Integrating Social and Knowledge Graphs with Time Decay Mechanisms*

Can Lin[1*] , Xiaoyang Han[1*], Zengyi Yu[2*], and Jiaxin Du[1] (✉)

[1] College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, 310023, China
jiaxin.joyce.du@gmail.com
[2] College of Educational Science and Technology, Zhejiang University of Technology, Hangzhou, 310023, China

Abstract. This paper introduces an innovative approach to recommender systems by integrating social connectivity enhancement, knowledge graph augmentation, and a temporal decay mechanism within a Graph Neural Network (GNN) architecture. Our proposed model, ISKG-TD, aims to capture the dynamic nature of user preferences by incorporating temporal information into the user-item interactions, thereby addressing the limitations of static models. By leveraging the social graph for user connections and the knowledge graph for item attributes, our model enhances the recommendation quality with rich contextual information. The integration of a time decay mechanism allows for the differentiation of interactions over time, ensuring the model remains sensitive to the evolving nature of user preferences. We conduct comprehensive experiments on sampled Yelp dataset, demonstrating the superiority of ISKG-TD over traditional and state-of-the-art models in terms of recommendation accuracy and relevance. The results highlight the effectiveness of our multi-graph approach and the potential of time-sensitive modeling in recommender systems. The code is available at https://github.com/codelover123hxy/ISKG_TD.

Keywords: Recommender Systems, Graph Neural Networks, Social Enhancement, Knowledge Graphs, Time Decay Mechanism.

1 Introduction

A recommender system (RS) augments the social recommendation process by aggregating and directing user recommendations [1]. These systems utilize implicit evaluations like reading durations and bookmark lists to filter content and highlight notable items, considering not only technical innovations but also incentive mechanisms, and personal privacy implications.

Traditional recommender systems methods primarily rely on users' preferences and behavioral patterns to predict and suggest items or content of interest, including content-based recommendation [2], collaborative filtering recommendation [3], and knowledge-based recommendation [4].

*These authors contribute to this work equally, and should be regarded as co-first authors

With the development of deep learning technology, Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and attention mechanisms are considered capable of effectively processing and learning complex features within large-scale unstructured data [5].

The application of Graph Neural Networks (GNNs) mainly focuses on mining complex interactions between users and items, as well as potential connections among items [6]. By constructing user-item interaction data into a graph structure, GNNs can effectively capture high-order connectivity information between users and items, thereby learning more accurate representations of users and items. This method leverages the strong representational learning capability of GNNs on graph data to understand users' interests and preferences at a deeper level, thereby enhancing the accuracy and personalization of recommendation systems.

In the field of recommender systems, the concept of time decay has been utilized to enhance Deep Neural Network (DNN) models [7] for more accurate and personalized recommendations. Although time decay has proven effective in DNN, it has not yet been applied in Graph Neural Networks (GNN). Given the unique advantages of GNN in handling complex relationships between users and items along with their attribute features, integrating time decay into GNN could further enhance the performance of recommender systems, especially in dealing with dynamically changing user preferences and real-time interaction data.

Social enhancement and knowledge graph augmentation use social connections [8] and domain knowledge [9] to improve RS accuracy and explainability. Although both approaches have achieved certain successes independently within recommender systems, their combined use is relatively less explored in existing research.

Based on the foregoing discussion, we propose an innovative approach that integrates social graphs, knowledge graphs, and time decay mechanisms into Graph Neural Networks (GNNs). The aim is to transition from traditional static modeling to dynamic temporal modeling of user-item interactions in recommender systems. Our main contributions can be summarized as follows:

1. **Time Decay for Dynamic Preferences:** Our model incorporates a time decay mechanism to account for the evolving nature of user preferences, enabling a more accurate representation of temporal dynamics in user-item interactions.

2. **Empirical Evidence:** We validate our model on diverse datasets, demonstrating its superior performance in capturing dynamic user preferences and outperforming existing recommendation systems.

The rest of this paper is organized as follows. Section II explains the problem formulation. Section III provides a detailed description of our proposed method. In Section IV, we present the experimental setup and results. Finally, we conclude the paper and suggest future research directions in Section V.

