
Knowledge Graph-augmented Sequential Recommendation with Adaptive Time-decay Kernels

Xuelian Zhang^{1,*}, Mian Ren² and Chunling Xiang³

¹*School of Computer Science, Jinjiang College of Sichuan University, Meishan, Sichuan, China*

²*School of Information Engineering, Sichuan Vocational and Technical College of Posts and Telecommunications, Chengdu, Sichuan, China*

³*School of Computer Science, Chengdu College of University of Electronic Science and Technology of China, Chengdu, Sichuan, China*

E-mail: zhangxuelian@scujj.edu.cn; renmian@sptc.edu.cn; 1004032778@qq.com

**Corresponding Author*

Received 27 November 2025; Accepted 27 January 2026

Abstract

To address the limitations of existing knowledge graph-enhanced recommendation systems – particularly their reliance on static fusion mechanisms that fail to capture the dynamic evolution of user interests and their inadequate modeling of heterogeneous information interactions – this paper proposes AdaTKGR, an adaptive time-decay weighted framework for knowledge graph-enhanced recommendations. First, a time-aware self-attention mechanism is introduced to effectively model temporal dependencies in user behavior sequences, thereby capturing fine-grained patterns of interest shift over time. Second, we integrate the RippleNet-style knowledge propagation strategy with a learnable temporal decay kernel, enabling dual-weighted representation learning based on both relational distance within the knowledge graph and temporal recency. Third, a cross-compression unit leveraging

Journal of Web Engineering, Vol. 25_4, 635–666.

doi: 10.13052/jwe1540-9589.2547

© 2026 River Publishers

low-rank bilinear transformations is designed to facilitate deep semantic interaction between user–item interaction embeddings and knowledge graph entity representations. Finally, a time-gated multi-task learning objective is formulated to dynamically balance the primary recommendation task with auxiliary knowledge graph link prediction, enhancing joint optimization. Extensive experiments are conducted on three benchmark datasets – Book-Crossing, Last-FM, and MovieLens-1M – where AdaTKGR achieves average improvements of 6.1% and 8.4% in HR@10 and NDCG@10, respectively, over the strongest baseline methods. Notably, the proposed framework exhibits enhanced generalization performance and interpretability, particularly under data-sparse conditions. This work presents a principled approach to jointly optimizing temporal dynamics modeling and semantic knowledge integration in recommender systems.

Keywords: Knowledge graph, sequential recommendation, temporal decay mechanism, self-attention mechanism, multi-task learning, feature fusion.

1 Introduction

In the era of information explosion, online resources are growing at an exponential rate. While users benefit from digital convenience, they are also confronted with the pervasive challenge of “information overload” [1, 2]. As a pivotal technology for information filtering, recommender systems mitigate this issue by analyzing users’ historical behaviors to deliver personalized suggestions [3]. Thereby, they serve as an effective solution to information overload [4–6]. Among the various recommendation techniques, collaborative filtering stands as one of the most classical approaches, widely adopted due to its simple model and strong interpretability. However, as platform scales expand, the inherent issues of data sparsity and cold start have become increasingly pronounced [6, 7].

To address these limitations, researchers have begun incorporating rich auxiliary information to enhance the feature representation of users and items [8–12]. Among these, knowledge graphs (KGs) have garnered significant attention for their powerful semantic representation capabilities. By encoding structured knowledge in the form of entity-relation-entity triples, KGs effectively integrate user behaviors, item attributes, and their high-order associations, offering a promising pathway to improve both the accuracy and interpretability of recommendations [12].

In recent years, with the growing recognition of the dynamic nature of user behaviors, temporal recommendation methods have emerged as a key research focus. These approaches treat historical user interactions as time series and capture interest evolution patterns by mining sequential dependencies [13].

In particular, self-attention-based sequential models, such as SAS-Rec [14], have achieved breakthrough performance across multiple benchmark datasets through global weighting mechanisms. Nevertheless, most existing sequential methods are confined to binary user–item interactions and fail to fully leverage the external semantic information provided by KGs, limiting their generalization capability under sparse data conditions.

Meanwhile, although KG-enhanced recommendation methods alleviate data sparsity to some extent, they exhibit notable shortcomings in modeling temporal dynamics. Existing KG-aware methods primarily follow three technical routes: static embedding fusion (e.g., CKE [15, 16]), which treats the KG as a static feature repository and fails to adapt to dynamic user interest shifts; path propagation and diffusion methods (e.g., RippleNet [17, 18]), which lack an inherent time-aware mechanism and struggle to reflect the principle that recent interactions tend to be more influential; and multi-task collaborative methods (e.g., MKR [19]), which, despite enhancing generalization through parameter sharing, still exhibit limited capacity for sequential modeling.

A systematic analysis of existing literature reveals that the core challenge in current recommender systems research lies in achieving deep and adaptive integration of temporal dynamics and knowledge semantics. Specifically, three key issues persist: (1) temporal signals and knowledge semantics are often processed in isolation, lacking effective interaction mechanisms; (2) time decay patterns are typically modeled using fixed parameters, failing to accommodate personalized demands across users and scenarios; and (3) weight allocation strategies in multi-task learning are often statically defined, unable to adapt dynamically during training.

To holistically address these challenges, this paper proposes an adaptive time-decay knowledge graph recommendation (AdaTKGR) framework. The main contributions of this work are as follows:

- A novel paradigm of “temporal-modulated knowledge fusion” is introduced, which employs temporal dynamics as control signals for the knowledge propagation process, establishing a theoretical foundation for temporal-knowledge synergy.

- A learnable piecewise decay kernel function is designed, enabling the model to adaptively capture multi-scale patterns in user interest evolution. A cross-compression unit is constructed to facilitate fine-grained bidirectional interaction between temporal signals and knowledge semantics.
- A time-gated multi-task optimization strategy is proposed, which dynamically balances the recommendation loss and the knowledge graph loss based on the training process, thereby enhancing both the efficiency and effectiveness of model training.

Comprehensive experiments are conducted on three public benchmark datasets. The results demonstrate that AdaTKGR outperforms state-of-the-art baseline models in terms of recommendation accuracy, generalization capability, and interpretability. This study not only offers a practical technical solution for the deep integration of knowledge graphs and temporal recommendation, but also lays a theoretical foundation for building next-generation adaptive recommender systems.

2 Related Works

In recent years, research on recommendation systems has gradually shifted from static modeling to dynamic temporal modeling. This shift aims to better capture the evolving patterns of user interests. Meanwhile, as a rich source of semantic information, knowledge graphs have been extensively integrated into recommendation systems to address the issue of data sparsity. This section reviews relevant research works from two perspectives: temporal recommendation methods and knowledge graph-enhanced recommendation methods. Additionally, it conducts an in-depth analysis of the limitations of existing methods, which leads to the research motivation of this paper.

2.1 Research on Temporal Recommendation Methods

Temporal recommendation methods predominantly revolve around the modeling of user behavior sequences. In the early stages, research efforts modeled consecutive interactions using Markov chains [20]. However, the limited state representation capacity of Markov chains made it arduous to capture long-term dependencies.

With the advent of deep learning, recurrent neural networks [21, 22] and temporal convolutional networks [23] were introduced into sequence recommendation. These models leverage their innate temporal processing

capabilities to learn the dynamic changes in user interests. In recent times, the self-attention mechanism [24] has emerged as a focal point of research due to its formidable sequence modeling capabilities. Studies such as SASRec [25] and BERT4Rec [26] have achieved remarkable results on multiple public datasets by assigning global attention weights.

Nevertheless, existing temporal recommendation methods suffer from notable theoretical limitations and practical drawbacks. Theoretically, the majority of these methods solely concentrate on the temporal order of interactions, overlooking the multi-scale characteristics inherent in the interest evolution process. User interests typically exhibit diverse patterns, including short-term bursts, medium-term stability, and long-term periodicity. Existing methods, however, often adopt a uniform temporal processing strategy, rendering them incapable of adaptively capturing these multi-scale features.

