applied sciences

MDPI

Article

GLARA: A Global–Local Attention Framework for Semantic Relation Abstraction and Dynamic Preference Modeling in Knowledge-Aware Recommendation

Runbo Liu, Lili He and Junhong Zheng *

College of Computer Science and Technology (College of Artificial Intelligence), Zhejiang Sci-Tech University, Hangzhou 310018, China; 202230603081@mails.zstu.edu.cn (R.L.); llhe@zju.edu.cn (L.H.)
* Correspondence: zjhist@zstu.edu.cn

Abstract: Knowledge graph-enhanced recommendation has gained increasing attention for its ability to provide structured semantic context. However, most existing approaches struggle with two critical challenges: the sparsity of long-tail relations in knowledge graphs and the lack of adaptability to users' dynamic preferences. In this paper, we propose GLARA, a novel recommendation framework that combines semantic abstraction and behavioral adaptation through a two-stage modeling process. First, a Virtual Relational Knowledge Graph (VRKG) is constructed by clustering semantically similar relations into higher-level virtual groups, which alleviates relation sparsity and enhances generalization. Then, a global Local Weighted Smoothing (LWS) module and a local Graph Attention Network (GAT) are integrated to jointly refine item and user representations: LWS propagates information within each virtual relation subgraph to improve semantic consistency, while GAT dynamically adjusts neighbor importance based on recent interaction signals. Extensive experiments on Last.FM and MovieLens-1M demonstrate that GLARA outperforms state-of-the-art methods, achieving up to 5.8% improvements in NDCG@20, especially in long-tail and cold-start scenarios. Additionally, case studies confirm the model's interpretability by tracing recommendation paths through clustered semantic relations. This work offers a flexible and interpretable solution for robust recommendation under sparse and dynamic conditions.

Keywords: knowledge graph embedding; graph attention network; virtual relation; long-tail sparsity; user preference modeling; recommendation systems; interpretable recommendation

Received: 18 April 2025
Revised: 23 May 2025
Accepted: 4 June 2025
Published: 6 June 2025

Citation: Liu, R.; He, L.; Zheng, J. GLARA: A Global–Local Attention Framework for Semantic Relation Abstraction and Dynamic Preference Modeling in Knowledge-Aware Recommendation. *Appl. Sci.* **2025**, *15*, 6386. https://doi.org/10.3390/app15126386

1. Introduction

In the era of big data, the exponential growth of online information has made recommendation systems indispensable for mitigating information overload. These systems have been widely applied across domains such as e-commerce, social media, and content delivery platforms. Traditional collaborative filtering (CF) techniques, which rely on historical user–item interaction data to infer user preferences, often suffer from two major issues: data sparsity and the cold-start problem [1–3].

To address these challenges, Knowledge Graphs (KGs)—structured semantic networks encoding entities and their interrelations—have been introduced into recommendation frameworks. By enriching item representations with rich entity attributes and relation semantics, KGs enable more informed inference [4]. For instance, in the movie domain, a KG can link items like Inception and Interstellar via shared relations such as "director" or "genre", thereby capturing semantic similarity beyond user behavior alone.

Current KG-based recommendation approaches can be grouped into three main categories: embedding-based methods, which learn vector representations of entities and relations path methods (e.g., TransE [5], CKE [6]); path-based methods, which mine semantic paths between users and items (e.g., RippleNet [7] and MCRec [8]); and graph neural network (GNN)-based methods [9], which leverage the power of graph structures to model user–item interactions. Methods such as KGAT [10] and KGINKGIN [11] have demonstrated the effectiveness of leveraging attention-based and relation-aware propagation mechanisms, respectively. Specifically, KGAT uses attention mechanisms to weight neighbors, and KGIN employs relation-aware propagation to separate collaborative and knowledge signals. These methods have improved recommendation performance by capturing multi-hop dependencies, but they still face two critical challenges: the sparsity of long-tail relations and the inability to effectively model dynamic user preferences.

Despite their success, these approaches still exhibit significant limitations: First, long-tail relationships—which dominate most real-world KGs—often occur infrequently, making it difficult to learn quality embeddings for rare relations [12,13]. Second, current models usually assume static user preferences, failing to capture users' evolving interests over time [14,15]. Third, existing GNN models often treat relations independently and ignore higher-level semantic clusters formed by related relations (e.g., "Director" + "Actor" -> "Creative Team").

To overcome these issues, we propose a novel recommendation framework, GLARA, which integrates a Virtual Relational Knowledge Graph (VRKG) with a Graph Attention Network (GAT). The VRKG is constructed by clustering semantically similar relations (e.g., "Director" and "Writer") into virtual relation groups (e.g., "Creative Team") using unsupervised methods. This abstraction mechanism not only alleviates the data sparsity caused by long-tail relations but also enhances semantic connectivity, ensuring better generalization in scenarios with sparse data. Furthermore, GLARA introduces a two-level optimization framework that synergistically combines both global and local perspectives: At the global level, a Local Weighted Smoothing (LWS) module aggregates semantic information across related nodes, promoting embedding convergence for semantically similar entities. This process ensures that the embeddings remain consistent across the graph, mitigating the sparsity of long-tail relations. At the local level, a GAT layer dynamically assigns attention weights to recent interactions, enabling the model to capture fine-grained, temporally adaptive user preferences. The combination of VRKG's global semantic abstraction and GAT's local attention-based adaptation offers a novel synergy that is not achievable by either approach alone. This integrated framework allows GLARA to effectively address both the long-tail problem and the dynamic nature of user preferences.

This synergistic integration enables GLARA to effectively bridge global semantic consistency and local behavioral dynamics, leading to more accurate and adaptive recommendations. Experimental results on two benchmark datasets demonstrate that our model significantly outperforms state-of-the-art baselines, especially in long-tail recommendation and dynamic preference modeling scenarios.

The main contributions of this work are summarized as follows:

- We propose GLARA, a novel recommendation framework that combines a Virtual Relational Knowledge Graph (VRKG) with a Graph Attention Network (GAT) to address both global semantic sparsity and local dynamic preference modeling.
- We design a hierarchical co-optimization architecture, where the global layer employs a Local Weighted Smoothing (LWS) strategy to align semantically related node embeddings, and the local layer uses attention-based interaction modeling to capture time-sensitive user interests.

- We conduct extensive experiments on two benchmark datasets (Last.FM and MovieLens-1M), and the results demonstrate that our model consistently outperforms state-of-the-art baselines in terms of both accuracy and robustness, particularly in long-tail and cold-start scenarios.

2. Related Work

Knowledge graph-enhanced recommendation has attracted significant attention in recent years, giving rise to a variety of techniques aimed at improving the expressiveness and generalizability of user and item representations. These methods can be broadly classified into three categories: embedding-based methods, path-based methods, and graph neural network (GNN)-based methods.

2.1. Embedding-Based Methods

Embedding-based methods [6,16–21] aim to map entities and relations in a knowledge graph into a low-dimensional vector space, enabling semantic similarity computations via vector operations. (Classical models like TransE [5]) learn such embeddings by minimizing the distance between head and tail entities via relation translation. Extensions such as TransR [22] and ComplEx [23] support more complex relation types via matrix or tensor decomposition.

In the context of recommendation, methods like CKE [6] combine collaborative filtering with KG embeddings, integrating TransR with matrix factorization to jointly learn user–item relevance.Other works (e.g., KTUP [17]) employ joint optimization frameworks to enhance both recommendation quality and KG completion.

Despite their efficiency, these methods face two core limitations:

- They primarily capture first-order relations, lacking mechanisms for higher-order semantic propagation.
- They operate under static embedding assumptions, failing to reflect dynamic user interests or contextual shifts over time.