2 Problem Formulation

The target of our research is to propose a novel approach for personalized recommendation. More specifically, we have defined a set of features and a prediction model that incorporates the input of social graph and knowledge graph. To enhance the recommendation effect, we have adopted three graphs in the model, the user-item interaction graph, the social graph, and the knowledge graph. To unearth temporal information within user-item interaction graphs, we employ a time decay mechanism in our model.

Initially, we define a user set $U = u_1, u_2, \cdots, u_m$ and an item set $V = v_1, v_2, \cdots, v_n$. Each user $u_i \in U$ demonstrates a vector that contains attributes in user, such as age, preferences and other demographic details. Similarly, each item $v_j \in V$ is represented by attributes like city, address, and other pertinent features.

Subsequently, certain properties of items can be categorized as entities and denoted as edge information. For instance, a movie related to action can be represented as (Movie, BelongTo, Action). Consequently, we introduce an entity set $E = e_1, e_2, \cdots, e_l$.

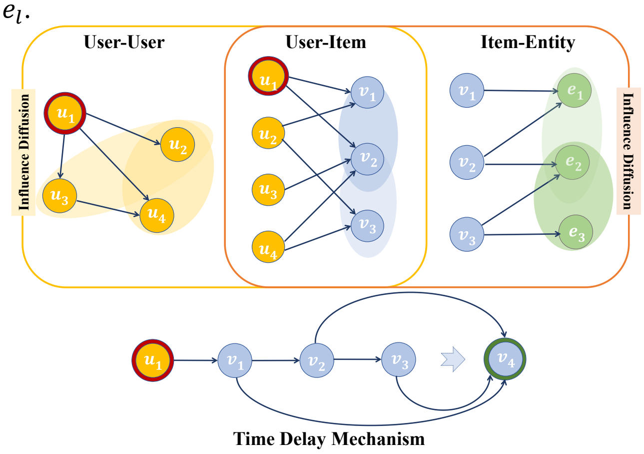

Fig. 1. Presentation of the temporal information in user-item interaction graph and item-entity interaction graph. The yellow nodes u_i mean users, blue nodes v_i represent items, and the green nodes e_i demonstrate entities.

As illustrated in Fig. 1, our approach harnesses time-sequential information and node interactions within these graphs. This strategy is designed to capture the nuances of users' long-term preferences and their temporal dynamics, providing a more nuanced and effective recommendation system. The three graphs and the time decay mechanism we mentioned above are elaborated as follows:

User-item Interaction Graph. To represent the collaborative signals between users and items, we have constructed a graph G_I , which is defined as $G_I = (U \cup V, E_I)$,where I is the user-item interaction matrix, delineating the user's rating preferences for items. For instance, the link $I_{iu} = score$ represents a specific score that user u rate items i.

Social Graph. In order to provide more accurate recommendations by leveraging the community-based interest alignment, we have introduced a social graph. The graph is represented as $G_S = (U, E_S)$ with E_S emerging from S, indicating the network of social connections within the user set U. if user u_i interacts with u_j, we assign $S_{ij} = 1$, otherwise $S_{ij} = 0$.

Knowledge Graph. The interaction between items and entities forms a knowledge graph. Specifically, if item v_j is related to entity e_f , we assign $K_{jf} = 1$, otherwise $K_{jf} = 0$. The graph is denoted as $G_K = (V \cup E, F_K)$, where F_K delineates the relationships between items and entities.

Time Decay Mechanism. Through the three graphs above, we can capture the deep information hide in the social relationship. However, as the user's interest evolves over time, we have denoted a time decay mechanism to differentiate the importance of user-item ratings across different time periods for recommendations. Formally, it has been defined as a function $TD(t)$.

Task Description. In this section, we delineate the task that we will undertake in this study.

Input. a set of graphs, encompassing the social graph G_S, the user-item interaction graph G_I and the knowledge graph G_K. The user-item interaction graph has time sequential information concerning rating.

Output. a predicting model designed to generate a ranked list of items that user may be interested in.

3 METHODOLOGY

In this section, we present the technical design of our proposed ISKG-TD model, accompanied by the overall model architecture depicted in. The ISKG-TD model employs a structured, multi-layered approach for generating embeddings and assessing relevance, with a significant focus on the Propagation Layers and Optimization Method.