Practically, current methods generally rely on fixed time decay functions or sliding window mechanisms, lacking the ability to model the personalized pace of interest evolution. More critically, most of these methods are confined to the binary user–item interaction relationship, failing to fully exploit the semantic correlations among items and external knowledge. This limitation results in suboptimal generalization capabilities in sparse data scenarios.

2.2 Knowledge Graph-Enhanced Recommendation Methods

Knowledge graph-enhanced recommendation methods can be categorized into three major technical paradigms based on distinct information fusion methodologies:

- **Static embedding fusion methods:** These methods regard knowledge graphs as static feature repositories. CKE [15] was the first to propose collaborative knowledge base embedding. It maps text, images, and structural knowledge into a shared latent space and then directly concatenates them to item vectors for matrix factorization, significantly enhancing representation in sparse scenarios. Nevertheless, the weights of multiple data sources need to be manually specified and lack the ability to adaptively adjust with the drift of user interests. SHINE employs a deep autoencoder to embed emotional networks, social networks, and profile (knowledge) networks into a unified space for celebrity recommendation. However, it lacks a dynamic weighting mechanism. DKN [20], tailored for the news domain, utilizes a knowledge-aware convolutional neural network to learn the embeddings of sentence texts. It then dynamically aggregates entity embeddings through attention

to derive user preferences. However, its KG embedding module and recommendation module are trained sequentially, precluding end-to-end optimization. Moreover, the attention weights only operate at the entity level and do not take into account the temporal dimension.

- **Path propagation and diffusion methods:** These methods strive to explicitly model the propagation process of user interests on knowledge graphs. RippleNet [17] made a groundbreaking contribution by conceptualizing user interests as “ripples” on the KG. Starting from the interacted item as a seed, it diffuses and aggregates the representations of multi-hop neighbors along relationship paths, thereby enhancing cold-start performance in an end-to-end manner. Although this mechanism demonstrates excellent performance in cold-start scenarios, it assigns equal weights to all hops, defying the intuitive principle that “the farther the distance, the weaker the signal.” Subsequent research efforts introduced attention gating [21] or GNN aggregation [27] to address this issue. However, these approaches still rely on static KGs and are unable to capture the temporal evolution of entity semantics. PGPR [28] exploits reinforcement learning to identify user–item paths in the KG, enhancing interpretability. However, it incurs high computational complexity and also neglects the temporal aspect.
- **Multi-task collaborative optimization methods:** These methods treat knowledge graph link prediction as an auxiliary task. MKR [19] achieves bidirectional enhancement of item and entity representations through cross-compression units. KGAT [29] further incorporates a graph attention network to aggregate high-order neighbors. Although these methods improve the model’s generalization ability, the multi-task weights are often manually set or determined using simple heuristic strategies. As a result, they cannot dynamically adjust the relative importance of the knowledge graph and the recommendation task during the training process.

3 Model Design

3.1 Problem Definition

In the recommendation system of this study, the user set is defined as $U = \{u_1, u_2, u_3, \dots, u_m\}$, and the item m set is defined as $V = \{v_1, v_2, v_3, \dots, v_m\}$, where m and n represent the total number of users and items, respectively. The elements in sets U and V are one-dimensional vectors

of length d . The user-item interaction matrix $Y \in R^{m \times n}$ is used to represent the interactions between users and items: if user u has interacted with item v , the elements in the matrix Y will be set to 1; if there is no interaction, then it will be 0.

In addition, the interaction matrix can be regarded as the user's action sequence $S_u = (s_1^u, s_2^u, s_3^u, s_4^u \dots s_n^u)$. During training, the model learns to predict the next interaction item based on the preceding n items. Specifically, the input to the model is constructed as $(s_1^u, s_2^u, s_3^u, s_4^u \dots s_{n-1}^u)$, and the user feature extraction module outputs a shifted version of the same length, $(s_2^u, s_3^u, s_4^u \dots s_n^u)$. This vector sequence is subsequently used as the user vector input for the recommendation module.

For the knowledge graph $G = \{(h, R, t) | h, t \in E, r \in R\}$, where E represents the entity set, R is the relation set, h represents the head entity, and t represents the tail entity. Relation r connects entities h and t along the direction from head to tail.

The problem of this study is defined as follows: Given the user-item interaction matrix Y , the knowledge graph G , and the rating records, this study aims to predict the potential interest probability of user u in item v that they have not interacted with before and simultaneously train the prediction function.

3.2 Model Framework

This study proposes an innovative recommendation system architecture, which integrates four core components that work collaboratively (Figure 1): the recommendation and prediction module, the knowledge graph-embedding module, the cross-compression unit, and the user feature extraction module. It aims to achieve more accurate modeling and prediction of the dynamic evolution of user interests.

Compared with traditional models, the innovation of this framework mainly lies in two aspects. First, it introduces an adaptive learning-based time decay weight mechanism, which can dynamically adjust the influence of users' historical behaviors in the model based on the time distance of interaction occurrence, thereby more precisely capturing the temporal evolution law of users' interests. Second, an efficient cross-compression unit was designed. Through low-rank bilinear transformation, deep feature interaction and integration between the recommendation system and the knowledge graph information were achieved, significantly improving the accuracy and interpretability of the recommendation.

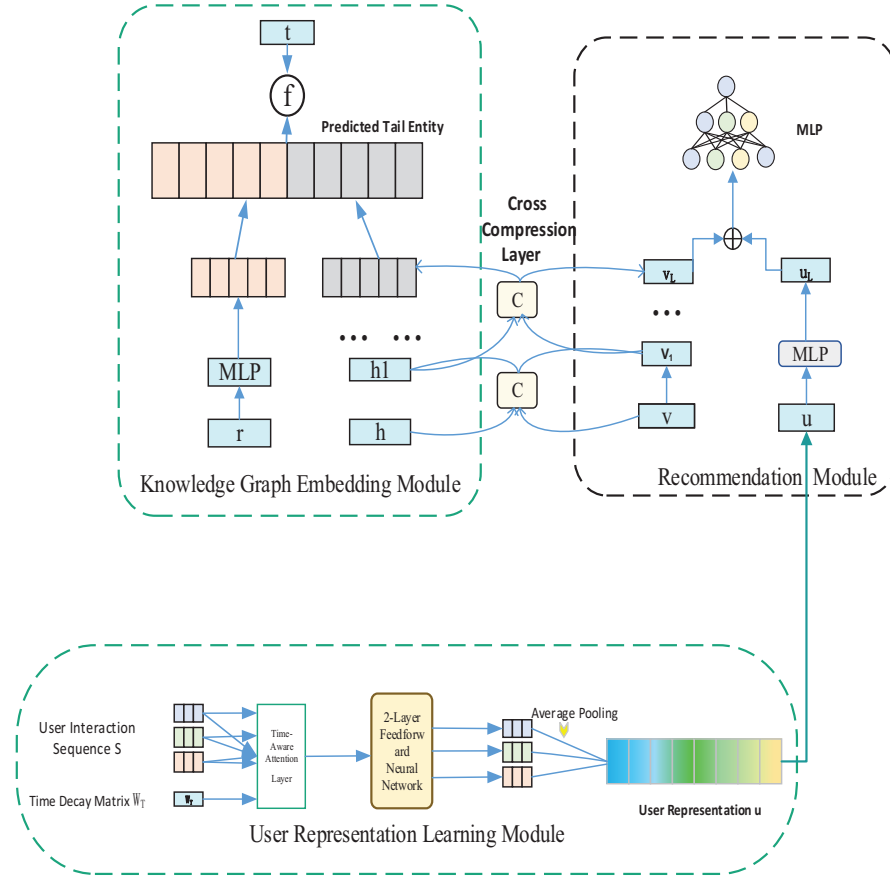


Figure 1 Model structure diagram.

The experimental results show that this framework is not only advanced in theoretical design, but also its effectiveness has been verified through empirical analysis on multiple public datasets. The specific description of the system architecture is as follows.

