typical KG-based recommendation, as shown in Fig. 1(a), if a user has watched the movie *Wonka*, its prequel *Charlie and the Chocolate Factory* may be recommended based on their sequel-prequel relation. However, when negative feedback is considered, as illustrated in Fig. 1(b), since the user disliked *Wonka*, its prequel should not be recommended. Moreover, *Incendies* should have a higher recommendation score than *Little Women*, because the user's low rating of *Wonka*, which is starred by Timothée Chalamet, may also indicate a decreased preference for the actor. **This demonstrates that focusing solely on positive interactions while ignoring negative feedback can adversely affect the accurate modeling of user preference.**

To address the above two issues, we propose Sign-aware Recommendation based on Virtual Semantic Knowledge Graph (SRVSKG), which simultaneously considers semantically similar relations and negative feedback in interaction records, enabling more accurate modeling of user preference. Specifically, the model first learns user and item embeddings through a virtual semantic subgraph collaborative representation module. For item embedding learning, we construct virtual semantic subgraphs through relation clustering and perform subgraph hierarchical feature extraction based on graph attention networks to capture cross-relation semantic associations. For user embedding learning, we emphasize user preference toward item attributes, further enhancing the accuracy of recommendation results. In addition, we also introduce a sign-aware learning module to leverage negative feedback (items disliked by users). By applying Laplacian matrix factorization, it learns features from both positive and negative feedback, helping to avoid recommending unfavorable content and enabling more comprehensive embeddings of user and item. Finally, the interaction features obtained from the virtual semantic subgraph collaborative representation module and the sign-aware learning module are deeply fused to enable precise modeling of user preference. The main contributions of this paper are as follows:

- We propose a virtual semantic subgraph collaborative representation module, which constructs virtual semantic subgraphs and performs hierarchical feature extraction using graph attention networks. This effectively captures latent semantic associations and generates informative item embedding. In addition, a user-attribute preference graph is constructed to emphasize users' preference for item attributes, thereby enriching user embedding learning.
- We introduce a sign-aware learning module to model the topological features of both positive and negative feedback through Laplacian matrix factorization. A Transformer architecture is employed to dynamically fuse signed information, effectively leveraging negative feedback from historical interactions and comprehensively learning user preference toward items. Finally, the two modules are fused to enhance the overall embedding learning.
- Extensive experiments on three public datasets prove the effectiveness and superiority of the proposed model.

2 Related Work

2.1 GNN-Based Knowledge Graph Recommendation System

Knowledge graph recommendation links historical interactions with the KG by aligning items with entities in the graph, forming a hybrid graph. They leverage the rich nodes and edges in the KG to learn relevant features, and GNNs are effective at aggregating high-order neighborhood information within the graph structure. Wang et al. proposed the KGAT [25] model, which employs graph convolutional networks (GCNs) to build a knowledge graph attention network, explicitly modeling high-order connections in the KG in an end-to-end manner, and using an attention mechanism to distinguish the importance of neighbor nodes. Wang et al. introduced the KGIN [19] model, which models user intent based on the KG and explicitly captures fine-grained user-item interactions within the KG, thereby enhancing both the interpretability and generalization ability of recommendations. Chen et al. proposed the CG-KGR [20] model, which introduces a co-guided mechanism and applies GCNs to extract more accurate personalized information from the KG, improving recommendation performance. Liu et al. presented the KANN [26] model, combining KG with attention mechanisms. It applies knowledge-aware attention to model movie review content and integrates KG information to construct user and item embeddings. Liu et al. proposed Mandari [27], which constructs a multi-modal temporal KG and employs graph attention networks to integrate heterogeneous subgraph embeddings with temporal preference patterns. Wang et al. proposed the MKGCN [6] model, which introduces a multi-view knowledge graph structure and uses GCNs to fuse information from different perspectives. Although these methods perform well in knowledge-aware recommendation, they still heavily rely on the quality of the knowledge graph and sparse interaction data. They fail to effectively capture the latent semantic correlations between different relation types, making it difficult to distinguish semantically similar relations, which limits improvements in user preference learning.

2.2 Sign-aware Recommendation System

Unlike traditional recommendation systems that rely solely on positive interactions, sign-aware recommendation systems leverage both positive and negative feedback. They learn user preference from positive samples while using negative samples to enhance preference modeling from another perspective. Huang et al. proposed SBGNN [28], which employs GNNs to learn signed bipartite graphs and designs a message aggregation function tailored for sign-aware learning to improve the modeling of both positive and negative interactions. PANE-GNN [29] divides the original interactions into two subgraphs based on positive and negative feedback, learning the features of both user interests and disinterests separately and then combining them to enhance recommendation performance. Huang et al. introduced SiGRec [30], which designs a novel sign-aware loss function capable of adaptively learning from various types of negative feedback. Li et al. proposed SLGNN [31], which constructs a GNN based on a signed Laplacian matrix to effectively model graph structures with both positive and negative edges, fully utilizing signed relational information. Chen et al. proposed the DCDL [32] framework, which establishes a dual-view collaboration structure with diffusion learning to explicitly address negative and positive feedback loops. This balanced feedback system dynamically adjusts the information flow between views while maintaining their complementary strengths. These methods combine sign-aware learning with recommendation to better mine user preference signals from historical interactions. However, most of them have limited integration with knowledge graphs, overlooking the rich prior knowledge contained within KG.

Our work integrates sign-aware learning with KG for recommendation, fully leveraging negative feedback while utilizing the rich semantic information in KG. This approach not only helps avoid recommending items that user dislikes but also enables comprehensive modeling of user preferences through additional contextual information.

3 Problem Formulation

In a typical recommendation scenario, we define $U = \{u_1, u_2, ..., u_m\}$ as the set of users with *m* denoting the number of users, and $I = \{i_1, i_2, \ldots, i_n\}$ as the set of items with *n* denoting the number of items. The historical user-item interactions are represented as a set $D = \{(u, i, y_{ui}) \mid u \in U, i \in I\}$, where $y_{ui} = 1$ indicates a positive interaction between user *u* and item *i*, $y_{ui} = -1$ indicates a negative interaction, and $y_{ui} = 0$ denotes no interaction between them. Inspired by [30], for datasets with rating scales ranging from 1 to 5, we treat ratings less than 5 as negative feedback, and the rest as positive feedback. We define the positive interaction set as $O^+ = \{(u, i) \mid u \in U, i \in I, y_{ui} = 1\}$ and the negative interaction set as $O^- = \{(u, i) \mid u \in U, i \in I, y_{ui} = -1\}$. We define the positive interaction graph as $G^+ = (N, O^+)$ and negative interaction graph as $G^- = (N, O^-)$, $N = U \cup I$ representing all users and items.

Furthermore, we define the knowledge graph as $G = \{(h, r, t) \mid h, t \in \mathcal{E}, r \in \mathcal{R}\}$, where each triple (*h*, *r*, *t*) consists of a head entity *h*, a relation *r*, and a tail entity *t*. Let $\mathcal{E}$ and $\mathcal{R}$ denote the sets of entities and relations, respectively. Each item $i \in I$ can be aligned to a corresponding entity $e \in \mathcal{E}$ in the knowledge graph. For instance, in the case of movie recommendation, the item *Little Women* can also be found as an entity with the same name in the KG.

