Electronic Commerce Research (2026) 26:2211–2240
https://doi.org/10.1007/s10660-025-09978-y

Dynamic heterogeneous graph convolutional networks for click-through rate prediction in recommender systems

Ying Jin[1] · **Yanwu Yang**[1] · **Baojun Ma**[2]

Accepted: 11 April 2025 / Published online: 18 May 2025

Abstract

Click-through rate (CTR) prediction is a critical component of recommender systems, helping infer the likelihood of a user's engagement (i.e., clicks) with a particular item. Previous studies have typically focused on leveraging either dynamic historical user behaviors or heterogeneous information for feature augmentation. However, relying solely on one aspect is insufficient to capture the intricate user-item dependencies. In this paper, we propose dynamic heterogeneous graph convolutional networks (DH-GCN) for CTR prediction, combining dynamic user-item interactions and heterogeneous information. Specifically, we construct three graphs: item knowledge graph, user-user graph, and user-item graph, and design a novel graph-to-graph learning method to realize the sharing of neighbors and relationships in the GCN framework. Moreover, we leverage multi-granularity time-sliced user-item graphs to capture evolving user preference trajectories. Experiments on three public datasets show that DH-GCN makes significant improvements over state-of-the-art baselines and achieves 0.01-level improvements in AUC, accuracy, and F1.

Keywords Click-through rate prediction · Graph convolutional networks · Dynamic user-item interactions · Heterogeneous information networks

✉ Baojun Ma
mabaojun@shisu.edu.cn

Ying Jin
jinying.isec@gmail.com

Yanwu Yang
yangyanwu.isec@gmail.com

[1] School of Management, Huazhong University of Science and Technology, Wuhan 430074, China

[2] Key Laboratory of Brain-Machine Intelligence for Information Behavior (Ministry of Education and Shanghai), School of Business and Management, Shanghai International Studies University, Shanghai 201620, China

1 Introduction

With the explosive growth of the Internet, users are increasingly confronted with the situation of information overload. Recommender systems have increasing significance in finding items matching users' interests [4, 72]. As an integral and indispensable component of recommender systems, click-through rate (CTR) prediction infers the probability of a user clicking on the target item [71]. An improvement of 0.1% in the CTR prediction accuracy would result in hundreds of millions of extra earnings [39].

In the literature on CTR prediction, plenty of studies have explored feature augmentation to address data sparsity and cold-start issues and improve model performance, mainly from two perspectives, namely dynamics and heterogeneity. The former enriches feature representation with temporal information during the historical interactions between users and items (or advertisements), such as users' behaviors and evolving user preferences over time [79]; while the latter enhances prediction models with various heterogeneous information in the application scenarios, such as user profiles, fruitful semantics about items and multi-type relationships among them [15, 30], and contextual factors, e.g., locations. Given the dynamic and heterogeneous nature of user-item interactions, it is widely acknowledged that dynamics and heterogeneity essentially complement to each other in the task of CTR prediction [18]. Graph neural networks (GNNs) have shown great power in representing heterogeneous information networks and accommodating multi-source information and higher-order feature interactions [8, 63, 65]. Recently, researchers have attempted to combine the two strengths to provide synergistic benefits (e.g., [37, 47]) by incorporating knowledge graphs of items into user's behavioral sequences. However, they handled each information network in a relatively monotonous manner, thus failed to explore deep relationships across heterogeneous graphs, which restricts the model's expressive capacity to capture complex interaction patterns [48]. Moreover, they explored time-dependent representations from the user's perspective, while neglecting dynamic interactions on the item side [13] and the multi-granularity variability of users' interactions [70], hindering the ability of capturing complex latent connections and the intricate nature of interactions [67]. This research is one of the first efforts in this direction, aiming to provide a unified model taking into account both dynamics (i.e., temporal dynamics of user preferences and multi-granularity time-sliced interactions) in users' behaviors and heterogeneity (i.e., multiple heterogeneous information networks) in the GNNs frameworks to improve CTR prediction.

There are several challenges to realize the full potential of dynamic heterogeneous interactions in CTR prediction. First, how to effectively capture complex interaction patterns from heterogeneous information networks is demanding because multi-aspects information interweaves with each other [67]. Second, how to capture structural and temporal information simultaneously poses another challenge in handling higher-order connections.

In this research, we propose dynamic heterogeneous graph convolutional networks (DH-GCN) exploiting complex interaction patterns and temporal dynamics

of user preferences to facilitate CTR prediction. DH-GCN incorporates heterogeneous information networks with a knowledge graph to represent item-related semantics, a user-user graph to discover collaborative signals based on preference similarity, and a user-item graph to capture the user-item dependencies on the temporal axis. To delve into correlations across these graphs, we devise a novel graph-to-graph learning method to distill information by mapping a node in one graph to that in another graph to obtain a unified neighborhood space in the Graph Convolutional Networks (GCN) framework. Moreover, considering dynamic user preferences and various item categories (e.g., daily necessities and holiday decorations), we eschew simply modeling the holistic behavior history and exploit multi-granularity time-sliced user-item graphs. We correlate a series of temporal graph representations corresponding to time intervals (e.g., weekly, monthly, yearly) to generate long-term representations using the sequential modeling method Receptance Weighted Key Value (RWKV) [46], which utilizes a linear attention mechanism and combines the advantages of Transformer and Recurrent Neural Networks (RNN). Experiments on three public datasets (i.e., Last.FM, MovieLens-1M, and E-Commerce) demonstrate that DH-GCN yields improvements of 0.02 in AUC and F1, and achieves the superiority with increases of 0.02 (Last.FM), 0.02 (MovieLens-1M), and 0.01 (E-Commerce), respectively in accuracy, comparing to state-of-the-art baselines. In addition, we conduct ablation studies to verify whether each component can boost the model performance. We find that representation learning on each information network and modeling multi-granularity time slices have positive impacts on CTR prediction performance.

Contributions from this research are summarized as follows. First, we propose dynamic heterogeneous graph convolutional networks (DH-GCN) for CTR prediction, which empowers structure-aware and time-aware representations reflective of interaction dynamics. Multi-granularity time-sliced user-item graph representation learning enables the discovery of both short-term and long-term user-item dependencies. Second, our graph-to-graph learning method integrates heterogeneous information, providing synergistic effects for a holistic understanding of user preferences. Third, DH-GCN demonstrates significant superiority over existing baselines in terms of AUC, accuracy, and F1, underscoring the effectiveness of our proposed model in capturing dynamic interactions and heterogeneous information.

The structure of this paper is outlined as follows. In Sect. 2, we provide a concise review of related literature. In Sect. 3, we introduce details of our proposed method. In Sect. 4, we report experimental results and analysis. Finally, in Sect. 5, we conclude key findings and discuss future research directions.

2 Related work

2.1 CTR prediction models

CTR prediction has emerged as a pivotal area of research, attracting extensive studies aimed at enhancing its accuracy and effectiveness. Among traditional

CTR prediction methods, logistic regression models rely on feature engineering to improve accuracy [33, 71], which can be easily implemented due to simplicity, but consider only the first-order interactions and have limited representation power [38]. Factorization Machines (FM) [31, 53, 68] are proficient in capturing feature interactions efficiently; however, they are unable to provide an adequate understanding of intricate patterns [51]. Deep Neural Networks (DNNs) provide significant advancements over traditional methods and can capture high-order feature interactions automatically [23, 25, 27], as exemplified by models such as Wide&Deep [14], DeepFM [21], and Deep & Cross Network (DCN) [56, 60]. For further information, refer to a detailed review on CTR prediction by Yang and Zhai [71].

Within the scope of CTR prediction, the interactions between users and items can be depicted as a heterogeneous bipartite graph, where nodes denote users and items, and edges denote interaction relationships. GNNs are well-suited to handle graph structures and can effectively model complex relationships [3]. GNNs propagate information along links in the graph and aggregate neighborhood information to update the node representation [50, 75]. For example, Li et al. [18] utilize a graph structure to represent features in multi-field categories, allowing for flexible and explicit modeling of complex feature interactions. Zhai et al. [75] combine the advantages of GNNs in graph learning and causality in interpretability for CTR prediction. This method integrates graph representations of features, users, and ads to enable causal inference among field features.

2.2 CTR prediction based on dynamic user-item interactions

To capture the time-varying interactions and the dynamic dependencies among features, researchers have increasingly modeled the evolving user behaviors and interests [10, 17, 34]. Zhou et al. [80] design a local activation unit to model latent user interests underlying concrete behaviors, enabling adaptive learning of user interest representations. Their method assigns higher weights to historical behaviors more relevant to a specific ad. Zhou et al. [79] point out that most previous works simply consider user behavior as a representation of interest, without capturing the dynamics of interest. They incorporate Gated Recurrent Unit (GRU) augmented with an attention-based update gate to capture the evolving user interests from historical behaviors. He et al. [26] apply contrastive learning to decouple multi-type behavior (e.g., add-to-cart, order) sequences and capture discrepancy and consistency characteristics among various behaviors to represent user interests.

The above research focuses on modeling temporal dynamics from the user's perspective. However, some studies observe the insufficiency of fixed item embeddings and focus on modeling dynamic item characteristics [36]. For example, Zhang et al. [77] handle the problem of sparse user behaviors and dynamically capture the characteristics of the item with interacted users and timestamps. Similar to us, Wang et al. [62] construct a time-evolving graph of the user's sequential behaviors and dig into the user's real-time interests, but they only model the dynamics from the user's view.

2.3 CTR prediction based on heterogeneous information

Heterogeneous information can alleviate the problem of data sparsity in CTR prediction by providing rich semantic information and various relationships.

User-item interactions are the most basic heterogeneous information, providing explicit or implicit feedback, such as clicks, or purchases. This type of network allows for the model to consider personal user interests and specific contextual factors, which are typically modeled by neural networks [11, 66]. For example, Zhang et al. [77] utilize Transformer to learn the item's multiple aspects, representing the candidate item through its interactions with users. Lyu et al. [42] propose a Multi Classifier CTR prediction model to model users' inherent click tendency and preference towards various items.

Many researchers have introduced various heterogeneous information into CTR prediction, which has proven effective in improving performance. For instance, Zhang et al. [78] design a Multi-Interactive Layer, which simultaneously considers user-item interactions and context information to model fine-grained features. Yang et al. [69] utilize straightforward and auxiliary information, including the compositional layouts, the visual images, and the interactions to enrich ad representations and improve the ad CTR prediction. Some other research (e.g., [22, 35]) augments the user-item graph with additional information such as user demographics, item characteristics, or contextual situations [44] to improve the performance of GNNs in CTR prediction.

Recent studies have gone a further step by incorporating knowledge graph as side information. Among the various heterogeneous networks, knowledge graph provides a much more comprehensive and sophisticated semantic representation [64, 82]. For example, Wang et al. [58] design a knowledge-aware Convolutional Neural Network (CNN) for news CTR prediction, which uses TransD [29] to generate knowledge graph embeddings and realize the alignment and combination of word-level and knowledge-level information. However, this method ignores high-order connections between knowledge entities. On the contrary, Feng et al. [16] bridge relational paths on the knowledge graph and apply Bidirectional Long Short Term Memory (Bi-LSTM) [20] to encode each path, modeling multiplex relationships between user behaviors and specific items. However, how to design meta-structures in the graph properly requires domain knowledge. Subsequent studies focus on embedding propagation among multi-hop neighbors. Wang et al. [57] construct ripple sets on the item knowledge graph to propagate user preferences, but the relationships are weakly personalized. Wang et al. [59] characterize neighborhood information and achieve high-order connection modeling on the knowledge graph under the GNNs framework. Subsequently, some research (e.g., [45, 83]) performs high-order propagation within the graph structure by utilizing the potential of contrastive learning, which contrasts positive pairs against negative pairs from multiple views.