User feature extraction module: This module takes the user–item interaction sequence as input. First, it maps the discrete item identifiers to low-dimensional dense vectors through the embedding layer. On this basis, a self-attention mechanism integrating time decay weights is adopted to analyze the long-range dependency relationships among elements in the

sequence, thereby explicitly modeling the temporal dynamics of user interests. Finally, the features are subjected to nonlinear transformation and enhancement through a two-layer point-based feedforward network, and a comprehensive feature vector that can reflect the shift of user interests is output.

Recommendation module: This module takes the user vector output by the user feature extraction module and the item feature vector enhanced by the cross-compression unit as input. The multi-layer perceptron (MLP) is adopted to conduct deep interaction and integration of the above features, thereby predicting the interaction probability of users towards the target item and completing the core recommendation task.

Knowledge graph-embedding module: This module is dedicated to learning the semantic representations of entities and relationships in knowledge graphs. The interaction and refinement between the head entity and the relation vector are carried out through the cross-compression unit, and the semantic correlation between it and the candidate tail entity is calculated by using the score function. The embedding learning of the knowledge graph is achieved by comparing positive and negative triples, providing rich semantic context information for recommendations.

Cross-compression unit: As a bridge connecting the recommendation module and the knowledge graph embedding module, this unit is the core for achieving bidirectional information flow. It achieves the sharing and deep integration of item features in the recommendation scenario and entity features in the knowledge graph by constructing a cross-feature matrix between the item vector and its corresponding head entity vector, and performing compression projection operations, thereby synergistically enhancing the feature representation capabilities of the two major modules and the overall performance of the model.

3.2.1 Sequence preprocessing and temporal modeling

In the input processing stage of the model, considering the inconsistency in the length of the user behavior sequence, it is necessary to perform a standardized transformation on the training sequence to adjust it to a fixed-length sequence. First, standardize the original sequence $S = (u_1, u_2, \dots, u_T)$, where T represents the actual length of the sequence, and let n be the predefined target length. When the sequence length is set, retain the nearest

n interaction items. When zero vectors are filled to length n at the left end of the sequence, a standardized sequence $S' = (pad, \dots, pad, u_1, u_2, \dots, u_T)$ is obtained, where the number of filled terms is $n - T$.

3.2.2 Adaptive time decay mechanism

To quantify the temporal distance of each interaction, the elapsed time is calculated as the difference between its timestamp and the current time, as shown in Equation (1):

$$\Delta t_{ij} = t_j - t_i \quad (1)$$

The traditional time decay model adopts a fixed decay coefficient, which makes it difficult to adapt to the diversity of user interest evolution in different application scenarios. This paper proposes that the piecewise learnable decay kernel function is:

$$\omega_t(\Delta t_{ij}) = \begin{cases} e^{-\lambda_0 \Delta t_{ij}} & \Delta t_{ij} \leq t_1 \\ e^{-\lambda_1 \Delta t_{ij}} & t_1 \leq \Delta t_{ij} \leq t_2 \\ e^{-\lambda_2 \Delta t_{ij}} & \Delta t_{ij} > t_2 \end{cases} \quad (2)$$

Among them, $\omega_t(\Delta t_i)$ is the attention weight at a time interval of t , t_1 and t_2 are the preset time boundary points, λ_0 , λ_1 and λ_2 are the learnable attenuation coefficients corresponding to the short-term, medium-term, and long-term stages respectively. This design enables the model to adaptively capture the evolution patterns of users' interests at different time scales.

Based on this, a time decay matrix $W_T \in R^{n \times n}$ is constructed, which explicitly encodes the time-dependent strength between any two interactions in the sequence and is defined as follows:

$$W_T(i, j) = \omega_t(|t_i - t_j|) \quad (3)$$

3.2.3 Embedding representation and position encoding

For item representation, an embedding matrix is employed to map each item into a low-dimensional space. The total number of items is denoted as $|V|$, and the embedding dimension as d_i . To enhance the temporal modeling capability of the sequence, a positional encoding matrix $P \in R^{n \times d}$ is introduced. For a preprocessed sequence s^u each element s_i^u corresponds to the s_i^u th row of the embedding matrix M , yielding the item embedding. Positional information is explicitly incorporated by adding the positional encoding matrix to the item embeddings, formulated as:

$$E = MS' + P \quad (4)$$

Here, $E \in R^{n \times d}$ denotes the enhanced sequence embedding representation, which encapsulates both the semantic information of the items and their positional information within the sequence.

3.2.4 Time-aware self-attention layer

Building upon the standard self-attention mechanism, a temporal decay matrix is incorporated for bias adjustment. First, the query (Q), key (K), and value (V) matrices are computed:

$$Q = EW_Q, \quad K = EW_K, \quad V = EW_V \quad (5)$$

By introducing the temporal decay matrix, the modified attention calculation is formulated as:

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{W_T \odot QK^T}{\sqrt{d_k}} \right) V \quad (6)$$

In this formulation, the embedding matrix represents the sequence data, while the query (Q), key (K), and value (V) vectors are obtained by linearly transforming the embedding matrix with learnable weight matrices W_Q , W_K and W_V , respectively; d_k denotes the dimension of the query and key vectors. Through this mechanism, self-attention dynamically assigns attention weights to each element in the sequence, thereby demonstrating significant advantages in handling sparse data and limited supervision.

To enhance the model's sensitivity to the dynamic evolution of user interests, we refine the conventional self-attention by introducing a temporal-decay factor that emphasizes more recent user behaviors. The improved attention computation is given in Equation (6).

3.2.5 Feature-enhanced feed-forward network

Although the sequential representations output by the self-attention layer capture global dependencies among elements, the feature space remains subject to linear constraints. To enhance the model's nonlinear representational capacity and achieve feature enhancement, this module introduces a two-layer pointwise feed-forward network, designed as follows:

Let the output of the self-attention layer be denoted as $S \in R^{n \times d}$, where n represents the sequence length and d the feature dimension. The feed-forward network applies identical nonlinear transformations to the features at each position independently. The specific computational process is as follows:

First layer of the feed-forward network:

$$F1 = \text{Re } LU(SW_1 + b_1) \quad (7)$$

where $W_1 \in R^{d_1 \times d_1}$ is the trainable weight matrix of the first layer, $b_1 \in R^{d_1}$ is the corresponding bias vector, and d_1 denotes the hidden layer dimension. Typically, $d_1 > d$ is set to increase the model capacity. The ReLU activation function, defined as $\text{ReLU}(x) = \max(0, x)$, provides the necessary nonlinear transformation capability.

Second-layer feed-forward computation:

$$F_2 = F_1 W_2 + b_2 \quad (8)$$

3.2.6 User representation generation

After processing through the aforementioned multi-layer neural network, we obtain the sequential representation $F_2 \in R^{n \times d}$, which incorporates rich temporal information. To transform this variable-length sequence into a fixed-dimensional user representation, this module employs an average pooling operation to aggregate features along the temporal dimension. The user representation vector $u \in R^d$ is derived by applying average pooling over the sequence dimension:

$$u = \frac{1}{n} \sum_{i=1}^n F_2(i) \quad (9)$$

where $F_2(i) \in R^d$ denotes the output feature at the i th position in the sequence.

3.3 Cross-compression Unit

The cross-compression unit serves as a bridge between the recommendation module and the knowledge graph embedding module, primarily achieving feature interaction and sharing through two key operations: cross operation and compression operation. Its core function lies in effectively integrating entity representations learned by the knowledge graph embedding module into the recommendation module, thereby leveraging the rich semantic information from the knowledge graph to enhance recommendation performance.

