

Graph Collaborative Filtering Model Combining Time Factor and Attention Mechanism

XIANGLIN ZUO, College of Computer Science and Technology, Jilin University, China

XIN HE, School of Artificial Intelligence (SAI), Jilin University, China

TIANHAO JIA, Key Laboratory of Symbolic Computation and Knowledge Engineering, College of Computer Science and Technology, Jilin University, China

YING WANG*, Key Laboratory of Symbolic Computation and Knowledge Engineering, College of Computer Science and Technology, Jilin University, China

Recently, with the triumph of deep learning, attention mechanism, and graph convolutional networks in their respective fields, using new representation learning techniques or introducing auxiliary information to improve the representation ability of embedding has become the core content of the recommendation algorithm research. Generally, most existing GNN-based recommendation methods recursively propagate embedding information on the graph structure and capture collaborative signals by exploring the high-level connectivity between users and items. Despite the great success, those methods do not consider the influence of temporal context on user preferences embedding information propagation, nor do they distinguish the contribution of different neighbor node information to the target node. In order to address the two problems, we propose a graph collaborative filtering model TAGCF combining time factors and attention based on the existing method. The model uses the time factor to integrate temporal information into the process of embedding information propagation and uses the attention mechanism to distinguish the influence of embedding information from different neighbors. The effectiveness of TAGCF, time information, and attention mechanism are verified through comparative experiments with multiple baseline methods on the two recommendation system datasets, MovieLens and Amazon-books.

JAIR Associate Editor: Prof. Chang-Dong Wang

JAIR Reference Format:

Xianglin Zuo, Xin He, Tianhao Jia, and Ying Wang. 2025. Graph Collaborative Filtering Model Combining Time Factor and Attention Mechanism. *Journal of Artificial Intelligence Research* 84, Article 8 (September 2025), 15 pages. doi: [10.1613/jair.1.18696](https://doi.org/10.1613/jair.1.18696)

1 Introduction

With the rapid development of Internet technology and the explosive growth of information, people can quickly obtain a large amount of data. Still, the excessive amount of information attenuates the efficiency of information utilization, leading to serious information overload problems. Recommendation systems are powerful ways to solve information overload. Compared with traditional technical methods such as portal websites and search engines, recommendation systems have two notable characteristics: initiative and personalization. At present,

*Corresponding Author.

Authors' Contact Information: Xianglin Zuo, ORCID: [0000-0002-1653-749X](https://orcid.org/0000-0002-1653-749X), zuoxl17@mails.jlu.edu.cn, College of Computer Science and Technology, Jilin University, Changchun, Jilin, China; Xin He, ORCID: [0000-0003-0559-1949](https://orcid.org/0000-0003-0559-1949), hexin20@mails.jlu.edu.cn, School of Artificial Intelligence (SAI), Jilin University, Changchun, Jilin, China; Tianhao Jia, tianhaojiapaper@163.com, Key Laboratory of Symbolic Computation and Knowledge Engineering, College of Computer Science and Technology, Jilin University, Changchun, Jilin, China; Ying Wang, wangying2010@jlu.edu.cn, Key Laboratory of Symbolic Computation and Knowledge Engineering, College of Computer Science and Technology, Jilin University, Changchun, Jilin, China.



This work is licensed under a [Creative Commons Attribution International 4.0 License](https://creativecommons.org/licenses/by/4.0/).

© 2025 Copyright held by the owner/author(s).

doi: [10.1613/jair.1.18696](https://doi.org/10.1613/jair.1.18696)

recommendation systems have been widely used in many fields, including e-commerce (L. Liu 2022; Xu and Sang 2022), social networking (Zuo et al. 2021), information retrieval (Shen and Jiang 2022), etc.

The core of the recommendation system algorithm is to predict the user's interest in items based on historical interaction data such as purchases, ratings, or clicks. The most widely used technology in the recommendation system algorithm is collaborative filtering, which assumes that users with similar historical behaviors have identical preferences on similar items. Based on the above assumption, the standard paradigm of collaborative filtering is to represent users and items in a learnable parameter vector, update and optimize the representation vector through historical interaction information, and predict the user's preference score for items (Gu et al. 2021; Z. Liu, L. Yang, et al. 2022; Z. Liu, Ma, et al. 2021; J. Wu et al. 2021).

The Neural Graph Collaborative Filtering (NGCF) (X. Wang et al. 2019) points out that the previously used embedding generation model (Choi et al. 2021; Guo et al. 2020; He, Liao, et al. 2017; Tay et al. 2018; H. Wang et al. 2015) do not explicitly encode the key collaborative signals that are implicit in user and item interaction records, so they can not reveal the behavior similarity between users and the attribute characteristics similarity between items. More specifically, most existing methods only use descriptive features (such as user ID, item attribute description, etc.) information when constructing embedded representations of users (items) without considering the interaction between user and item, which means that user-item interactions are only used to define the objective function of model training. NGCF is a deep collaborative filtering model based on the user-item interaction bipartite graph to mine the collaborative signals in the user-item interaction information to obtain a better embedding representation. The high-order connectivity in the structure of NGCF captures the collaborative signal, uses the structure information of the graph to embed and propagate, iterates and updates the representation to obtain better user and item representations.

Although the depth graph collaborative filtering recommendation model can obtain a more expressive embedding vector than the previous methods, it does not distinguish the contribution of different neighboring nodes to the target node when aggregating the propagation embeddings of first-order neighbors, which will lead to inaccurate embedding representation. At the same time, most current GNN-based collaborative filtering recommendation algorithms (Bin and Sun 2021; J. Liu et al. 2021; Niu 2022; Z. Xia et al. 2021) do not consider the influence of temporal context on user preferences. For a user node on the interactive bipartite graph, the strength of the relationship between the user and the item is equally important with the time when the interaction occurs.

Therefore, to solve the two problems mentioned above, this paper proposes a graph collaborative filtering recommendation model TAGCF that integrates time factors and attention mechanism. To consider the impact of temporal context information on user preferences, TAGCF integrates temporal information into the embedded information propagation process through a time factor. The propagation information of neighboring nodes is multiplied by the corresponding time factor, which is obtained by calculating the time of the current interaction. Furthermore, this paper uses the attention mechanism to distinguish the different contributions of neighbor nodes and generates corresponding attention coefficients for different neighbor nodes. The significant contributions of this paper are summarized as follows:

- We propose **TAGCF**, a novel time-aware graph collaborative filtering model. It incorporates temporal information through a time factor and leverages an attention mechanism to adaptively weight neighbor contributions.
- We systematically investigate different aggregation strategies (GCN, GraphSage, KGAT) and fusion approaches (direct splicing and combination based on attention mechanism), providing insights into their effectiveness within TAGCF.
- Extensive experiments on two real-world datasets verify the effectiveness of TAGCF, and ablation studies highlight the roles of the time factor and attention mechanism.