2.2. Path-Based Methods

Path-based methods [8,24–27] explicitly explore multi-hop semantic paths connecting users and items to uncover latent interests. For example, RippleNet [7] propagates user preferences along KG paths rooted at items the user has interacted with, while PGPR [28] formulates path selection as a reinforcement learning problem.

These methods offer strong interpretability, as they reveal why a user might be linked to a given item. However, their effectiveness is hindered by the following:

- The need for manually crafted meta-paths or domain-specific rules, which limit generalizability across domains.
- High computational overhead, especially as the number of hops increases exponentially.
- A loose coupling between path selection and recommendation objectives, which may lead to suboptimal performance.

2.3. GNN-Based Methods

The graph neural network (GNN)-based methods [10,11,29–34] have become a dominant paradigm in knowledge-aware recommendation by leveraging message-passing mechanisms to aggregate multi-hop neighborhood information. For instance, KGAT [10] integrates both user–item interactions and KGs into a heterogeneous graph and uses attention mechanisms to weight neighbors, while CKAN [33] separates collaborative and knowledge signals using dual-channel aggregation.

These models excel at capturing long-range dependencies, but several challenges remain:

- Many assume independent treatment of relation types, which ignores higher-level semantic structures (e.g., related roles like "Actor" and "Director").
- Noise propagation from irrelevant neighbors may degrade representation quality.
- Most GNNs rely on static graph structures, limiting their ability to model temporal dynamics in user behavior.

3. Problem Formulation

In a typical recommendation scenario, the user–item interaction data can be modeled as a bipartite graph:

$$\mathcal{G}_I = (\mathcal{U}, \mathcal{I}, \mathcal{E}_I) \tag{1}$$

where $\mathcal{U}$ and $\mathcal{I}$ denote the sets of users and items, respectively, and $\mathcal{E}_I \in \mathcal{U} \times \mathcal{I}$ represents observed interactions (e.g., clicks, ratings). The interactions are stored in a binary matrix $\mathbf{R} \in \{0,1\}^{|\mathcal{U}| \times |\mathcal{I}|}$, where

$$\mathbf{R}_{ui} = \begin{cases} 1, & \text{if user u interacted with item i} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

To incorporate external semantic information, a knowledge graph (KG) is defined as

$$\mathcal{G}_K = (\mathcal{E}_K, \mathcal{R}_K, \mathcal{T}_K) \tag{3}$$

where $\mathcal{E}_K$ is the set of entities (e.g., items, actors, and genres), $\mathcal{R}_K$ is the set of relation types, (e.g., "belongs to a category", "has a tag", etc.), and $\mathcal{T}_K = \{(h, r, t)\}$ is the set of relational triples, with head entity h, tail entity t, and relation $r \in \mathcal{R}_K$.

Given both the interaction graph $\mathcal{G}_I$ and the knowledge graph $\mathcal{G}_K$, the objective is to learn low-dimensional embeddings $\mathbf{e}_u \in \mathbb{R}^d$ and $\mathbf{e}_i \in \mathbb{R}^d$ for each user $u \in \mathcal{U}$ and item $i \in \mathcal{I}$, such that a scoring function $\hat{y}_{ui}$ estimates the likelihood of user u interacting with item i:

$$\hat{y}_{ui} = f(\mathbf{e}_u, \mathbf{e}_i) = \mathbf{e}_u^\top \mathbf{e}_i \tag{4}$$

The final list of personalized recommendations is generated based on $\hat{y}_{ui}$.

The task of the recommendation model is to rank unobserved items for each user by predicting their relevance scores and recommending the top-K items.

Despite recent progress, two major challenges remain in this task:

- Long-tail sparsity: Many relations in real-world KGs follow a long-tail distribution. Rare relations (e.g., "coproducer") occur infrequently, making it difficult to learn meaningful embeddings and resulting in poor generalization for cold-start or niche items.
- Dynamic preference modeling: User preferences are not static but evolve over time. Most models use fixed embeddings that fail to reflect recent behavioral shifts or short-term interests.

4. The Proposed Model

We propose a unified recommendation framework named GLARA that integrates a Virtual Relational Knowledge Graph (VRKG), a Local Weighted Smoothing (LWS) module, and a Graph Attention Network (GAT) to jointly model global semantics and local dynamic preferences, and we present the overview of the model in Figure 1.

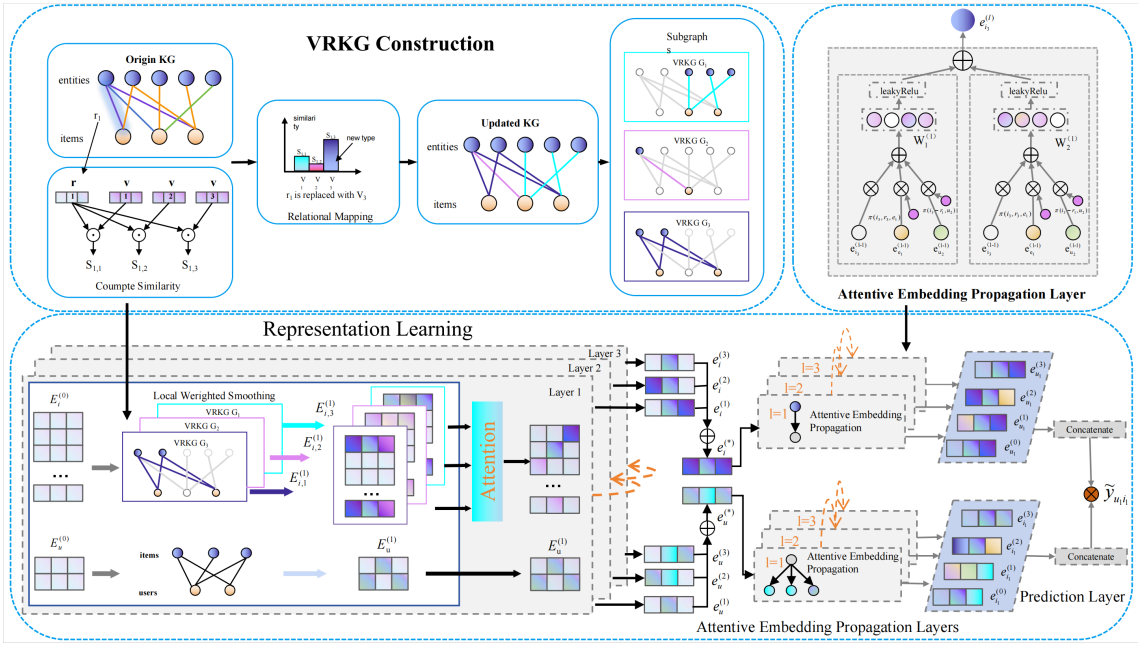

Figure 1. Overview of the proposed GLARA model.

4.1. Virtual Relational Knowledge Graph (VRKG) Construction

In real-world knowledge graphs (KGs), the relation set $\mathcal{R}$ often follows a long-tail distribution, where many relations occur infrequently and lead to sparse semantic connections. The virtual centers for clustering are initialized using the k-means clustering algorithm, which groups relations based on their semantic similarity. The number of clusters K is selected through cross-validation to determine the optimal balance between abstraction and generalization. Specifically, we evaluate different values of K and select the one that minimizes the reconstruction error while maintaining a compact and coherent representation of the relations. This clustering approach not only alleviates the data sparsity caused by long-tail relations but also enhances semantic connectivity, ensuring better generalization across infrequent relations.