Specifically, the item vector v and the head entity vector h undergo feature interaction within the cross-compression unit. First, the two are combined to form a cross-feature matrix $C \in R_{d \times d}$, where each element C_{ij} represents the interaction strength between corresponding dimensions of the item vector v and the head entity vector h . Subsequently, a compression operation projects this cross-feature matrix back into the latent representation space, generating

updated item and head entity vectors – denoted as v' and h' respectively – thereby achieving feature fusion and representation refinement.

$$C_l = v_l h_l^T = \begin{bmatrix} v_l^1 h_l^1 & v_l^1 h_l^2 & \dots & v_l^1 h_l^{d_2} \\ v_l^2 h_l^1 & v_l^2 h_l^2 & \dots & v_l^2 h_l^{d_2} \\ \dots & \dots & \dots & \dots \\ v_l^{d_2} h_l^1 & v_l^{d_2} h_l^2 & \dots & v_l^{d_2} h_l^{d_2} \end{bmatrix} \quad (10)$$

$$v_{l+1} = C_l W^V + C_l^T W^{E \rightarrow V} + b_l^V \quad (11)$$

$$h_{l+1} = C_l W^{V \rightarrow E} + C_l^T W^E + b_l^E \quad (12)$$

In the above formula, $C_l \in R^{d_2 \times d_2}$ denotes the cross-feature matrix at layer l . The trainable weight vectors W_l^V , $W_l^{E \rightarrow V}$, $W_l^{V \rightarrow E}$ and W_l^E are respectively responsible for capturing item–item, item–entity, entity–item, and entity–entity feature interactions. This process not only deepens the model’s understanding of item and entity characteristics but also boosts the recommendation system’s performance through feature sharing.

3.4 Recommendation Prediction and Multi-task Training

3.4.1 Recommendation prediction

In the recommendation module, the user feature vector u is taken as input and transformed through a fully connected mapping $\text{Re } LU(W_u \cdot u + b_u)$ to obtain a d -dimensional higher-order representation u^L . On the item side, the cross-compression unit is retained, which performs low-rank interaction between the initial item vector v and the corresponding KG entity h , outputting a same-dimensional vector v^L .

$$u^L = MLP(u) = \text{ReLU}(W_u \cdot u + b_u) \quad (13)$$

$$v^L = CC^L(v, h)[v] \quad (14)$$

The recommendation prediction module is responsible for comprehensively processing user representations and item information to output the probability of interaction between a user and a target item. Specifically, the user vector $u \in R^d$ output by the user feature extraction module and the optimized item vector $v^L \in R^d$ from the cross-compression unit are subjected to feature extraction, forming a joint representation $z = u \oplus v \in R^{2d}$, where \oplus denotes the vector concatenation operation.

This joint representation is then fed into a multi-layer perceptron (MLP) for deep feature interaction and nonlinear transformation:

$$\hat{y}_{uv} = \sigma(MLP(z)) \quad (15)$$

$\sigma(\cdot)$ represents the sigmoid function, ensuring the output lies within the range $[0, 1]$, denoting the predicted interaction probability between user u and item v . The MLP consists of two fully connected layers, with a *ReLU* activation function applied between them to enhance nonlinear modeling capability.

3.4.2 Knowledge graph link prediction

To improve the accuracy of item semantic representations, the model concurrently performs knowledge graph link prediction. This task utilizes the optimized entity vector h^L output by the cross-compression unit and the relation embedding vector r to predict the tail entity \hat{t} through a specific scoring function.

For a given triple (h, r, t) , its plausibility score is calculated as follows:

$$\hat{t} = M^K \left(\begin{bmatrix} h_L \\ r_L \end{bmatrix} \right) \quad (16)$$

$$\text{score}(h, r, t) = f_{KG}(t, \hat{t}) \quad (17)$$

3.4.3 Adaptive multi-task loss function

To achieve collaborative optimization between the recommendation task and knowledge graph learning, this paper proposes an adaptive multi-task loss function defined as follows:

$$L_2 = L_{RS} + g(\tau) \cdot L_{KG} \quad (18)$$

The recommendation loss LRS adopts a time-weighted binary cross-entropy form:

$$L_{RS} = - \sum_{u,v} W_T [y_{uv} \log \hat{y}_{uv} + (1 - y_{uv}) \log(1 - \hat{y}_{uv})] \quad (19)$$

where W_T is the time decay weight matrix, y_{uv} denotes the ground-truth interaction label between user u and item v , and \hat{y}_{uv} represents the predicted interaction probability.

The knowledge graph loss LKG employs a margin-based ranking loss:

$$L_{KG} = \sum_{(h,r,t) \in G} \sum_{(h,r,t') \notin G} \max(0, \gamma + \text{score}(h, r, t') - \text{score}(t, \hat{t}(h, r))) \quad (20)$$

where $\gamma > 0$ is the margin hyperparameter, and t' denotes the negative-sampled tail entity.

The time-gated function $g(\tau)$ employs a learnable Sigmoid curve to ensure a smooth transition, as detailed below:

$$g(\tau) = \sigma(\vartheta - \beta\tau) = \frac{1}{1 + e^{-(\vartheta - \beta\tau)}} \quad (21)$$

Here, τ represents the training epoch number, while β_0 and β_1 are learnable parameters that control the initial weight and decay rate, respectively. This design enables the model to prioritize knowledge graph semantic learning in the early stages of training and gradually shift focus toward recommendation task optimization as training progresses. Prior to training, β_0 and β_1 are initialized to 2.0 and 0.02, respectively. During training, these parameters are automatically updated via gradient back-propagation without requiring manual intervention.

4 Experiments and Analysis

4.1 Data Set

This paper focuses on the practical exploration of CRT and Top-K tasks on three public datasets derived from real application scenarios. These datasets are the Last.FM dataset in the music field, the Book-Crossing dataset in the book field, and the MovieLens-1M dataset in the film field. They each have their own characteristics in terms of scale and sparsity.

The Last.FM dataset: It covers the listening records of 1872 users to 3846 pieces of music on the Last.FM (<https://www.last.fm>) online music platform, providing rich user behavior data for music recommendation research.

Book-Crossing dataset [31]: Derived from the online book community Book-Crossing, it not only contains over a hundred thousand comments of users on books but also reflects the degree of users' preferences through users' ratings of books (including 10 levels of displayed ratings and one level of implicit ratings). It is one of the representative datasets with relatively high data sparsity (<https://archive.ics.uci.edu/ml/datasets/Book-Crossing>).

The MovieLens-1M dataset [32]: It is an open-source dataset based on users' ratings of movies, which includes feedback from users on movie websites, with ratings ranging from 1 to 5 (<https://grouplens.org/datasets/movielens/>).

Table 1 The statistics of each data set

| Data set | Users | Items | Interactions | Entities | Relations | Triple | Sparsity |
|---------------|---------|--------|--------------|----------|-----------|---------|----------|
| Last.FM | 1,872 | 3,846 | 42,346 | 9,366 | 60 | 15,518 | 99.41% |
| Book-Crossing | 17,860 | 14,976 | 139,746 | 77,903 | 25 | 151,500 | 99.95% |
| MovieLens-1M | 138,159 | 16,954 | 13,501,622 | 102,567 | 32 | 499,474 | 95.81% |

For the purposes of this study, the movie dataset is preprocessed so that records with a rating of 3 or above are treated as positive samples (labeled as 1), and the rest are treated as negative samples (labeled as 0). Similarly, in the music and book datasets, if a user rates an item (music or book), the user is considered to have interacted with the item and is labeled as positive (1). Whereas if the user has not rated the item, the user has little or no interest in the item and is labeled as negative (0). Table 1 details the statistics of each dataset.

4.2 Evaluation Criteria

To comprehensively evaluate the overall performance of the model, this study adopted industry-recognized evaluation metrics for two types of tasks: click-through rate (CTR) prediction and Top-K recommendation. All indicators are calculated based on the model's prediction results on the test set.

4.2.1 Evaluation indicators for CTR prediction tasks

CTR prediction is essentially a binary classification problem. We adopt the following two core indicators:

AUC (area under the ROC curve): It measures the overall ability of a model to make correct predictions, is insensitive to category imbalance, and is the gold-standard indicator for evaluating ranking performance. Its calculation is based on the area under the ROC curve, and the calculation formula is shown as follows:

$$AUC = \frac{\sum_{i=1}^P \sum_{j=1}^N I(p_i > p_j)}{P \times N} \quad (22)$$

Among them, $N+$ and $N-$ respectively represent the number of positive and negative samples in the test set, p_i and p_j respectively represent the predicted scores of the i th positive sample and the j th negative sample, and $I(\cdot)$. It is an indicator function. When the condition is met, its value is 1; otherwise, it is 0.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (23)$$

Among them, TP (true positive) and TN (true negative) respectively represent the numbers of positive and negative samples that are correctly predicted, while FP (false positive) and FN (false negative) respectively represent the numbers of positive and negative samples that are wrongly predicted.