2 Related Work

2.1 Recommendation System

The recommendation system generates a list of recommended items for users or predicts the user's rating of specific items, mainly by analyzing the historical behavior preferences of users with similar interests or extracting features from item description information. According to the above two recommendation ideas, Adomavicius and Tuzhilin in 2005 (Adomavicius and Tuzhilin 2005) divided the traditional recommendation algorithm into three types, namely the recommendation algorithm based on collaborative filtering, the recommendation algorithm based on content and hybrid recommendation algorithm. In which the recommendation algorithm based on collaborative filtering is the most used technology in the modern recommendation.

The algorithms based on collaborative filtering assume that users with common interests and preferences in the past will also maintain the same preferences in the future. These algorithms generate recommendations based on a large number of interactive relationships between users and items to mine the user's potential preferences, the item's potential features and the potential relationship between users and items. For example, Matrix Factorization (MF) (Koren et al. 2009) is a classic method that decomposes the high-dimensional original matrix into two low-dimensional matrices and uses the two new matrices to calculate the prediction score. There are many extensions on the basis of matrix factorization, such as Bayesian Personalized Ranking Matrix Factorization (BPRMF) (Rendle et al. 2012), which proposes a general optimization objective for personalized ranking Bayesian personalized ranking optimization. NeuMF (He, Liao, et al. 2017) uses a neural network as an interaction function to break through the defects of traditional inner products and proposes a recommendation model that combines traditional matrix factorization and multi-layer perceptron.

2.2 Graph Based Recommendation System

Another line of research (Huang et al. 2021; S. Wang et al. 2020; L. Xia et al. 2021) exploits the user-item interaction graph to infer user preference. Early efforts, such as ItemRank (Gori et al. 2007) and BiRank (He, Gao, et al. 2016), adopt the idea of label propagation to capture the collaborative filtering effect. These methods define the labels to score items for a user as her interacted items and propagate the labels on the graph. As the recommendation scores are obtained based on the similarity between the historical items and the target item, these methods essentially belong to neighbor-based methods. However, these methods are conceptually inferior to collaborative filtering methods since there is a lack of model parameters to optimize the objective function of recommendation. With the development of GNNs (Gong et al. 2023; Z. Zhao et al. 2023) in recommendation systems in recent years, there have been many GNN-based methods (Fan et al. 2019; He, Deng, et al. 2020; L. Wu et al. 2019) that have achieved excellent performance in recommendation systems.

The method HOP-Rec (J.-H. Yang et al. 2018) alleviates the problem by combining the GNN-based method and embedding-based method. It first performs random walks to enrich a user's interactions with multi-hop connected items. Then it trains MF with BPR objective based on the enriched user-item interaction data to build the recommendation model. The superior performance of HOP-Rec over MF proves that incorporating the connectivity information is beneficial to obtain better representations in capturing the collaborative filtering effect. However, we argue that HOP-Rec does not fully explore the high-order connectivity, which is only utilized to enrich the training data rather than directly contributing to the model's embedding function. Moreover, the performance of HOP-Rec depends heavily on the random walks, which require careful tuning efforts, such as setting a proper decay factor.

While models such as T-GCN (L. Zhao et al. 2019) and DGCF (Li et al. 2020) explicitly address temporal dynamics, they are either designed for sequential prediction tasks or emphasize dynamic preference disentanglement. In contrast, TAGCF integrates temporal factors directly into the message passing process of graph collaborative filtering, offering a lightweight yet effective way to capture time-aware neighbor interactions.

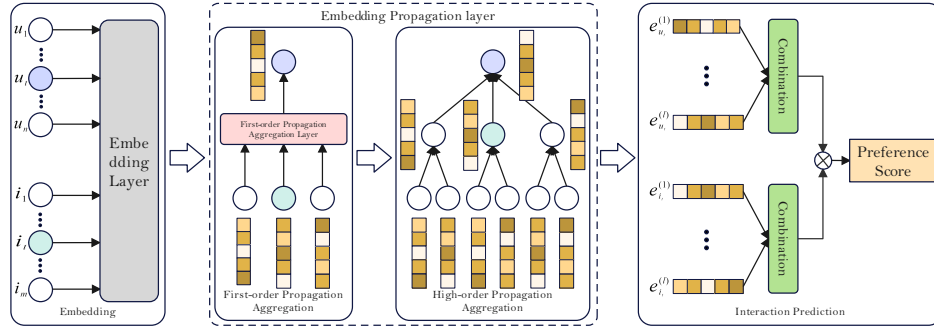


Fig. 1. The Architecture of TAGCF Model

3 TAGCF

The overall architecture of the TAGCF model is shown in Figure 1, and each part will be introduced separately below.

3.1 Embedding Layer

The primary role of the embedding layer is to generate the corresponding initial embedding of the user and item based on their IDs. The embedding layer in this paper uses the embedding vector $e_u \in \mathbb{R}^d$ ($e_i \in \mathbb{R}^d$) to represent a user u (item i), where d represents the dimension size of the embedding vector. E is a matrix containing the representations of all users and items:

$$E = [e_{u_1}, e_{u_2}, \dots, e_{u_n}, e_{i_1}, e_{i_2}, \dots, e_{i_m}] \quad (1)$$

where n is the number of users and m is the number of items, $[e_{u_1}, e_{u_2}, \dots, e_{u_n}]$ is the initial representation of all users and $[e_{i_1}, e_{i_2}, \dots, e_{i_m}]$ is the initial representation of all items. The model we propose in this paper then utilizes the user-item bipartite graph to propagate and aggregate the embedding information of neighboring nodes to update the embedding representation, and explicitly integrates the time context and high-order similarity into the embedding representation to make the embedding representation more accurate and effective.

3.2 Time Factor and Attention Embedding Propagation Layer

In this section, we build an information propagation layer that fuses the time factor and attention mechanism to obtain the representation of users and items.