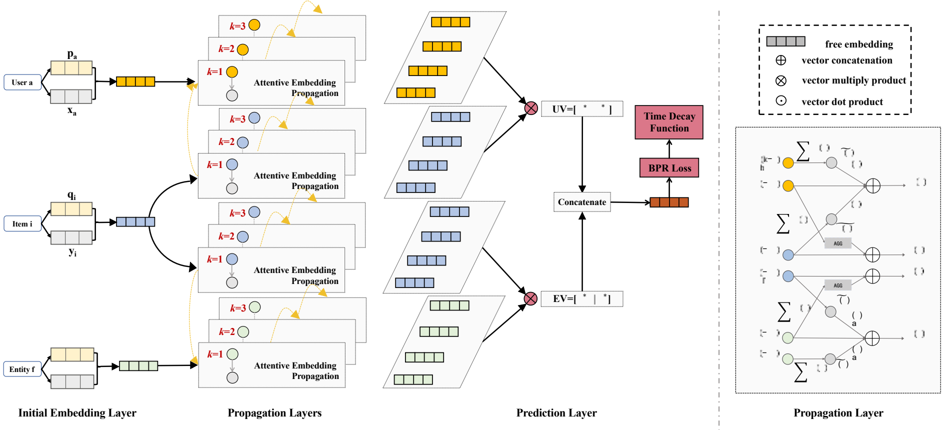

Fig. 2. Illustration of the proposed ISKG-TD model. The left subfigure shows model framework of ISKG-TD, and the right subfigure illustrates the embedding propagation layer of ISKG-TD.

The ISKG-TD model uses an initial embedding layer to map users, items, and related entities into a unified low-dimensional vector space. The fusion layer then enriches these vectors by combining user and item attributes, capturing complex interactions. However, the ISKG-TD's efficacy lies in its propagation layers that refine embeddings using relevant information, as depicted in Fig. 2, thus improving recommendation quality. Additionally, we introduce one crucial mechanism: a time decay mechanism in the loss function, to further enhance recommendation performance.

3.1 Initial Embedding Layer

In the initial embedding layer, user, item, and entity nodes are encoded into low-dimensional vectors $p_u^{(0)}$, $q_i^{(0)}$, and $o_f^{(0)}$ in R^d and encapsulated in embedding matrices P, Q and O. This forms a three-dimensional space in the ISKG-TD framework, serving as the basis for further updates through aggregation and propagation.

3.2 Fusion Layer

The Fusion Layer is responsible for collecting and aggregating neighbors' information, which plays a crucial role in amalgamating features from diverse sources to augment the representational capacity of the model.

For each user a, the fusion layer combines the user's embedding vector p_a with its corresponding feature vector x_a, resulting in u_a^0:

$$u_a^0 = g(W_a \cdot [p_a, x_a]) \tag{1}$$

where W_a represents a transformation matrix and $g(x)$ denotes a nonlinear transformation function.

Similarly, for item embedding and item-entity association, v_i^0 and e_f^0 are calculated like user above.

3.3 Propagation Layers

In this framework, embeddings for users (u_a^k), items (v_i^k), and entities (e_f^k) are refined over iterations. Starting from $k = 0$, embeddings evolve through propagation functions until reaching depth K.

For any given item i and its embedding at the kth layer v_i^k, we construct the next layer embedding v_i^{k+1} from the interest graph G_I as follows:

$$\widetilde{v_{ia}^{k+1}} = AGG_u(u_a^k, \forall a \in I_i) = \sum_{a \in I_i} \eta_{ia}^{k+1} u_a^k \tag{2}$$

$$v_{ia}^{k+1} = \widetilde{v_{ia}^{k+1}} + v_{ia}^k \tag{3}$$

Here, I_i denotes the set of users who have provided ratings for item i, and u_a^k is the embedding of user a at layer k. The aggregation weight is denoted by n_{ia}^{k+1}, and the updated item embedding v_i^{k+1} combines the aggregated neighbor embeddings with the item's previous layer embedding.

For incorporating knowledge from the knowledge graph, the formula to update embedding is similar, which we can get v_{if}^{k+1}.