4.2.2 Top-K recommendation task evaluation index

In the Top-K recommendation scenario, we use two key metrics, namely the widely used precision (Precision@K) and recall (Recall@K), to evaluate the recommendation effect of the model.

The precision rate Precision@K focuses on the quality of the recommendation results, calculating how many of the Top-K items recommended for the user are truly of interest to the user. Its calculation formula is shown in the following.

$$\text{Precision@K} = \frac{1}{|U|} \sum_{u=1}^{|U|} \frac{|R_u(K) \cap T_u|}{K} \quad (24)$$

In the formula, U represents the total number of test users, $R_u(K)$ is the set of the Top-K items recommended for user u , and T_u is the set of items that user u actually interacts with on the test set.

The recall rate Recall@K measures the ability of the recommendation results to cover users' interests. It calculates how many items that users are truly interested in are successfully recommended by the model. The calculation formula is shown as follows:

$$\text{Recall@K} = \frac{1}{|U|} \sum_{u=1}^{|U|} \frac{|R_u(K) \cup T_u|}{|T_u|} \quad (25)$$

In this study, we set $K = 10$ for evaluation. All experiments were independently repeated three times, and their average values and 95% confidence intervals were reported.

4.3 Comparing Baselines

To ensure the reliability of the proposed model's performance, this study adopted multiple knowledge graph-based recommendation models as benchmarks (baselines) for comparative analysis. All models were evaluated and tested on the same dataset.

- DKN [20]: This model realizes the fusion of the knowledge graph and deep learning. Its core lies in using the knowledge graph embedding

method to take entity embedding and word embedding as multiple information channels and integrating the information from these channels into a convolutional neural network (CNN) architecture for prediction.

- RippleNet [17]: It is a classical model based on the propagation mechanism, which propagates users' personalized preferences in the knowledge graph in a ripple diffusion manner. The model uses the embedding of neighbor nodes to update the potential features of users, thereby enriching the representation dimension of users and improving the accuracy and effectiveness of the recommendation system.
- MKR [19]: This model adopts the framework of multi-task learning and integrates knowledge graph feature learning and the recommendation module into two interrelated tasks for alternating learning through the cross-compression unit. The two tasks complement each other and effectively compensate for the lack of information coverage of each other, thus enhancing the model's accuracy in predicting users' un-interacted items.
- KGCN [36]: This model combines the knowledge graph and the graph neural network. It uses graph convolution technology to capture the high-order information of the project's neighbor nodes, then generates an accurate item embedding representation, and realizes an efficient recommendation function based on this.
- CKAN [37]: The CKAN model proposed by Wang et al. is based on the graph attention network, which highlights the importance of collaborative information in the interaction between users and items and combines it with the knowledge graph to obtain the feature representation of users and items.

4.4 Parameter Settings

This study conducted experiments under the PyTorch 1.11 deep learning framework. The hardware environment was configured with an Intel Xeon Gold 6248R CPU and an NVIDIA A100 GPU (80 GB video memory).

The experiment was conducted based on three public datasets, including Last.FM, Book-Crossing, and MovieLens-1M.

In the data preprocessing stage, the historical interaction records of each user are first arranged in ascending order according to the timestamp to construct a user behavior sequence with strict temporal relationships. According to the characteristics of different datasets, the maximum sequence lengths are set respectively (Last.FM: 400, Book-Crossing: 300, MovieLens-1M: 500).

Table 2 Hyperparameter settings

| Hyperparameter | Notation | Range of Values | Final Value | Notes |
|-------------------------|-------------------------------------|----------------------|-------------|---|
| Embedding dimension | d | {32, 64, 128} | 64 | Shared dimension for users, items, and entities |
| Maximum sequence length | n | {200, 300, 400, 500} | * | Adjusted according to the dataset |
| Batch size | B | {128, 256, 512} | 512 | Training batch size |
| Initial learning rate | η | {0.1, 0.01, 0.001} | 0.001 | Parameter for the Adam optimizer |
| Decay coefficients | $(\lambda_0, \lambda_1, \lambda_2)$ | – | Learnable | Time decay parameters |
| Time thresholds | (τ_1, τ_2) | (7, 30) | Fixed | In units of days |
| KG propagation order | K | {1, 2, 3} | 3 | Number of knowledge graph neighborhood aggregation layers |
| Gating initial values | (θ, β) | – | (2.0, 0.02) | Initialization for learnable parameters |

For sequences exceeding the length, the most recent interaction records are retained, and for sequences with insufficient lengths, zero vectors are used for filling to ensure the uniformity of the input dimensions.

On this basis, strictly following the chronological order, the training set, validation set, and test set are divided in a ratio of 6:2:2 to effectively avoid future information leakage problems. Meanwhile, to reduce the impact of cold start issues, users and items with less than five interactions are pre-excluded, effectively improving data quality.

To verify the effectiveness of the proposed AdaTKGR framework for the system, this section provides a detailed description of the experimental parameter settings. Table 2 provides a complete list of the key hyperparameter configurations of the model and their optimal value ranges.

Training strategy and optimization: During the model training process, we used the Adam optimizer to update the parameters. Its hyperparameters were set as $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 10^{-8}$. The initial learning rate was 0.001.

To prevent overfitting, multiple regularization strategies were employed during the training process. Dropout technology with a ratio of 0.2 was applied in the feed-forward network layer. Meanwhile, L2 regularization constraints with a weight attenuation coefficient of 10^{-5} were imposed, and gradient clipping with a maximum limit of 2.0 was performed on the gradient norm.

To dynamically adjust the learning step size, we introduced a learning rate scheduling mechanism based on the performance plateau. When the validation set loss fails to improve for five consecutive epochs, the learning rate is halved.

The training process adopts an early-stop strategy. If the increase in the AUC index of the validation set is less than 1×10^{-4} for 10 consecutive epochs, the training will be terminated in advance, and the maximum number of training rounds is limited to 100 epochs.

4.5 Analysis of Experimental Results

4.5.1 Model comparison test

To comprehensively evaluate the effectiveness of the AdaTKGR model, this subsection systematically compares the performance of the proposed model with five baseline methods on three benchmark datasets and conducts an in-depth analysis of the results from the perspective of statistical significance.

To verify the findings proposed in this paper, Table 3 presents the AUC and ACC index results of each model in the CTR prediction task. Table 4 shows the AUC and ACC results of each model in CTR prediction.