Despite the significant progress made in CTR prediction, only a few studies combine dynamic user-item interactions with heterogeneous information. Peng et al. [47] explore high-order propagation of user interests in the knowledge graph and apply Bi-LSTM to capture evolving user interests. Li et al. [37] mine paths connecting

interacted items within the knowledge graph to learn the dynamic representations of user interests.

Existing methods incorporate heterogeneous information to enrich the contextual understanding of user behaviors and preferences. However, they generally model each information network independently to aggregate user or item neighborhood information. It is worth noting that there exist interactive patterns across heterogeneous networks, potentially concealing the interactions between user- and item-neighbors [48]. Therefore, previous research is limited to exploring deep connections across heterogeneous graphs, which restricts the model's expressive capacity to capture complex interaction patterns. What's more, most studies focus on propagating entity information within the knowledge graph, without fully exploiting historical user-item interactions [12]. They pay great attention to structural relationships and semantic information in the knowledge graph but lack an elaborate design for the most fundamental user-item interactions. This oversight neglects the significant importance of discovering user preferences and behavior patterns in relation to items. While many studies model historical behavior sequences to learn user interest representations, they often overlook the dynamic nature of items interacting with different users over time. Moreover, these studies typically model the entire timeline, which limits their ability to capture both long-term user preferences and short-term interest changes [13].

In our research, we propose dynamic heterogeneous graph convolutional networks (DH-GCN) exploiting temporal dynamics of user preferences and complex interaction patterns to facilitate CTR prediction. First, we construct multiple information networks (i.e., a knowledge graph, a user-user graph and a user-item graph). Second, we exploit multi-granularity time-sliced user-item graphs to discover both short- and long-term user-item dependency. Third, we design a graph-to-graph learning method in the GCN framework to explore correlations across multi-source heterogeneous information networks, instead of dealing with each information network in isolation or neglecting potential connections across heterogeneous networks. This method allows us to derive enriched representations of users and items for CTR prediction.

3 Dynamic heterogeneous graph convolutional networks

In this section, we first describe the architecture of DH-GCN, and then elaborate on details of each component.

3.1 Model architecture

Figure 1 shows the overall architecture of DH-GCN, which consists of four main modules. The first component constructs heterogeneous information networks, including a knowledge graph, user-user graph, and user-item graph. The second component, representation learning for dynamic heterogeneous networks employs hierarchical multi-head self-attention [54] and GCN to learn enriched

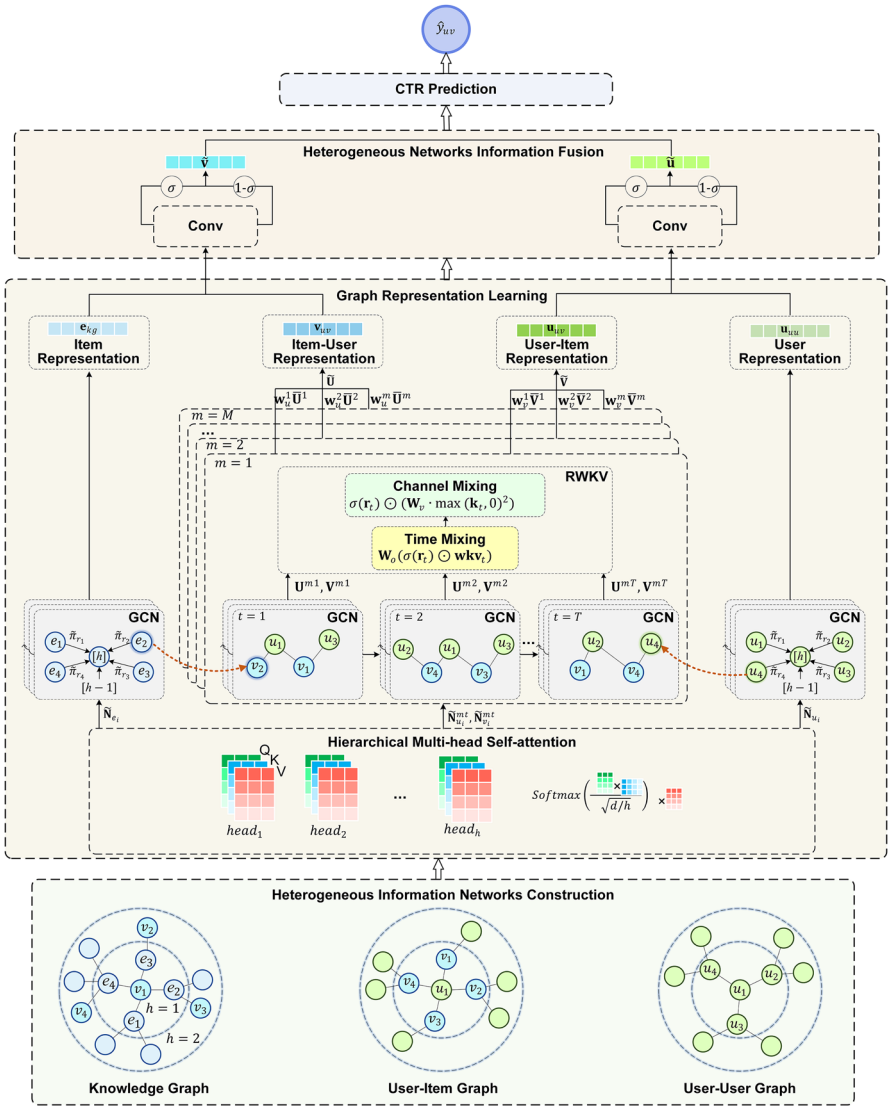


Fig. 1 Architecture of DH-GCN

representations, in which graph-to-graph learning (red dashed lines) maps one node in one graph to that in another graph to obtain sharing neighborhood space. Then, dynamic representations learned from multi-granularity time-sliced user-item graphs are correlated by the sequential modeling method RWKV. The third component, heterogeneous networks information fusion fuses information from heterogeneous networks through convolution operations. The final component outputs the predicted click probability.

3.2 Heterogeneous information networks construction

In order to explore rich semantic relatedness, we elaborately construct heterogeneous information networks, including a knowledge graph $\mathcal{G}_{kg}$, user-user graph $\mathcal{G}_{uu}$ and user-item graph $\mathcal{G}_{uv}$ to enhance CTR prediction.

Knowledge graph. A knowledge graph is a structured representation of information that consists of entities and their relations in a graph format. The knowledge graph $\mathcal{G}_{kg}$ is composed of massive entity-relation-entity (h, r, t) triplets, where $h \in \mathcal{E}$ is the head entity, $r \in \mathcal{R}$ is the relation entity, and $t \in \mathcal{E}$ is the tail entity. For example, the triplet (The Great Gatsby, written_by, F. Scott Fitzgerald) illustrates that the literary work "The Great Gatsby" is written by the author "F. Scott Fitzgerald". To construct the item knowledge graph, we first link the items and their attributes to external knowledge sources, such as Microsoft Satori, and then iteratively extend the graph through multi-hop connections, thereby enhancing its depth and breadth. Each item can be aligned with the entity in the knowledge graph, that is, the item set $\mathcal{V} \subseteq \mathcal{E}$. Thus, items' information is enriched by benefiting from the attributes and connections within the knowledge graph.

User-user graph. We also construct a user-user graph $\mathcal{G}_{uu}$ to learn user preference similarity. If users u_i and u_j have common clicked items, there exists an edge from u_i to u_j. The weight from u_i to u_j can be calculated as $w_{ij} = \frac{|S^{u_i} \cap S^{u_j}|}{|S^{u_j}|}$, where S^{u_i} and S^{u_j} denote the clicked item set of u_i and u_j, respectively.

User-item graph. In a typical sequential recommendation scenario, a temporal bipartite graph $\mathcal{G}_{uv}$ is used to present user-item interactions, where nodes represent users or items and edges represent users' feedback to items with timestamps. A set of users is defined as $\mathcal{U} = \{u_1, u_2, \ldots, u_{|\mathcal{U}|}\}$ and a set of items is defined as $\mathcal{V} = \{v_1, v_2, \ldots, v_{|\mathcal{V}|}\}$, where $|\mathcal{U}|$ and $|\mathcal{V}|$ are the number of users and items, respectively. The user-item interaction matrix is defined as $Y = \{y_{uv} | u \in \mathcal{U}, v \in \mathcal{V}\}$, where y_{uv}=1 denotes user u interacts with item v; otherwise y_{uv}=0.

To learn the dynamic evolutionary user-item interaction patterns, we construct the multi-granularity continuous-time bipartite graphs based on the user-item graph $\mathcal{G}_{uv}$. We slice the holistic timeline into T intervals, with each time slice comprising user-item interactions occurring within the corresponding interval. A certain interval length represents a time granularity. Therefore, under the granularity m, the time-sliced interaction collections can be denoted as $\mathcal{T}^m = \{\mathcal{T}^{m1}, \mathcal{T}^{m2}, \cdots, \mathcal{T}^{mT}\}$, where the t-th time-slice $\mathcal{T}^{mt}$ corresponds to the user-item bipartite graph $\mathcal{G}^{mt}$. Multi-granularity graph representation learning is based on $\mathcal{T} = \{\mathcal{T}^1, \mathcal{T}^2, \cdots, \mathcal{T}^M\}$ with the number of M. It is notable that in the case where there are no interactions between users and items at that time slice, corresponding representations are not learned and updated.

Consider an e-commerce platform where the entire timeline spans one year. At a monthly granularity (m as month), suppose a user purchases several spring clothes in March (slice $\mathcal{T}^{m3}$), which shows user's preferences at a coarse-grained level; while at a weekly granularity (m as week), the user purchases such as a new pair of jeans and a light jacket with a discount in the first week in March (slice $\mathcal{T}^{m9}$), which indicates user's preferences at a fine-grained level.

Given the knowledge graph $\mathcal{G}_{kg}$, user-user graph $\mathcal{G}_{uu}$, and user-item graph $\mathcal{G}_{uv}$, we aim to learn a prediction function $\hat{y}_{uv} = \mathcal{F}\big(u, v, \mathcal{G}_{kg}, \mathcal{G}_{uu}, \mathcal{G}_{uv}, \mathcal{T}; \Theta\big)$, where $\hat{y}_{uv}$ denotes the predicted probability that the user u will interact with the item v, and Θ denotes the model parameters of prediction function $\mathcal{F}$. Notations used in this paper are summarized in Table 1. Matrices are denoted by bold uppercase letters, and vectors are denoted by bold lowercase letters.

3.3 Representation learning for dynamic heterogeneous networks

We first present how to embed heterogeneous information networks and then introduce hierarchical multi-head self-attention to distinctively prioritize neighboring nodes. Subsequently, GCN and the sequential modeling method RWKV are used to learn representations of heterogeneous information and dynamic interactions.

3.3.1 Heterogeneous information networks embedding

We first map heterogeneous information networks to low-dimensional embeddings [2, 7]. Embeddings of nodes and relations between them in each network are randomly initialized, allowing the model to progressively learn task-specific embeddings through training.