4.1.1. Virtual Relation Clustering

Given the original relation embedding matrix $\mathbf{R} = \left[\mathbf{r}_1, \ldots, \mathbf{r}_{|\mathcal{R}|}\right] \in \mathbb{R}^{d \times |\mathcal{R}|}$, we define a virtual relation center matrix $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_K] \in \mathbb{R}^{d \times K}$, where K is the number of virtual relation clusters. Each original relation $\mathbf{r}_p$ is softly matched to virtual centers via temperature-scaled attention:

$$\alpha_{pk} = \frac{\exp\left(\frac{\mathbf{r}_p^\top \mathbf{v}_k}{\tau}\right)}{\sum_{j=1}^{K} \exp\left(\frac{\mathbf{r}_p^\top \mathbf{v}_j}{\tau}\right)}, \quad \forall k \in \{1, \ldots, K\}, \tag{5}$$

where τ is the temperature parameter controlling assignment sharpness.

The reconstructed virtual relation embedding for $\mathbf{r}_p$ is then defined as

$$\tilde{\mathbf{r}}_p = \sum_{k=1}^{K} \alpha_{pk} \cdot \mathbf{v}_k. \tag{6}$$

To ensure clustering consistency and semantic coherence, we impose a regularization term that penalizes reconstruction loss:

$$\mathcal{L}_{\text{cluster}} = \sum_{p=1}^{|\mathcal{R}|} \left\| \mathbf{r}_p - \tilde{\mathbf{r}}_p \right\|_2^2. \tag{7}$$

This regularization term, $\mathcal{L}_{\text{cluster}}$, is used during the clustering phase to encourage the model to learn consistent and coherent virtual relation clusters. However, it is not directly included in the final loss function since its effect is implicitly integrated into the global optimization framework through the clustering process. The primary optimization focuses on improving the recommendation task, while $\mathcal{L}_{\text{cluster}}$ ensures the semantic consistency of the virtual relation embeddings during the clustering step. This approach allows the model to focus on the recommendation objective while maintaining the integrity of the semantic relationships learned during clustering.

4.1.2. Knowledge Graph Reconstruction

After computing virtual relation embeddings, we replace the original triples $(h, r, t) \in \mathcal{T}$ with virtualized triples $(h, \tilde{\mathbf{r}}, t) \in \tilde{\mathcal{T}}$. To formalize this, we define a projection operator $\phi : \mathcal{R} \to \mathbb{R}^d$, where

$$\phi(r_p) = \tilde{\mathbf{r}}_p. \tag{8}$$

The reconstructed knowledge graph becomes

$$\tilde{\mathcal{G}} = (\mathcal{E}, \tilde{\mathcal{R}}, \tilde{\mathcal{T}}), \quad \tilde{\mathcal{T}} = \{(h, \phi(r), t) \mid (h, r, t) \in \mathcal{T}\}. \tag{9}$$

This transformation improves embedding quality and facilitates better generalization over sparse relations by reducing high-variance gradients from tail edges.

4.1.3. VRKG Construction

To support parallel computation and semantic disentanglement, we further partition $\tilde{\mathcal{G}}$ into K disjoint subgraphs $\{\mathcal{G}_1, \ldots, \mathcal{G}_K\}$, each corresponding to a virtual relation group.

Let

$$k^* = \arg\max_k \alpha_{pk}, \quad \forall r_p \in \mathcal{R}. \tag{10}$$

Then, subgraph $\mathcal{G}_k$ is defined as

$$\mathcal{G}_k = \left\{ (h, \tilde{\mathbf{r}}_p, t) \mid (h, r_p, t) \in \mathcal{T},\ \arg\max_j \alpha_{pj} = k \right\}. \tag{11}$$

The complete VRKG is expressed as the union of all virtual–relation subgraphs:

$$\mathcal{G}_{\text{VRKG}} = \bigcup_{k=1}^{K} \mathcal{G}_k. \tag{12}$$

This subgraph partitioning allows the model to learn disentangled embeddings under semantically consistent supervision, improving both scalability and representation modularity.

4.2. Local Weighted Smoothing and Representation Learning

To promote embedding consistency and mitigate relation-level sparsity, we introduce a Local Weighted Smoothing (LWS) module. This component aggregates information from neighboring entities within each virtual relation subgraph, enabling semantic propagation and contextual enrichment. LWS ensures that semantically related entities across virtual relation clusters converge to similar embeddings, thus improving the robustness of the

model in scenarios dominated by long-tail relations. The smoothing process effectively bridges gaps between infrequent relations by aggregating their related neighborhood information. This synergy with the VRKG abstraction ensures that both long-tail and frequently occurring relations contribute to embedding learning, providing a more consistent and effective representation for downstream recommendation tasks. LWS is applied over the VRKG constructed in Section 4.1 and serves as the global-level encoder of our model.

4.2.1. Neighborhood Definition and Similarity Computation

Given a virtual relation subgraph $\mathcal{G}_k = (\mathcal{E}, \mathcal{V}_k, \mathcal{T}_k)$, we define the local neighborhood of entity h as

$$\mathcal{N}_k(h) = \{t \mid (h, \tilde{\mathbf{r}}, t) \in \mathcal{T}_k\}. \tag{13}$$

To measure local semantic coherence, we compute the pairwise similarity between h and each neighbor $t \in \mathcal{N}_k(h)$ using dot-product:

$$\omega_{ht} = \frac{\mathbf{e}_h^{(0)\top} \mathbf{e}_t^{(0)}}{\left\|\mathbf{e}_h^{(0)}\right\| \cdot \left\|\mathbf{e}_t^{(0)}\right\|}, \tag{14}$$

where $\mathbf{e}_h^{(0)}$ and $\mathbf{e}_t^{(0)}$ are the initial embeddings of h and t, respectively.

4.2.2. Weighted Embedding Aggregation

The smoothed embedding of entity h under virtual relation group k is defined as a similarity-weighted average over its neighbors:

$$\mathbf{e}_{h,k}^{(1)} = \sum_{t \in \mathcal{N}_k(h)} \gamma_{ht} \cdot \mathbf{e}_t^{(0)}, \tag{15}$$

where the normalized attention weight γ_{ht} is computed as

$$\gamma_{ht} = \frac{\exp(\omega_{ht})}{\sum_{t' \in \mathcal{N}_k(h)} \exp(\omega_{ht'})}. \tag{16}$$

Then, we combine the original embedding with the smoothed result through a residual connection:

$$\mathbf{u}_{h,k}^{(1)} = \mathbf{e}_h^{(0)} + \lambda \cdot \mathbf{e}_{h,k}^{(1)}, \tag{17}$$

where λ is a smoothing coefficient that controls the influence of neighbors.

4.2.3. Multi-Hop Smoothing Propagation

We repeat the smoothing process for L iterations (or layers). At each layer l, the smoothed representation is computed as

$$\mathbf{u}_{h,k}^{(l)} = \mathbf{u}_{h,k}^{(l-1)} + \lambda \cdot \sum_{t \in \mathcal{N}_k(h)} \gamma_{ht}^{(l)} \cdot \mathbf{u}_{t,k}^{(l-1)}. \tag{18}$$

To avoid scale explosion and maintain numerical stability, we apply normalization at each step:

$$\hat{\mathbf{u}}_{h,k}^{(l)} = \frac{\mathbf{u}_{h,k}^{(l)}}{\left\|\mathbf{u}_{h,k}^{(l)}\right\|_2}. \tag{19}$$

4.2.4. Representation Output for Entities

To obtain the final representation of entity h, we aggregate over all virtual relation subgraphs:

$$\mathbf{e}_h^{\text{LWS}} = \sum_{k=1}^{K} \beta_k \cdot \hat{\mathbf{u}}_{h,k}^{(L)}, \tag{20}$$

where β_k denotes the learned importance of subgraph $\mathcal{G}_k$ and is computed via a softmax-based attention:

$$\beta_k = \frac{\exp\left(\mathbf{q}^\top \cdot \hat{\mathbf{u}}_{h,k}^{(L)}\right)}{\sum_{j=1}^{K} \exp\left(\mathbf{q}^\top \cdot \hat{\mathbf{u}}_{h,j}^{(L)}\right)}, \tag{21}$$

with $\mathbf{q} \in \mathbb{R}^d$ being a learnable query vector.