To better learn user embedding, we construct a user-attribute preference graph $G_{u,a} = \{(u, a) \mid u \in U, a \in A_u\}$. In this graph, the entity connected to item *i* in the KG are treated as attribute *a*. The user-attribute preference graph includes all first-order entities connected in the knowledge graph *G* between user *u* and the *i* they have positively interacted with. The formal definition of attribute *a* is as follows, where I_u represents the set of items with positive interactions with user *u*:

$$A_u = \{a \mid \exists i \in I_u, (i, r, a) \in G\} \tag{1}$$

The recommendation model in this paper can be formally defined as Equation 2. Given the input positive interaction graph G^+, negative interaction graph G^-, and knowledge graph *G*, it outputs the predicted recommendation score $\hat{y}_{ui}$ between the predicted user *u* and the item *i* that they have not interacted with before.

$$\hat{y}_{ui} = f\left(\{u, i \mid G, G^+, G^-, A_u\}\right) \tag{2}$$

4 Methods

This section provides a detailed explanation of the SRVSKG model architecture, as illustrated in Fig. 2. The model consists of four main modules: (1) A virtual semantic subgraph collaborative representation module that combines semantic subgraph message aggregation and user-attribute preference enhancement. This module aims to mine rich semantic information from the KG and enhance user preference modeling; (2) A sign-aware learning module that learns features from both the positive and negative interaction graphs. This module fully leverages negative feedback to more comprehensively learn user preference for items; (3) A prediction module that fuses features from the above two modules and performs Bayesian Personalized Ranking (BPR) on user and item embeddings; (4) An optimization module that adopts multi-task joint training to enhance overall performance.

4.1 Virtual Semantic Subgraph Collaborative Representation

The virtual semantic subgraph collaborative representation module is designed to learn more comprehensive user and item embeddings. First, we perform virtual semantic relation dividing (4.1.1) to obtain the structure of the virtual semantic subgraphs. Then, we generate item embedding (4.1.2) through graph attention hierarchical feature extraction on GCN to mine the potential semantic association. In the user embedding generation progress (4.1.3), we construct a user-attribute preference graph to enhance user embedding by emphasizing user preference for attributes, thereby achieving more comprehensive user feature learning.

4.1.1 Virtual Semantic Relation Dividing

In this section, we aim to acquire virtual relations for KG relations and obtain virtual semantic subgraphs. We first divide the various relations in the KG into s virtual semantic relations based on their semantic similarity. The KG is then reorganized according to these new semantic relations to

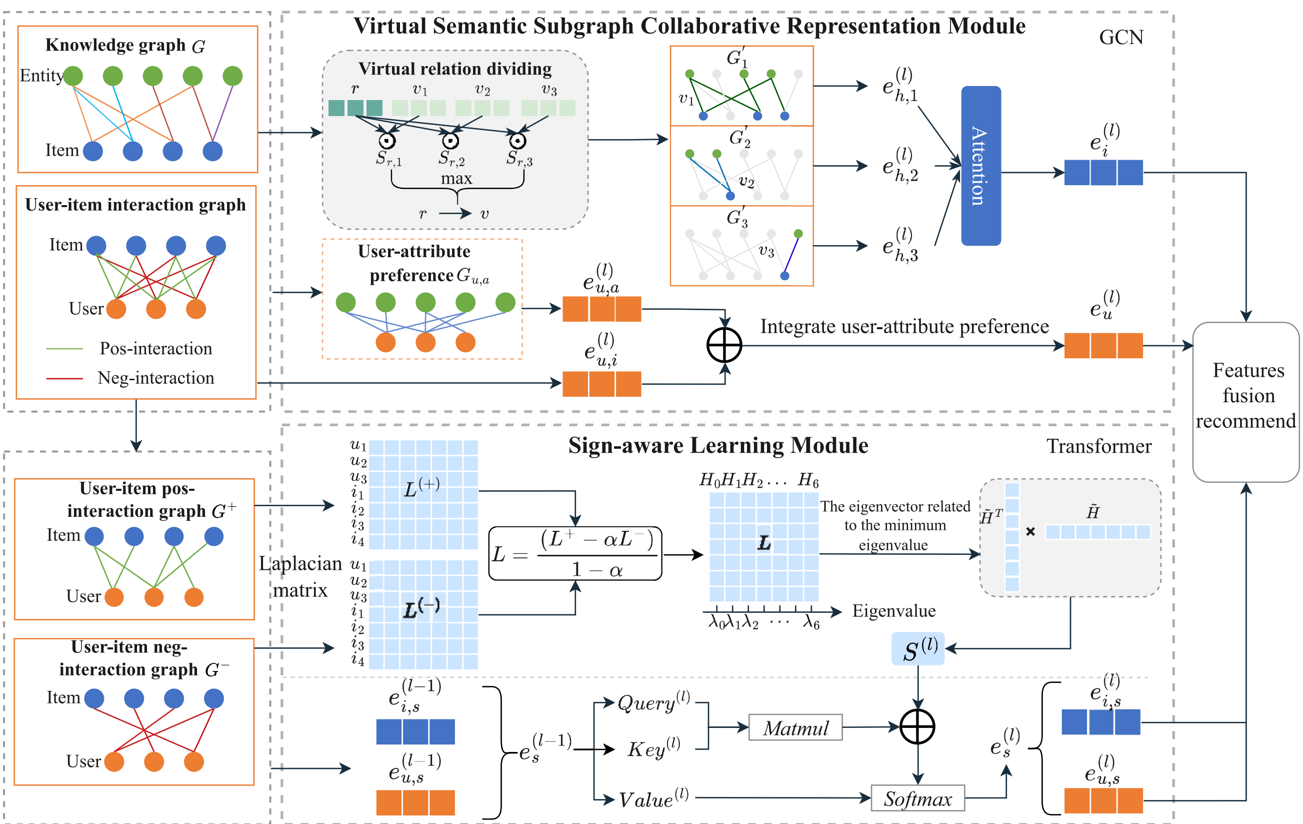


Fig. 2 Illustration of the proposed SRVSKG model: the virtual semantic subgraph collaborative representation module learns **item embedding** through graph attention hierarchical feature extraction on virtual subgraphs G', which is divided from the knowledge graph G, and learns **user embedding** from the user-attribute preference graph $G_{u,a}$ and user-item interaction graph; the sign-aware learning module extracts signed features via Laplacian matrix factorization on the user-item signed graph (G^+ and G^-) and further learns **signed embedding** through a Transformer. Finally, the embeddings are fused for recommendation

generate virtual semantic subgraphs, allowing item embedding to be learned on the same semantic subgraph.

We adopt a cluster-based approach to perform the division of virtual semantic relation. Additionally, we design an unsupervised learning method to fully capture the latent semantics of original relations within the neural network. Firstly, initialize the virtual semantic relation matrix *V*.

$$V = (v_1, v_2, \ldots, v_s),\ V \in R^{s \times d} \tag{3}$$

where v_s represents the *s*-th virtual semantic relation and is a *d*-dimensional vector. The parameter *s* is a predefined hyperparameter indicating the number of virtual semantic relations, and its optimal value may vary across different KGs. We randomly initialize the *V* using a normal distribution and assign equal initial weights to each virtual relation. During the training process, we automatically optimize and adjust them through gradient descent.

For each original relation *r*, we need to determine which virtual semantic relation it should be assigned to. Therefore, we compute the cosine similarity *g*(*r*, *v*) between the original relation *r* and each virtual semantic relation *v*:

$$g(r, v) = \frac{r \cdot v}{\|r\| \| v\|} \tag{4}$$

where $r \in R^d$ is also a *d*-dimensional vector. The similarity scores between the original relation *r* and each virtual semantic relation are represented as a similarity vector S_r. The virtual semantic relation with the highest score is selected as the mapped relation $r^{'}$ of the original relation *r*.

$$S_r = (g(r, v_1), g(r, v_2), \ldots, g(r, v_s)) \tag{5}$$

$$r^{'} = \mathrm{argmax}_{S=1,2,\ldots,s}(g(r, v_1), g(r, v_2), \ldots, g(r, v_S)) \tag{6}$$

Using $r^{'}$ replace *r*, we obtain a new virtual semantic knowledge graph $G^{'}$:

$$G^{'} = \left\{(h, r^{'}, t \mid h, t \in \mathcal{E}, r^{'} \in V)\right\} \tag{7}$$

$$G^{'} \to G_1^{'}, G_2^{'}, \ldots, G_s^{'} \tag{8}$$

In the virtual semantic knowledge graph $G^{'}$, *s* virtual semantic subgraphs can be obtained based on the dividing of virtual semantic relations. For instance, as illustrated in the virtual relation dividing part of Fig. 2, for a given relation *r* (where $s = 3$), S_r is concretely represented as $S_{r,1}$, $S_{r,2}$ and $S_{r,3}$. The original relation *r* will be replaced by the virtual relation *v* with the highest score, while simultaneously generating 3 virtual semantic subgraphs $G_1^{'}$, $G_2^{'}$ and $G_3^{'}$. The virtual semantic relation matrix *V* is jointly updated with the model parameters θ during training, so that after multiple iterations, *V* will be more adaptable to real semantic information and accurately classify similar semantic relations.