The embeddings of users and items in user-item graph $\mathcal{G}_{uv}$ consist of three types of embeddings: initial embedding, temporal embedding, and positional encoding. We adopt temporal embedding [81] to learn continuous time-dependent interactions, in which each timestamp, such as hour, week, and month is projected into a fixed-dimensional vector using a learnable embedding layer. We also use positional encoding [54] to differentiate the positional information sorted by interaction time in the user or item neighborhood, which enhances the model's learning of order information. Positional encoding is implemented using sine and cosine functions with different frequencies. The embedding operation process of user u and item v are shown in Eq. 1.

$$\begin{cases} \mathbf{u} = \mathbf{u}_0 + \mathbf{t}_u + \mathbf{p}_u \\ \mathbf{v} = \mathbf{v}_0 + \mathbf{t}_v + \mathbf{p}_v \end{cases}, \tag{1}$$

where $\mathbf{u} \in \mathbb{R}^d$ and $\mathbf{v} \in \mathbb{R}^d$ are vector representations, $\mathbf{u}_0 \in \mathbb{R}^d$ and $\mathbf{v}_0 \in \mathbb{R}^d$ are initialized embeddings, $\mathbf{t}_u \in \mathbb{R}^d$ and $\mathbf{t}_v \in \mathbb{R}^d$ are temporal embeddings, $\mathbf{p}_u \in \mathbb{R}^d$ and $\mathbf{p}_v \in \mathbb{R}^d$ are positional encodings of user u and item v, respectively. d denotes the embedding dimension.

3.3.2 Hierarchical multi-head self-attention

Given the knowledge graph $\mathcal{G}_{kg}$, user-user graph $\mathcal{G}_{uu}$, and user-item graph $\mathcal{G}_{uv}$, we introduce hierarchical multi-head self-attention, which characterizes the importance of neighbors with the same connectivity order [1]. The attention computation is a scaled

Table 1 Notations and descriptions

Notations	Descriptions
u	User
v	Item
$\mathcal{U} = \{u_1, u_2, \ldots, u_{\lvert\mathcal{U}\rvert}\}$	A set of users with the number of $\lvert\mathcal{U}\rvert$
$\mathcal{V} = \{v_1, v_2, \ldots, v_{\lvert\mathcal{V}\rvert}\}$	A set of items with the number of $\lvert\mathcal{V}\rvert$
$Y = \{y_{uv} \lvert u \in \mathcal{U}, v \in \mathcal{V}\}$	The user-item interaction matrix
$\mathcal{G}_{uv}$	User-item graph
$\mathcal{G}_{kg}$	Knowledge graph
$\mathcal{G}_{uu}$	User-user graph
$\mathcal{E}$	The set of head and tail entities in the knowledge graph
$\mathcal{R}$	The set of relations in the knowledge graph
(h, r, t)	The knowledge triplet, where $h \in \mathcal{E}$ is the head entity, $r \in \mathcal{R}$ is the relation entity, and $t \in \mathcal{E}$ is the tail entity
S^{u_i}	The set of items clicked by user u_i
w_{ij}	The weight from u_i to u_j in $\mathcal{G}_{uu}$
$\hat{y}_{uv}$	The predicted probability that user u will click item v
$\mathcal{F}$	The function of click-through rate prediction
Θ	Parameters of the prediction function $\mathcal{F}$
d	The embedding dimension
$\mathbf{u}$, $\mathbf{v}$	Vector representations of u and v, respectively
$\mathbf{u}_0$, $\mathbf{v}_0$	Initialized embeddings of u and v, respectively
$\mathbf{t}_u$, $\mathbf{t}_v$	Temporal embeddings of u and v, respectively
$\mathbf{p}_u$, $\mathbf{p}_v$	Positional encodings of u and v, respectively
$\mathbf{W}^O$,$\mathbf{W}_i^Q$,$\mathbf{W}_i^K$,$\mathbf{W}_i^V$	Parameter matrices of the projection transformation in multi-head self-attention
$\mathcal{N}(e_i)$	Neighbors of entity e_i
$\widetilde{\mathbf{N}}_{e_i}$, $\widetilde{\mathbf{N}}_{u_i}$, $\widetilde{\mathbf{N}}_{u_i}^{mt}$	Attentive neighbor matrices generated from $\mathcal{G}_{kg}$, $\mathcal{G}_{uu}$, $\mathcal{G}_{uv}$, respectively
$\mathcal{T} = \{\mathcal{T}^1, \mathcal{T}^2, \cdots, \mathcal{T}^M\}$	Time slices of user-item interactions with M granularities
$\mathcal{T}^{mt}$	The t-th time slice of user-item interactions under granularity m
$\mathcal{G}^{mt}$	The user-item bipartite graph corresponding to $\mathcal{T}^{mt}$
$MultiHead(\cdot)$	Multi-head self-attention
r_{e_i,e_j}	The relationship between entities e_i and e_j
$\pi_{r_{e_i,e_j}}$	The relevance score between user u and relationship r_{e_i,e_j}
$g(\cdot)$	The function of calculating the relevance score (e.g., inner product)
$\mathbf{e}_i$	The representation of entity e_i
$\mathbf{e}_{\mathcal{N}(e_i)}$	The neighborhood information representation of entity e_i learned from $\mathcal{G}_{\text{kg}}$
h	The number of propagation layers in GCN
$f(\cdot)$	Aggregator
$\mathbf{e}_{kg}$	The item representation generated from $\mathcal{G}_{kg}$
γ	The nonlinear activation function such as ReLU
$\mathbf{W}$, $\mathbf{W}_1$, $\mathbf{W}_2$	Parameter matrices in the aggregator
$\lvert\lvert$	The concatenation operator
$\odot$	The element-wise product operator

Table 1 (continued)

Notations	Descriptions
$\pi_{r_{u_i,u_j}}$	The relevance score between users u_i and u_j generated from $\mathcal{G}_{uu}$
$\mathbf{u}_{\mathcal{N}(u_i)}$	The neighborhood information representation of user u_i learned from $\mathcal{G}_{uu}$
$\mathbf{u}_{uu}$	The user representation generated from $\mathcal{G}_{uu}$
$\pi^{mt}_{r_{u_i,v_i}}$	The relevance score between user u_i and interacted item v_i in $\mathcal{G}^{mt}$
$\mathbf{u}^{mt}_{\mathcal{N}(u_i)}$	The neighborhood information representation of user u_i learned from $\mathcal{G}^{mt}$
$\mathbf{U}^m = [\mathbf{U}^{m1};\mathbf{U}^{m2};\cdots;\mathbf{U}^{mT}]$	The user-specific representation matrices under the granularity m
$\mathbf{V}^m = [\mathbf{V}^{m1};\mathbf{V}^{m2};\cdots;\mathbf{V}^{mT}]$	The item-specific representation matrices under the granularity m
$\mathbf{r}_t$	The receptance of past information in RWKV at time t
$\mathbf{k}_t$	The key vector in RWKV at time t
$\mathbf{v}_t$	The value vector in RWKV at time t
$\mathbf{wkv}_t$	The attention score in RWKV at time t
$\mathbf{o}_t$	The output of RWKV at time t
$\boldsymbol{\mu}_r, \boldsymbol{\mu}_k, \boldsymbol{\mu}_v$	Weights controling information updates in RWKV
$\mathbf{w}, \mathbf{x}$	Time decay vectors in RWKV
$\mathbf{W}_r, \mathbf{W}_k, \mathbf{W}_v, \mathbf{W}_o$	Weight matrices trainable in RWKV
Θ^m_U, Θ^m_V	Parameters trainable in RWKV
$\mathbf{u}_{uv}$	The user representation generated from $\mathcal{G}_{uv}$
$\mathbf{v}_{uv}$	The item representation generated from $\mathcal{G}_{uv}$
$\sigma(\cdot)$	The sigmoid function
$\widetilde{\mathbf{u}}$	The final user representation
$\widetilde{\mathbf{v}}$	The final item representation
$\mathcal{L}$	The complete loss function
$\mathcal{J}$	The cross-entropy loss
P	The negative uniformed sampling distribution
N^u	The number of each user's negative samples
λ	The L 2 regularization coefficient

dot-product computation between a query ($\mathbf{Q}$) and a set of key ($\mathbf{K}$) -value ($\mathbf{V}$) pairs (Eq. 2).

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = Softmax\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V} \tag{2}$$

Each head attention from different projection subspaces is computed independently in parallel, then they are concatenated together and linearly projected once more to obtain the output (Eq. 3).

$$\begin{cases} MultiHead(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = Concat\big(head_1, \ldots, head_h\big)\mathbf{W}^O \\ head_i = Attention\Big(\mathbf{Q}\mathbf{W}^Q_i, \mathbf{K}\mathbf{W}^K_i, \mathbf{V}\mathbf{W}^V_i\Big) \end{cases}, \tag{3}$$

where $\mathbf{W}^O$,$\mathbf{W}^Q_i$,$\mathbf{W}^K_i$,$\mathbf{W}^V_i$ are parameter matrices in the projection transformation.

For a knowledge graph containing fruitful connections and side information for items [5, 83], it is imperative to differentiate the importance of adjacent entities in describing and characterizing user preferences [28, 52]. The user-user graph considers similar behaviors indicating shared interests among users [74]. Obviously, neighboring users contribute unequally to the target user's preference similarity. For time-sliced user-item graphs, the historical clicked items reflect the user's behavioral motivations and personal interests. Therefore, it is essential to assign different weights to neighbors in each graph through multi-head self-attention (Eq. 4).

$$\begin{cases} \widetilde{\mathbf{N}}_{e_i} = MultiHead\left(\mathbf{N}_{e_i}, \mathbf{N}_{e_i}, \mathbf{N}_{e_i}\right) \\ \widetilde{\mathbf{N}}_{u_i} = MultiHead\left(\mathbf{N}_{u_i}, \mathbf{N}_{u_i}, \mathbf{N}_{u_i}\right) \\ \widetilde{\mathbf{N}}^{mt}_{u_i} = MultiHead\left(\mathbf{N}^{mt}_{u_i}, \mathbf{N}^{mt}_{u_i}, \mathbf{N}^{mt}_{u_i}\right) \end{cases}, \tag{4}$$

where $\mathbf{N}_{e_i}$ and $\widetilde{\mathbf{N}}_{e_i}$ denote neighbor matrices of entity e_i in $\mathcal{G}_{kg}$ before and after the update, respectively; $\mathbf{N}_{u_i}$ and $\widetilde{\mathbf{N}}_{u_i}$ denote neighbor matrices of user u_i in $\mathcal{G}_{uu}$ before and after the update, respectively; $\mathbf{N}^{mt}_{u_i}$ and $\widetilde{\mathbf{N}}^{mt}_{u_i}$ denote neighbor matrices of user u_i in $\mathcal{G}^{mt}$ before and after the update, respectively.

In the representation learning on multi-granularity time-sliced user-item graphs, multi-head self-attention is also used to differentiate the neighboring nodes of items. The learning process of the item side is similar to the third formula in Eq. 4, and thus will not be repeated here.

3.3.3 Representation learning on heterogeneous information networks

In this section, we introduce representation learning on heterogeneous information networks, which is designed to capture evolutionary interaction patterns and pivotal factors influencing user behaviors and preferences. Instead of modeling independent intra-graph connections, the graph-to-graph learning method maps the node from one graph to another. Specifically, it integrates the neighbors from different spaces of the given node into a unified space. Therefore, the model can incorporate complex heterogeneous relations across networks. It should be mentioned that the following aggregation processes contain information from different subspaces, which is not redundantly described next.