The final smoothed embedding $\mathbf{e}_h^{\text{LWS}}$ integrates multi-hop neighborhood signals under different semantic views, and serves as the input for downstream user–item interaction modeling.

4.3. Graph Attention Embedding Representation

While the LWS module captures global semantic consistency via virtual subgraph smoothing, it lacks the ability to dynamically adapt to evolving user preferences. To address this, we incorporate a Graph Attention Network (GAT) to model the local user–item interaction graph with adaptive neighbor weighting.

4.3.1. Interaction Graph and Input Embeddings

Given the LWS-optimized embeddings $e_u^{(L)}$ and $e_i^{(L)}$ for user u and item I, we apply a GAT layer to compute attention coefficients α_{ui} reflecting the relevance of each neighbor. This attention mechanism dynamically adjusts the importance of recent interactions, allowing the model to capture fine-grained, temporally adaptive user preferences. By focusing on more relevant and recent interactions, GAT refines the model's understanding of user behavior and preferences, complementing the global smoothing provided by LWS. The combination of VRKG's semantic abstraction and GAT's attention mechanism allows GLARA to address both global semantic consistency and local behavioral dynamics, offering an innovative approach that adapts to evolving user preferences over time.:

$$\alpha_{ui} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^T[\mathbf{e}_u^{(L)} || \mathbf{e}_i^{(L)}]))}{\sum_{j \in \mathcal{N}(u)} \exp(\text{LeakyReLU}(\mathbf{a}^T[\mathbf{e}_u^{(L)} || \mathbf{e}_j^{(L)}]))} \tag{22}$$

where $||$ denotes vector concatenation, a is a learnable weight vector, and $\mathcal{N}(u)$ is the set of items user u has interacted with.

This mechanism allows the model to assign higher weights to recently or frequently interacted items, thus capturing fine-grained user interest shifts.

4.3.2. Attention-Guided Aggregation

The final embedding of a user is computed as a weighted sum of the embeddings of interacted items:

$$\mathbf{e}_u^{\text{GAT}} = \sigma\left(\sum_{i \in \mathcal{N}(u)} \alpha_{ui} \cdot \mathbf{e}_i^{(L)}\right) \tag{23}$$

where $\sigma(\cdot)$ is a non-linear activation function such as LeakyReLU or ReLU. Similarly, for item i, we compute

$$\mathbf{e}_i^{\text{GAT}} = \sigma\left(\sum_{u \in \mathcal{N}(i)} \alpha_{iu} \cdot \mathbf{e}_i^{(L)}\right) \tag{24}$$

4.3.3. Multi-Layer Attention Propagation

To capture higher-order dependencies, we stack multiple GAT layers, where the output of layer $l-1$ serves as the input to layer l:

$$\mathbf{e}_u^{(l)} = \sigma\left(\sum_{i \in \mathcal{N}(u)} \alpha_{ui}^{(l)} \cdot \mathbf{e}_i^{(l-1)}\right), \mathbf{e}_i^{(l)} = \sigma\left(\sum_{u \in \mathcal{N}(i)} \alpha_{iu}^{(l)} \cdot \mathbf{e}_u^{(l-1)}\right). \tag{25}$$

In our implementation, we adopt a two-layer GAT architecture ($L = 2$) to balance expressiveness and computational cost. Through layer-wise propagation, GAT enables the model to integrate multi-hop semantic and behavioral signals, while dynamically adjusting the importance of neighbors based on learned attention.

4.4. Prediction

After obtaining the final representations for users and items from the GAT module, we define a scoring function to estimate the likelihood of user–item interactions. Let $\mathbf{e}_u^{\text{GAT}}, \mathbf{e}_i^{\text{GAT}} \in \mathbb{R}^d$ denote the final GAT-enhanced embeddings for user u and item i, respectively.

Scoring Function

The predicted preference score of user u for item i is computed via a dot product scoring function. By integrating the global semantic abstraction of VRKG and the local dynamic adaptation of GAT, GLARA generates more accurate and contextually relevant predictions. While the objective function combines the recommendation loss and regularization terms, the regularization term L_{cluster} is not explicitly included in the final loss function. Instead, it plays an important role during the virtual relation clustering process, ensuring that the clustering is semantically consistent. The main focus of the loss function is to optimize the recommendation accuracy, while L_{cluster} ensures the quality of the virtual relation embeddings by regulating the clustering step. This design allows the model to concentrate on improving the recommendation task while maintaining the integrity of the semantic relationships learned during clustering. The synergy between these components ensures that both long-tail relations and recent user preferences contribute to the final prediction, providing a more personalized and adaptive recommendation:

$$\hat{y}_{ui} = f(u, i) = \left(\mathbf{e}_u^{\text{GAT}}\right)^\top \mathbf{e}_i^{\text{GAT}}. \tag{26}$$

This formulation assumes that user preferences are proportional to the similarity between latent representations in the shared embedding space.

Alternatively, a bilinear scoring function can be adopted for higher expressiveness:

$$\hat{y}_{ui} = \left(\mathbf{e}_u^{\text{GAT}}\right)^\top \mathbf{W}_s \, \mathbf{e}_i^{\text{GAT}}, \quad \mathbf{W}_s \in \mathbb{R}^{d \times d}, \tag{27}$$

where $\mathbf{W}_s$ is a learnable interaction weight matrix.

4.5. Optimization

We adopt a pairwise learning strategy to train the model, using the Bayesian Personalized Ranking (BPR) loss to maximize the margin between observed and unobserved

interactions. Specifically, for each user u, we sample a positive item $i \in \mathcal{I}_u^+$ and a negative item $j \in \mathcal{I}_u^-$, and minimize the following loss:

$$\mathcal{L}_{\text{BPR}} = -\sum_{(u,i,j)\in\mathcal{D}} \log \sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \lambda \|\Theta\|^2, \tag{28}$$

where $\sigma(\cdot)$ is the sigmoid function, $\mathcal{D}$ is the set of training triplets, Θ denotes the model parameters, and λ is a regularization coefficient.

The model is optimized via mini-batch stochastic gradient descent (SGD) with back-propagation. All components—including VRKG construction, LWS propagation, GAT attention, and scoring—are trained end-to-end.

5. Experiments

We conduct an empirical study to demonstrate the effectiveness of the proposed methodology, with a particular focus on how the synergistic combination of VRKG, LWS, and GAT addresses both global semantic consistency and local dynamic preferences. The experimental results answer the following research questions, which are designed to validate the effectiveness of each component in overcoming long-tail sparsity and modeling dynamic user preferences:

- RQ1: How does this paper's method perform in terms of recommendation performance compared to state-of-the-art knowledge graph methods?
- RQ2: What is the contribution of the key components of this paper's approach to model performance?
- RQ3: How do hyperparameters such as the number of iterations of LWS, number of GAT layers, embedding dimension, etc.) affect the model performance?
- RQ4: How does this paper's approach explore user preferences and provide intuitive interpretability?

5.1. Experiment Settings

5.1.1. Datasets

We evaluate the proposed GLARA model on two publicly available benchmark datasets commonly used in knowledge-aware recommendation research:

- Last.FM: This dataset contains user listening records for music tracks, along with associated artist and tag metadata. We follow previous work [ref] by extracting 2000 users, 8302 items, and 23,355 interactions, supplemented by a domain-specific knowledge graph built from music-related entities (e.g., genre, singer, and album).
- MovieLens-1M (ML-1M): This widely used dataset contains approximately 1 million user movie ratings. We binarize the interactions by retaining only ratings ≥ 4 as positive feedback. The dataset includes 6040 users, 3706 movies, and 996,314 interactions. To construct the knowledge graph, we align movie entities with external sources such as IMDb and DBpedia, integrating side information like genres, directors, actors, and production companies.