Information Propagation. An edge on the user-item bipartite graph represents an interaction. For an interaction pair of user-item (u, i) , the information propagated from item embedding to user embedding is defined as:

$$I_{u \leftarrow i} = f(e_u, e_i, t_{u,i}) \quad (2)$$

where $I_{u \leftarrow i}$ represents the embedded information propagated from item i to user u , e_u and e_i are user embedding and item embedding respectively, and $t_{u,i}$ is the time when the interaction occurs. $f(\cdot)$ is the information propagation function. The input of $f(\cdot)$ is e_u , e_i and $t_{u,i}$. The main function of $f(\cdot)$ is to filter the embedded representation of neighboring nodes based on the time context and attention mechanism to construct the propagation to the target Embedded information of the node. The information propagation function $f(\cdot)$ in this

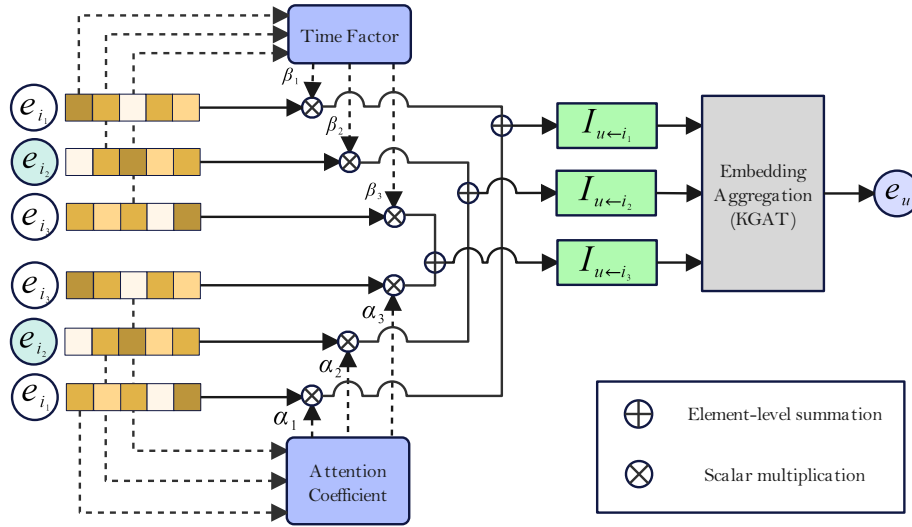


Fig. 2. The First-order Propagation Aggregation Process.

paper is defined as follows:

$$f(e_u, e_i, t_{u,i}) = \alpha_{u,i}(W_1 e_i) + \beta_{u,i}(W_2 e_i) \quad (3)$$

where $W_1 \in \mathbb{R}^{d' \times d}$, $W_2 \in \mathbb{R}^{d' \times d}$ are trainable $d' \times d$ dimensional parameter matrices, which are mainly used to extract e_i and e_u for information propagation. $\beta_{u,i}$ is a time factor based on the temporal context, which is calculated based on the interaction time between the user and its first-order neighboring items, $\alpha_{u,i}$ is an attention coefficient, which is calculated based on the similarity between the user's embedding and its first-order neighboring item embedding. This dual-weight structure allows the model to independently adjust different features of the input, supporting the dual-attention mechanism. W_1 and W_2 enable the model to focus on different aspects of the data, which helps capture more complex relationships in the attention mechanism.

Time-based Embedding Propagation Term. The time context information of the interaction between the user and the item plays an essential role in learning user representation. Therefore, the corresponding weighting factor should be generated based on the time information of the interaction to distinguish the difference in the time of different interactions while spreading the embedded information of items. The time factor designed in this paper is defined as follows:

$$T_{u,i} = \text{sigmoid}\left(\frac{t_{u,i} - t_{\min_u}}{t_{\max_u} - t_{\min_u}}\right) \quad (4)$$

$$\beta_{u,i} = \frac{e^{T_{u,i}}}{\sum_{i \in N_u} e^{T_{u,i}}} \quad (5)$$

where $t_{u,i}$ is the time of interaction between user u and item i , t_{\min_u} is the earliest interaction time between u and all its first-order neighbors, and t_{\max_u} is the latest interaction time between u and all its first-order neighbors. The time used here is the Unix timestamp. Since the time of different interactions of the same user may be very different, the *Sigmoid*(\cdot) function is used to scale the time data to (0.5, 1). $T_{u,i}$ is the corresponding time score, which can represent the relative time when the interaction between user u and item i occurred. Then all the time scores $T_{u,i}$ of the same user are normalized by the softmax function to obtain the corresponding time factor $\beta_{u,i}$.

Attention-based Embedding Propagation Term. The attention-based embedding propagation term mainly distinguishes the contribution of different nodes by calculating the attention coefficient between the user and its first-order neighboring nodes. The definition of the attention coefficient in this paper is as follows:

$$A_{u,i} = e_u^T * e_i \quad (6)$$

$$\alpha_{u,i} = \frac{e^{A_{u,i}}}{\sum_{i \in N_u} e^{A_{u,i}}} \quad (7)$$

where $A_{u,i}$ is the attention score corresponding to user u and item i , which is mainly obtained by calculating the similarity between the user and its first-order neighboring items. $\alpha_{u,i}$ represents the user's preference for different neighboring items. The larger the $\alpha_{u,i}$ is, the more the user's preference is related to the item, and the Attention-based Embedding Propagation Term will propagate more embedding information of the item to the user's embedding representation. In our design, we adopt the dot product to calculate item-to-user attention rather than an MLP-based scoring function. This choice is motivated by two considerations: (i) the dot product is computationally more efficient, making it well suited for large-scale recommendation scenarios; (ii) it naturally captures vector-space similarity, which aligns with the embedding structure of our model.