Knowledge graph representation learning. We employ GCN [59] to capture user's high-order preference propagation in the knowledge graph. The model aggregates a node's neighboring information and updates its representation based on context representation.

User preferences are inherently personalized; for instance, in a purchase scenario, one user may value the brand of the product, while another user may pay more attention to the style of the product. Accordingly, we introduce a relation-aware attention mechanism that models the significance of relationships between entities for each user [19]. For user u and item v, neighborhood information adjacent to v is aggregated with different attention weights (Eq. 5).

$$\begin{cases} \mathbf{e}_{\mathcal{N}(e_i)} = \sum_{e_j \in \mathcal{N}(e_i)} \tilde{\pi}_{r_{e_i,e_j}} \mathbf{e}_j \\ \tilde{\pi}_{r_{e_i,e_j}} = \frac{\exp(\pi_{r_{e_i,e_j}})}{\sum_{e_j \in \mathcal{N}(e_i)} \exp(\pi_{r_{e_i,e_j}})} \\ \pi_{r_{e_i,e_j}} = g(u, \mathbf{r}_{e_i,e_j}) \end{cases}, \tag{5}$$

where $\mathbf{e}_i$ denotes the representation of entity e_i, $\mathbf{e}_{\mathcal{N}(e_i)}$ denotes neighborhood information representation of e_i. $\widetilde{\pi}_{r_{e_i,e_j}}$ is the normalized result of $\pi_{r_{e_i,e_j}}$, which is calculated by using the function $g(\cdot)$, such as the inner product. The relevance score $\pi_{r_{e_i,e_j}}$ represents the importance of relationship r_{e_i,e_j} for user u.

Take the knowledge graph $\mathcal{G}_{kg}$ in Fig. 1 as an example. The computation of the neighborhood information representation of item v_1 (i.e., $\mathbf{e}_{\mathcal{N}(v_1)}$), which has four neighboring entities (i.e., e_1, e_2, e_3 and e_4), is carried out in three steps: relevance scores $\pi_{r_{v_1,e_1}}$, $\pi_{r_{v_1,e_2}}$, $\pi_{r_{v_1,e_3}}$ and $\pi_{r_{v_1,e_4}}$ are first calculated by using the third formula in Eq. 5, which are then fed into the second formula to make normalization and generate attention weights $\widetilde{\pi}_{r_{v_1,e_1}}$, $\widetilde{\pi}_{r_{v_1,e_2}}$, $\widetilde{\pi}_{r_{v_1,e_3}}$ and $\widetilde{\pi}_{r_{v_1,e_4}}$, finally these weights are fed into the first formula to obtain $\mathbf{e}_{\mathcal{N}(v_1)}$.

To achieve the item representation with high-order structural and semantic information, we update the entity representation $\mathbf{e}_i$ with its neighborhood representation $\mathbf{e}_{\mathcal{N}(e_i)}$. For GCN-based representation learning in the knowledge graph, the h-hop aggregation process is implemented using the aggregator $f(\cdot)$, as shown in Eq. 6.

$$\mathbf{e}_i[h] = f\Big(\mathbf{e}_i[h-1], \mathbf{e}_{\mathcal{N}(e_i)}[h-1]\Big). \tag{6}$$

There are three common types of aggregators $f(\cdot)$, including sum, concat, and bi- interaction aggregators, which are evaluated in Sect. 4.5.1.

(1) Sum aggregator [32] sums the entity representation and its neighborhood representation before nonlinear transformation (Eq. 7).

$$f_{sum} = \gamma\Big(\mathbf{W}\Big(\mathbf{e}_i + \mathbf{e}_{\mathcal{N}(e_i)}\Big)\Big), \tag{7}$$

where γ denotes the nonlinear function such as ReLU, and $\mathbf{W}$ denotes the parameter matrix.

(2) Concat aggregator [24] concatenates the entity representation and its neighborhood representation, followed by nonlinear transformation (Eq. 8).

$$f_{concat} = \gamma(\mathbf{W}(\mathbf{e}_i || \mathbf{e}_{\mathcal{N}(e_i)})), \tag{8}$$

where $||$ is the concatenation operator.

(3) Bi-interaction aggregator [61] considers the combination of a single interaction between two representations, applying both summation and element-wise product operators (Eq. 9).

$$f_{bi-interaction} = \gamma\left(\mathbf{W}_1\left(\mathbf{e}_i + \mathbf{e}_{\mathcal{N}(e_i)}\right)\right) + \gamma\left(\mathbf{W}_2\left(\mathbf{e}_i \odot \mathbf{e}_{\mathcal{N}(e_i)}\right)\right), \tag{9}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are parameter matrices.

Thus, each entity incorporates the initial representation and its neighborhood information. After iterative multi-hop propagation in the knowledge graph, we obtain the item representation $\mathbf{e}_{kg}$.

User-user graph representation learning. Using a GCN model similar to the one employed for the knowledge graph, each user's representation is enriched by aggregating representations of similar users (Eq. 10).

$$\begin{cases} \mathbf{u}_{\mathcal{N}(u_i)} = \sum_{u_j \in \mathcal{N}(u_i)} \tilde{\pi}_{r_{u_i,u_j}} \mathbf{u}_j \\ \tilde{\pi}_{r_{u_i,u_j}} = \frac{\exp(\pi_{r_{u_i,u_j}})}{\sum_{u_j \in \mathcal{N}(u_i)} \exp(\pi_{r_{u_i,u_j}})} \\ \pi_{r_{u_i,u_j}} = g\left(\mathbf{u}_i, w_{ij}\mathbf{r}_{u_i,u_j}\right) \end{cases}, \tag{10}$$

where $\mathbf{u}_{\mathcal{N}(u_i)}$ denotes the neighborhood information representation of user u_i, and $\pi_{r_{u_i,u_j}}$ denotes the relevance score between users u_i and u_j.

Take the user-user graph $\mathcal{G}_{uu}$ in Fig. 1 as an example. The computation of the neighborhood information representation of user u_1 (i.e., $\mathbf{u}_{\mathcal{N}(u_1)}$), who has three neighbors (i.e., u_2, u_3 and u_4), is carried out in three steps: neighboring scores $\pi_{r_{u_1,u_2}}$, $\pi_{r_{u_1,u_3}}$, and $\pi_{r_{u_1,u_4}}$ are first calculated by using the third formula in Eq. 10, which are then fed into the second formula to make normalization and generate attention weights $\tilde{\pi}_{r_{u_1,u_2}}$, $\tilde{\pi}_{r_{u_1,u_3}}$, and $\tilde{\pi}_{r_{u_1,u_4}}$, finally these weights are fed into the first formula to obtain $\mathbf{u}_{\mathcal{N}(u_1)}$.

Then, aggregators propagate information and incorporate preferences from similar users (Eq. 11). The final aggregation layer generates the learned user representation $\mathbf{u}_{uu}$ from the user-user graph.

$$\mathbf{u}_i[h] = f\left(\mathbf{u}_i[h-1], \mathbf{u}_{\mathcal{N}(u_i)}[h-1]\right) \tag{11}$$

Multi-granularity time-sliced user-item graph representation learning. It is worth noting that temporal dependencies between users and items underlying dynamic interactions cannot be overlooked [73]. Therefore, we design multi-granularity time-sliced user-item graphs to capture evolutionary representations for both the user and item sides. Concretely, we employ GCN to distill structural connection information at each time slice (Eq. 12).

$$\begin{cases} \mathbf{u}^{mt}_{\mathcal{N}(u_i)} = \sum_{v_i^{mt} \in \mathcal{N}(u_i^{mt})} \tilde{\pi}^{mt}_{r_{u_i,v_i}} \mathbf{v}_i^{mt} \\ \tilde{\pi}^{mt}_{r_{u_i,v_i}} = \frac{\exp(\pi^{mt}_{r_{u_i,v_i}})}{\sum_{v_i^{mt} \in \mathcal{N}(u_i^{mt})} \exp(\pi^{mt}_{r_{u_i,v_i}})} \\ \pi^{mt}_{r_{u_i,v_i}} = g\left(\mathbf{u}_i^{mt}, \mathbf{v}_i^{mt}\right) \end{cases}, \tag{12}$$

where $\mathbf{u}^{mt}_{\mathcal{N}(u_i)}$ denotes neighborhood information representation of user u_i learned from $\mathcal{G}^{mt}$, and $\pi^{mt}_{r_{u_i,v_i}}$ denotes the neighboring score between user u_i and interacted item v_i.

Take the second time slice under the granularity m in Fig. 1 as an example. The computation of the neighborhood information representation of user u_1 (i.e., $\mathbf{u}^{m2}_{\mathcal{N}(u_1)}$), who has two neighboring items (i.e., v_3 and v_4) is carried out in three steps: relevance scores $\pi^{m2}_{r_{u_1,v_3}}$ and $\pi^{m2}_{r_{u_1,v_4}}$ are first calculated by using the third formula in Eq. 12, which are then input into the second formula to make normalization and generate attention weights $\widetilde{\pi}^{m2}_{r_{u_1,v_3}}$ and $\widetilde{\pi}^{m2}_{r_{u_1,v_4}}$, finally these weights are fed into the first formula to obtain $\mathbf{u}^{m2}_{\mathcal{N}(u_1)}$.

Then we aggregate user-item interaction information, as shown in Eq. 13.

$$\mathbf{u}^{mt}_i[h] = f\left(\mathbf{u}^{mt}_i[h-1], \mathbf{u}^{mt}_{\mathcal{N}(u_i)}[h-1]\right) \tag{13}$$

After high-order propagation on each time-sliced graph $\mathcal{G}^{mt}$, we obtain the user-specific representation matrices under the granularity m, i.e., $\mathbf{U}^m = [\mathbf{U}^{m1};\mathbf{U}^{m2};\cdots;\mathbf{U}^{mT}]$. Analogously, we have the item-specific representation matrices, i.e., $\mathbf{V}^m = [\mathbf{V}^{m1};\mathbf{V}^{m2};\cdots;\mathbf{V}^{mT}]$. It is desirable to develop an efficacious method to model the dynamics of user and item representations across different time slices. The sequential modeling method RWKV exhibits powerful capability in parallelized training and efficient inference. RWKV integrates the advantages of RNNs and Transformers, eschewing quadratic-complexity dot-product attention mechanism and reformulating a linear attention mechanism. It consists of recurrent structures with time-mixing and channel-mixing blocks. For each user representation $\mathbf{u}^{mt}$ in $\mathbf{U}^{mt}$, the time-mixing block captures the temporal dependency in a recurrent fashion, as shown in Eq. 14.

$$\begin{cases} \mathbf{r}_t = \mathbf{W}_r(\boldsymbol{\mu}_r\mathbf{u}^{mt} + (1-\boldsymbol{\mu}_r)\mathbf{u}^{m(t-1)})) \\ \mathbf{k}_t = \mathbf{W}_k(\boldsymbol{\mu}_k\mathbf{u}^{mt} + (1-\boldsymbol{\mu}_k)\mathbf{u}^{m(t-1)})) \\ \mathbf{v}_t = \mathbf{W}_v(\boldsymbol{\mu}_v\mathbf{u}^{mt} + (1-\boldsymbol{\mu}_v)\mathbf{u}^{m(t-1)})) \\ \mathbf{wkv}_t = \frac{\sum_{i=1}^{t-1} e^{-(t-1-i)\mathbf{w}+\mathbf{k}_i}\mathbf{v}_i + e^{\mathbf{x}+\mathbf{k}_t}\mathbf{v}_t}{\sum_{i=1}^{t-1} e^{-(t-1-i)\mathbf{w}+\mathbf{k}_i} + e^{\mathbf{x}+\mathbf{k}_t}} \\ \mathbf{o}_t = \mathbf{W}_o(\sigma(\mathbf{r}_t) \odot \mathbf{wkv}_t) \end{cases}, \tag{14}$$

where $\mathbf{r}_t$ means the receptance of past information; $\mathbf{k}_t$ and $\mathbf{v}_t$ play the role of $\mathbf{K}$ and $\mathbf{V}$ in traditional attention, respectively; $\mathbf{wkv}_t$ represents the attention score; $\mathbf{o}_t$ denotes the output at time step t; $\boldsymbol{\mu}_r$, $\boldsymbol{\mu}_k$ and $\boldsymbol{\mu}_v$ are weights controling information updates, balancing current and previous inputs; $\mathbf{w}$ and $\mathbf{x}$ are time decay vectors; $\mathbf{W}_r$, $\mathbf{W}_k$, $\mathbf{W}_v$ and $\mathbf{W}_o$ are trainable weight matrices.