Consistent with previous research, we converted the logged data into user–item pairs as active interaction data and used Microsoft Satori to match the headers of all the triples in each dataset with the item IDs to construct a knowledge graph. The basic statistics of the two datasets are shown in Table 1.

All datasets are split into training (80%), validation (10%), and test (10%) sets using the leave-one-out strategy, where for each user, the most recent interaction is held out for testing.

Table 1. Dataset Statistics.

Dataset	User–Item Interaction			Knowledge Graph		
	#User	#Item	#Rating	#Entity	#Relation	#Triplet
Last.FM	1872	3846	42,346	9366	60	15,518
Movie-1M	6036	2347	753,772	6729	7	20,195

5.1.2. Methods of Comparison

To validate the effectiveness of the proposed method, we compare it with the following knowledge graph-based recommendation methods:

- FM [35]: A classical latent factor model that captures pairwise feature interactions using second-order terms, widely applied in recommendation and ranking tasks.
- NFM [36]: An extension of FM that replaces manual interaction modeling with a neural network, enabling the capture of higher-order and nonlinear feature interactions.
- CKE [6]: A knowledge-aware recommendation model that integrates collaborative filtering with knowledge graph embedding (TransR), modeling both structural, and semantic item features.
- KGAT [10]: A graph neural network-based model that jointly learns embeddings from user–item interactions and KG triples via an attention-guided message-passing framework.
- KGIN [11]: A recent GNN-based model that separates collaborative and knowledge signals using relation-aware propagation and semantic-level attention mechanisms.
- VRKG4Rec [34]: A virtual relational KG-based model that clusters original relations into high-level virtual categories to alleviate sparsity and enhance recommendation performance.
- LightGCN [37]: A simplified graph convolutional model that focuses on pure neighbor aggregation without feature transformation or nonlinear activation, achieving strong performance on collaborative filtering tasks.
- Wide & Deep [38]: A hybrid model combining linear (wide) and deep (nonlinear) components to jointly learn low- and high-order interactions, widely adopted in industrial recommender systems.

5.1.3. Evaluation Metrics

We use Recall@K and NDCG@K as evaluation metrics, where K is set to 20. These metrics are widely used in the evaluation of recommendation systems and can effectively measure the accuracy and diversity of recommendation lists.

5.1.4. Parameter Setting

We implemented the model using PyTorch 1.8.2, with the embedding dimension set to 64, the optimizer to Adam, the learning rate to 0.001, and the batch size to 1024. The number of iterations Q for LWS was defaulted to 3, and the number of layers L for GAT was defaulted to 2. The regularization factor Λ was set to 0.0001.

For the other hyperparameters, such as the number of virtual clusters K, the temperature τ, and the smoothing coefficient λ, these were selected using grid search. The value of K was determined through cross-validation to find the optimal number of clusters that balances generalization and clustering consistency. The temperature τ controls the sharpness of the soft attention assignment during clustering, with lower values leading to more focused assignments. The smoothing coefficient λ controls the weight of the smoothed embeddings, balancing the influence of the original graph structure and the smoothed

representation. These hyperparameters were fine-tuned based on preliminary experiments to ensure stable convergence and optimal recommendation accuracy.

5.2. Performance Comparison (RQ1)

We first compare the recommended performance of the proposed method with the existing methods on the two datasets. Table 2 shows the performance of different methods on Recall@20 and NDCG@20.

Table 2. Overall comparison of performance.

Model	Last.FM		MovieLens-1M	
	Recall20	NDCG20	Recall20	NDCG20
FM	0.1402	0.0772	0.2911	0.2798
NFM	0.1497	0.0805	0.2759	0.2484
CKE	0.2695	0.1625	0.3130	0.2918
KGAT	0.2059	0.1671	0.2637	0.2305
KGIN	0.3549	0.2169	0.3150	0.1935
VRKG4Rec	0.3878	0.2302	0.3412	0.3192
LightGCN	0.2246	0.1326	0.2730	0.2366
Wide & Deep	0.1748	0.1103	0.2683	0.2305
GLARA	0.3953	0.2482	0.3627	0.3325

As shown in the table, our proposed method outperforms all baselines in most cases on both datasets. This demonstrates that the integration of the VRKG-based global smoothing mechanism and the GAT-based local attention mechanism enables the model to capture complex user–item interactions more effectively, thereby enhancing recommendation performance. We perform the following analysis and discussion:

First, the improvement can be attributed to the design of the Virtual Relational Knowledge Graph (VRKG), which consolidates numerous original relations in the knowledge graph into a small set of virtual relations. These virtual relations not only uncover semantically related edges but also help encode more informative and task-relevant knowledge for downstream recommendation. Additionally, the Local Weighted Smoothing (LWS) mechanism generates item embeddings by aggregating features from their neighbors, focusing on transforming relational knowledge into neighborhood-aware item representations. This design encourages closer proximity between semantically connected entities in the embedding space.

Second, the Graph Attention Network (GAT) component enhances both user and item embeddings via a dynamic attention mechanism. The GAT can adaptively assign importance weights to neighboring nodes, for example by prioritizing recent interactions to reflect users' evolving preferences. Furthermore, it aggregates multi-hop semantic signals (e.g., connections via directors, actors, etc.) through stacked layers. Compared to traditional GCNs with fixed-weight aggregation, the GAT effectively suppresses noise propagation and improves embedding expressiveness. Importantly, the GAT and LWS are complementary: while LWS mitigates the sparsity of long-tail relations through virtual relation clustering, GAT refines fine-grained interaction modeling via attention-based aggregation. This synergistic architecture balances global semantic consistency and local behavioral dynamics, leading to both accuracy and adaptability in recommendation.

Among the baselines, FM and NFM perform poorly on both datasets due to their inability to utilize external knowledge graph information, limiting their capacity to learn expressive item embeddings. Although CKE incorporates first-order knowledge via TransR, it lacks the capacity to model multi-hop semantics. In contrast, KGAT and KGIN apply GNN-based propagation to capture higher-order neighbor signals. However, the effectiveness

of this propagation heavily depends on graph structure and domain characteristics—for instance, KGAT performs poorly on Last.FM, possibly due to over-smoothing or noise accumulation in sparse relational paths.

Interestingly, CKE and KGIN exhibit dataset-specific performance trade-offs: KGIN outperforms CKE on Last.FM, while the opposite is observed on MovieLens-1M. This may be because the knowledge graph of MovieLens-1M primarily consists of shallow, one-hop triples, which TransR in CKE can exploit more effectively. In contrast, KGIN may introduce noise during multi-hop propagation, resulting in degraded performance.

5.3. Ablation Experiments (RQ2)

To investigate the contribution of each key component in our proposed GLARA model, we conduct an ablation study. In addition to comparing the performance of the full model with the ablated versions, statistical significance tests (such as *t*-tests) were performed to ensure that the observed differences were statistically significant. The performance differences between the full model and the ablated versions were consistently significant, with *p*-values of less than 0.05. Moreover, the standard deviation across 10 runs was calculated to assess the stability of the results.

- GLARA w/o VRKG: Disable virtual relationship clustering and keep the original relationship type.
- GLARA w/o LWS: Disable LWS smoothing and use the raw knowledge graph directly.
- GLARA w/o GAT: Replacing GAT attention with mean pooling.

The results in Table 3 show that the full model performs significantly better than all variants:

Table 3. Performance comparison of different GLARA versions.