Information Aggregation. After obtaining the embedded information propagated from the first-order neighboring items to the user, the embedded information needs to be aggregated to enhance the representation ability of the user node u . To this end, we propose three different information aggregation functions, namely GCN-based aggregation, GraphSage-based aggregation, and KGAT-based aggregation. Each is defined as follows:

$$\mathcal{F}_{gcn} = \text{LeakyRelu}(W e_u + \sum_{i \in N_u} I_{u \leftarrow i}) \quad (8)$$

$$\mathcal{F}_{graphsage} = \text{LeakyRelu}(W e_u || \sum_{i \in N_u} I_{u \leftarrow i}) \quad (9)$$

$$\begin{aligned} \mathcal{F}_{kgat} = & \gamma_1 * \text{LeakyRelu}(W e_u + \sum_{i \in N_u} I_{u \leftarrow i}) + \\ & \gamma_2 * \text{LeakyRelu}(W e_u \odot \sum_{i \in N_u} I_{u \leftarrow i}) \end{aligned} \quad (10)$$

where N_u is the set of all first-order neighboring items of user u , and $\sum_{i \in N_u} I_{u \leftarrow i}$ is the sum of embedding information propagated by all neighboring items. $W \in \mathbb{R}^d$ is a trainable parameter matrix used to extract useful information in e_u . $\text{LeakyRelu}(\cdot)$ is the activation function, an improved version of the Relu function, which can handle input values less than 0 and can solve the problem of neuron "death". $||$ represents the concatenate operation. γ_1 and γ_2 are two coefficients used to control the influence of two kinds of aggregated information, and \odot is element-level multiplication.

Focus on the above three different aggregation methods, we design three corresponding model variants in the experimental part and conduct experiments, respectively. The experimental results prove that the \mathcal{F}_{kgat} aggregation method is the most effective one. \mathcal{F}_{kgat} can retain more original embedding information through $W e_u + \sum_{i \in N_u} I_{u \leftarrow i}$. And for similar users and items, $W e_u \odot \sum_{i \in N_u} I_{u \leftarrow i}$ can propagate more information. The two functions complement each other, and maximize the embedded representation ability.

Figure 2 is a detailed description of the first-order propagation aggregation process. In the traditional graph convolutional neural network, the attenuation coefficient of information propagation uses the Laplacian norm $\frac{1}{\sqrt{(|N_u| + |N_i|)}}$. In the depth map collaborative filtering model, the information propagation adds $e_i \odot e_u$, but does not consider the time factor and the attention mechanism. The embedded information propagation aggregation layer

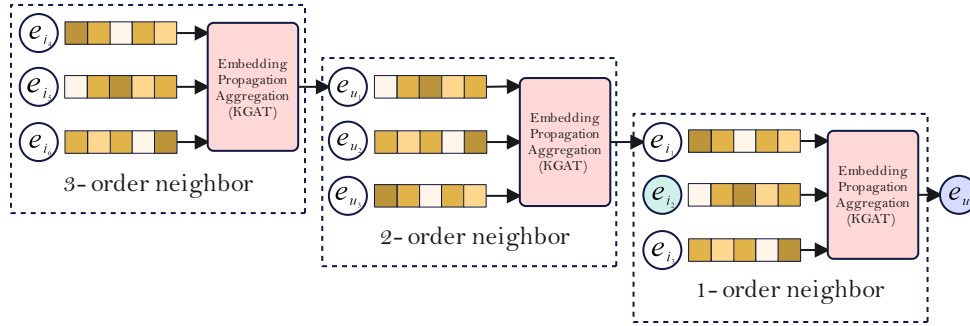


Fig. 3. The High-order Propagation Aggregation Process.

in this paper adds the time factor and the attention mechanism to the computation of embedded representation and achieves better results.

High-order Propagation Aggregation. After obtaining the first-order embedding through the first-order propagation aggregation layer, it is necessary to continue the high-order propagation aggregation of the embedding to enhance the representations of the nodes. Figure 3 shows the specific process of third-order propagation aggregation. Users (or items) can receive embedded information propagated from their first-order neighbors through first-order propagation aggregation. The process of L -order propagation aggregation is defined as follows:

$$e_u^{(l)} = \gamma_1 * \text{LeakyRelu}(W^{(l)} e_u^{(l-1)}) + \sum_{i \in N_u} I_{u \leftarrow i}^{(l)} + \gamma_2 * \text{LeakyRelu}(W^{(l)} e_u^{(l-1)}) \odot \sum_{i \in N_u} I_{u \leftarrow i}^{(l)} \quad (11)$$

The embedding information spread by neighboring nodes is defined as follows:

$$I_{u \leftarrow i}^{(l)} = \alpha_{u,i}^{(l)} (W_1^{(l)} e_i^{(l-1)}) + \beta_{u,i}^{(l)} (W_2^{(l)} e_i^{(l-1)}) \quad (12)$$

Where $e_u^{(l-1)} (e_i^{(l-1)})$ is the user(item) embedding generated by $l - 1$ aggregation layer, which includes the information of itself and $l - 1$ order neighboring nodes, $e_u^{(l)} (e_i^{(l)})$ is the user(item) embedding that includes the information of itself and l order neighboring nodes. $W^{(l)} \in \mathbb{R}^{d^{(l)} \times d^{(l-1)}}$, $W_1^{(l)} \in \mathbb{R}^{d^{(l)} \times d^{(l-1)}}$, $W_2^{(l)} \in \mathbb{R}^{d^{(l)} \times d^{(l-1)}}$ are the parameter matrix of l aggregation layer, which are used to extract useful information from the embedding. $\alpha_{u,i}^{(l)}$ is the attention coefficient of layer l , $\beta_{u,i}^{(l)}$ is the time factor of layer l , $I_{u \leftarrow i}^{(l)}$ is the embedded information propagated by the l -order neighboring nodes.

3.3 Interaction Prediction

In order to express the importance of the embedding representations in different layers, we use the self-attention mechanism to assign corresponding weights to different embedding representations of different layers in this paper. The formula to obtain the final user representation e_u^* is defined as follows:

$$A_u^{(j)} = (W_u * e_u^{(j)} + b_u), j \in (1, 2, \dots, l) \quad (13)$$

$$\theta_j = \frac{e^{A_u^{(j)}}}{\sum_{k=1}^l e^{A_u^{(k)}}}, j \in (1, 2, \dots, l) \quad (14)$$

Algorithm 1 TAGCF with KGAT Aggregation**Require:** User-item graph $G(U, I, E)$; embedding dimension d ; number of layers L ; timestamps $\{t_{u,i}\}$ **Ensure:** Final user embeddings $\{e_u^*\}$, item embeddings $\{e_i^*\}$