The experimental results show that: In the CTR prediction task, the AdaTKGR model proposed in this paper has achieved significant performance improvements on all three benchmark datasets. Specifically, on the MovieLens-1M dataset, the AUC index of the model reached 0.941, and

Table 3 Performance evaluation of different models on CTR prediction using AUC and accuracy

| Model | MovieLens-1M | | Book-Crossing | | Last.FM | |
|-----------|--------------|-------|---------------|-------|---------|-------|
| | AUC | ACC | AUC | ACC | AUC | ACC |
| RippleNet | 0.925 | 0.874 | 0.824 | 0.771 | 0.843 | 0.790 |
| DKN | 0.879 | 0.856 | 0.795 | 0.745 | 0.821 | 0.771 |
| MKR | 0.917 | 0.869 | 0.812 | 0.757 | 0.834 | 0.780 |
| KGCN | 0.931 | 0.879 | 0.835 | 0.777 | 0.858 | 0.800 |
| CKAN | 0.910 | 0.835 | 0.842 | 0.783 | 0.866 | 0.798 |
| AdaTKGR | 0.941 | 0.890 | 0.854 | 0.795 | 0.871 | 0.810 |

Table 4 Comparison of top-10 recommended performance (Precision@10/Recall@10)

| | MovieLens-1M | | Book-Crossing | | Last.FM | |
|-----------|--------------|-----------|---------------|-----------|--------------|-----------|
| | Precision@10 | Recall@10 | Precision@10 | Recall@10 | Precision@10 | Recall@10 |
| RippleNet | 0.162 | 0.288 | 0.118 | 0.203 | 0.135 | 0.234 |
| DKN | 0.149 | 0.265 | 0.105 | 0.184 | 0.128 | 0.221 |
| MKR | 0.158 | 0.281 | 0.112 | 0.195 | 0.132 | 0.228 |
| KGCN | 0.165 | 0.294 | 0.122 | 0.211 | 0.141 | 0.245 |
| CKAN | 0.159 | 0.283 | 0.124 | 0.215 | 0.144 | 0.251 |
| AdaTKGR | 0.168 | 0.301 | 0.128 | 0.223 | 0.147 | 0.258 |

the accuracy rate reached 0.890, which increased by 1.08% and 1.25% respectively compared with the optimal baseline model. On the data-sparse Book-Crossing dataset, the AUC and accuracy rates reached 0.854 and 0.795 respectively, with improvements of 1.43% and 1.53%. It also maintains a stable performance advantage on the Last-FM dataset. These results fully demonstrate the effectiveness of the model in the click-through rate prediction task.

In the Top-K recommendation tasks, AdaTKGR also demonstrated outstanding performance. On the two key metrics of Precision@10 and Recall@10, the model performed significantly better than all baseline methods on the three datasets. Especially on the sparse dataset Book-Crossing, the precision and recall rates reached 0.128 and 0.223 respectively, increasing by 3.23% and 3.72% compared with the optimal baseline. This result highlights the strong adaptability of the model in data-scarce scenarios.

4.5.2 Ablation experiment

In order to systematically evaluate the contribution of each key component of AdaTKGR, we design four groups of ablation models and conduct comparative experiments on three public datasets, Book-Crossing, Last. FM and MovieLens-1M. In the experiment, the embedding dimension $d = 64$, the maximum number of training rounds 100, and the patience of early stopping 10 were uniformly set, and the other hyperparameters were consistent with the full model. All results are taken from the mean of three independent replicates of the experiment, and 95% confidence intervals are given.

- AdaTKGR-Full represents the complete model. In this paper, the time decay kernel, cross-compression unit, and time-gated multitasking loss are simultaneously enabled as the performance upper limit.
- AdaTKGR-T represents the removal of the time kernel, setting the time decay weight matrix WT to the identity matrix, with all historical

Table 5 Ablation analysis results in terms of AUC and accuracy

| Model Variant | MovieLens-1M | | Last.FM | Average Decline |
|---------------|---------------|-------------|-------------|-----------------|
| | Book-Crossing | | | in Performance |
| | AUC/ACC | AUC/ACC | AUC/ACC | AUC/ACC |
| AdaTKGR-Full | 0.941/0.890 | 0.854/0.795 | 0.871/0.810 | – |
| AdaTKGR-T | 0.923/0.872 | 0.833/0.776 | 0.852/0.792 | 2.1%/2.2% |
| AdaTKGR-G | 0.935/0.883 | 0.840/0.782 | 0.862/0.802 | 0.9%/1.0% |
| AdaTKGR-C | 0.936/0.885 | 0.835/0.778 | 0.858/0.798 | 1.3%/1.4% |

behavior weights being the same, verifying the necessity of time dynamic modeling.

- AdaTKGR-G represents the fixed multi-task gating weight. Let the time gating function $g(\tau)$ of Equation (17) = 0.5 and no longer adaptively adjust with the training process to test the contribution of the dynamic weight balance mechanism.
- AdaTKGR-C represents the removal of cross-compression units. Feature fusion is performed using a simple vector concatenation operation $[u^L|v^K]$ to investigate the influence of low-rank bilinear fusion on heterogeneous information interaction.
- The experimental results show that AdaTKGR-Full achieved optimal performance on all three datasets, verifying the effectiveness of the overall framework.
- Furthermore, the effectiveness of each component of the model highly depends on the data characteristics. When the data is abundant, time-series dynamic modeling is the key to improving performance. When the data is sparse, it becomes particularly important to deeply integrate external knowledge with adaptive optimization strategies.
- AdaTKGR-T exhibited the most significant performance degradation on all datasets, especially on the sparse dataset Book-Crossing, where the AUC decreased by 2.1%. This fully demonstrates the crucial role of explicit temporal modeling in capturing the dynamic evolution of user interests. On the dense dataset MovieLens-1M, the performance degradation was relatively small (1.8%), indicating that rich interaction data can make up for the lack of time-series information to a certain extent.
- The performance loss of AdaTKGR-G is more obvious in sparse scenarios (Book-Crossing: -1.4% , Last.FM: -0.9%), verifying the importance of adaptive task weight adjustment when data is scarce.

As the training progresses, the fixed weights cannot flexibly adjust the learning focus of the recommendation task and the knowledge graph task, resulting in the model converging to a sub-optimal solution.

- The performance degradation (1.9%) of AdaTKGR-A on Book-Crossing was significantly higher than that on MovieLens-1M (0.5%), demonstrating the special value of deep feature interaction in sparse data. When user behavior data is limited, the semantic information extracted from the knowledge graph can effectively make up for the deficiency of collaborative signals through fine-grained feature interaction.

4.5.3 Hyperparameter sensitivity analysis

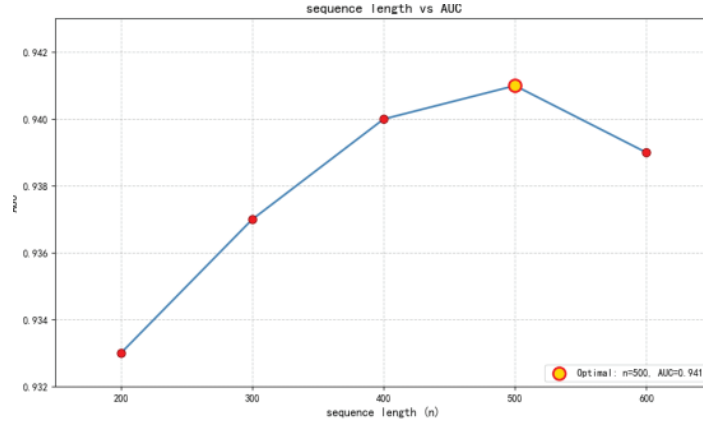
To deeply explore the impact of model hyperparameters on performance, this study conducted a systematic sensitivity analysis on the sequence length n and embedding dimension d of two key hyperparameters. All experiments were completed on the MovieLens-1M dataset, with AUC as the main evaluation metric.

The sequence length n determines the range of historical behaviors considered by the model and has a significant impact on capturing users' long-term interest patterns. We tested the performance variations of $n \in \{100, 200, 300, 400, 500\}$, and the results are shown in Figure 2(a).

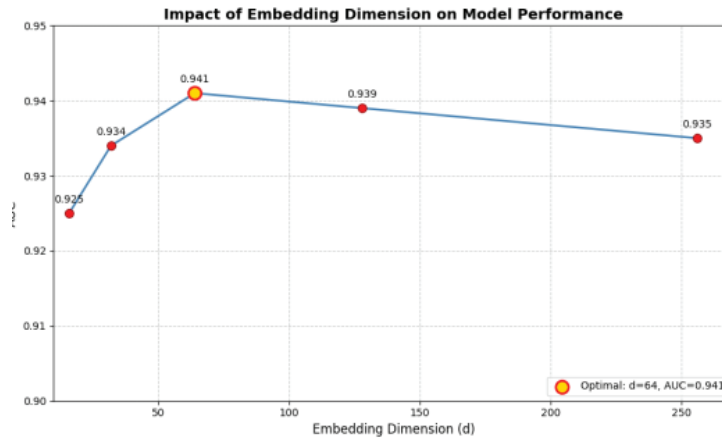
The embedding dimension d directly affects the representational ability of the model. We evaluated the performance of $d \in \{16, 32, 64, 128, 256\}$, and the results are shown in Figure 2(b).