Model Version	Recall@20	Recall@20 Std Dev	*p*-Value (Recall)	NDCG@20	NDCG@20 Std Dev	*p*-Value (NDCG)
w/o VRKG	0.3379	0.013	0.03	0.3162	0.012	0.04
w/o LWS	0.3251	0.015	0.02	0.3085	0.013	0.03
w/o GAT	0.3432	0.017	0.045	0.3216	0.014	0.05
Full Model	0.3627	0.013	-	0.3325	0.011	-

The ablation results clearly indicate that each component plays a critical role in the overall effectiveness of the model. Specifically, removing any one of VRKG, LWS, or GAT results in a noticeable drop across all metrics on both datasets, though to varying degrees. The reasons for these degradations are analyzed below:

1. Removal of VRKG (w/o VRKG):
   When we eliminate the Virtual Relational Knowledge Graph, the model relies solely on the raw relations from the original knowledge graph. In this setting, long-tail relations are treated as independent and isolated, lacking semantic clustering or abstraction. As a result, the model suffers from relation sparsity, leading to poor generalization for infrequent entity pairs. This is especially detrimental for low-frequency or cold-start items, where VRKG's semantic grouping plays a vital role in enhancing connectivity and embedding robustness.
2. Removal of LWS (w/o LWS):
   Without Local Weighted Smoothing, the model loses its semantic denoising and neighbor regularization mechanism. The original LWS module helps smooth item embeddings by incorporating information from semantically similar neighbors within each virtual relation group. Removing it disrupts the global semantic consistency in the representation space and makes the model more sensitive to noise from sparse

or noisy connections. In particular, the model becomes overly dependent on the raw structure of the graph, which can be unstable and noisy, especially in datasets with long-tail distributions.

3. Removal of the GAT (w/o the GAT):
   Disabling the Graph Attention Network results in the most significant performance drop. This is because the GAT serves as the core mechanism for modeling local dynamic user behavior. Without it, the model can no longer assign adaptive importance to recent or behaviorally relevant interactions, nor can it effectively capture temporal preference shifts. Additionally, the lack of attention leads to uniform weighting of neighbors, which is both inefficient and susceptible to irrelevant or noisy signals.

This highlights that the GAT is essential for capturing personalized, time-sensitive interaction patterns, and complements the global smoothing of LWS by focusing on high-resolution local signals.

Taken together, these results confirm that VRKG, LWS, and the GAT each contribute unique and complementary strengths to the model. Their integration leads to a balanced architecture that captures both global semantic regularities and local dynamic preferences, and removing any one of them disrupts this balance, thus validating the overall design of GLARA.

5.4. Parameter Sensitivity Analysis (RQ3)

To evaluate the sensitivity of our model to key hyperparameters, we analyze the impact of varying the number of LWS iterations Q and GAT layers L on overall recommendation performance. This analysis provides further insight into how the balance between global semantic smoothing (via LWS) and local attention-based adaptation (via GAT) affects the model's ability to adapt to evolving user preferences and effectively handle long-tail data distributions. We conduct experiments on the Last.FM and MovieLens-1M datasets with the number of virtual relations K set to 3. Figure 2 shows the performance on both datasets for Recall@20 and NDCG@20.

As shown in Figure 2a,b, we fix the number of GAT layers L and vary the number of iterations Q in the range 1, 2, 3, 4. We notice that the performance curve also increases and then decreases on the Last.FM dataset. The results show that as Q increases, items will be closer to similar neighbors in the embedding space will benefit from the item representation in the recommendation task. However, as Q increases, the node embeddings will be too close to differentiate, thus impairing the model performance. In addition, the performance tends to decrease when iterating over MovieLens-1M with increasing Q, because the knowledge graph of MovieLens-1M is richer and has fewer entities than Last.FM, and the dense connectivity makes the embeddings more prone to over-smoothing.

In the context of this study, the number of iterations, Q, was fixed at a particular value, and the value of L was varied within a specific range. The results demonstrated an initial enhancement in performance, followed by a subsequent decline. Specifically, as L increases from 1 to 2, the attention mechanism is able to capture higher-order semantic information (e.g., key connections in long-tail relationships) more accurately by dynamically assigning the neighbor weights, thus significantly improving the recommendation effect. However, when L = 3, although the attention mechanism mitigates the oversmoothing problem of uniform aggregation, the attention weights of some paths in deep propagation may fail due to semantic dilution or noise interference, resulting in a slight performance degradation. This phenomenon is more pronounced in sparsely-connected KGs (e.g., Last.FM), where the complexity of the deep propagation paths exacerbates the difficulty of attention allocation, and in densely-connected KGs (e.g., MovieLens-1M), where attentional redundancy may further erode the gains of the deep structure.

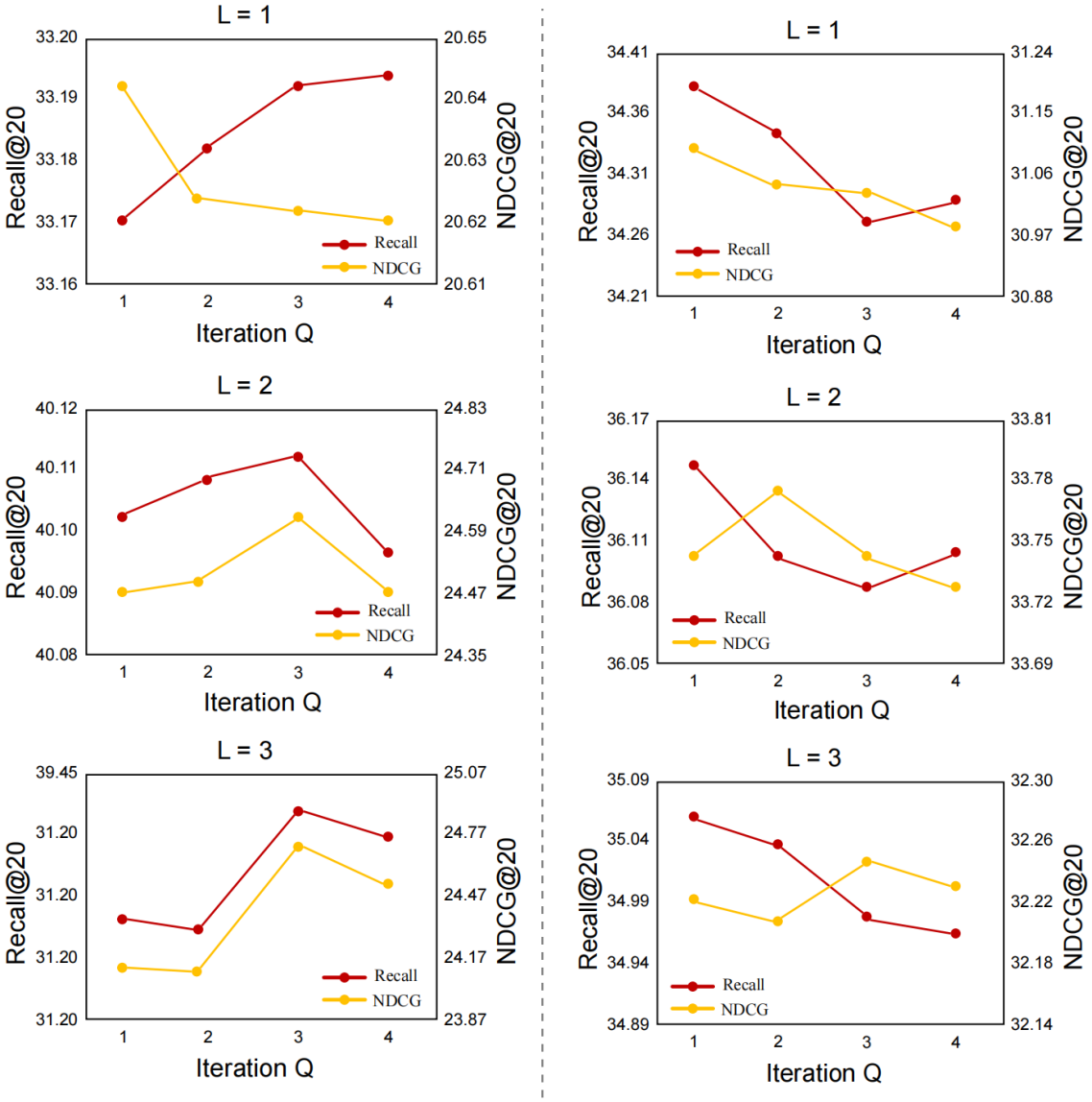

Figure 2. (**a**) Last.FM. (**b**) Movie-1M.