4.1.2 Item Embedding Generation

To fully leverage the KG for generating item embedding, we design a message aggregation mechanism on virtual semantic subgraphs, which updates node features by capturing high-order structural information within the subgraph space. We first focus on a single iteration of message aggregation in one layer.

For an entity *h* in a virtual semantic subgraph $G_s^{'}$, we learn the structural information of all its neighbor nodes. Intuitively, neighbors that are more similar to *h* carry more important information, so we use the similarity score between the entity *h* and a neighbor node *t* as the weight for message propagation.

$$\pi(h, t) = e_h^0 \cdot \tanh(e_t^0) \tag{9}$$

e_h^0 represents the feature embedding of entity *h* in the first iteration of single-layer message aggregation, typically initialized with its ID. The $\tanh$ function is the activation function. Then, we use the *softmax* function to normalize the coefficients of neighbor nodes in the subgraph connected to *h*.

$$\pi(h, t) = \frac{\exp(\pi(h, t))}{\sum_{(h,t^{'}) \in N_s(h)} \exp(\pi(h, t^{'}))} \tag{10}$$

The neighbor node information $e_{N_s(h)}^0$ during the propagation is represented as follows:

$$e_{N_s(h)}^0 = \textstyle\sum_{t \in N_s(h)} \pi(h, t) e_t^0 \tag{11}$$

$N_s(h)$ represents the set of neighbor nodes of entity *h* within the subgraph $G_s^{'}$. e_t^0 represents the embedding of neighbor node *t* in the first iteration, which is typically initialized as the node's ID.

Then, we aggregate the obtained neighbor information with the node's feature embedding to obtain the updated node embedding e_h^1 after one iteration:

$$e_h^1 = agg(e_h^0, e_{N_s(h)}^0) = e_h^0 + e_{N_s(h)}^0 \tag{12}$$

To further optimize the node embedding learning and make the structural information in the virtual semantic space more explicit, we repeat the above weighted propagation operation for multiple iterations:

$$e_h^q = agg(e_h^{q-1}, e_{N_s(h)}^{q-1}) = e_h^{q-1} + e_{N_s(h)}^{q-1} \tag{13}$$

where q denotes the current iteration number. Representing the above operation as a function $f_{agg}(\cdot)$, the semantic subgraph message aggregation can be expressed by the following formula:

$$e_{h,s} = f_{agg}\left(\{(e_h, e_t) \mid t \in N_s(h)\}\right) \tag{14}$$

We perform multiple iterations of message aggregation within the s semantic subgraphs to obtain first-order neighbor feature embedding across multiple semantic levels, denoted as $e_{h,s}^{(0)}$. These s feature embeddings encode the item from s different perspectives. To fuse the information from multiple semantic levels, we apply an attention mechanism to generate the final first-order item embedding $e_i^{(1)}$.

$$e_i^{(1)} = e_h^{(1)} = w_1 e_{h,1}^{(0)} + w_2 e_{h,2}^{(0)} + \cdots + w_s e_{h,s}^{(0)} \tag{15}$$

where w_s denotes the attention weight for the neighbor information in the virtual semantic subgraph $G_s^{'}$, representing the importance of that semantic information. w_s is a learnable parameter that is continuously updated during training.

GNN can mine high-order information in the propagation between layers. In order to utilize the multi hop neighbor information in KG, we stack multiple layers of GNN, and the item embedding generated by the l-th layer of GNN are represented as:

$$e_i^{(l)} = e_h^{(l)} = \sum w_s f_{agg}\left(\left\{(e_h^{(l-1)}, e_t^{(l-1)} \mid t \in N_s(h))\right\}\right) \tag{16}$$

For example, as shown in Fig. 2, we iteratively learn three l-th layer neighbor information $e_{h,1}^{(l)}$, $e_{h,2}^{(l)}$ and $e_{h,3}^{(l)}$ from the three virtual semantic subgraphs, then fuse them as the item embedding $e_i^{(l)}$ via our attention mechanism.

4.1.3 User Embedding Generation Enhanced by User-attribute Preference

In the process of virtual semantic subgraph collaborative representation, we use positive interaction information $O^+ = \{(u, i) \mid u \in U, i \in I, y_{ui} = 1\}$ to learn user embedding. For user u, collaborative information is used to learn the impact of the items that u has previously interacted with, and message aggregation is also performed to obtain the collaborative user embedding $e_{u,i}^{(0)}$:

$$e_{u,i}^{(0)} = f_{agg}\left(\left\{(e_u^0, e_i^0) \mid (u, i) \in O^+\right\}\right) \tag{17}$$

where e_u^0 is initialized as the user ID embedding, and e_i^0 is initialized as the item ID embedding. It is worth noting that item i is aligned with entity h in the KG, which means $e_i^{(l)} = e_h^{(l)}$.

To enhance the learning of user preference over item attributes, inspired by Satori[33], an intention-aware model, we introduce a user-attribute preference graph. This graph includes all first-order entities (attributes) connected to the items that user u has positively interacted with.

We use user-attribute preference information to emphasize the degree of preference for different attributes. For example, if user u watched the movie *Titanic* starred by Leonardo and Kate, and also watched the movie *The Great Gatsby* starred by Leonardo and Kerry, it can be considered that user u has more preference for the attribute "Leonardo" more than the attribute "Kate". In order to explore users' preference in attributes, we also perform message aggregation operations on the user-attribute preference graph $G_{u,a}$ to obtain attribute level user embedding $e_{u,a}^{(0)}$:

$$e_{u,a}^{(0)} = f_{agg}\left(\left\{(e_u^0, e_a^0) \mid a \in A_u\right\}\right) \tag{18}$$

The essence of attribute a is an entity, therefore the embedding of attribute a is the same as that of entity h, which means $e_a^{(l)} = e_h^{(l)}$.

The embeddings obtained from historical interaction records and from the user-attribute preference graph reflect different aspects of the user preference. Therefore, we aggregate the two to obtain the final user embedding(as shown in the part of integrating user-attribute preference in the Fig. 2):

$$e_u^{(1)} = e_{u,i}^{(0)} + e_{u,a}^{(0)} \tag{19}$$

The user embedding generated by the l-th layer of the GNN is:

$$\begin{aligned} e_u^{(l)} = f_{agg}&\left(\left\{(e_u^{(l-1)}, e_i^{(l-1)}) \mid (u, i) \in O^+\right\}\right) \\ &+ f_{agg}\left(\left\{(e_u^{(l-1)}, e_a^{(l-1)}) \mid a \in A_u\right\}\right) \end{aligned} \tag{20}$$

It can be observed that in the process of generating item and user embeddings, the item i, entity h, and attribute a share the same feature embedding. This indicates that information can be propagated in parallel across the three graphs during training. We simultaneously train $e_i^{(l)}$ and $e_u^{(l)}$, and by stacking multiple GNN layers, high-order structural information from the KG can be propagated into the user embedding, thereby improving recommendation performance.

4.2 Sign-aware Learning

This section introduces the sign-aware learning module, which fully encodes signed feature through Laplacian matrix decomposition (4.2.1), and dynamically fuses positive and negative interaction features using a Transformer architecture (4.2.2), resulting in more comprehensive user and item embeddings.