The experimental results show that:

- When the sequence length increases from 100 to 300, the model performance shows a significant improvement trend, with the AUC rising from 0.928 to 0.937, an increase of 0.9%. Within the range of 300 to 500, the performance improvement tends to level off, reaching a peak of 0.941 at 500, which is only a 0.4% increase compared to the value at 300. After the sequence length exceeds 500, due to the introduction of noise information and the increase in computational complexity, the performance slightly declines. The results show that moderate sequence length expansion helps the model establish a more complete user interest profile, but overly long sequences will introduce historical noise and increase the risk of overfitting. Based on the balance consideration of efficiency and effect, this study ultimately selects $n = 500$ as the optimal configuration.
- When the dimension increases from 16 to 64, the AUC shows stable growth, rising from 0.925 to 0.941. The optimal performance is achieved



(a)



(b)

Figure 2 (a) Analysis of the impact of sequence length on model performance. (b) Analysis of the impact of embedding dimension on model performance.

when $d = 64$, at which point the model has sufficient representational ability while maintaining computational efficiency. When the dimension continues to increase to 128 and 256, the AUC decreases to 0.939 and 0.935 respectively, indicating the occurrence of overfitting. The results show that the selection of embedding dimensions needs to strike a balance between model capacity and generalization ability. Too low a dimension limits the model's expressive power, while too high a dimension is prone to overfitting. In this study, $d = 64$ demonstrates the best comprehensive performance within the test range.

4.5.4 Discussion

Based on the aforementioned systematic experimental verification, this section conducts an in-depth analysis and discussion of the performance of the AdaTKGR model. Starting from each key component of the model architecture, it explains the internal mechanism of its performance advantages.

- Performance analysis of time series modeling: The experimental results show that AdaTKGR performs particularly well on the MovieLens-1M dataset with strong dynamics, which is mainly attributed to its innovative time-aware mechanism. Specifically, the model achieves adaptive capture of the evolution pattern of user interests through a learnable piecewise decay kernel function. Unlike the traditional method that uses a fixed attenuation coefficient, the parameters λ_0 , λ_1 , and λ_2 in this model can dynamically adjust the attenuation rates of short-term, medium-term, and long-term interests according to specific scenarios. This design enables the model to exhibit stronger robustness when facing the changing interest patterns of different types of users. Especially in scenarios where user behavior patterns undergo significant changes, the time decay matrix WT can effectively increase the weight of recent interactions and ensure that the recommendation results are consistent with the user's current interests.
- Discussion on the knowledge fusion mechanism: On the sparse Book-Crossing dataset, AdaTKGR demonstrates significant advantages, mainly attributed to its deep feature fusion design. The cross-compression unit achieves fine-grained feature interaction between the recommendation domain and the knowledge domain through low-rank bilinear transformation. Specifically, this mechanism not only takes into account the direct correlation between item features and entity features but also captures the bidirectional dependency between features through transposition operations. This deep integration approach is particularly important in scenarios with sparse data. When the collaborative filtering of signals is insufficient, the model can obtain supplementary information from the structured semantics of the knowledge graph, effectively alleviating the cold start problem. Furthermore, the low-rank constraint in the feature interaction process ensures that the model maintains its expressiveness while avoiding the risk of overfitting.
- Evaluation of multi-task optimization strategies: The adaptive multi-task learning mechanism is another core innovation of AdaTKGR. Through the dynamic adjustment of the time gating function $g(\tau)$, the model achieves an intelligent balance between recommendation tasks

and knowledge graph tasks during the training process. In the early stage of training, the model focuses on learning the structured semantics of the knowledge graph and establishing stable entity relationship representations. As the training progresses, the focus gradually shifts to optimizing the recommendation performance. This step-by-step optimization strategy not only accelerates model convergence but also significantly enhances the final performance. Experimental data show that this mechanism works particularly well on moderately sparse datasets such as Last-FM, proving its unique value in balancing data-driven and knowledge-driven approaches.

- Validation of model generalization capability: The performance across datasets validates the powerful generalization ability of AdaTKGR. Whether on dense MovieLens-1M, highly sparse Book-Crossing, or moderately sparse Last-FM datasets, the model maintains stable performance advantages. This strong generalization stems from the following aspects: firstly, the time decay mechanism provides a universal modeling method for the dynamics of user behavior; secondly, the cross-compression unit establishes a transferable feature interaction framework; finally, adaptive multi-task learning ensures that the model can automatically adjust the optimization focus according to the characteristics of the data. These designs enable AdaTKGR to adapt to different recommendation scenarios and data distributions, demonstrating excellent practicality.

The comparative analysis with the baseline model revealed several important findings. Compared with path propagation methods such as RippleNet, AdaTKGR avoids the interference of outdated interests through explicit time modeling. Compared with multi-task learning methods such as MKR, its dynamic gating mechanism provides more refined optimization control. Compared with graph neural network methods such as KGCN, the cross-compression unit achieves a deeper level of feature interaction.

These comparisons not only demonstrate the advanced nature of the method proposed in this paper but also provide important inspirations for the future development of recommendation systems: an effective recommendation model needs to simultaneously consider the collaborative design of three dimensions – temporal dynamics, knowledge fusion, and optimization strategies.

In summary, AdaTKGR, through the organic integration of three key technologies – time series awareness, knowledge fusion, and multi-task optimization – significantly enhances recommendation performance while

maintaining the model's generalization ability. Future research directions include exploring more complex temporal pattern modeling, introducing multimodal knowledge information, and optimizing the knowledge transfer mechanism in cross-domain recommendation scenarios.

5 Conclusions and Future Work

Aiming at the limitation that existing knowledge graph recommendation algorithms struggle to simultaneously capture the time-series evolution of user interests and the deep interaction of heterogeneous information, this paper proposes a knowledge graph-enhanced recommendation framework, AdaTKGR, with adaptive time decay weight. The framework employs time-aware self-attention to mine user preference shifts and uses cross-compression units to achieve low-rank bilinear fusion of user-item interaction features and knowledge graph semantics. Thus, the model can leverage both timeliness and structured user-item representations, significantly enhancing the recommendation accuracy and interpretability.

Experimental results demonstrate that AdaTKGR outperforms other state-of-the-art methods on three public datasets, Book-Crossing, Last.FM, and MovieLens-1M, verifying its effectiveness.

In future work, we will consider introducing richer multivariate attributes (such as type, label, etc.) on the item side to enhance the item representation. We will explore more efficient graph neural networks or Transformer variants to further improve the extraction efficiency and expression ability of user and item features.

Acknowledgement

This work was supported in part by Youth Fund Project of Sichuan University Jinjiang College (QNJJ-2025-A08); in part by Meishan Science and Technology Bureau Project (2024KJZD164); in part by Network and Data Security Key Laboratory of Sichuan Province, University of Electronic Science and Technology of China Chengdu (No. NDS2023-3).

References

- [1] Liu Zewei. Research on Social Recommendation Methods Based on Graph Neural Networks[D]. Tianjin University of Technology, 2024. DOI: 10.27360/d.cnki.gtlgy.2024.000653.