We employ an attention mechanism to calculate the weights η_{ia} and η_{if} in Eqs. (2), using the following attention functions:

$$\eta_{ia}^{k+1} = MLP_{att_a}\left(\left[v_{ia}^{k}, u_{a}^{k}\right]\right) \tag{4}$$

The attention network adopts a MultiLayer Perceptron (MLP). Subsequent normalization of attention weights is performed as follows:

$$\eta_{ia}^{k+1} = \frac{exp\left(\eta_{ia}^{k+1}\right)}{\sum_{b\in I_i} exp(\eta_{ib}^{k+1})} \tag{5}$$

For each user a, their representation at layer k is denoted as u_a^k. In the context of both the social structure G_S and the affinity graph G_I, the embedding of a user at layer $k+1$, u_a^{k+1}, merges influence from these networks: influence spread in G_S and interest propagation in G_I. The term $\widetilde{p_a^{k+1}}$ represents the combined social influence embedding, and $\widetilde{q_a^{k+1}}$ summarizes the item-based interest embedding at the next layer. The evolution of each user's embedding u_a^{k+1} is thus articulated as:

$$u_a^{k+1} = u_a^k + \left(\gamma_{a1}^{k+1}\widetilde{p_a^{k+1}} + \gamma_{a2}^{k+1}\widetilde{q_a^{k+1}}\right) \tag{6}$$

with $\widetilde{p_a^{k+1}}$ computed through

$$\widetilde{p_a^{k+1}} = \sum_{b\in S_a} \alpha_{ab}^{k+1} u_b^k \tag{7}$$

and $\widetilde{q_a^{k+1}}$ is calculated similarly.

where α_{ab}^{k+1} and β_{ai}^{k+1} denote the scores reflecting social and interest influence. The former is calculated as:

$$\alpha_{ab}^{k+1} = MLP_{social}\left(\left[u_a^k, u_b^k\right]\right) \tag{8}$$

Post-calculation of node-specific attentive weights, these are inputted into the graph attention framework, permitting us to formulate the graph attention weights γ_{al}^{k+1} $(l = 1, 2)$ as:

$$\gamma_{a1}^{k+1} = MLP_{att}\left(\left[u_a^k, \widetilde{p_a^k}\right]\right) \tag{9}$$

Similarly, in the enhancement phase involving the knowledge graph, we got e_f^{k+1}.

3.4 Concatenation Layers

In the K-level propagation process, embeddings for users (u_a^k), items (v_i^k), and entities (e_f^k) are collected at each level up to $k = K$. The comprehensive embedding for a user a is $u_a^* = [u_a^0| \ldots |u_a^K]$, integrating representations across layers. Similarly, for items within a social graph, the final embedding is $v_{ia}^* = [v_{ia}^0| \ldots |v_{ia}^K]$, and for entities, $\mathrm{e}_\mathrm{f}^* = \left[e_f^0| \ldots |e_f^K\right]$. For items in a knowledge graph, the comprehensive embedding is $\mathrm{v}_\mathrm{if}^* = \left[v_{if}^0| \ldots |v_{if}^K\right]$. This approach synthesizes features enhanced by both social and knowledge graphs, streamlining the representation of users, items, and entities across multiple layers.

The Concatenation Layers thus enable the synthesis of feature sets that reflect enhancements from the social (Eq.(14)) and knowledge graphs (Eq.(15)) as follows:

$$UV = [u_a^* | v_{ia}^*] \tag{10}$$

$$EV = [e_f^* | v_{if}^*] \tag{11}$$

We used vector splicing to get the final the feature vector (Eq.(16)) as follows:

$$v_{ai}^* = UV + EV \tag{12}$$

3.5 Prediction Layers

Finally, we conduct inner product of the ultimate user and item embeddings, so as to predict their matching score:

$$\widehat{I_{ai}} = v_{ai}^{*T} v_{ai}^* \tag{13}$$

3.6 Optimization Method

We adopt the Bayesian Personalized Ranking (BPR) loss [30] for optimizing all trainable parameters, which includes the embeddings and convolution weights. Loss function is defined as:

$$L_{origin} = \min_{\Theta} \sum_{(a,i,j)\in I} - ln\left(\sigma\left(\widehat{I_{ai}} - \widehat{I_{aj}}\right)\right) + \rho\left|\left|\Theta\right|\right|^2 \tag{14}$$

where $I = \{ (a,i,j) \mid (a,i) \in I^+, (a,j) \in I^- \}$ denotes the training set, I^+ signifies the collection of positive instances (known user-item interactions), while I^- represents the negative instances (user-item pairs not observed and hence sampled from the set I).