4.2.1 Signed Feature Encoding

Spectral feature [34, 35] is an important concept in graph theory, typically referring to the Laplacian matrix of a graph or its eigenvalues and related properties, which can capture the global structural information of the graph. When applied in common GNN or Transformers [36], they can enhance the modeling capability for graph data. Inspired by related studies, we unify the spectral features of the positive graph G^+ and the negative graphs G^- as sign-aware features to enhance the learning of positive and negative feedback information. Specifically, we first combine the Laplacian matrices of G^+ and G^- into a sign-aware Laplacian matrix L:

$$L = \frac{L^+ - \alpha L^-}{1 - \alpha} \tag{21}$$

where L^+ and L^- denote the Laplacian matrices of the positive and negative interaction graphs. α is a hyperparameter that controls the influence of the negative interaction graph. This hyperparameter is crucial for encoding sign-aware features. When $\alpha > 0$, the model tends to recommend items that are farther away from the negative feedback data; when $\alpha < 0$, the model may partially include negatively interacted items in the recommendation. We set the range of α to $(-1, 1)$ to learn negative feedback information more objectively, and an appropriate alpha value can greatly improve recommendation accuracy.

According to graph theory, the eigenvalues and eigenvectors of L can be represented as:

$$L = H^T \Lambda H \tag{22}$$

where H denotes the eigenvector of the Laplacian matrix, and Λ denotes its corresponding eigenvalue. When the eigenvalue is large, the corresponding eigenvector captures high-frequency and local information in the graph. Conversely, when the eigenvalue is small, the corresponding eigenvector focuses on global structure. The eigenvector associated with the smallest eigenvalue $\tilde{\Lambda}$ is denoted as $\tilde{H}$ and it represents the global structure of the graph. Therefore, we use $\tilde{H}$ to encode the sign-aware feature $S^{(l)}$ of the graph:

$$S^{(l)} = \gamma \tilde{H}^T \tilde{H} \tag{23}$$

γ is a learnable positive parameter that is continuously updated during the training process.

4.2.2 Transformer with Sign-aware Features

After obtaining the sign-aware features, multiple iterative learning steps are required to derive the user and item embeddings. The Transformer architecture [37, 38], composed of self-attention modules and feed-forward neural networks, demonstrates strong modeling capabilities in capturing complex user–item interactions. By taking user and item features as input, the Transformer can compute the similarity between users and items while aggregating information from other entities. Therefore, we adopt a multilayer Transformer architecture to dynamically fuse the sign-aware features.

In this section, we design a multilayer Transformer with sign-aware features to perform sign-aware learning. In the l-th Transformer layer, the input and update of the module are defined as follows:

$$Query^{(l)} = Key^{(l)} = Value^{(l)} = e_s^{(l-1)}, e_s^{(l-1)} \in R^{(m+n)\times d} \tag{24}$$

$$e_s^{(l)} = softmax\left(\frac{Query^{(l)}(Key^{(l)})^T}{\sqrt{d}} + S^{(l)}\right) Value^{(l)} \tag{25}$$

where $Query^{(l)}$, $Key^{(l)}$ and $Value^{(l)}$ correspond to the query, key, and value in the Transformer, respectively. $e_s^{(l-1)}$ denotes the user–item embedding obtained at the l-th Transformer layer, and $e_s^{(0)}$ represents the initial user–item embedding, which are formed by concatenating the initial user and item embedding. d is the dimensionality of the feature embedding. Compared with traditional Transformers, we omit the projection matrices for $Query^{(l)}$, $Key^{(l)}$ and $Value^{(l)}$, which significantly reduces the training complexity.

$$e_s^{(l)} = e_{u,s}^{(l)} \parallel e_{i,s}^{(l)}. \tag{26}$$

4.3 Multi-feature Fusion Recommendation

The virtual semantic subgraph collaborative representation module generates user embedding $e_u^{(l)}$ and item embedding $e_i^{(l)}$, while the sign-aware learning module produces sign-aware user embedding $e_{u,s}^{(l)}$ and sign-aware item embedding $e_{i,s}^{(l)}$.

We design a multi-feature fusion mechanism that sums up the two embeddings and calculates the predicted recommendation scores $\hat{y}_{ui}$ for user u and item i using the widely used inner product method:

$$\hat{y}_{ui} = (e_u^{(l)} + e_{u,s}^{(l)})^T (e_i^{(l)} + e_{i,s}^{(l)}). \tag{27}$$

4.4 Optimization

To enable the joint optimization of the virtual semantic subgraph collaborative representation module and the sign-aware learning module, we define two sets of training samples:

$$D^+ = \left\{(u,i,j) \mid (u,i) \in G^+, y_{uj} = 0\right\} \tag{28}$$

$$D^- = \left\{(u,i,j) \mid (u,i) \in G^-, y_{uj} = 0\right\} \tag{29}$$

We adopt BPR loss to optimize our model, with the goal of making the predicted scores of items that users have interacted with higher than those that they have not interacted with. The losses generated by the virtual semantic subgraph collaborative representation module and sign-aware learning module are as follows:

$$\mathcal{L}_{VSKG} = -\sum\nolimits_{(u,i,j)\in D^+} \ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) \tag{30}$$

$$\begin{aligned}\mathcal{L}_{Sign} = &- \sum\nolimits_{(u,i,j)\in D^+} \ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) \\ &+ \sum\nolimits_{(u,i,j)\in D^-} \ln \sigma(\hat{y}_{ui} - \hat{y}_{uj})\end{aligned} \tag{31}$$

The objective function for joint training of two modules is as follows:

$$\mathcal{L} = \mathcal{L}_{VSKG} + \mathcal{L}_{Sign} + \lambda \left\|\theta\right\|_2^2 \tag{32}$$

where θ is the set of model parameters, and λ is the hyperparameter controlling the regularization term. During training, we alternately optimize $\mathcal{L}_{VSKG}$ and $\mathcal{L}_{Sign}$, and adopt the Adam optimizer to minimize the overall loss.

5 Experiments

Extensive experiments were conducted in this section to validate the effectiveness of the proposed model. The experimental results aim to answer the following research questions:

(1) How does the proposed SRVSKG perform on recommendation tasks compared to state-of-the-art baseline models? (2) How do the different modules of SRVSKG contribute to the overall model performance? (3) How do different hyperparameter settings affect the performance of the model? (4) From an intuitive perspective, how does SRVSKG explore user preference and uncover latent semantics?

5.1 Experimental Settings

5.1.1 Datasets

We conduct experiments on three real-world datasets: MovieLens-1M[1], Amazon-Book[2] and Yelp[3]. These datasets, collected from the domains of movies, books and restaurants respectively, contain both positive and negative feedback, and vary in scale. The statistics of the datasets are shown in Table 1.

MovieLens-1M is a movie recommendation dataset that contains approximately one million explicit ratings from 6,040 users. Each user rates movies with an integer score ranging from 1 to 5.

The Amazon-book dataset is derived from the widely-used product recommendation dataset Amazon-Review, focusing specifically on book ratings. Following previous studies [25], we apply a 10-core filtering on the Amazon-book dataset, retaining users and items with at least ten interaction records to ensure data quality.

The Yelp dataset is officially released by the review website Yelp, which includes users' real ratings of restaurants and other businesses, with a rating range of integers from 1 to 5. Following the previous research [25] and actual data situation, we perform 20-core processing on the Yelp dataset, retaining users and projects with at least twenty interaction records to ensure the quality of the data.

Table 1 Statistics of datasets

Dataset	User-Item Interaction				Knowledge Graph		
	User	Item	Interaction	Pos:Neg	Entity	Relation	Triplet
MovieLens-1M	6,040	3,952	1,000,209	1:3.42	6,729	7	20,195
Amazon-book	70,679	24,915	846,434	1:1.04	88,572	39	2,557,746
Yelp	44,911	62,987	2,079,250	1:1.62	65,825	43	1,847,489

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

[2] http://jmcauley.ucsd.edu/data/amazon.

[3] https://business.yelp.com/data/resources/open-dataset/.

To convert explicit feedback into implicit feedback, we follow the recent work [30], treating high ratings (score = 5) as positive feedback, and all other interactions as negative feedback. For MovieLens-1M, we use Microsoft Satori [17] to align item IDs with head entities in the knowledge graph triplets. For Amazon-book, we map items to Freebase [14] entities using title matching to construct the knowledge graph for each dataset. For Yelp, we extract knowledge from the official local business information network (including restaurants' types, cities, stars, etc.) as KG data.