The channel-mixing block uses the squared activation function to increase nonlinearity, which can be formulated as

$$\begin{cases} \mathbf{r}_t = \mathbf{W}_r(\boldsymbol{\mu}_r \mathbf{u}^{mt} + (1-\boldsymbol{\mu}_r)\mathbf{u}^{m(t-1)})) \\ \mathbf{k}_t = \mathbf{W}_k(\boldsymbol{\mu}_k \mathbf{u}^{mt} + (1-\boldsymbol{\mu}_k)\mathbf{u}^{m(t-1)})) \\ \mathbf{o}_t = \sigma(\mathbf{r}_t) \odot (\mathbf{W}_v \cdot \max(\mathbf{k}_t, 0)^2) \end{cases}. \tag{15}$$

The process of correlating time-sliced representations described above can be summarized as

$$\overline{\mathbf{U}}^m = RWKV\left(\mathbf{U}^m; \Theta_U^m\right)|_{T+1}, \quad \overline{\mathbf{V}}^m = RWKV\left(\mathbf{V}^m; \Theta_V^m\right)|_{T+1}. \tag{16}$$

where $RWKV(\cdot)|_{T+1}$ learns the hidden representations $\overline{\mathbf{U}}^m$ and $\overline{\mathbf{V}}^m$ for the next recurrent time slice; Θ_U^m and Θ_V^m denote trainable model parameters.

After modeling time-dependent evolution under both coarse granularities and fine granularities, we have the respective user and item representation matrices $\overline{\mathbf{U}} = [\overline{\mathbf{U}}^1; \overline{\mathbf{U}}^2; \cdots; \overline{\mathbf{U}}^M]$ and $\overline{\mathbf{V}} = [\overline{\mathbf{V}}^1; \overline{\mathbf{V}}^2; \cdots; \overline{\mathbf{V}}^M]$. Instead of simple concatenation or summation, we adaptively aggregate multi-granularity representations with different weights (Eq. 17). These adaptive weights are learned from 1×1 convolution layers $Conv(\cdot)$ followed by the softmax operation [40].

$$\begin{cases} \widetilde{\mathbf{U}} = \sum_{m=1}^{M} \mathbf{w}_u^m \overline{\mathbf{U}}^m \\ \left[\mathbf{w}_u^1; \mathbf{w}_u^2; \cdots; \mathbf{w}_u^m\right] = Softmax\left(Conv\left(\overline{\mathbf{U}}\right)\right) \\ \widetilde{\mathbf{V}} = \sum_{m=1}^{M} \mathbf{w}_v^m \overline{\mathbf{V}}^m \\ \left[\mathbf{w}_v^1; \mathbf{w}_v^2; \cdots; \mathbf{w}_v^m\right] = Softmax(Conv(\overline{\mathbf{V}})) \end{cases}, \tag{17}$$

where $\widetilde{\mathbf{U}}$ and $\widetilde{\mathbf{V}}$ denote the fusion of multi-granularity representations; $\mathbf{w}_u^m$ and $\mathbf{w}_v^m$ refer to the weights of the user and item for a given granularity, respectively.

As a result, we obtain the learned user's representation $\mathbf{u}_{uv}$ and item's representation $\mathbf{v}_{uv}$ from the user-item graph.

3.4 Heterogeneous networks information fusion

After obtaining the user-side and item-side representations from heterogeneous networks, the ensuing task is how to trade off the contributions from different perspectives [9, 16]. To further inspire the strength of DH-GCN, we implement a sophisticated fusion through a learnable convolutional gate (Eq. 18).

$$\begin{cases} \widetilde{\mathbf{u}} = \sigma(Conv(\mathbf{u}_{uu} + \mathbf{u}_{uv})) \odot \mathbf{u}_{uu} + (1 - \sigma(Conv\left(\mathbf{u}_{uu} + \mathbf{u}_{uv}\right))) \odot \mathbf{u}_{uv} \\ \widetilde{\mathbf{v}} = \sigma(Conv(\mathbf{e}_{kg} + \mathbf{v}_{uv})) \odot \mathbf{e}_{kg} + (1 - \sigma(Conv\left(\mathbf{e}_{kg} + \mathbf{v}_{uv}\right))) \odot \mathbf{v}_{uv} \end{cases} \tag{18}$$

To be specific, the first summation operation of two-aspect representations refers to the initial integration. The convolutional gate regulates the amount of information between different channels and transports information in a flexible and efficient manner [41, 55]. The sigmoid function $\sigma(\cdot)$ returns the weights between 0 and 1, enhancing the model's non-linear capability. Consequently, we obtain the final user representation $\widetilde{\mathbf{u}}$ and item representation $\widetilde{\mathbf{v}}$.

3.5 CTR Prediction and loss function

DH-GCN generates user and item representations learned from the knowledge graph, user-user graph, and multi-granularity time-sliced user-item graphs. Finally, the sigmoid function predicts the probability of the user clicking the item (Eq. 19).

$$\widehat{y}_{uv} = \sigma\left(\widetilde{\mathbf{u}}, \widetilde{\mathbf{v}}\right) \tag{19}$$

During the training stage, the loss function with a negative sampling strategy is given as

$$\mathcal{L} = \sum_{u \in \mathcal{U}} \left(\sum_{v : y_{uv}=1} \mathcal{J}\left(y_{uv}, \widehat{y}_{uv}\right) - \sum_{i=1}^{N^u} \mathbb{E}_{v_i \sim P(v_i)} \mathcal{J}\left(y_{uv_i}, \widehat{y}_{uv_i}\right) \right) + \lambda \parallel \mathcal{F} \parallel_2^2, \tag{20}$$

where $\mathcal{L}$ denotes the complete loss function, $\mathcal{J}$ denotes cross-entropy loss, P denotes a negative uniformed sampling distribution, N^u denotes the number of each user's negative samples, and λ controls the L 2 regularization.

4 Experiments

In this section, we present an experimental evaluation on three public datasets to compare our proposed model (i.e., DH-GCN) with ten state-of-the-art baselines.

4.1 Datasets

We utilize three datasets from different scenarios to evaluate the performance of DH-GCN.

Last.FM[1] contains music artists' listening records from more than 1 thousand users from the Last.FM online music system. Each user is associated with a list of their most popular artists and the corresponding play counts. Moreover, the dataset includes user-generated tags that can be utilized to construct content vectors.

MovieLens-1M[2] is a widely used dataset for movie recommendation. It contains approximately 0.75 million ratings from 6036 users for 2245 movies from the MovieLens website.

E-Commerce[3] contains about twenty thousand users' browsing records on around sixteen thousand items and corresponding attributes. Collected from a prominent multi-category online store between October 2019 and April 2020, this dataset captures various user behaviors including "view", "cart", "remove_from_cart" or "purchase".

[1] https://grouplens.org/datasets/hetrec-2011/.

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

[3] https://www.kaggle.com/datasets/dschettler8845/recsys-2020-ecommerce-dataset?select=val.parquet.

We transform datasets with explicit feedback into implicit feedback (0 for no click and 1 for click). For each dataset, interaction records of each user are split into 6/2/2 as the train/validation/test ratio.

For datasets Last.FM and MovieLens-1M, we construct knowledge graphs following [59]. A subset of triplets from the Microsoft Satori knowledge graph is selected based on specific criteria such as relation names containing relevant keywords ("movie" or "musician") and a confidence level greater than 0.9. Then, Satori IDs of valid items are collected by matching their names with the tail of triplets. Items with no matches or multiple matches are excluded for simplicity. After obtaining the item IDs, they are matched with the head of all triplets in the sub-knowledge graph, and well-matched triplets are selected as the final knowledge graph for each dataset. For the dataset E-Commerce, we construct the knowledge graph on items and their attributes, such as category and brand. The statistics of the three processed datasets are shown in Table 2.

4.2 Baselines

To evaluate the effectiveness of our proposed method, we compare DH-GCN with the following state-of-the-art baselines:

BPR-MF [49] combines the strengths of matrix factorization with personalized ranking, learning by maximizing posterior probabilities derived from Bayesian analysis. It can effectively capture users' relative preferences by optimizing pairwise ranking loss.

CKE [76] utilizes TransR to learn knowledge graph embeddings within a unified Bayesian framework. This method allows for the joint learning of knowledge graph embeddings and collaborative filtering signals.

RippleNet [57] exploits the knowledge graph as side information and simulates user preference propagation in the knowledge graph similar to water ripples.

KGAT [61] investigates the combination of the knowledge graph and user-item graph. It employs the attention mechanism to discriminate the importance of neighboring entities during propagation.

KGCN [59] captures users' potential interests, recursively propagating high-order relations. It aggregates and incorporates neighborhood information with bias in the knowledge graph.

Table 2 Basic statistics of the datasets

Statistics		Last.FM	MovieLens-1 M	E-Commerce
User-item interaction	#Users	1,348	6,036	19,962
	#Items	8,443	2,445	15,877
	#Interactions	62,201	753,772	622,004
Knowledge graph	#Entities	9,366	90,279	15,877
	#Relations	60	12	6
	#Triplets	31,036	1,241,995	135,354

DIN [80] overcomes the limitations of a fixed-length user interest representation by adaptively learning representations based on historical behaviors, tailored to each candidate item. This method enhances the model's expressive ability, capturing nuances and relevancies specific to different items.

DIEN [79] points out little works capture the changing trend of interest. It utilizes GRU and an attention-based update gate to capture diverse user interests and their evolving patterns and activate relative interests.

SDIM [6] is an efficient end-to-end method for capturing long-term user behaviors. By sampling from various hash functions, it creates hash signatures for items in the user behavior sequence and target items, directly capturing user interest through shared hash signatures.

DisenCTR [62] introduces a novel method for dynamic CTR prediction by borrowing the concept of disentangled representation learning. Contrary to condensing various interests into a single representation, it learns disentangled representations that accurately capture multi-aspect and evolving interests.

FinalMLP [43] shows that a well-tuned model with two parallel MLPs can achieve surprisingly good performance. The integration of multi-view feature gating and bilinear interaction fusion facilitates effective stream-level interactions.