- 1: Initialize embeddings $e_u, e_i \in \mathbb{R}^d$ for all $u \in U, i \in I$
- 2: **for** $l = 1$ to L **do**
- 3: **for** each interaction $(u, i) \in E$ **do**
- 4: $T_{u,i} \leftarrow \sigma \left(\frac{t_{u,i} - t_{\min}^u}{t_{\max}^u - t_{\min}^u} \right)$
- 5: $\beta_{u,i} \leftarrow \text{softmax}(T_{u,i})$
- 6: $A_{u,i} \leftarrow e_u^{(l-1)} \cdot e_i^{(l-1)}$
- 7: $\alpha_{u,i} \leftarrow \text{softmax}(A_{u,i})$
- 8: $I_{u \leftarrow i}^{(l)} \leftarrow \alpha_{u,i}(W_1^{(l)} e_i^{(l-1)}) + \beta_{u,i}(W_2^{(l)} e_i^{(l-1)})$
- 9: **end for**
- 10: **for** each user $u \in U$ **do**
- 11: $e_u^{(l)} \leftarrow \gamma_1 \cdot \text{LeakyReLU}(W e_u^{(l-1)} + \sum I_{u \leftarrow i}^{(l)})$
- 12: $+ \gamma_2 \cdot \text{LeakyReLU}(W e_u^{(l-1)} \odot \sum I_{u \leftarrow i}^{(l)})$
- 13: **end for**
- 14: (Symmetric update for items $e_i^{(l)}$)
- 15: **end for**
- 16: **for** each user u **do**
- 17: **for** $j = 1$ to L **do**
- 18: $A_u^{(j)} \leftarrow W_f e_u^{(j)} + b$
- 19: $\theta_j \leftarrow \text{softmax}(A_u^{(j)})$
- 20: **end for**
- 21: $e_u^* \leftarrow \theta_1 e_u^{(1)} \parallel \dots \parallel \theta_L e_u^{(L)}$
- 22: **end for**
- 23: (Similarly compute e_i^*)
- 24: Prediction: $\hat{y}_{u,i} \leftarrow (e_u^*)^\top e_i^*$
- 25: Loss: $L \leftarrow \sum_{(u,i,j) \in \Omega} -\ln \sigma(\hat{y}_{u,i} - \hat{y}_{u,j}) + \lambda \|\Phi\|_2^2$

$$e_u^* = (\theta_1 * e_u^{(1)} \parallel \theta_2 * e_u^{(2)} \parallel \dots \parallel \theta_L * e_u^{(L)}) \quad (15)$$

where $W_u \in \mathbb{R}^{(n_u \times d)}$ and $b_u \in \mathbb{R}^{(n_u \times 1)}$ are the parameter matrix and bias term of the force network. $A_u^{(j)}$ is the self-attention of user's embedding representation in j -th layer. θ_j is the normalized attention coefficient corresponding to the embedding representation of the user in j -th layer. Through the combination based on the attention mechanism, the embedding propagation layer not only can be used to enhance the quality of the final embedding representations, but also can adjust l to control the scope of propagation. A similar definition is provided for the final embedding representation e_i^* of an item.

The interaction function used in the TAGCF model is the inner product, which is defined as follows:

$$\hat{y}_{u,i} = e_u^{*T} * e_i^* \quad (16)$$

The advantage of using the inner product as the interaction function is that it does not require additional trainable parameters, reduces the computational cost of the model, and can make the model pay more attention to the learning of the embedded propagation layer parameters.

At last, in order to train the parameters of the recommendation model, we choose the pairwise BPR loss as the optimization objective, which is defined as follows:

$$L = \sum_{(u,i,j) \in \Omega} -\ln \delta(\widehat{y}_{u,i} - \widehat{y}_{u,j}) + \lambda \|\Phi\|_2^2 \quad (17)$$

where $\delta(\cdot)$ is the sigmoid function, and Φ is all the trainable parameters in the TAGCF model. λ is the regularization coefficient to control the strength of the L_2 norm regularization term. $\Omega = (u, i, j) | (u, i) \in \Omega^+, (u, j) \in \Omega^-$ is a paired training dataset, $(u, i) \in \Omega^+$ represents the interactions between users and items are observed (positive), $(u, j) \in \Omega^-$ represents the interactions between users and items are not observed (negative). The optimization objective of the model is to make $(\widehat{y}_{u,i} - \widehat{y}_{u,j})$ as large as possible, then the value of $-\ln \delta(\widehat{y}_{u,i} - \widehat{y}_{u,j})$ becomes smaller so that the overall loss of the recommendation algorithm will become lower. The detailed implementation of TAGCF is shown in Algorithm 1.

3.4 Computational Complexity Analysis

Compared with NGCF, TAGCF introduces only minor additional complexity. Both models follow a message-passing paradigm with overall time complexity of $\mathcal{O}(L|E|d)$, where L is the number of layers, $|E|$ is the number of user-item interactions, and d is the embedding dimension. The extra operations in TAGCF mainly stem from the time factor and dual-weight attention, which require lightweight projection matrices and coefficient calculations. These incur only a small constant overhead, leaving the asymptotic complexity unchanged. In practice, TAGCF adds less than 10% additional training time per epoch compared to NGCF under the same experimental settings, while delivering notable performance gains.

4 Experiment

In this section, we mainly perform experiments related to the TAGCF algorithm. We will first introduce the dataset and baseline methods. Then, we will analyze the results of a series of model comparison experiments, which verifies the superiority of our proposed method and the role of temporal context information and attention mechanism.

4.1 Datasets

Table 1. Statistics of MovieLens-1M and Amazon-books dataset.

Dataset	User	item	rating	Density
MovieLens-1M	6,040	3,952	1,000,209	0.0419
Amazon-books	8,373	19,994	456,564	0.0027

In order to evaluate the effectiveness of the TAGCF model, two benchmark datasets are used in this experiment: MovieLens-1M and Amazon-books (2018). The statistical summary of these two datasets is shown in Table 1.

MovieLens-1M dataset: Contains 1000209 rating records of 3952 movies by 6040 users, where the user ID ranges from 1 to 6040, the movie ID ranges from 1 to 3952, and the rating uses a five-star rating (1-5). The timestamp is expressed in Unix timestamps, and each user has at least 20 score records, which ensures that the dataset is not too sparse.

Amazon-books (2018) dataset: As the Amazon-books (2018) dataset is very large and extremely sparse, in order to increase the density of the data, at first, the original data was filtered. The processed Amazon-books (2018) contains 456564 rating records of 8373 users on 19994 items, among which the user ID ranges from 1 to 8373, the item ID ranges from 1 to 19994, and the rating is a five-star rating (1-5). The timestamp is expressed in Unix timestamps, and each user has at least 15 score records.