User's interest changes with time. Therefore, to give higher weights to the most recent ratings, a time decay function is incorporated into the ratings. These time functions determine the appropriate time weights and provide high weights for recently rated items. In Table 1, a list of time decay functions is given.

Table 1. List of Time-Decay Functions.

Function	Mathematical Expression	References
Exponential	$e^{-\mu\lvert T_{u_a},k\rvert}$	[10]
Concave	$\alpha^{\lvert T_{u_a},k\rvert}$	[11]
Convex	$1 - \beta^{t-\lvert T_{u_a,k}\rvert}$	[11]
Linear	$1 - \frac{\lvert T_{u_a,k}\rvert}{t}$	[11]
Logistic	$\frac{1}{1 + e^{-\lambda * \lvert T_{u_a,k}\rvert}}$	[12], [13]
Power	$\lvert T_{u_a,k}\rvert^{-\omega}$	[13]

Here, $\mu, \alpha, \beta, \omega, \gamma$, and λ are the tuning parameters, while t is the difference between the recent timestamp and the timestamp at which the item is rated. Further details on parameter configurations will be elaborated in the experimental discussions.

The time-weighted ratings can be retrieved as follows:

$$\widehat{T_{a\iota}} = \widehat{I_{a\iota}} \cdot TD(t) \tag{15}$$

We then obtained a loss utilizing MSE loss with respect to the time decay as follows:

$$L_{time} = \frac{1}{N}\sum_{i=1}^{N}\left(\widehat{T_{a\iota}} - \widehat{I_a}\right)^2 \tag{16}$$

Here, $\widehat{I_a}$ denotes the labels of the training sets.

In conclusion, our final loss is defined as:

$$L = L_{origin} + L_{time} \tag{17}$$

4 Experiments

To verify the ISKG-TD framework's robustness, we perform comprehensive experiments on a dataset (Yelp), which has been pretreated and uploaded to https://huggingface.co/datasets/codeloverhxy/yelp. These experiments are aimed at responding to the following crucial queries:

- **RQ1**: How does the ISKG-TD model's performance compare with that of the leading-edge recommender systems?
- **RQ2**: How do ISKG-TD's core components impact the model's performance?
- **RQ3**: How does altering the time decay functions impact ISKG-TD's efficacy?

4.1 Experimental Settings

Datasets. To evaluate the effectiveness of the ISKG-TD framework, we utilize one benchmark dataset and generate our dataset by sampling randomly from it. Detailed statistics for a dataset are presented in Table2.

Table 2. statistics of the datasets.

	#Users	#Items	#Entities	#Time	#User-Item Interaction Graph	#Social Graph	#Knowledge Graph
Yelp	444	6841	822	9029	9029	1520	34558

Utilizing the Yelp dataset, we construct a multi-dimensional graph structure that represents the complicated interactions among users, businesses (as items), and their associated attributes (forming entities).

In our **knowledge graph**, businesses are represented as items, with their location and category attributes extracted to form entities. This extraction process elucidates the multifaceted nature of each business and its potential categorization within the graph.

Our **social graph** is derived from the ‘friends‘ attribute within the user data, establishing social connections that reflect the network of interactions and influences among users.

The **user-item interaction graph** is crafted from user reviews, specifically utilizing the star ratings given to businesses. These ratings not only represent user preferences but also serve as the predictive target for our recommendation model’s performance evaluation.

Each user, item, and entity is depicted as a node within the corresponding graphs, where edges signify relationships like friendships, reviews, or attribute associations. The rating score, pivotal to our analysis, is the numerical value that users assign to items, reflecting their satisfaction or experience.

To further refining the dataset, we filter out users and items with fewer than two ratings or links, ensuring data quality and engagement. We then split each user’s interactions into 80% for training and 20% for testing, providing a thorough view of user activity. Additionally, within the training set, a random 10% of interactions are used as a validation set for hyperparameter tuning and model optimization before final evaluation on the test set.

We implement our ISKG-TD model in PyTorch. In RQ3, ISKG-TD introduces a manual tuning function ($TD(t)$), pivotal for model adjustment.

Baselines. To demonstrate the effectiveness, we compare our proposed ISKG-TD with various advanced baseline methods. The baselines contain classical methods like **BPR** [14], **CKE** [15], and knowledge graph enhanced method **KGAT** [9].