4.3 Experimental settings

We evaluate the CTR prediction model using three metrics: Area Under ROC Curve (AUC), accuracy (ACC), and F1 Score (F1). AUC represents the area under the ROC (Receiver Operating Characteristic) curve, which is a plot of the true positive rate against the false positive rate at various threshold settings, with higher values indicating better model performance. ACC measures the proportion of instances that are correctly classified among the total instances. F1 is calculated as the harmonic mean of precision and recall, providing a balance between the two metrics.

Each experiment conducted in this study is repeated 5 times and the average performance is reported. All models are trained with Adam Optimizer. The selection of hyper-parameters depends on optimization with AUC on the validation set.

The learning rate is tuned amongst the set of $\{10^{-2}, 5\times10^{-3}, 10^{-3}, 5\times10^{-4}, 10^{-4}, 5\times10^{-5}\}$ during the training process. The coefficient of L 2 regularization is searched from $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$. The batch size is chosen from {64, 256, 1024}. The segmentation of time intervals is selected from {15 days, 1 month, 2 months, …, 1 year}. The multi-head self-attention is configured with 8 heads. The embedding size is fixed to 16. The dropout rate is set to 0.1. The function $g(\cdot)$ is set as the inner product.

4.4 Model comparison

The performance of our model and baselines are presented in Table 3. Note that each bold value in Table 3 represents the best model performance in terms of an evaluation metric (e.g., AUC, ACC and F1). From Table 3, we summarize the following observations.

Table 3 Model comparison in CTR prediction

Model	Last.FM			MovieLens-1 M			E-Commerce		
	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1
BPR-MF	0.820	0.756	0.833	0.819	0.757	0.755	0.852	0.784	0.821
CKE	0.855	0.792	0.821	0.849	0.765	0.757	0.881	0.799	0.836
RippleNet	0.842	0.792	0.818	0.848	0.762	0.747	0.876	0.810	0.837
KGAT	0.858	0.837	0.889	0.773	0.690	0.717	0.807	0.705	0.784
KGCN	0.837	0.814	0.873	0.849	0.769	0.760	0.870	0.798	0.826
DIN	0.797	0.729	0.795	0.857	0.770	0.760	0.856	0.782	0.813
DIEN	0.801	0.748	0.812	0.858	0.767	0.756	0.868	0.797	0.825
SDIM	0.828	0.788	0.842	0.795	0.757	0.761	0.864	0.782	0.818
DisenCTR	0.774	0.711	0.728	0.822	0.761	0.762	0.878	0.790	0.796
FinalMLP	0.852	0.824	0.871	0.811	0.755	0.763	0.869	0.795	0.824
DH-GCN	0.873	0.858	0.904	0.874	0.790	0.786	0.896	0.821	0.855
Improve	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.01	0.02

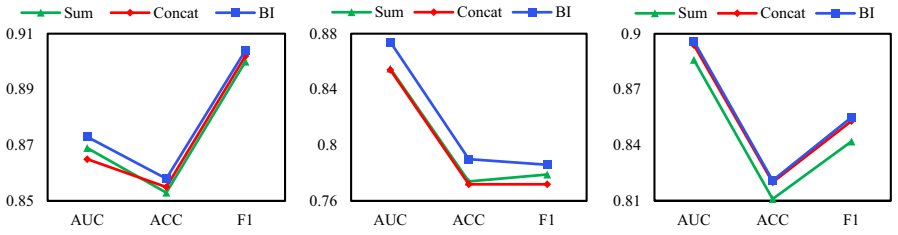
(1) DH-GCN achieves the best performance on these three datasets. As an illustration, DH-GCN outperforms baselines by 0.02 in AUC on three datasets. Such performance gap can be attributed to the joint exploration of dynamic interactions and heterogeneous information fusion. What's more, DH-GCN yields larger improvements on MovieLens-1M and E-Commerce than Last.FM, indicating its capability of handling sparse scenarios effectively.

(2) In general, the matrix factorization method BPR-MF shows relatively unsatisfactory performance. This is in conformity with our expectations, as it makes use of limited information.

(3) Among the baselines, models leveraging heterogeneous information exhibit superior overall performance, highlighting the importance of integrating knowledge from multiple perspectives. Relying on a single source of information often fails to comprehensively capture user behavior patterns and preferences, especially when behavior data is sparse or cold-start issues arise, thereby limiting the model's predictive performance.

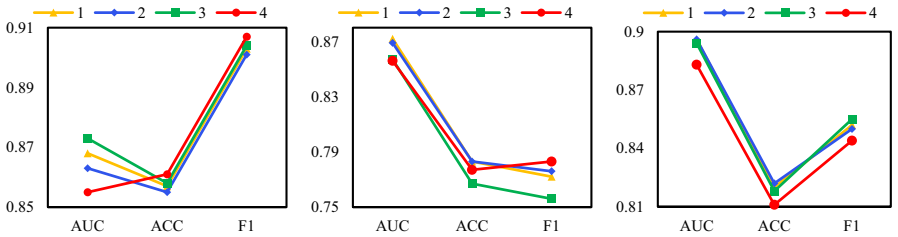
(4) From an overall perspective, methods that take into account historical user behaviors tend to perform better, suggesting the rationality of exploring dynamics underlying user-item interactions. DIN and DIEN have a similar structure, with the latter being superior. This can be attributed to effectively capturing sequential interests.

(5) Compared with other baselines, FinalMLP performs relatively well, indicating the effectiveness of explicitly modeling feature interactions.



(a) Last.FM (b) MovieLens-1M (c) E-Commerce

Fig. 2 Influences of aggregators



(a) Last.FM (b) MovieLens-1M (c) E-Commerce

Fig. 3 Influences of the number of GCN layers

4.5 Hyper-parameter analysis

In this section, we evaluate the influences of hyper-parameters on prediction performance, including aggregators, the number of GCN layers, and the neighbor size.

4.5.1 Influences of aggregators

We explore how different aggregators affect the results. As shown in Fig. 2, the bi-interaction (BI) aggregator yields the best performance. This could be attributed to the fact that single-interaction aggregators are insufficient to aggregate self and context representations. Our results prove that combining single-interaction operators can enhance the model's learning capability.

4.5.2 Influences of the number of DH-GCN layers

We evaluate the effect of varying the number of propagation layers from 1 to 4 (Fig. 3). We observe that by incorporating multiple layers of embedding propagation, DH-GCN can capture information from a wider range of neighbors and higher-order connections explicitly. However, continuously increasing stacked layers may not

always lead to better performance. This is consistent with our intuition that an excessively deep receptive field may inevitably introduce some irrelevant noise. Thus, a modest number of layers can often suffice to achieve the desired performance.

4.5.3 Influences of the neighbor size

We assess how the size of incorporating neighboring entities influences the prediction performance. We can observe that DH-GCN performs best when the neighbor size is set from 8 to 32 in most cases as shown in Fig. 4. This illustrates that increasing the neighbor size appropriately provides sufficient information for exploring latent connections and extending users' potential interests.

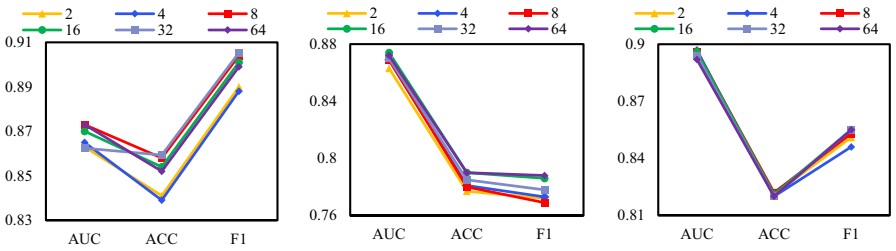
4.6 Ablation study

To get deep insights into design rationality, we clarify how different key components influence the model performance, including the impacts of heterogeneous information, graph representation learning, and multi-granularity time-sliced user-item graphs.

4.6.1 Influences of heterogeneous information

We evaluate and analyze the efficacy of incorporating heterogeneous information. Specifically, we consider the following model variants of DH-GCN: "w/o UV" denotes the absence of the user-item graph, "w/o KG" removes the external knowledge graph about items, and "w/o UU" means not injecting the user-user graph with similar preferences.

By comparing the performance of model variants in Fig. 5, we can observe that removing any aspect information results in performance decline. In addition to basic user-item interactions, heterogeneous information networks complement each other and bring synergistic effects.



(a) Last.FM (b) MovieLens-1M (c) E-Commerce

Fig. 4 Influences of the neighbor size

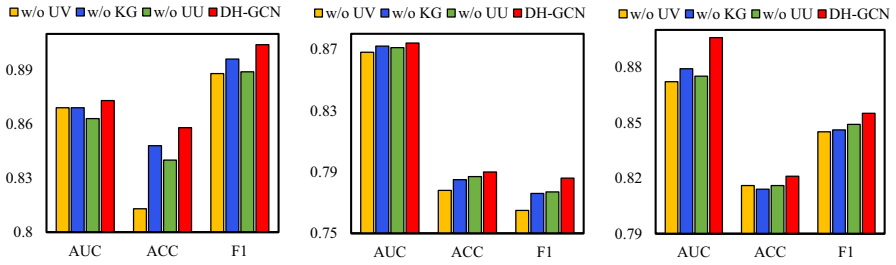


Fig. 5 Influences of heterogeneous information

4.6.2 Influences of graph representation learning

We conduct an ablation study to validate whether graph representation learning can boost performance. First, we investigate the necessity of multi-head self-attention in each network. "w/o MA" represents not utilizing multi-head self-attention to distinguish the importance of neighbors. Second, we assess the contributions of the joint learning component. "w/o G2G" replaces the graph-to-graph learning method with isolated individual graph learning. "w/o GCN" means replacing GCN with simple mean operations to aggregate neighbors.

From Fig. 6, we have the following observations. First, model variants do not suffer severe performance degradation, revealing the superiority and stability of our architecture. Overall, graph-to-graph learning contributes more to the model's accuracy. Second, according to the results of "w/o MA", multi-head self-attention shows a positive impact, proving indispensable for discerning the importance of neighbors. Third, during the graph representation learning phase, we verify that the graph-to-graph learning component is capable of correlating intricate relations across heterogeneous networks and that GCN allows for capturing higher-order connectivities.

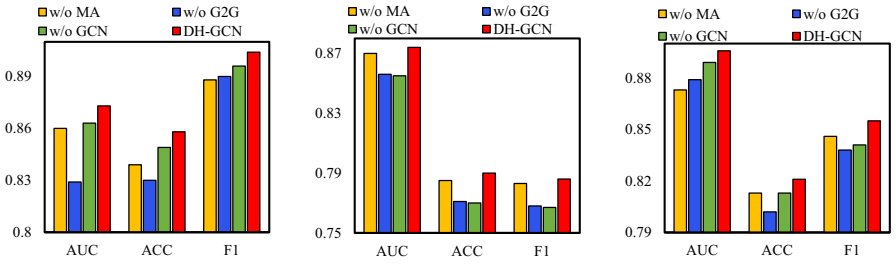


Fig. 6 Influences of graph representation learning

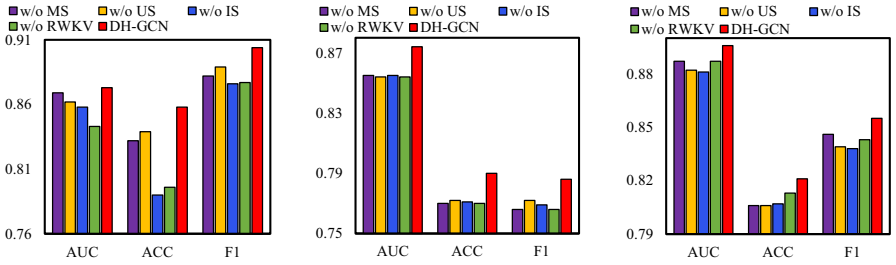
4.6.3 Influences of multi-granularity time-sliced user-item graphs

We explore whether multi-granularity time-sliced user-item graph representation learning can generate more expressive representations. We conduct an ablation study on the following variants: "w/o MS" represents modeling the whole timeline instead of time slices, "w/o US" removes the modeling of user-side dynamic interactions, "w/o IS" removes the counterpart item-side dynamic user-item interactions, "w/o RWKV" means not utilizing the sequential modeling method RWKV to capture sequential patterns and replacing it with the mean operations on the representations generated from each time slices.