In summary, as the layer L increases, the model performance initially improves significantly and subsequently declines uniformly, independent of the number of iterations Q. This indicates that the GAT layer L is the primary determinate of model performance, and its attentional mechanism enhances the directionality of information propagation, yet does not fully address the inherent limitations of deep propagation. The number of layers L determines the depth of the receptive field, and stacking L layers enables the item embedding to fuse multi-hop neighbor information. The number of iterations Q controls the similarity between the embedding and its local neighborhood by adjusting the smoothing degree of the first-order neighbors. The attentional mechanism further enhances the weight assignment of critical neighbors; however, peak performance is still limited by the reasonable choice of the number of layers L.

5.5. Interpretability Analysis (RQ4)

To evaluate the interpretability of GLARA, we conduct a case study by visualizing a real recommendation scenario using the Last.FM dataset. As shown in Figure 3, we select a

specific user u_{57} and a recommended item i_{33}, to illustrate the preference propagation path from historical interactions to the recommended item.

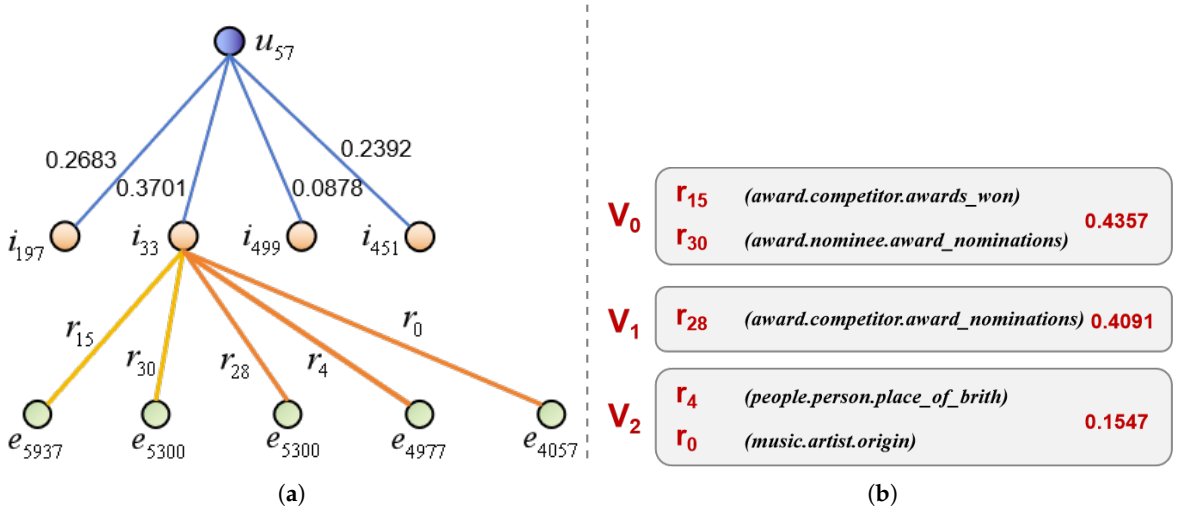

Figure 3. Case study from the Last.FM dataset. (**a**) Shows user u_{57}'s preferences and clustering relationships for interaction items. Best viewed in color. (**b**) Shows the semantics of the relationships in the case study and the corresponding virtual relationships.

Figure 3 presents the multi-hop structure of u_{57}'s interaction history, including user nodes (blue), item nodes (orange), and knowledge entities (green), along with their semantic clustering relationships. Figure 3b illustrates the semantic relations associated with this case and their corresponding virtual relation mappings, showing how original KG relations such as $r_{15}$and r_{30} are abstracted into the same virtual relation v_0. These two original relations are both associated with "awards", reflecting a coherent semantic group. Similarly, relations aligned with v_2 pertain to "location"-related semantics.

We observe that the model clearly identifies preference propagation paths from historical items to i_{33} via semantically enriched virtual connections. Furthermore, through the attention mechanism, the model assigns higher weights to items or paths more aligned with the user's preferences. For instance, the attention score assigned to i_{33} indicates a stronger relevance to user u_{57}, which implies that the model deems it more informative in capturing current interests.

In Figure 3b, attention scores are displayed for each virtual relation during the embedding fusion process. These scores reflect the degree of attention user u_{57} pays to different semantic aspects when generating the recommendation. Notably, the model assigns a higher score to virtual relation v_0(awards), suggesting that awards are more influential than location information in this user's preference. This provides a clear semantic explanation for why item i_{33} is recommended—it matches the user's interest in award-related content.

In addition, Table 4 shows the exposure frequencies of different virtual relations across the dataset. The distribution indicates that the use of virtual relations balances the exposure of originally sparse relations, helping mitigate the long-tail problem. This further confirms that VRKG not only improves model interpretability but also enhances knowledge coverage.

Table 4. Shows the number of exposure of the virtual relations.

Virtual Relation	V_0	V_1	V_2
Expouse Count	14,145	12,493	4398

6. Conclusions

In this paper, we proposed GLARA, a novel recommendation framework that combines a Virtual Relational Knowledge Graph (VRKG) and Graph Attention Network (GAT) to enhance both the semantic expressiveness and behavioral adaptability of recommendation systems. Specifically, we introduced a global-level Local Weighted Smoothing (LWS) module to mitigate relation sparsity and promote semantic cohesion, and a local-level attention mechanism to model user-specific interaction dynamics.

To address the long-tail distribution problem in knowledge graphs, we designed a virtual relation clustering strategy that aggregates infrequent or semantically similar relations into higher-level abstractions, improving knowledge coverage without manual path engineering. Furthermore, the GAT module adaptively adjusts attention weights based on recent user behaviors, allowing the model to effectively capture temporal preference shifts.

Comprehensive experiments on two benchmark datasets, Last.FM and MovieLens-1M, demonstrate that GLARA outperforms a variety of strong baselines across multiple evaluation metrics. Ablation studies further confirm the unique and complementary contributions of each component. A case study illustrates the model's ability to generate interpretable recommendations via virtual relation tracing and attention weight analysis.

Overall, this work offers a unified and flexible solution for combining global semantics and local dynamics in knowledge-aware recommendation. In future work, we plan to extend our framework to multi-modal knowledge graphs and explore reinforcement learning-based interaction policies for further personalization.

However, we acknowledge that there are recent advancements in contrastive learning approaches, such as Knowledge Graph Contrastive Learning and Sparse Group Lasso systems, as well as LLM-based retrieval methods. These methods have shown promise in recommendation tasks but typically require extensive hardware resources, such as large GPU clusters, which were not feasible in the current study. As a result, we have not included these methods in our comparison. We plan to explore these techniques in future work, where we will investigate their integration with GLARA and examine their performance with respect to the hardware and computational constraints involved. This is an important direction for enhancing the scalability and robustness of our model in real-world industrial applications.