Evaluation Metrics. In our study focused on Top-N recommendations, we measure the model’s performance using two key metrics: Recall@N, and NDCG@N. By default, we set N to 20. Recall@N quantify the correlation, and NDCG@N assesses the ranking accuracy of prevalent items. Enhanced metrics imply superior performance. We conduct each experiment quintuply to ensure equity in evaluation, subsequently averaging the rankings across all items.

4.2 Recommendation Performance Comparison (RQ1)

In this study, we focus on evaluating the performance of Top-K recommendation systems and present a comparative analysis of our model against other baseline models in Table 3.

Table 3. Performance Comparison on Yelp Datasets.

	BPR	CKE	KGAT	ISKG-TD	%Improv.
#Users	0.1423	0.1437	0.1442	0.1511*	4.79%
#Items	0.1458	0.1405	0.1447	0.1523*	3.89%

The ISKG-TD model demonstrates remarkable performance on the Yelp dataset, surpassing baseline metrics by showing notable improvements in Recall (4.79%) and NDCG (3.89%). It effectively captures high-order collaborative signals and social interactions, enhanced by integrating social relationships and knowledge graphs with an attention mechanism and contrastive learning.

Contrasted with matrix factorization methods like BPR and NFM, the ISKG-TD model seamlessly integrates insights from social relationships and knowledge graphs, further augmented by an attention mechanism and contrastive learning, boosting its representational learning capacity and underscoring the pivotal role of GNNs within the recommendation system landscape.

Moreover, the ISKG-TD model's multi-tier architecture, spanning from the Initial Embedding Layer to the Fusion and Propagation Layers, furnishes the model with the depth and versatility required to capture and interpret the complex web of interactions inherent in the dataset. This architectural nuance ensures the model's adeptness in navigating the intricate user-item relationship landscape, thereby setting a new benchmark in recommendation system performance.

4.3 Ablation Study of ISKG Framework (RQ2)

We dissect the ISKG-TD model to ascertain the individual contributions of its core components towards its overall performance on the Yelp dataset. This ablation study is crucial for understanding the significance of each component within the model's architecture. The results of this analysis are presented in Table 4, offering a granular view of how each element contributes to the efficacy of the ISKG-TD model.

Table 4. Ablation of key components in ISKG-TD on Yelp.

Item	**Recall**	**NDCG**
ISKG-TD (full model)	0.1511	0.1523
Ablation of Loss Function		
w/o L_{origin}	0.1478	0.1489
w/o L_{time}	0.1485	0.1496
Ablation of Graph Embedding and Attention Mechanism		
w/o self-attention mechanism	0.1493	0.1504
w/o social graph enhancement	0.1480	0.1491
w/o knowledge graph enhancement	0.1482	0.1493

The ablation study, as articulated in Table 4, accurately deconstructs the ISKG-TD model to unveil the intrinsic value of each component. The outset with the full model establishes a baseline, boasting Recall and NDCG metrics of 0.1511 and 0.1523, respectively.

The "Ablation of Loss Function" segment reveals the critical nature of Lorigin and Ltime to the model's predictive prowess. Their exclusion leads to a tangible decline in performance, signifying their indispensable role in the model's optimization landscape.

Further exploration into the "Ablation of Graph Embedding and Attention Mechanism" underscores the self-attention mechanism's efficacy in feature discernment, and the pivotal role of social and knowledge graph enhancements in contextual enrichment. The decrement in performance metrics, upon their exclusion, attests to their fundamental contributions towards the model's nuanced understanding and representation of complex interactions.

In summary, this ablation study not only highlights the synergistic contribution of each component to the ISKG-TD model's performance but also paves the path for targeted enhancements by pinpointing the elemental contributors to its success.

4.4 Impact of time decay function TD(t) on ISKG-TD's Efficacy (RQ3)

We investigate the effect of different time decay functions on the performance of the ISKG-TD model. Time decay functions are crucial for capturing the temporal dynamics in user-item interactions. The functions tested include Exponential, Concave, Convex, Linear, Logistic, and Power, with their impacts on the Recall and NDCG metrics summarized in Table 5.

Table 5. Impact of Time-Decay Functions on ISKG-TD.