- [2] Tang J X, Wang K. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding[C]// Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining. New York: ACM, 2018: 565–573.
- [3] Xiang L. Practice of Recommender Systems [M]. Beijing: Posts & Telecom Press, 2012.
- [4] Tan Q Y, Zhang J W, Yao J C, et al. Sparse-Interest Network for Sequential Recommendation [C]// Proceedings of the 14th ACM International Conference on Web Search and Data Mining. New York: ACM, 2021: 598–606.
- [5] Ma J X, Zhou C, Cui P, et al. Learning Disentangled Representations for Recommendation [C]// Advances in Neural Information Processing Systems, 2019: 5712–5723.
- [6] Liu Junliang, Li Xiaoguang. Advances in Personalized Recommendation System Technology [J]. Computer Science, 2020, 47(7): 47–55.
- [7] Li K, Wan P Z, Zhang D Z. Collaborative Filtering Recommendation Algorithm Based on Improved User Similarity Measure and Scoring Forecast [J]. Journal of Chinese Computer Systems, 2018, 39(3): 567–571.
- [8] Chang, L., et al. “Review of Recommendation Systems Based on Knowledge Graph.” [J]. CAAI Transactions on Intelligent Systems 14.2 (2019): 207-216.
- [9] Guo Q, Zhuang F Z, Qin C, et al. A Survey on Knowledge Graph-Based Recommender Systems [J]. IEEE Transactions on Knowledge and Data Engineering, 2020, 34(8): 3549–3568.
- [10] Qin C, Zhu H S, Zhuang F Z, et al. A Survey on Knowledge Graph-Based Recommender Systems [J]. Scientia Sinica: Informations, 2020, 50(7): 937–956.
- [11] Zhu D L, Wen Y, Wan Z C. Review of Recommendation Systems Based on Knowledge Graph [J]. Data Analysis and Knowledge Discovery, 2021, 5(12): 1–13.
- [12] Zhao Y, Liu L, Wang H, et al. A Survey of Knowledge Graph-Based Recommender Systems [J]. Journal of Frontiers of Computer Science and Technology, 2023, 17(4): 771–791.
- [13] Shuai Jie, Zhang Kun, Wu Le, et al. A Review-Aware Graph Contrastive Learning Framework for Recommendation [C] // Proc of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: Association for Computing Machinery, 2022: 1283–1293.

- [14] Wang H, Zhang F, Hou M, et al. SHINE: Signed Heterogeneous Information Network Embedding for Sentiment Link Prediction [C] // Proc of the 11th ACM International Conference on Web Search and Data Mining. New York: Association for Computing Machinery, 2018: 592–600.
- [15] Zhang F, Yuan N J, Lian D, et al. Collaborative Knowledge Base Embedding for Recommender Systems [C] // Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, Aug 13–17, 2016. New York: ACM, 2016: 353–362.
- [16] Wang H, Zhang F, Zhang M, et al. Knowledge-Aware Graph Neural Networks with Label Smoothness Regularization for Recommender Systems [C] // Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019: 968–977.
- [17] Wang H, Zhang F, Wang J, et al. RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems [C] // Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Oct 22–26, 2018. New York: ACM, 2018: 417–426.
- [18] Zheng Chenwang. Research on Point-of-Interest Recommendation Algorithm Based on User Dynamic Preferences and Attention Mechanism [D]. Beijing Jiaotong University, 2021. DOI: 10.26944/d.cnki.gbfju.2021.001734.
- [19] Wang H, Zhang F, Zhao M, et al. Multi-Task Feature Learning for Knowledge Graph Enhanced Recommendation [C] // Proc of the 2019 World Wide Web Conference. New York: Association for Computing Machinery, 2019: 2000–2010.
- [20] Wang H, Zhang F, Xie X, et al. DKN: Deep Knowledge-Aware Network for News Recommendation [C] // Proceedings of the 2018 World Wide Web Conference. 2018: 1835–1844.
- [21] Cho K, Van Merriënboer B, Gulcehre C, et al. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation [J]. arXiv:1406.1078, 2014.
- [22] Tang Hong, Fan Sen, Tang Fan, et al. Recommendation Algorithm Integrating Knowledge Graph and Attention Mechanism [J]. Computer Engineering and Applications, 2022, 58(05): 94–103.
- [23] Cheng Huasong, Xiong Caiquan, KE Yuanzhi, et al. Neural Network Model for News Recommendation Based on Knowledge Graph [J]. Journal of Hubei University of Technology, 2023, 38(04): 82–87.

- [24] Wang H, Zhang F, Wang J, et al. RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems [C] // Proceedings of the 27th ACM International Conference on Information and Knowledge Management. 2018: 417–426.
- [25] Wei Jishu. Graphic Retrieval Cross Modal Entity Oriented Alignment Research [D]. Qilu University of Technology, 2023. DOI: 10.27278/d.cnki.gsdqc.2023.000668.
- [26] Wang X, Wang D, Xu C, et al. Explainable Reasoning over Knowledge Graphs for Recommendation [C] // Proc of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2019), Paris, France, July 2019. New York: ACM, 2019: 295–304.
- [27] Wang H, Zhang F, Zhao M, et al. Multi-Task Feature Learning for Knowledge Graph Enhanced Recommendation [C] // The World Wide Web Conference. 2019: 2000–2010.
- [28] Xian Y, Fu Z, Muthukrishnan S, et al. Reinforcement Knowledge Graph Reasoning for Explainable Recommendation [C] // Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, Jul 21–25, 2019. New York: ACM, 2019: 285–294.
- [29] Wang X, He X, Cao Y, et al. Knowledge Graph Attention Network for Recommendation [C] // Proc of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: Association for Computing Machinery, 2019: 950–958.
- [30] Cao Haodong, Wang Haitao, He Jianfeng. Date-Aware Sequence Recommendation Algorithm Integrating Local Information of Sequences [J]. *Computer Engineering and Science*, 2024, 46(04): 734–742.
- [31] Davagdorj K, Park K H, Ryu K H. A Collaborative Filtering Recommendation System for Rating Prediction [C] // Proceedings of the 15th International Conference on IHHMSP in Conjunction with the 12th International Conference on FITAT, Jilin, Jul 18–20, 2019. Singapore: Springer, 2019: 265–271.
- [32] Harper F M, Kontan J A. The MovieLens Datasets: History and Context [J]. *ACM Transactions on Interactive Intelligent Systems*, 2015, 5(4): 1–19.
- [33] Yang Ran. Recommendation System Based on Partial Correlation Modeling Research [D]. South China University of Technology, 2021. DOI: 10.27151/d.cnki.ghnlu.2021.001609.

- [34] Feng S, Li X, Zeng Y, et al. Personalized Ranking Metric Embedding for Next New POI Recommendation [C] // IJCAI'15 Proceedings of the 24th International Conference on Artificial Intelligence. New York: ACM, 2015: 2069–2075.
- [35] Kang W C, McAuley J. Self-Attentive Sequential Recommendation [C] // 2018 IEEE International Conference on Data Mining (ICDM). IEEE, 2018: 197–206.
- [36] Wang H, Zhao M, Xie X, et al. Knowledge Graph Convolutional Networks for Recommender Systems [C] // Proc of the 2019 World Wide Web Conference. New York: Association for Computing Machinery, 2019: 3307–3313.
- [37] Wang Z, Lin G, Tan H, et al. CKAN: Collaborative Knowledge-aware Attentive Network for Recommender Systems [C] // Proc of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: Association for Computing Machinery, 2020: 219–228.

Biographies



Xuelian Zhang received her master's degree in information security from Chengdu University of Information Technology in 2018. She currently works as a lecturer in the College of Computer Science at Sichuan University Jinjiang College, where she also serves as an assistant in the University Computer Fundamentals Teaching and Research Section. Her research areas include data mining, deep learning, and social network analysis.



Mian Ren received her bachelor's degree in information security from Chengdu University of Information Technology in 2015 and her master's degree in information security from the same university in 2018. Currently, the author serves as a Lecturer in the School of Information Engineering at Sichuan Vocational and Technical College of Posts and Telecommunications, working as a faculty member specializing in information security. The author's research areas include information security, deep learning, multi-modal fusion, and large model security.



Chunling Xiang received her master's degree in Internet of Things technology from Chengdu University of Information Technology in 2016. She began her career in 2016 at Chengdu Xinan Eureka Co., Ltd., where she was engaged in information security-related work. Currently, she serves as a faculty member in Information Security at the Department of Cyberspace Security, Chengdu College of University of Electronic Science and Technology of China. Her research focuses on applied cryptography security, side-channel security, and chip security. She has authored four patents and participated in the development of two industry standards.