For each dataset, we randomly select 80% of the interactions as the training set and use the remaining 20% as the test set.

5.1.2 Evaluation Metrics

During the evaluation, we adopt a combination of Top-K recommendation and Click-Through Rate (CTR) prediction. For each user in the test set, we treat items that have not been interacted with during training process as candidate test items and use the trained model to predict scores for all these items. For example, for a user u_m, its candidate test item set is $I' = \{i \mid i \in I, y_{u_m i} = 0\}$. In the Top-K recommendation setting, we generate a list of K recommended items for each user and use widely adopted evaluation metrics including Recall and NDCG. Recall reflects the retrieval ability of the recommendation system, while NDCG is an important metric for evaluating the ranking quality of the recommended results. We set the recommendation list length to 20. In the CTR prediction setting, we use the widely adopted metric AUC (Area Under the ROC Curve) to measure the model's ability to distinguish between positive and negative samples. Higher values of Recall and NDCG, as well as an AUC closer to 1, indicate better model performance.

5.1.3 Baselines

To demonstrate the effectiveness of the proposed SRVSKG, we compare it with three categories of recommendation models, including factorization-based models (FM, NFM), GNN and knowledge graph-based models (KGAT, KGIN, CG-KGR, VRKG, KGTN), and sign-aware models (SLGNN, SIGformer):

FM [39]: A classic feature interaction model that efficiently captures second-order interactions among sparse features. In the recommendation scenario, we use the IDs of users, items, and knowledge entities as input features.

NFM [40]: An extension of FM that leverages a neural network to model second-order feature interactions in a nonlinear fashion, enhancing its expressive power and enabling it to capture more complex feature relations.

KGAT [25]: A representative GNN-based recommendation model that integrates knowledge graphs with user–item interaction data. It explicitly models high-order connections in the KG in an end-to-end manner and employs an attention mechanism to distinguish the importance of neighbors, improving recommendation performance.

KGIN [19]: A GNN-based model that introduces user-level entity graphs and models user–item relations from the perspective of fine-grained intent and long-range semantics through GNN.

CG-KGR [20]: A knowledge graph-enhanced recommendation model that constructs a coupled graph of user–item interactions and the KG, and applies GNN for information fusion.

VRKG [18]: A GNN-based recommendation model that learns user and item embeddings over the KG and designs a Local Weighted Smoothing (LWS) mechanism to improve recommendation accuracy.

KGTN [41]: A recent GNN-based KG recommendation model that uses Graph Transformer for global intent modeling and performs contrastive learning under different intent-aware views.

SLGNN [31]: A sign-aware GNN recommendation model that introduces a Laplacian-based GNN framework to capture the structural information of signed graphs.

SIGformer [36]: A state-of-the-art sign-aware recommendation model that utilizes a Transformer architecture to perform sign-aware graph learning and recommendation.

5.1.4 Parameter Setting

For our model, we set the embedding dimension to 64 and define positive feedback as ratings of 5. The Adam optimizer is used to optimize the model parameters, with a batch size of 1024. We set the number of message aggregation iterations q to 3, the number of GNN layers to 2, and the number of sign-aware Transformer layers to 3. The hyperparameter α, which controls the influence of the negative interaction graph, is set to -0.2. we set the regularization hyperparameter $\lambda = 10^{-5}$, and the learning rate $lr = 10^{-4}$.

For baseline models, we follow SiReN [42] in treating positive feedback as positive interactions for sign-agnostic models. The implementations of FM and NFM follow the settings in KGAT [25], and other hyperparameters are consistent with the default configurations in their original papers.

All models are trained for 1000 epochs, with evaluation performed every 10 epochs, and an early stopping strategy is applied to avoid overfitting.

5.2 Performance Comparison

Table 2 shows the performance comparison between our proposed SRVSKG and all baseline models in terms of Recall@20, NDCG@20, and AUC. The optimal results are highlighted in bold, while the suboptimal results are underlined. We analyze and discuss the results as follows:

(1) GNN-based and knowledge graph-enhanced models outperform factorization-based models, indicating that the strong representation capabilities of GNNs and the rich semantic information brought by knowledge graphs significantly enhance recommendation performance. Among sign-aware models, SIGformer, which adopts a Transformer architecture, outperforms SLGNN, which uses GCN, showing the better adaptability of Transformer to sign-aware structures. SIGformer achieves the second-best performance on MovieLens-1M and Yelp, and outperforms others in two metrics on Amazon-book, demonstrating the effectiveness of sign-aware learning in improving recommendation accuracy.

(2) SRVSKG outperforms all baseline models on MovieLens-1M and Yelp, and ranks second only to the best baseline on Amazon-book, illustrating that incorporating virtual semantic subgraphs and sign-aware learning into KG-based recommendation improves personalization. Specifically, on MovieLens-1M, our model achieves significant improvements, with Recall@20 and NDCG@20 increased by 23.29% and 25.86%, and AUC improved by 1.11%. On Amazon-book, SRVSKG achieves the best performance on Recall@20 and NDCG@20 with improvements of 6.82% and 0.65%, respectively. On the Yelp dataset, three metrics achieve improvements of 7%, 3.76%, and 0.4% respectively. These gains can be attributed to two key aspects: (a.) Dividing the knowledge graph into virtual semantic relation subgraphs enables the identification of semantically similar relations and reduces noise caused by redundant relations. Additionally, modeling user preference for attributes enhances user representation learning. (b.) The Transformer-based module captures global structure in the sign-aware graph, fully utilizing the rich information in the negative graph while balancing positive and negative feedback.

(3) SRVSKG performs slightly lower than SIGformer and CG-KGR on AUC, as it focuses more on generating high-quality recommendation lists rather than optimizing global ranking. This explains the modest AUC gains on MovieLens-1M. Under sparse interactions, the global ranking performance gains from virtual semantic relation dividing are less pronounced. This is an important direction for future improvements. It is also worth noting that although SRVSKG lags slightly behind CG-KGR in AUC on Amazon-book, CG-KGR shows instability, as it only outperforms factorization-based models on MovieLens-1M.

5.3 Ablation Study

To investigate the impact of different components of SRVSKG on the overall model performance, we design three ablation variants: w/o VSKG, w/o SIG, and w/o SIGA. Specifically, w/o VSKG removes the virtual semantic subgraph collaborative representation module (VSKG), retaining only the sign-aware learning module (SIG) to learn from both positive and negative interactions. w/o SIG excludes the SIG module and makes recommendations solely based on the VSKG module. w/o SIGA is designed to examine the effectiveness of the user-attribute enhancement. It removes the SIG module and replaces the user-attribute preference embedding in the VSKG module with a standard user representation. The performance of these variants on the three datasets is shown in Table 3.

Impact of the virtual semantic subgraph collaborative representation module. When the virtual semantic subgraph collaborative representation module is removed, we consistently observe performance degradation, which confirms the contribution of this module to overall performance. This also highlights the necessity of incorporating the knowledge graph into recommendation systems.