Function	Exponential	Concave	Convex	Linear	Logistic	Power
Recall	0.1472	0.1485	0.1498	0.1501	0.1504	**0.1527**
NDCG	0.1483	0.1496	0.1509	0.1512	0.1515	**0.1538**

The Power decay function outperforms others in the ISKG-TD model, with higher Recall and NDCG values of 0.1527 and 0.1538. It models the temporal relevance of interactions more effectively than the Exponential or Linear functions, which either overemphasize recent interactions or treat them uniformly. This balanced attenuation aligns better with the natural decay of user interest, highlighting the importance of choosing the appropriate time decay function to capture user-item interaction dynamics and improve model efficacy.

5 Conclusion & Future Work

In our work, we have successfully integrated social enhancement and knowledge graph augmentation with a time decay mechanism, proposing a novel recommender system framework. The essence of this framework lies in the refinement of the BPR loss function, employing various time decay functions to capture the dynamic modeling of temporal changes, thus reflecting users' preferences more authentically. Extensive experiments conducted on real datasets substantiate the rationality and efficacy of our model.

Future efforts will concentrate on incorporating hypergraph structures to enhance our recommender system framework. Transitioning from traditional graphs to hypergraph representations will enable more intricate interaction modeling, including a wider array of user-item-entity relationships. Moreover, considering the three-dimensional spatial relationship between the recommendation system and the user's physical environment. we will also attempt to integrate 3D semantic scenes [16] into the topological structure [17] of the recommendation system to further enhance the accuracy and experience of the system.

Acknowledgments. The work was supported by the Zhejiang Provincial Natural Science Foundation of China (No. LQ24F020038), and was supported by the Xinmiao Talent Program (No. 2023R403030).

References

1. Resnick, P., Varian, H.R.: Recommender systems. Communications of the ACM 40(3), 56–58 (1997)
2. Balabanović, M., Shoham, et al.: Fab: content-based, collaborative recommendation. Communications of the ACM 40(3), 66–72 (1997)
3. Herlocker, J.L., Konstan, et al.: Explaining collaborative filtering recommendations. In: Proceedings of the 2000 ACM conference on Computer supported cooperative work. pp. 241–250 (2000)
4. Burke, R.: Knowledge-based recommender systems. Encyclopedia of library and information systems 69(Supplement 32), 175–186 (2000)
5. Yao, J., Wu, T., Zhang, et al.: Improving depth gradientcontinuity in transformers: A comparative study on monocular depth estimation with cnn. arXiv preprint arXiv:2308.08333 (2023)
6. Wu, S., Sun, et al.: Graph neural networks in recommender systems: a survey. ACM Computing Surveys 55(5), 1–37 (2022)
7. Jain, G., Mahara, et al.: Td-dnn: A time decay based deep neural network for recommendation system. Applied Sciences 12(13), 6398 (2022)
8. Wu, L., Li, et al.: Diffnet++: A neural influence and interest diffusion network for social recommendation. IEEE Transactions on Knowledge and Data Engineering 34(10), 4753–4766 (2020)
9. Wang, X., He, et al.: Kgat: Knowledge graph attention network for recommendation. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 950–958 (2019)
10. Xu, G., Tang, et al.: A collaborative filtering recommendation algorithm based on user confidence and time context. Journal of Electrical and Computer Engineering 2019 (2019)
11. Ding, Y., Li, et al.: Time weight collaborative filtering. In: Proceedings of the 14th ACM international conference on Information and knowledge management. pp. 485–492 (2005)
12. Zhang, L., Zhang, et al.: Ur: A user-based collaborative filtering recommendation system based on trust mechanism and time weighting. In: 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS). pp. 69–76. IEEE (2019)
13. Larrain, S., Trattner, et al.: Good times bad times: A study on recency effects in collaborative filtering for social tagging. In: Proceedings of the 9th ACM Conference on Recommender Systems. pp. 269–272 (2015)
14. Rendle, S., Freudenthaler, et al.: Bpr: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618 (2012)
15. Zhang, F., Yuan, et al.: Collaborative knowledge base embedding for recommender systems. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 353–362 (2016)
16. Yao, J., Li, et al.: Ndc-scene: Boost monocular 3d semantic scene completion in normalized devicecoordinates space. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9421–9431. IEEE Computer Society (2023)
17. Yao, J., Pan, et al.: Building lane-level maps from aerial images. In: ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and SignalProcessing (ICASSP). pp. 3890–3894. IEEE (2024)