From Fig. 7, we summarize the following observations. First, modeling multi-granularity time-sliced user-item graphs is better than a whole graph. Second, "w/o US" and "w/o IS" are inferior to the complete version, conforming to the efficacy of capturing the dynamics of both the user and item sides. Third, "w/o RWKV" validates that correlating sequential representations can boost performance. The possible reason is that RWKV can discover the dependency underlying user-item interactions.

5 Conclusion and future work

In this paper, we propose dynamic heterogeneous graph convolutional networks (DH-GCN) integrating dynamic user-item interactions and heterogeneous information for CTR prediction in recommender systems. We design a graph-to-graph learning method with a sharing neighborhood space to distill complex correlations across heterogeneous information networks. Moreover, we develop a multi-granularity time-sliced user-item graph representation learning method to simultaneously capture structural and temporal dependencies underlying user-item interactions. Experiments conducted on three public datasets demonstrate the superiority of DH-GCN; meanwhile, the usefulness of graph-to-graph learning in integrating heterogeneous information and the effectiveness of multi-granularity time-sliced graph representation learning in capturing dynamic user preferences are verified.



(a) Last.FM (b) MovieLens-1M (c) E-Commerce

Fig. 7 Influences of multi-granularity time-sliced user-item graphs

In future research, we intend to delve into real-time updates of knowledge graph, ensuring that prediction models are continuously informed by the latest semantics. Moreover, it is also worthwhile exploring roles of multi-modal information for CTR prediction. With the increasing availability of multimedia contents (e.g., image, audio and video), there is a growing interest in incorporating multi-modal features to perform high-quality representation learning. Furthermore, inspired by the developments of self-supervised pre-training in natural language processing and computer vision, it is a promising perspective to perform data augmentations by randomly omitting certain items in user-item interactions to generate supervision signals and enhance model's predictive capability and adaptability.

Acknowledgements We are thankful to the editor and anonymous reviewers who provided valuable suggestions that led to a considerable improvement in the organization and presentation of this manuscript. This work is partially supported by the National Natural Science Foundation of China (Nos. 72171093, 72172092), Shanghai Key Laboratory of Brain-Machine Intelligence for Information Behavior (22dz2261100), and the Fundamental Research Funds for the Central Universities (No. 41005067).

Author contributions Conceptualization: Yanwu Yang, Baojun Ma; Methodology: Ying Jin, Yanwu Yang; Formal analysis and investigation: Ying Jin; Writing—original draft preparation: Ying Jin, Yanwu Yang; Writing—review and editing: Baojun Ma; Funding acquisition: Yanwu Yang, Baojun Ma; Resources: Yanwu Yang, Baojun Ma; Supervision: Yanwu Yang, Baojun Ma.

Data availability The Last.FM dataset is available at https://grouplens.org/datasets/hetrec-2011/. The MovieLens-1 M dataset is available at https://grouplens.org/datasets/movielens/1m/. The E-Commerce dataset is available at https://www.kaggle.com/datasets/dschettler8845/recsys-2020-ecommerce-dataset?select=val.parquet.

Declarations

Conflict of interest On behalf of all authors, the authors state that there is no conflict of interest.

References

1. Abdalla, H. B., Gheisari, M., & Awlla, A. H. (2024). Hybrid self-attention BiLSTM and incentive learning-based collaborative filtering for E-commerce recommendation system. *Electronic Commerce Research*. https://doi.org/10.1007/s10660-024-09888-5
2. Ali, Z., Huang, Y., Ullah, I., Feng, J., Deng, C., Thierry, N., Khan, A., Jan, A. U., Shen, X., Rui, W., & Qi, G. (2022). Deep learning for medication recommendation: A systematic survey. *Data Intelligence, 5*(2), 303–354.
3. Bahi, A., Gasmi, I., Bentrad, S., & Khantouchi, R. (2024). MycGNN: Enhancing recommendation diversity in E-commerce through mycelium-inspired graph neural network. *Electronic Commerce Research*. https://doi.org/10.1007/s10660-024-09911-9
4. Bazargani, M., et al. (2024). Group deep neural network approach in semantic recommendation system for movie recommendation in online networks. *Electronic Commerce Research*. https://doi.org/10.1007/s10660-024-09897-4
5. Cao, X., Shi, Y., Yu, H., Wang, J., Wang, X., Yan, Z., & Chen, Z. (2021). DEKR: Description enhanced knowledge graph for machine learning method recommendation. Paper presented at the Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021).
6. Cao, Y., Zhou, X., Feng, J., Huang, P., Xiao, Y., Chen, D., & Chen, S. (2022). Sampling is all you need on modeling long-term user behaviors for CTR prediction. Paper presented at the Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM 2022), Atlanta, GA, USA.

7. Chan, T. H., Wong, C. H., Shen, J., & Yin, G. (2023). Source-aware embedding training on heterogeneous information networks. *Data Intelligence, 5*(3), 611–635.
8. Chen, C., Cai, F., Hu, X., Chen, W., & Chen, H. (2021). HHGN: A hierarchical reasoning-based heterogeneous graph neural network for fact verification. *Information Processing & Management, 58*(5), Article 102659.
9. Chen, Q., Zhao, H., Li, W., Huang, P., & Ou, W. (2019). Behavior sequence transformer for e-commerce recommendation in Alibaba. Paper presented at the Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data (DLP-KDD 2019), Anchorage, AK, USA.
10. Chen, T., Yin, H., Nguyen, Q. V. H., Peng, W. C., Li, X., & Zhou, X. (2020). Sequence-aware factorization machines for temporal predictive analytics. Paper presented at the 2020 IEEE 36th International Conference on Data Engineering (ICDE 2020), Dallas, TX, USA.
11. Chen, X., Tang, Q., Hu, K., Xu, Y., Qiu, S., Cheng, J., & Lei, J. (2022). Hybrid CNN based attention with category prior for user image behavior modeling. Paper presented at the Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2022), Madrid, Spain.
12. Chen, Y., Yang, Y., Wang, Y., Bai, J., Song, X., & King, I. (2022). Attentive knowledge-aware graph convolutional networks with collaborative guidance for personalized recommendation. Paper presented at the 2022 IEEE 38th International Conference on Data Engineering (ICDE 2022), Kuala Lumpur, Malaysia.
13. Chen, Z., Zhang, W., Yan, J., Wang, G., & Wang, J. (2021). Learning dual dynamic representations on time-sliced user-item interaction graphs for sequential recommendation. Paper presented at the Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM 2021).
14. Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., & Ispir, M. (2016). Wide & deep learning for recommender systems. Paper presented at the Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS 2016), Boston, MA, USA.
15. Fan, S., Zhu, J., Han, X., Shi, C., Hu, L., Ma, B., & Li, Y. (2019). Metapath-guided heterogeneous graph neural network for intent recommendation. Paper presented at the Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2019), Anchorage, AK, USA.
16. Feng, Y., Lv, F., Hu, B., Sun, F., Kuang, K., Liu, Y., Liu, Q., & Ou, W. (2020). MTBRN: Multiplex target-behavior relation enhanced network for click-through rate prediction. Paper presented at the Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM 2020).
17. Feng, Y., Lv, F., Shen, W., Wang, M., Sun, F., Zhu, Y., & Yang, K. (2019). Deep session interest network for click-through rate prediction. Paper presented at the Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI 2019), Macao, China.
18. Gao, H., Kong, D., Lu, M., Bai, X., & Yang, J. (2018). Attention convolutional neural network for advertiser-level click-through rate forecasting. Paper presented at the Proceedings of the 2018 World Wide Web Conference (WWW 2018), Lyon, France.
19. Gao, Y., Li, Y.-F., Lin, Y., Gao, H., & Khan, L. (2020). Deep learning on knowledge graph for recommender system: A survey. arXiv:2004.00387.
20. Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks, 18*(5–6), 602–610.
21. Guo, H., Tang, R., Ye, Y., Li, Z., & He, X. (2017). DeepFM: A factorization-machine based neural network for CTR prediction. Paper presented at the Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI 2017), Melbourne, Australia.
22. Guo, W., Su, R., Tan, R., Guo, H., Zhang, Y., Liu, Z., Tang, R., & He, X. (2021). Dual graph enhanced embedding neural network for CTR prediction. Paper presented at the Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2021).
23. Guo, X., Lin, W., Li, Y., Liu, Z., Yang, L., Zhao, S., & Zhu, Z. (2020). DKEN: Deep knowledge-enhanced network for recommender systems. *Information Sciences, 540*, 263–277.
24. Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive representation learning on large graphs. Paper presented at the Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, California, USA.

25. He, J., Mei, G., Xing, F., Yang, X., Bao, Y., & Yan, W. (2020). DADNN: Multi-scene CTR prediction via domain-aware deep neural network. arXiv:2011.11938.
26. He, T., Li, K., Chen, S., Wang, H., Liu, Q., Wang, X., & Wang, D. (2023). DMBIN: A Dual multi-behavior interest network for click-through rate prediction via contrastive learning. Paper presented at the Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2023), Taipei, Taiwan.
27. Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. Paper presented at the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, HI, USA.
28. Huang, W., Wu, J., Song, W., & Wang, Z. (2022). Cross attention fusion for knowledge graph optimized recommendation. *Applied Intelligence, 52*(9), 10297–10306.
29. Ji, G., He, S., Xu, L., Liu, K., & Zhao, J. (2015). Knowledge graph embedding via dynamic mapping matrix. Paper presented at the Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2015), Beijing, China.
30. Jin, Y., & Yang, Y. (2025). A survey on knowledge graph-based click-through rate prediction. *Expert Systems with Applications*, 281, Article 127501.
31. Juan, Y., Zhuang, Y., Chin, W.-S., & Lin, C.-J. (2016). Field-aware factorization machines for CTR prediction. Paper presented at the Proceedings of the 10th ACM Conference on Recommender Systems (RecSys 2016), Boston, Massachusetts, USA.
32. Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. Paper presented at the 5th International Conference on Learning Representations (ICLR 2017), Toulon, France.
33. Kumar, R., Naik, S. M., Naik, V. D., Shiralli, S., Sunil, V. G., & Husain, M. (2015). Predicting clicks: CTR estimation of advertisements using logistic regression classifier. Paper presented at the 2015 IEEE International Advance Computing Conference (IACC 2015), Banglore, India.
34. Li, C., Liu, Z., Wu, M., Xu, Y., Zhao, H., Huang, P., Kang, G., Chen, Q., Li, W., & Lee, D. L. (2019). Multi-interest network with dynamic routing for recommendation at Tmall. Paper presented at the Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM 2019), Beijing, China.
35. Li, F., Yan, B., Long, Q., Wang, P., Lin, W., Xu, J., & Zheng, B. (2021). Explicit semantic cross feature learning via pre-trained graph neural networks for CTR prediction. Paper presented at the Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021).
36. Li, X., Wang, C., Tong, B., Tan, J., Zeng, X., & Zhuang, T. (2020). Deep time-aware item evolution network for click-through rate prediction. Paper presented at the Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM 2020), Virtual Event, Ireland.
37. Li, Y., Guo, X., Lin, W., Zhong, M., Li, Q., Liu, Z., Zhong, W., & Zhu, Z. (2023). Learning dynamic user interest sequence in knowledge graphs for click-through rate prediction. *IEEE Transactions on Knowledge and Data Engineering, 35*(1), 647–657.
38. Li, Z., Cui, Z., Wu, S., Zhang, X., & Wang, L. (2019). Fi-GNN: Modeling feature interactions via graph neural networks for CTR prediction. Paper presented at the Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM 2019), Beijing, China.
39. Ling, X., Deng, W., Gu, C., Zhou, H., Li, C., & Sun, F. (2017). Model ensemble for click prediction in Bing search ads. Paper presented at the Proceedings of the 26th International Conference on World Wide Web Companion (WWW 2017), Perth, Australia.
40. Liu, S., Huang, D., & Wang, Y. (2019). Learning spatial fusion for single-shot object detection. arXiv:1911.09516.
41. Liu, Y., Li, B., Zang, Y., Li, A., & Yin, H. (2021). A knowledge-aware recommender with attention-enhanced dynamic convolutional network. Paper presented at the Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM 2021), Virtual Event, Queensland, Australia.
42. Lyu, S., Cai, H., Zhang, C., Ling, S., Shen, Y., Zeng, X., Gu, J., Zhang, G., & Zhang, H. (2022). See clicks differently: Modeling user clicking alternatively with multi classifiers for CTR prediction. Paper presented at the Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM 2022), Atlanta, GA, USA.