Table 2 Performance comparison. R@20 and N@20 denote Recall@20 and NDCG@20, respectively

Model	MovieLens-1M			Amazon-book			Yelp		
	R@20	N@20	AUC	R@20	N@20	AUC	R@20	N@20	AUC
FM	20.98	16.30	0.9182	8.40	4.04	0.8056	5.18	3.07	0.9004
NFM	15.66	11.36	0.8973	7.62	3.72	0.8510	4.95	2.86	0.8917
KGAT	27.37	21.12	0.9450	12.81	6.20	0.8607	6.61	3.85	0.9466
KGIN	28.20	21.40	0.9446	14.49	7.31	0.8651	8.78	5.23	0.9486
CG-KGR	21.09	14.58	0.9275	20.36	<u>13.91</u>	<u>0.9312</u>	8.80	5.57	0.9468
VRKG	30.36	23.86	0.9518	11.62	7.38	0.8888	7.72	4.52	0.9464
KGTN	26.08	24.01	0.9372	8.97	6.04	0.8731	5.15	2.74	0.9012
SLGNN	7.19	5.97	0.7589	6.82	3.05	0.7184	3.05	2.36	0.7168
SIGformer	<u>30.44</u>	<u>24.05</u>	<u>0.9519</u>	<u>22.73</u>	12.57	**0.9343**	<u>9.03</u>	<u>5.58</u>	<u>0.9502</u>
SRVSKG	**37.53**	**30.27**	**0.9625**	**24.28**	**14.00**	0.9303	**9.66**	**5.79**	**0.9540**

Table 3 Ablation results. R@20 and N@20 denote Recall@20 and NDCG@20, respectively

Model	MovieLens-1M			Amazon-book			Yelp		
	R@20	N@20	AUC	R@20	N@20	AUC	R@20	N@20	AUC
w/o VSKG	36.65	29.49	0.9622	23.84	13.80	0.9261	9.27	5.43	0.9525
w/o SIG	30.85	24.20	0.9510	15.85	8.19	0.8903	6.12	3.58	0.9466
w/o SIGA	30.47	23.90	0.9501	15.34	8.08	0.8837	6.06	3.56	0.9464
SRVSKG	**37.53**	**30.27**	**0.9625**	**24.28**	**14.00**	**0.9303**	**9.66**	**5.79**	**0.9540**

Impact of the sign-aware learning module. As shown in Table 3, removing the sign-aware learning module leads to a significant performance drop, indicating the importance of the information embedded in negative interaction data. Moreover, the performance degradation on the Amazon-book and Yelp dataset is more pronounced than on MovieLens-1M, suggesting that negative feedback becomes increasingly crucial in sparser datasets.

Impact of user-attribute preference enhancement. By comparing the performance of the w/o SIGA and w/o SIG variants, it is evident that emphasizing user-attribute preference in the virtual semantic subgraph collaborative representation module helps improve model performance. Learning users' preference for different attributes facilitates the generation of more effective user embedding and enhances the accuracy of recommendation.

5.4 Parameters Study

5.4.1 The Impact of the Number of Virtual Semantic Relations s

In this section, we investigate the impact of the number of virtual semantic relations on model performance. Due to the much smaller size of the MovieLens-1M dataset compared to the other two datasets, we conducted experiments within the range of virtual semantic relation number $s = \{1, 2, 3, 4\}$ for MovieLens-1M and within the range $s = \{1, 2, 3, 4, 5, 6, 7\}$ for the other two datasets. For each virtual semantic relation, a corresponding subgraph is constructed. When $s = 1$, it indicates that no virtual semantic relation division is applied. Figure 3 illustrates the model performance on the three datasets. When $s = 3$, SRVSKG achieves the best performance on the MovieLens-1M dataset, while the optimal performance on the Amazon-book and Yelp is achieved when $s = 4$. The following conclusions can be drawn from the observation:

1. When the number of virtual semantic relations is greater than 1, the model consistently outperforms the version without semantic relation dividing. This demonstrates that constructing virtual semantic relation subgraphs and performing semantic subgraph message aggregation can effectively exploit the rich semantic information contained in the knowledge graph.
2. The optimal number of virtual semantic relations for the MovieLens-1M dataset is smaller than that for the Amazon-book and Yelp, indicating that datasets with a greater variety of relation types require more virtual semantic relations to uncover latent semantic information. Overall, the impact of the number of virtual semantic relations on performance follows a trend of initial increase and subsequent decrease. This suggests that clustering relations helps capture semantic correlations. As the number of relations increases further, highly correlated relations may be separated into different clusters, which hinders semantic mining.

5.4.2 The impact of α

According to Equation 21, the hyperparameter α controls the relative contribution of the positive and negative graphs in the signed feature. To analyze the impact of α on model performance and explore its optimal value, we conduct experiments in the range $[-0.8, 0.8]$ with a step size of 0.2. Figure 4 shows the Recall scores of the model under different α values. On the three datasets, the model exhibits a trend of initially increasing then decreasing. The optimal value of α is always relatively small, which indicates that an appropriate value of α can effectively leverage the information brought by negative feedback and thus improve the model's performance. After a performance decline on the MovieLens-1M dataset, it will rise again and tend to stabilize. The nodes with performance decline on the Amazon-book dataset and the Yelp dataset are later than MovieLens-1M, indicating that the convergence speed of performance will be slower for larger datasets.

5.5 Case Study

This section provides an intuitive illustration of our proposed method from the perspective of actual users and items. Taking user u_{56}, the positive feedback item i_{23}, and the negative feedback item i_{540} in the MovieLens-1M

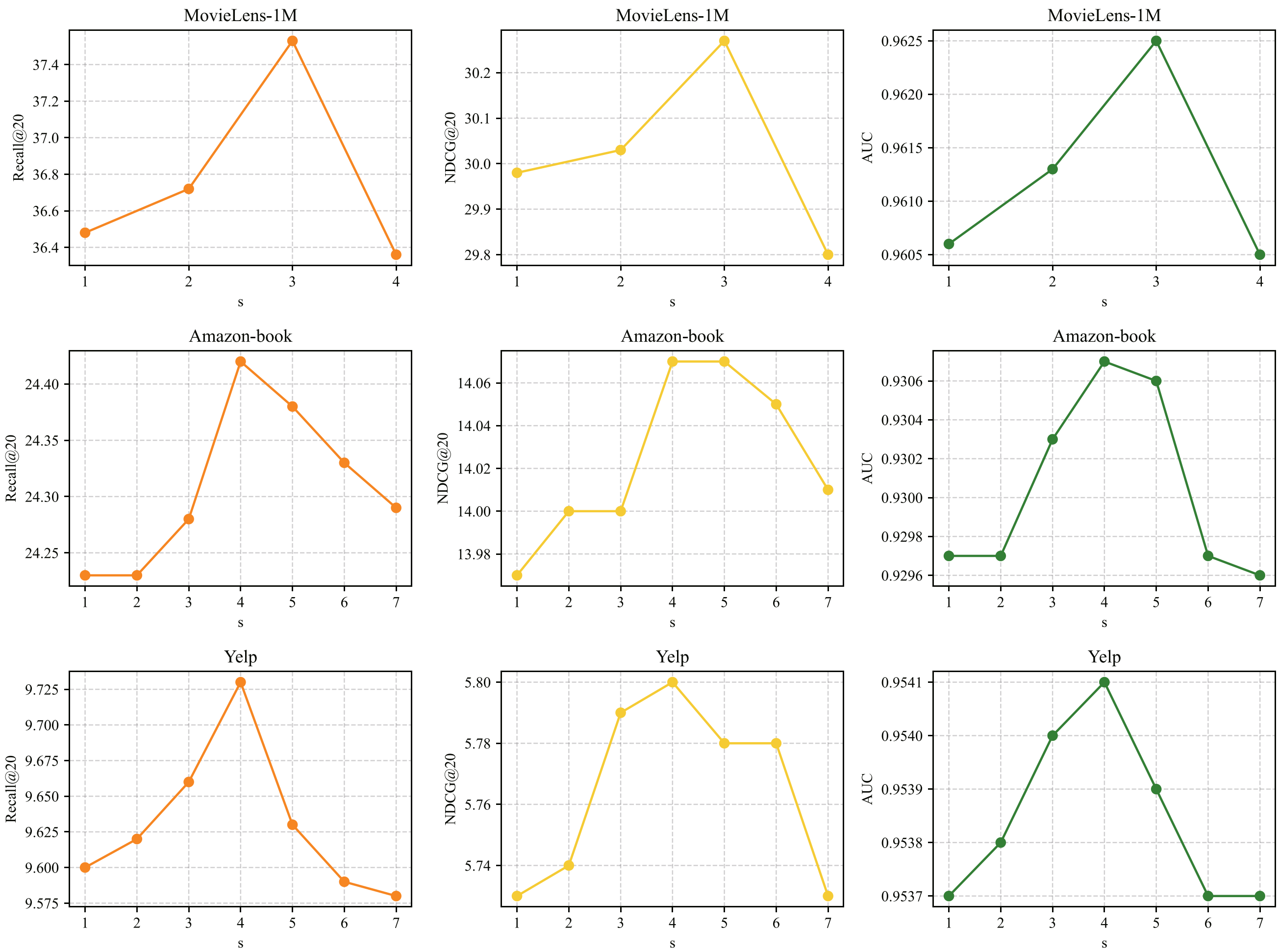


Fig. 3 The influence of hyperparameter s

dataset as an example, the visualization results are shown in Fig. 5.