4.2 Baselines

In order to prove the effectiveness of the TAGCF model proposed in this paper, seven baseline methods are selected to compare with TAGCF, namely MF, NeuMF, CMN, HOP-Rec, PinSage, GC-MC, and NGCF. The specific introduction of each baseline method is as follows:

MF (Koren et al. 2009): Matrix factorization is a model-based collaborative filtering method, which decomposes the sparse original rating matrix into two low-dimensional matrices. The matrix factorization is applied to the Top-N recommendation task as one of the comparison methods of TAGCF by replacing Bayesian personalized ranking loss with the original loss function.

NeuMF (He, Liao, et al. 2017): NeuMF is a typical recommendation algorithm based on deep learning, which uses a neural network as an interaction function to break through the defects of traditional inner products and proposes a recommendation model that combines traditional matrix factorization and multi-layer perceptrons. In this paper, we replace the loss function in NeuMF with Bayesian Personalized Ranking loss as a comparison method of our model.

CMN(Ebesu et al. 2018): CMN is a memory-based hybrid model that combines the global structure of the hidden factor model and the local structure based on the neighborhood and uses the memory network and attention mechanism to encode the complexity of users and items.

HOP-Rec (J.-H. Yang et al. 2018): HOP-Rec is currently a relatively advanced GNN-based recommendation model. It uses random walks on the two interactive graphs to mine high-level relationships and obtains high-level information from each user's neighborhood items.

PinSage (Ying et al. 2018) : PinSage combines random walk and GCN to generate the embedding vector of the node while taking into account the graph structure and the characteristic information of the node and then adopts a novel training strategy to improve the robustness of the model and speed up the model convergence.

GC-MC (Berg et al. 2017): GC-MC is a graph autoencoder framework for matrix completion tasks in recommendation systems, and it is also a GNN-based recommendation method. It uses a GCN encoder that includes a graph convolutional layer to generate representations of users and items by spreading information on the bipartite graph of users and items, and then uses a bilinear decoder to predict the score.

NGCF (X. Wang et al. 2019): NGCF propagates embedded representations and effectively captures the high-order connectivity relationships by utilizing the structure of the interactive graph of users and items. At the same time, it combines the collaborative signal with the display method. NGCF is the predecessor of the model proposed in this paper. It has made certain improvements, but it does not consider time context information and does not use the attention mechanism to distinguish the importance of neighbors.

4.3 Baseline Comparison Experiment

In this section, we compared the TAGCF model with the baseline methods. The results of the comparative experiment are shown in Table 2. To provide a clearer understanding, we next analyze the performance of each baseline in detail:

- NeuMF uses a neural network as an interactive model, which has a nonlinear fitting ability, and overcomes the shortcomings of the previous inner product operation. Compared with the performance of traditional matrix factorization method MF, NeuMF achieves a handful of improvements.
- CMN is a hybrid model based on deep learning. Compared with NeuMF, an attention mechanism is included. The attention mechanism can effectively distinguish the importance of different neighboring users, so the experimental effect is also better than NeuMF.
- HOP-Rec enriches user-item interaction data and expands training data by performing random walks on the two interactive graphs to mine high-level relationships. Since the effect of the recommendation model is usually very dependent on the sparseness of the data, expanding the training data can significantly

improve the effectiveness of the recommendation, and the performance of HOP-Rec is greater than the three models mentioned above.

- The experimental effects of PinSage and GC-MC models are similar because the two models have many similarities in generating representations of users and items. For example, both models disseminate information on the interactive bipartite graph of user-item, and both use GCN to fuse information from neighbor nodes to obtain richer vector representations of users and items.
- NGCF is the best overall effect among the GNN-based recommendation models, and it is also the best among all the baseline methods because NGCF combines embedding representations of different propagation distances through splicing. The embedding representations of various propagation distances encode different information, which can retain more information than PinSage using only the output of the last layer.
- In general, the model TAGCF proposed in this paper has achieved the best recommendation effect on the four evaluation indicators of the two datasets. Specifically, compared with the optimal baseline method NGCF, on the MovieLens-1M dataset, TAGCF has an increase of 13.58% in Precision and 16.28% in NDCG; On the Amazon-Books dataset, TAGCF has an increase of 3.20% in Precision and 3.64% in NDCG. The reasons for the improvement of the TAGCF model can be attributed to the fact that TAGCF considers the impact of time context information on user interests, and use the attention mechanism to measure the impact of different neighboring nodes on the target user.

Table 2. Experimental results of the baseline models.

Model	MovieLens-1M				Amazon-Books			
	Recall	Precision	Hit	NDCG	Recall	Precision	Hit	NDCG
MF	0.04206	0.05401	0.4912	0.1132	0.04768	0.01704	0.2371	0.07281
NeuMF	0.04310	0.05527	0.5005	0.1156	0.04836	0.01750	0.2379	0.07302
CMN	0.04395	0.05613	0.5167	0.1205	0.04892	0.01765	0.2389	0.07396
HOP-Rec	0.04879	0.06341	0.5497	0.1276	0.05017	0.01798	0.2483	0.07603
PinSage	0.04629	0.06296	0.5319	0.1271	0.04936	0.01774	0.2406	0.07427
GC-MC	0.04537	0.06273	0.5204	0.1269	0.04967	0.01783	0.2429	0.07538
NGCF	0.05405	0.06353	0.5784	0.1302	0.05230	0.01844	0.2577	0.07754
TAGCF	0.05689	0.07216	0.6045	0.1514	0.05342	0.01903	0.2624	0.08036
Improvement	5.25%	13.58%	4.51%	16.28%	2.14%	3.20%	1.81%	3.64%

4.4 Ablation Study

4.4.1 Effect of different aggregation functions. In the process of embedding propagation aggregation in section 3.2, three different information aggregation functions are proposed: GCN-based aggregation, GraphSage-based aggregation, and KGAT-based aggregation. The three models are tested on two datasets, and the result is shown in Figure 4.

By observing the experimental results, it can be found that the experimental effects of the TAGCF-kgat model are the best on the four evaluation indicators of the two datasets. This is because the $f_{kgat}(\cdot)$ aggregate function can not only retain more original embedded information but also spread more information from neighboring nodes to the target node. Therefore, the aggregation function based on KGAT is selected as the aggregation method for the embedding propagation layer.