43. Mao, K., Zhu, J., Su, L., Cai, G., Li, Y., & Dong, Z. (2023). FinalMLP: An enhanced two-stream MLP model for CTR prediction. Paper presented at the Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2023), Washington, DC, USA.
44. Noorian, A. (2024). Integrating user reviews and risk factors from social networks in a multi-objective recommender system. *Electronic Commerce Research*. https://doi.org/10.1007/s10660-024-09944-0
45. Ong, R. K., Qiu, W., & Khong, A. W. H. (2023). Quad-tier entity fusion contrastive representation learning for knowledge aware recommendation system. Paper presented at the Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM 2023), Birmingham, United Kingdom.
46. Peng, B., Alcaide, E., Anthony, Q. G., Albalak, A., Arcadinho, S., Cao, H., Cheng, X., Chung, M., Grella, M., Kranthikiran, G., He, X., Hou, H., Kazienko, P., Kocoń, J., Kong, J., Koptyra, B., Lau, H., Mantri, K. S. I., Mom, F., Saito, A., Tang, X., Wang, B., Wind, J. S., Wozniak, S., Zhang, R., Zhang, Z., Zhao, Q., Zhou, P., Zhu, J., & Zhu, R. (2023). RWKV: Reinventing RNNs for the transformer era. Paper presented at the Findings of the Association for Computational Linguistics: EMNLP 2023 (EMNLP 2023), Singapore.
47. Peng, W., Cheng, J., Wang, Z., Zhao, M., & Wu, X. (2023). DS-KGAT: A deep session GAT with knowledge enhancement for CTR prediction. Paper presented at the 2023 IEEE 3rd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA 2023), Chongqing, China.
48. Qu, Y., Bai, T., Zhang, W., Nie, J., & Tang, J. (2019). an end-to-end neighborhood-based interaction model for knowledge-enhanced recommendation. Paper presented at the Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data (DLP-KDD 2019), Anchorage, AK, USA.
49. Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2009). BPR: Bayesian personalized ranking from implicit feedback. Paper presented at the Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI 2009), Montreal, Quebec, Canada.
50. Song, C., Shu, K., & Wu, B. (2021). Temporally evolving graph neural network for fake news detection. *Information Processing & Management, 58*(6), Article 102712.
51. Song, W., Shi, C., Xiao, Z., Duan, Z., Xu, Y., Zhang, M., & Tang, J. (2019). AutoInt: Automatic feature interaction learning via self-attentive neural networks. Paper presented at the Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM 2019), Beijing, China.
52. Tang, X., Wang, T., Yang, H., & Song, H. (2019). AKUPM: Attention-enhanced knowledge-aware user preference model for recommendation. Paper presented at the Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2019), Anchorage, AK, USA.
53. Tao, Z., Wang, X., He, X., Huang, X., & Chua, T.-S. (2020). HoAFM: A high-order attentive factorization machine for CTR prediction. *Information Processing & Management, 57*(6), Article 102076.
54. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. Paper presented at the Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, California, USA.
55. Wang, C., Zhu, Y., Liu, H., Ma, W., Zang, T., & Yu, J. (2021). Enhancing user interest modeling with knowledge-enriched itemsets for sequential recommendation. Paper presented at the Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM 2021).
56. Wang, F., Gu, H., Li, D., Lu, T., Zhang, P., & Gu, N. (2023). Towards deeper, lighter and interpretable cross network for CTR prediction. Paper presented at the Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM 2023), Birmingham, United Kingdom.
57. Wang, H., Zhang, F., Wang, J., Zhao, M., Li, W., Xie, X., & Guo, M. (2018). RippleNet: Propagating user preferences on the knowledge graph for recommender systems. Paper presented at the Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM 2018), Torino, Italy.
58. Wang, H., Zhang, F., Xie, X., & Guo, M. (2018). DKN: Deep knowledge-aware network for news recommendation. Paper presented at the Proceedings of the 2018 World Wide Web Conference (WWW 2018), Lyon, France.

59. Wang, H., Zhao, M., Xie, X., Li, W., & Guo, M. (2019). Knowledge graph convolutional networks for recommender systems. Paper presented at the Proceedings of the 2019 World Wide Web Conference (WWW 2019), San Francisco, CA, USA.
60. Wang, R., Fu, B., Fu, G., & Wang, M. (2017). Deep & cross network for Ad click predictions. Paper presented at the Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2017), Halifax, NS, Canada.
61. Wang, X., He, X., Cao, Y., Liu, M., & Chua, T.-S. (2019). KGAT: Knowledge graph attention network for recommendation. Paper presented at the Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2019).
62. Wang, Y., Qin, Y., Sun, F., Zhang, B., Hou, X., Hu, K., Cheng, J., Lei, J., & Zhang, M. (2022). DisenCTR: Dynamic graph-based disentangled representation for click-through rate prediction. Paper presented at the Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2022), Madrid, Spain.
63. Wang, Y., Zhang, R., Yang, Q., Zhou, Q., Zhang, S., Fan, Y., Huang, L., Li, K., & Zhou, F. (2024). FairCare: Adversarial training of a heterogeneous graph neural network with attention mechanism to learn fair representations of electronic health records. *Information Processing & Management, 61*(3), Article 103682.
64. Wang, Z., & Li, Y. (2023). Knowledge graph-based graph neural network models for multi-perspective modeling of group preferences. *Electronic Commerce Research*. https://doi.org/10.1007/s10660-023-09771-9
65. Wang, Z., Lin, G., Tan, H., Chen, Q., & Liu, X. (2020). CKAN: Collaborative knowledge-aware attentive network for recommender systems. Paper presented at the Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2020), China.
66. Wu, C., Wu, F., Lyu, L., Huang, Y., & Xie, X. (2022). FedCTR: Federated native Ad CTR prediction with cross-platform user behavior data. *ACM Transactions on Intelligent Systems and Technology, 13*(4), 1–19.
67. Xia, L., Huang, C., Xu, Y., Dai, P., Zhang, X., Yang, H., Pei, J., & Bo, L. (2021). Knowledge-enhanced hierarchical graph transformer network for multi-behavior recommendation. Paper presented at the Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI 2021).
68. Xiao, J., Ye, H., He, X., Zhang, H., Wu, F., & Chua, T.-S. (2017). Attentional factorization machines: Learning the weight of feature interactions via attention networks. Paper presented at the Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI 2017), Melbourne, Australia.
69. Yang, X., Deng, T., Tan, W., Tao, X., Zhang, J., Qin, S., & Ding, Z. (2019). Learning compositional, visual and relational representations for CTR prediction in sponsored search. Paper presented at the Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM 2019), Beijing, China.
70. Yang, Y., Huang, C., Xia, L., Liang, Y., Yu, Y., & Li, C. (2022). Multi-behavior hypergraph-enhanced transformer for sequential recommendation. Paper presented at the Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2022), Washington DC, USA.
71. Yang, Y., & Zhai, P. (2022). Click-through rate prediction in online advertising: A literature review. *Information Processing & Management, 59*(2), Article 102853.
72. Yang, Y., Zhang, C., Zhao, K., & Wang, Q. (2023). The shifting role of information processing and management in interdiscipline development: From a collection of tools to a crutch? *Information Processing & Management, 60*(4), Article 103388.
73. Yang, Y., Zhao, K., Zeng, D. D., & Jansen, B. J. (2022). Time-varying effects of search engine advertising on sales: An empirical investigation in E-commerce. *Decision Support Systems, 163*, Article 113843.
74. Zeng, D., Liu, Y., Yan, P., & Yang, Y. (2021). Location-aware real-time recommender systems for brick-and-mortar retailers. *INFORMS Journal on Computing, 33*(4), 1608–1623.
75. Zhai, P., Yang, Y., & Zhang, C. (2023). Causality-based CTR prediction using graph neural networks. *Information Processing & Management, 60*(1), Article 103137.
76. Zhang, F., Yuan, N. J., Lian, D., Xie, X., & Ma, W.-Y. (2016). Collaborative knowledge base embedding for recommender systems. Paper presented at the Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2016).

77. Zhang, J., Lin, F., Yang, C., & Wang, W. (2022). Deep multi-representational item network for CTR prediction. Paper presented at the Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2022), Madrid, Spain.
78. Zhang, K., Qian, H., Cui, Q., Liu, Q., Li, L., Zhou, J., Ma, J., & Chen, E. (2021). Multi-interactive attention network for fine-grained feature learning in CTR prediction. Paper presented at the Proceedings of the 14th ACM International Conference on Web Search and Data Mining (WSDM 2021), Israel.
79. Zhou, G., Mou, N., Fan, Y., Pi, Q., Bian, W., Zhou, C., Zhu, X., & Gai, K. (2019). Deep interest evolution network for click-through rate prediction. Paper presented at the Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence (AAAI 2019/IAAI 2019/EAAI 2019), Honolulu, Hawaii, USA.
80. Zhou, G., Zhu, X., Song, C., Fan, Y., Zhu, H., Ma, X., Yan, Y., Jin, J., Li, H., & Gai, K. (2018). Deep interest network for click-through rate prediction. Paper presented at the Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2018), London, United Kingdom.
81. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond Efficient transformer for long sequence time-series forecasting. Paper presented at the Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI 2021).
82. Zhu, Z., Zhang, D., Li, L., Li, K., Qi, J., Wang, W., Zhang, G., & Liu, P. (2023). Knowledge-guided Multi-granularity GCN for ABSA. *Information Processing & Management, 60*(2), Article 103223.
83. Zou, D., Wei, W., Mao, X.-L., Wang, Z., Qiu, M., Zhu, F., & Cao, X. (2022). Multi-level cross-view contrastive learning for knowledge-aware recommender system. Paper presented at the Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2022), Madrid, Spain.