Figure 5(a) shows the user's preference for both positive and negative items, where the values between the user and items represent the predicted scores generated by the model - the higher the score, the more the user tends to prefer the item. Figure 5(b) presents a heatmap of virtual semantic relation dividing, where original relations are mapped to the virtual relation with the highest similarity score.

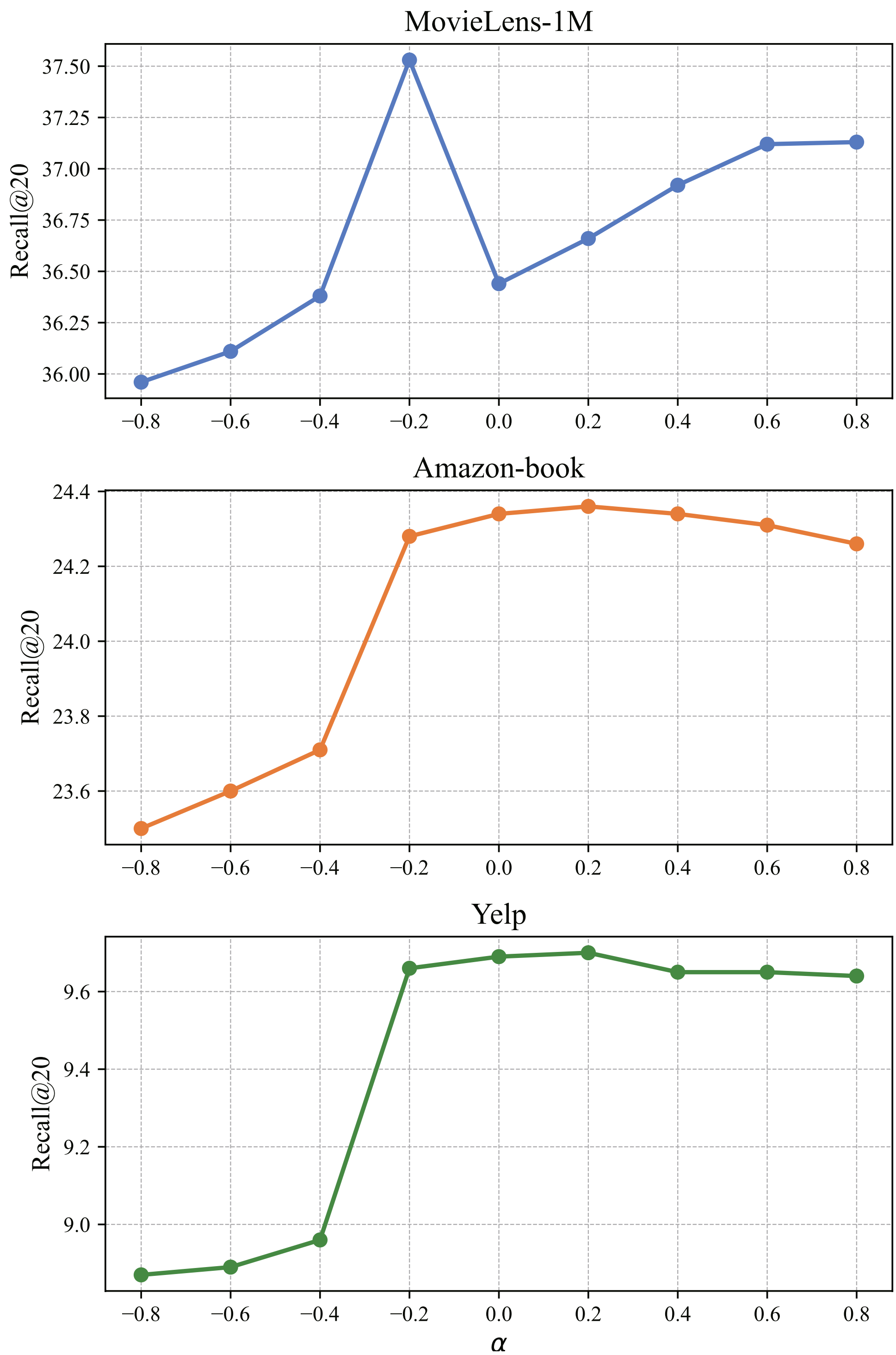
The data in Fig. 5 generally align with our expectations. As shown in Fig. 5(a), the positive item i_{23} receives the highest predicted score, indicating a strong preference from user u_{56}, while i_{540} has the lowest score, as it corresponds to a negative feedback item that the user dislikes. Item i_{540} is linked to i_{161} through the director entity e_{4373}, and it is evident that i_{161} also receives a low score. This demonstrates that learning from negative feedback items enables the model to obtain a more comprehensive understanding of user preference and avoid recommending related undesired items.

From the analysis of Fig. 5(b), we observe that the relations writer and director are clustered into virtual relation v_1, language is mapped to v_2, and star and country are grouped into v_3. This shows that the model successfully learns the underlying semantic similarity among different relations, which is consistent with our earlier analysis.

6 Conclusion

The proposed SRVSKG model integrates a virtual semantic subgraph collaborative representation module and a sign-aware learning module. The virtual semantic subgraph collaborative representation module enables more comprehensive item and user embeddings. In the item embedding learning process, it first constructs virtual semantic relation subgraphs over the knowledge graph (KG), and then applies hierarchical feature extraction via a graph attention network to capture multi-level semantic signals. In the user embedding learning process, the model emphasizes user-attribute preference, enhancing

Fig. 4 The influence of hyperparameter α



the learning of personalized interests. Additionally, SRVSKG introduces a sign-aware learning module to leverage negative interaction information. By performing Laplacian matrix decomposition, it captures the topological features of both positive and negative interaction graphs. A Transformer architecture is employed to dynamically fuse symbolic features, improving the accuracy of user preference modeling. Experiments on two public datasets validate the effectiveness of the proposed model and its individual components. In future work, we plan to further investigate how rich entity relations in the knowledge graph can enhance recommendation performance, simultaneously exploring the use of

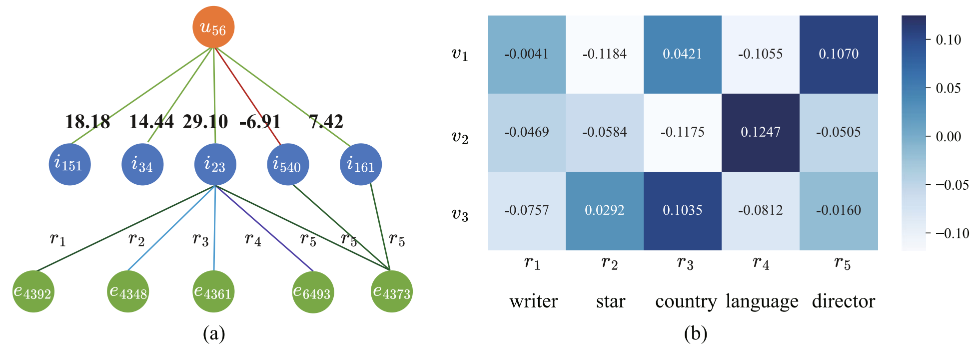


Fig. 5 Visual representation of case studies in the MovieLens-1M dataset. (**a**) indicates users' preference for positive and negative items, with higher scores indicating higher preference contribution. (**b**) is the heatmap for dividing virtual semantic relations

lighter methods to optimize the virtual semantic learning module and reduce model complexity.

Acknowledgements This work was supported by National Natural Science Foundation of China major projects under grant numbers (No.62137001) and National Science and Technology Major Project(grant. 2024ZD0608200).

Author Contributions Not applicable.