4.4.2 Effect of different embedding combination methods. In section 3.3, the prediction model combines different levels of embeddings to generate the user (item) representation. We conduct comparative experiments on the direct splicing combination and the attention-based combination method, and verify the effectiveness of the attention-based combination. The corresponding models are constructed based on these two different combination

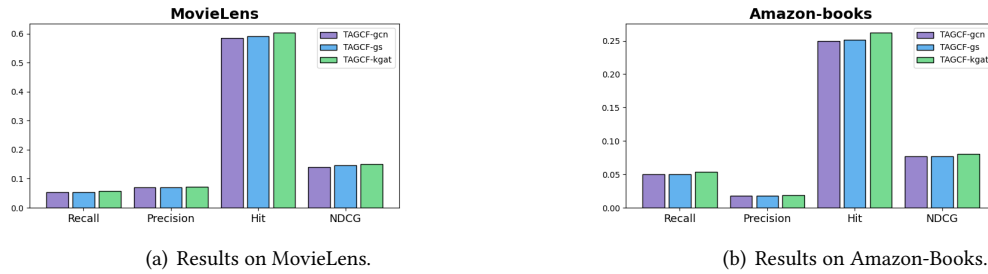


Fig. 4. Experimental results of three aggregation methods.

methods, which are respectively denoted as TAGCF-con and TAGCF-attention. The result of the experiments with these two models on two datasets is shown in Figure 5.

By observing the experimental results, it can be found that the experimental effect of TAGCF-attention is better than that of TAGCF-con, which also verifies that the use of the attention-based combination method can further improve the representation ability of the final embedding obtained, and then achieve a better recommendation effect. Therefore, the TAGCF model adopts an attention-based combination instead of the traditional direct splicing combination method.

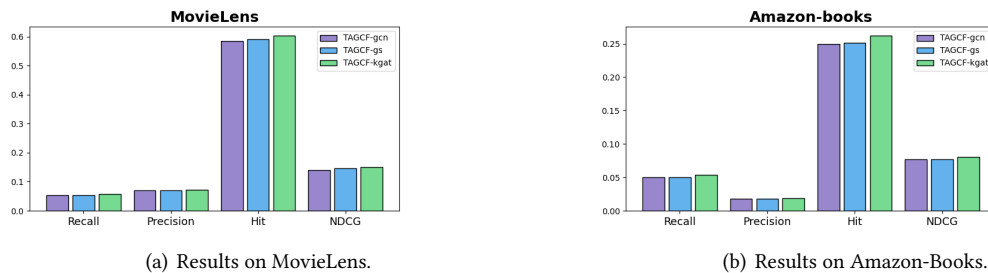


Fig. 5. Experimental results of two combination methods.

4.4.3 Effect of time factor and attention coefficient. In order to verify the effectiveness of the proposed model combining time factor and attention mechanism, this section sets up three different models for comparison experiments with NGCF, namely AGCF, TGCF, TAGCF. AGCF is a model that only uses the attention mechanism in the embedding propagation layer of section 3.2. TGCF is a model that uses only time context information in the embedding propagation layer of section 3.2. TAGCF is a model that uses attention mechanism and time context information simultaneously. NGCF uses neither the attention mechanism nor the time context information. The four models were tested on two datasets, and the experimental result is shown in Figure 6.

By observing the experimental results on the two datasets, it can be found that TGCF and AGCF have improved in the four evaluation indicators compared with NGCF. It is verified that the addition of time context information and attention mechanism can improve the recommendation effect of the model. The improvement of TGCF is more evident than AGCF, indicating that the time factor is more critical for generating embedding representations than the attention coefficient. The TAGCF model combines the advantages of two models (TGCF and AGCF) and achieves the best experimental results, which proves that the representation ability of the embedding vector

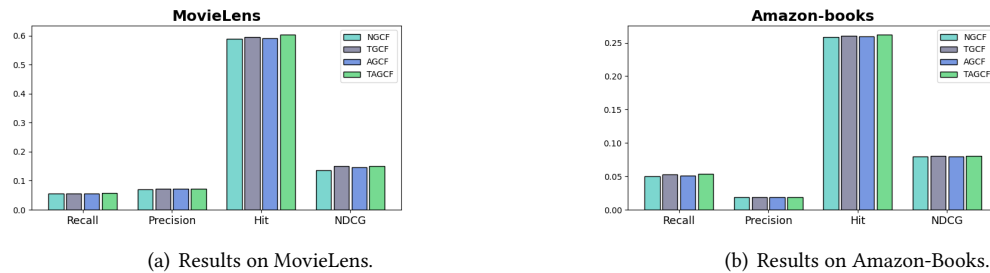


Fig. 6. Experimental results of four variant models.

obtained by the propagation layer which introduces time content information and attention mechanism is the best, and at the same time, verifies the value and significance of this paper.

5 Conclusion

One of the leading research directions of recommendation systems in recent years is to improve the representation learning ability of the model. In order to utilize the structural information and high-level relationships of graph structure data and consider temporal context information, this paper proposes a graph collaborative filtering model TAGCF based on NGCF, which integrates time factors and attention mechanism.

The proposed model uses the time factor to integrate temporal context information into the process of embedded information dissemination, which can capture the changes in user preference characteristics over time. The attention mechanism is used to distinguish the influence of different neighboring nodes on the target node by measuring the similarity between users and the items, and a higher propagation weight is assigned to the more similar neighboring node, and the embedded information of the corresponding item (user) contributes more to the user's (item's) representation. In addition, the use of the attention mechanism to combine the embedded representations obtained by different propagation layers can further improve the representation ability of the final embedding. Through the baseline method for comparative experiments, TAGCF achieved the best experimental results on the four evaluation indicators of the two datasets, which verified the effectiveness of the TAGCF model proposed in this paper.