Author Contributions: Conceptualization, L.H. and J.Z.; methodology, R.L. and J.Z.; software, J.Z. and R.L.; validation, J.Z. and R.L.; writing—original draft preparation, R.L.; writing—review and editing, L.H. and J.Z.; funding acquisition and resources, L.H. and J.Z.; supervision, L.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China (Grant No. 2024YFB3312602).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in this article; further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Isinkaye, F.O. Matrix factorization in recommender systems: Algorithms, applications, and peculiar challenges. *IETE J. Res.* **2023**, *69*, 6087–6100. [CrossRef]
2. Zhou, Z.; Zhang, L.; Yang, N. Contrastive collaborative filtering for cold-start item recommendation. In Proceedings of the ACM Web Conference 2023, Austin, TX, USA, 30 April–4 May 2023; pp. 928–937.

3. Kumar, S.; Chauhan, V.K.; Upadhyay, D.; Singh, S.; Tripathi, P. Enhancing Recommender Systems to Alleviate Data Sparsity and the Cold Start Problem. In Proceedings of the 2023 12th International Conference on System Modeling & Advancement in Research Trends (SMART), Moradabad, India, 22–23 December 2023; pp. 486–491.
4. Huai, Z.; Tao, J.; Che, F.; Yang, G.; Zhang, D. Knowledge graph enhanced recommender system. *arXiv* **2021**, arXiv:2112.09425.
5. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 2787–2795. Available online: https://proceedings.neurips.cc/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf (accessed on 1 June 2025).
6. Zhang, F.; Yuan, N.J.; Lian, D.; Xie, X.; Ma, W.Y. Collaborative knowledge base embedding for recommender systems. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 353–362.
7. Wang, H.; Zhang, F.; Wang, J.; Zhao, M.; Li, W.; Xie, X.; Guo, M. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, 22–26 October 2018; pp. 417–426.
8. Hu, B.; Shi, C.; Zhao, W.X.; Yu, P.S. Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 1531–1540.
9. Gao, Y.; Li, Y.F.; Lin, Y.; Gao, H.; Khan, L. Deep learning on knowledge graph for recommender system: A survey. *arXiv* **2020**, arXiv:2004.00387.
10. Wang, X.; He, X.; Cao, Y.; Liu, M.; Chua, T.S. Kgat: Knowledge graph attention network for recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 950–958.
11. Wang, X.; Huang, T.; Wang, D.; Yuan, Y.; Liu, Z.; He, X.; Chua, T.S. Learning intents behind interactions with knowledge graph for recommendation. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 878–887.
12. Liu, Y.; Gu, F.; Gu, X.; Wu, Y.; Guo, J.; Zhang, J. Resource recommendation based on industrial knowledge graph in low-resource conditions. *Int. J. Comput. Intell. Syst.* **2022**, *15*, 42. [CrossRef]
13. Yang, Y.; Huang, C.; Xia, L.; Li, C. Knowledge graph contrastive learning for recommendation. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 11–15 July 2022; pp. 1434–1443.
14. Wang, X.; Chen, J.; Wu, F.; Wang, J. Exploiting Global Semantic Similarities in Knowledge Graphs by Relational Prototype Entities. *arXiv* **2022**, arXiv:2206.08021.
15. Zhang, J.; Liang, S.; Sheng, Y.; Shao, J. Temporal knowledge graph representation learning with local and global evolutions. *Knowl.-Based Syst.* **2022**, *251*, 109234. [CrossRef]
16. Ai, Q.; Azizi, V.; Chen, X.; Zhang, Y. Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms* **2018**, *11*, 137. [CrossRef]
17. Cao, Y.; Wang, X.; He, X.; Hu, Z.; Chua, T.S. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In Proceedings of the The World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 151–161.
18. Palumbo, E.; Rizzo, G.; Troncy, R. Entity2rec: Learning user-item relatedness from knowledge graphs for top-n item recommendation. In Proceedings of the Eleventh ACM Conference on Recommender Systems, Como, Italy, 27–31 August 2017; pp. 32–36.
19. Wang, H.; Zhang, F.; Xie, X.; Guo, M. DKN: Deep knowledge-aware network for news recommendation. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 1835–1844.
20. Wang, H.; Zhang, F.; Zhao, M.; Li, W.; Xie, X.; Guo, M. Multi-task feature learning for knowledge graph enhanced recommendation. In Proceedings of the The World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 2000–2010.
21. Zhang, Y.; Ai, Q.; Chen, X.; Wang, P. Learning over knowledge-base embeddings for recommendation. *arXiv* **2018**, arXiv:1803.06540.
22. Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; Zhu, X. Learning entity and relation embeddings for knowledge graph completion. In Proceedings of the AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; Volume 29.
23. Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; Bouchard, G. Complex embeddings for simple link prediction. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 2071–2080.
24. Catherine, R.; Cohen, W. Personalized recommendations using knowledge graphs: A probabilistic logic programming approach. In Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, 15–19 September 2016; pp. 325–332.
25. Fan, S.; Zhu, J.; Han, X.; Shi, C.; Hu, L.; Ma, B.; Li, Y. Metapath-guided heterogeneous graph neural network for intent recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 2478–2486.

26. Ma, W.; Zhang, M.; Cao, Y.; Jin, W.; Wang, C.; Liu, Y.; Ma, S.; Ren, X. Jointly learning explainable rules for recommendation with knowledge graph. In Proceedings of the The World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 1210–1221.
27. Zhao, H.; Yao, Q.; Li, J.; Song, Y.; Lee, D.L. Meta-graph based recommendation fusion over heterogeneous information networks. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 635–644.
28. Xian, Y.; Fu, Z.; Muthukrishnan, S.; De Melo, G.; Zhang, Y. Reinforcement knowledge graph reasoning for explainable recommendation. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 285–294.
29. Cao, X.; Shi, Y.; Yu, H.; Wang, J.; Wang, X.; Yan, Z.; Chen, Z. DEKR: Description enhanced knowledge graph for machine learning method recommendation. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual, 11–15 July 2021; pp. 203–212.
30. Tang, X.; Wang, T.; Yang, H.; Song, H. AKUPM: Attention-enhanced knowledge-aware user preference model for recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 1891–1899.
31. Wang, H.; Zhang, F.; Zhang, M.; Leskovec, J.; Zhao, M.; Li, W.; Wang, Z. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 968–977.
32. Wang, H.; Zhao, M.; Xie, X.; Li, W.; Guo, M. Knowledge graph convolutional networks for recommender systems. In Proceedings of the The World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 3307–3313.
33. Wang, Z.; Lin, G.; Tan, H.; Chen, Q.; Liu, X. CKAN: Collaborative knowledge-aware attentive network for recommender systems. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual, 25–30 July 2020; pp. 219–228.
34. Lu, L.; Wang, B.; Zhang, Z.; Liu, S.; Xu, H. Vrkg4rec: Virtual relational knowledge graph for recommendation. In Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, Singapore, 27 February–3 March 2023; pp. 526–534.
35. Rendle, S.; Gantner, Z.; Freudenthaler, C.; Schmidt-Thieme, L. Fast context-aware recommendations with factorization machines. In Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, Beijing, China, 24–28 July 2011; pp. 635–644.
36. He, X.; Chua, T.S. Neural factorization machines for sparse predictive analytics. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, Japan, 7–11 August 2017; pp. 355–364.
37. He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; Wang, M. Lightgcn: Simplifying and powering graph convolution network for recommendation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual, 25–30 July 2020; pp. 639–648.
38. Cheng, H.T.; Koc, L.; Harmsen, J.; Shaked, T.; Chandra, T.; Aradhye, H.; Anderson, G.; Corrado, G.; Chai, W.; Ispir, M.; et al. Wide & deep learning for recommender systems. In Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, Boston, MA, USA, 15 September 2016; pp. 7–10.