Availability of data and materials https://github.com/Xianyihou123/SRVSKG

Declarations

Conflicts of interest The authors have no conflict of interest to declare that are relevant to the content of this article.

References

1. Yang F, Du H, Zhang X, Yang Y, Wang Y (2025) Self-supervised category-enhanced graph neural networks for recommendation. Knowl-Based Syst 311:113109
2. He X, Liao L, Zhang H, Nie L, Hu X, Chua TS (2017) Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, pp. 173–182
3. Wang H, Wang J, Zhao M, Cao J, Guo, M (2017) Joint topic-semantic-aware social recommendation for online voting. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 347–356
4. Yu C, Zhao J, Wu X, Luo Y, Xiao Y (2025) Tpgrec: text-enhanced and popularity-smoothing graph collaborative filtering for long-tail item recommendation. Neurocomputing. https://doi.org/10.1016/j.neucom.2025.129539
5. Li M (2025) Optimization of artificial intelligence recommendation algorithm based on knowledge graph and machine learning: computer thermal optimization. Therm Sci Eng Prog 59:103390
6. Wang X, Zhang Z, Shen G, Lai S, Chen Y, Zhu S (2025) Multi-view knowledge graph convolutional networks for recommendation. Appl Soft Comput 169:112633
7. Xie L, Hu Z, Cai X, Zhang W, Chen J (2021) Explainable recommendation based on knowledge graph and multi-objective optimization. Complex Intell Syst 9:1–12
8. Lin Z, Qu L (2023) Collaborative knowledge-aware recommendation based on neighborhood negative sampling. Inf Syst 115(May):1–119
9. Li X, Henriksson A, Duneld M, Nouri J, Wu Y (2024) Evaluating embeddings from pre-trained language models and knowledge graphs for educational content recommendation. Future Internet 16(1):12
10. Sardina J, Kelleher JD, O'Sullivan D (2024) Twig: Towards pre-hoc hyperparameter optimisation and cross-graph generalisation via simulated kge models. IEEE
11. Bordes A, Usunier N, Garcia-Duran A, Weston J, Yakhnenko O (2013) Translating embeddings for modeling multi-relational data. Curran Associates Inc
12. Wang Z, Zhang J, Feng J, Chen Z (2014) nowledge graph embedding by translating on hyperplanes. In: National Conference on Artificial Intelligence
13. Ji G, He S, Xu L, Liu K, Zhao J (2015) nowledge graph embedding via dynamic mapping matrix. In: Meeting of the Association for Computational Linguistics & the International Joint Conference on Natural Language Processing
14. Lin H, Liu Y, Wang W, Yue Y, Lin Z (2017) Learning entity and relation embeddings for knowledge resolution. Procedia Comput Sci 108:345–354
15. Wang X, Wang D, Xu C, He X, Chua TS (2019) Explainable reasoning over knowledge graphs for recommendation. Proceedings of the AAAI Conference on Artificial Intelligence 33(01):5329–5336
16. Ma W, Zhang M, Cao Y, Jin W, Wang C (2019) Jointly learning explainable rules for recommendation with knowledge graph
17. Xian Y, Fu Z, Muthukrishnan S, Melo GD, Zhang Y (2019) Reinforcement knowledge graph reasoning for explainable recommendation. ACM
18. Lu L, Wang B, Zhang Z, Liu S, Xu H (2023) Vrkg4rec: virtual relational knowledge graph for recommendation. In: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, pp. 526–534
19. Wang X, Huang T, Wang D, Yuan Y, Chua TS (2021) Learning intents behind interactions with knowledge graph for recommendation
20. Chen Y, Yang Y, Wang Y, Bai J, Song X (2021) King I Attentive knowledge-aware graph convolutional networks with collaborative guidance for personalized recommendation

21. Li Z, Liu H, Zhang Z, Liu T, Xiong NN (2021) Learning knowledge graph embedding with heterogeneous relation attention networks. IEEE Transactions on Neural Networks and Learning Systems PP(99)
22. Zhao J, Zhou Z, Guan Z, Zhao W, Ning W, Qiu G, He X (2019) DSIntentgc: a scalable graph convolution framework fusing heterogeneous information for recommendation. ACM
23. Qu Y, Bai T, Zhang W, Nie J, Tang J (2019) An end-to-end neighborhood-based interaction model for knowledge-enhanced recommendation. In: Proceedings of the 1st International Workshop on Deep Learning Practice for High-dimensional Sparse Data, pp. 1–9
24. Jinpeng C, Haiyang L, Xudong Z, Fan Z, Senzhang W, Kaimin W, Jiaqi J (2024) Sr-hetgnn: session-based recommendation with heterogeneous graph neural network. Knowl Inf Syst 2(66):1111–1134
25. Wang X, He X, Cao Y, Liu M, Chua TS (2019) Kgat: Knowledge graph attention network for recommendation. ACM
26. Liu Y, Miyazaki J (2023) Knowledge-aware attentional neural network for review-based movie recommendation with explanations. Neural Comput Appl 35(3):2717–2735
27. Liu X, Li X, Cao Y, Zhang F, Jin X, Chen JMandari: (2023) Multi-modal temporal knowledge graph-aware sub-graph embedding for next-poi recommendation. In: 2023 IEEE International Conference on Multimedia and Expo (ICME), pp. 1529–1534 . https://doi.org/10.1109/ICME55011.2023.00264
28. Huang J, Shen H, Cao Q, Tao S, Cheng X (2021) Signed bipartite graph neural networks
29. Liu Z, Wang C, Xu J, Wu C, Zheng K, Song Y, Mou N, Gai K (2023) Pane-gnn: Unifying positive and negative edges in graph neural networks for recommendation. arXiv preprint arXiv:2306.04095
30. Huang J, Xie R, Cao Q, Shen H, Zhang S, Xia F, Cheng X (2023) Negative can be positive: signed graph neural networks for recommendation. Inf Process Manag 60:103403
31. Li Y, Qu M, Tang J, Chang Y (2023) Signed laplacian graph neural networks. Proc AAAI Conf Artif Intell 37:4444–4452
32. Chen J, Wang J, Zhang F, Cao Y, Wei K, Huang L, Zhu H, Kou F, Huang H (2025) Dual-view collaboration fusion on diffusion learning for bundle recommendation. Inf Fusion 122:103154
33. Chen J, Cao Y, Zhang F, Sun P, Wei K (2022) Sequential intention-aware recommender based on user interaction graph. In: Proceedings of the 2022 International Conference on Multimedia Retrieval. ICMR '22, pp. 118–126. Association for Computing Machinery, New York, NY, USA . https://doi.org/10.1145/3512527.3531390 .
34. Nikiforov V (2010) Spectral graph theory
35. Shuman DI, Narang SK, Frossard P, Ortega A, Vandergheynst P (2013) The emerging field of signal processing on graphs: extending high-dimensional data analysis to networks and other irregular domains. IEEE Signal Process Mag 30(3):83–98
36. Chen S, Chen J, Zhou S, Wang B, Han S, Su C, Yuan Y, Wang C (2024) Sigformer: Sign-aware graph ransformer for recommendation. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1274–1284
37. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S (2021) An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations
38. Zuo Y, Zhang Y, Zhang Q, Zhang W (2025) Knowledge-aware recommendation based on hypergraph representation learning and transformer model optimization. Appl Intell 55(5):1–17
39. Rendle S, Gantner Z, Freudenthaler C (2011) Schmidt-Thieme, L.: Fast context-aware recommendations with factorization machines. In: International Acm Sigir Conference on Research & Development in Information Retrieval, p. 635
40. He X, Chua TS (2017) Neural factorization machines for sparse predictive analytics. ACM SIGIR FORUM 51((cd)):355–364
41. Zou D, Wei W, Zhu F, Xu C, Zhang T, Huo C (2024) nowledge enhanced multi-intent transformer network for recommendation. In: Companion Proceedings of the ACM Web Conference 2024, pp. 1–9
42. Seo C, Jeong KJ, Lim S, Shin WY (2024) Siren: sign-aware recommendation using graph neural networks. IEEE Trans Neural Netw Learn Syst 35(4):15