6 Acknowledgments

This work was supported by a grant from the National Natural Science Foundation of China under grants (No.62272191, 62372211), and the International Science and Technology Cooperation Program of Jilin Province (No.20240402067GH)

References

- G. Adomavicius and A. Tuzhilin. 2005. "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions." *IEEE transactions on knowledge and data engineering*, 17, 6, 734–749.
- R. v. d. Berg, T. N. Kipf, and M. Welling. 2017. "Graph convolutional matrix completion." *arXiv preprint arXiv:1706.02263*, 1.
- S. Bin and G. Sun. 2021. "Matrix factorization recommendation algorithm based on multiple social relationships." *Mathematical Problems in Engineering*, 2021, 1–8.
- I. Y. Choi, Y. U. Ryu, and J. K. Kim. 2021. "A recommender system based on personal constraints for smart tourism city." *Asia Pacific Journal of Tourism Research*, 26, 4, 440–453.
- T. Ebesu, B. Shen, and Y. Fang. 2018. "Collaborative memory network for recommendation systems." In: *The 41st international ACM SIGIR conference on research & development in information retrieval*, 515–524.

- W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin. 2019. "Graph neural networks for social recommendation." In: *The world wide web conference*, 417–426.
- M. Gong, H. Zhou, A. K. Qin, W. Liu, and Z. Zhao. 2023. "Self-Paced Co-Training of Graph Neural Networks for Semi-Supervised Node Classification." *IEEE Transactions on Neural Networks and Learning Systems*, 34, 11, 9234–9247. doi:10.1109/TNNLS.2022.3157688.
- M. Gori, A. Pucci, V. Roma, and I. Siena. 2007. "Itemrank: A random-walk based scoring algorithm for recommender engines." In: *IJCAI*. Vol. 7, 2766–2771.
- T. Gu, H. Chen, C. Bin, L. Chang, and W. Chen. 2021. "Neighborhood attentional memory networks for recommendation systems." *Scientific Programming*, 2021, 1–10.
- Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, and Q. He. 2020. "A survey on knowledge graph-based recommender systems." *IEEE Transactions on Knowledge and Data Engineering*, 1.
- X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang. 2020. "Lightgcn: Simplifying and powering graph convolution network for recommendation." In: *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, 639–648.
- X. He, M. Gao, M.-Y. Kan, and D. Wang. 2016. "Birank: Towards ranking on bipartite graphs." *IEEE Transactions on Knowledge and Data Engineering*, 29, 1, 57–71.
- X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. 2017. "Neural collaborative filtering." In: *Proceedings of the 26th international conference on world wide web*, 173–182.
- C. Huang et al.. 2021. "Knowledge-aware coupled graph neural network for social recommendation." In: *Proceedings of the AAAI Conference on Artificial Intelligence* 5. Vol. 35, 4115–4122.
- Y. Koren, R. Bell, and C. Volinsky. 2009. "Matrix factorization techniques for recommender systems." *Computer*, 42, 8, 30–37.
- X. Li, M. Zhang, S. Wu, Z. Liu, L. Wang, and P. S. Yu. 2020. "Dynamic graph collaborative filtering." In: *2020 IEEE international conference on data mining (ICDM)*. IEEE, 322–331.
- J. Liu, W.-H. Choi, and J. Liu. 2021. "Personalized movie recommendation method based on deep learning." *Mathematical Problems in Engineering*, 2021, 1–12.
- L. Liu. 2022. "e-commerce personalized recommendation based on machine learning technology." *Mobile Information Systems*, 2022.
- Z. Liu, L. Yang, Z. Fan, H. Peng, and P. S. Yu. 2022. "Federated social recommendation with graph neural network." *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13, 4, 1–24.
- Z. Liu, Y. Ma, Y. Ouyang, and Z. Xiong. 2021. "Contrastive Learning for Recommender System." *arXiv preprint arXiv:2101.01317*, 1.
- Y. Niu. 2022. "Collaborative Filtering-Based Music Recommendation in Spark Architecture." *Mathematical Problems in Engineering*, 2022.
- S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. 2012. "BPR: Bayesian personalized ranking from implicit feedback." *arXiv preprint arXiv:1205.2618*, 1.
- D. Shen and Z. Jiang. 2022. "Online Teaching Course Recommendation Based on Autoencoder." *Mathematical Problems in Engineering*, 2022.
- Y. Tay, L. Anh Tuan, and S. C. Hui. 2018. "Latent relational metric learning via memory-based attention for collaborative ranking." In: *Proceedings of the 2018 World Wide Web Conference*, 729–739.
- H. Wang, N. Wang, and D.-Y. Yeung. 2015. "Collaborative deep learning for recommender systems." In: *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 1235–1244.
- S. Wang et al.. 2020. "Graph learning approaches to recommender systems: A review." *arXiv preprint arXiv:2004.11718*, 1.
- X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua. 2019. "Neural graph collaborative filtering." In: *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, 165–174.
- J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, and X. Xie. 2021. "Self-supervised graph learning for recommendation." In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 726–735.
- L. Wu, P. Sun, Y. Fu, R. Hong, X. Wang, and M. Wang. 2019. "A neural influence diffusion model for social recommendation." In: *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, 235–244.
- L. Xia, Y. Xu, C. Huang, P. Dai, and L. Bo. 2021. "Graph meta network for multi-behavior recommendation." In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 757–766.
- Z. Xia, W. Zhang, and Z. Weng. 2021. "Social recommendation system based on hypergraph attention network." *Computational Intelligence and Neuroscience*, 2021, 1–12.
- L. Xu and X. Sang. 2022. "E-Commerce Online Shopping Platform Recommendation Model Based on Integrated Personalized Recommendation." *Scientific Programming*, 2022.
- J.-H. Yang, C.-M. Chen, C.-J. Wang, and M.-F. Tsai. 2018. "HOP-rec: high-order proximity for implicit recommendation." In: *Proceedings of the 12th ACM Conference on Recommender Systems*, 140–144.
- R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec. 2018. "Graph convolutional neural networks for web-scale recommender systems." In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 974–983.
- L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li. 2019. "T-GCN: A temporal graph convolutional network for traffic prediction." *IEEE transactions on intelligent transportation systems*, 21, 9, 3848–3858.

- Z. Zhao, Z. Yang, C. Li, Q. Zeng, W. Guan, and M. Zhou. 2023. "Dual Feature Interaction-based Graph Convolutional Network." *IEEE Transactions on Knowledge and Data Engineering*, 35, 9, 9019–9030. doi:[10.1109/TKDE.2022.3220789](https://doi.org/10.1109/TKDE.2022.3220789).
- L. Zuo, S. Xiong, X. Qi, Z. Wen, and Y. Tang. 2021. "Communication-based book recommendation in computational social systems." *Complexity*, 2021, 1–10.

Received 24 March 2025; accepted 27 August 2025