Dynamic Heterogeneous Graph Convolutional Networks for Click-Through Rate Prediction in Recommender Systems

Ying Jin

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

E-mail address: jinying.isec@gmail.com

Yanwu Yang

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

E-mail address: yangyanwu.isec@gmail.com

Baojun Ma*

**Corresponding author*

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

E-mail address: mabaojun@shisu.edu.cn

Dynamic Heterogeneous Graph Convolutional Networks for Click-Through Rate Prediction in Recommender Systems

Abstract: Click-through rate (CTR) prediction is a critical component of recommender systems, helping identify the user who is likely to interact with a particular item. Prior studies have been dedicated to feature augmentation for CTR prediction by primarily leveraging heterogeneous information (e.g., social information) and historical user-item interactions. However, focusing solely on heterogeneous information or historical interactions is insufficient to capture intricate dynamics of user behaviors. In this paper, we propose Dynamic Heterogeneous Graph Convolutional Networks (DH-GCN) for CTR prediction, combining heterogeneous information and dynamic user-item interactions. Specifically, we construct three graphs: user-item interaction graph, item knowledge graph, and user-user graph, and design a novel graph-to-graph learning component to realize the sharing of neighbors and relationships in the GCN framework. Moreover, we leverage multi-granularity time-sliced user-item interaction graphs to capture evolving user preference trajectories. Experiments on three public datasets show that DH-GCN makes significant improvements over existing state-of-the-art methods and achieves 0.01-level improvements in AUC, Accuracy, and F1.

Keywords: Click-through rate prediction; Dynamic user-item interactions; Graph convolutional networks; Graph-to-graph learning; User preference trajectories

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 [1]. 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 [2]. An improvement of 0.1% in the CTR prediction accuracy would result in hundreds of millions of extra earnings [3].

In CTR prediction, researchers have primarily leveraged two sources for feature augmentation including heterogeneous information, and dynamic user-item interactions. The former incorporates user profiles, fruitful semantics about items and multi-type relations among them [4], and contextual information (e.g., locations) [5]. Recently, Graph neural networks (GNNs) have shown a great power in representing heterogeneous information networks accommodating multi-source information and higher-order feature interactions [6-8]. However, previous research handle collaborative signals separately, hindering the ability of capturing complex latent connections and the intricate nature of interactions [9]. The latter aims to understand evolving user preferences and interests over time by exploring the temporal user-item interactions and weighing the historical behavior sequence relative to the target item [10]. However, prior research predominantly explore users' time-dependent representations, neglecting the item side [11] and the multi-granularity variability of user selections [12]. Capturing the evolutionary patterns of user-item interactions allows for timely tracking of changes. Given the heterogeneous and dynamic nature of user-item interactions, it is crucial to combine the two strengths to provide synergistic benefits [13]. Recently, a few researchers (e.g., [14, 15] provide semantic information about items in user's historical sequence. However, these methods consider either heterogeneous information on the item side or temporal dynamics on the user side in a relatively monotonous manner and fail to explore deep relationships across heterogeneous graphs, which restricts the model's expressive capacity to capture complex interaction patterns.

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 multiple information sources is demanding because multi-aspects information interweaves with each other [9]. Second, how to capture structural and temporal information simultaneously poses another challenge in handling the variability.

In this research, we propose Dynamic Heterogeneous Graph Convolutional Networks (DH-GCN) that exploits complex interaction patterns and temporal dynamics of user preferences to facilitate CTR prediction. DH-GCN incorporates multi-source heterogeneous information networks with a user-item interaction graph to capture the user-item dependency on the temporal axis, a knowledge graph to represent item-related semantics, and a user-user graph to discover collaborative signals based on preference similarity. To delve into correlations across these graphs, we devise a novel graph-to-graph learning component to distill information by mapping a node in one graph to that in another graph to obtain a unified neighborhood space in the 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 interaction graphs. We correlate a series of temporal graph representations corresponding to time intervals (e.g., weekly, monthly, yearly) to generate long-term representations using RWKV [16], which utilizes a linear attention mechanism and combines the advantages of Transformer and 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 on three datasets. It achieves the superiority with increases of 0.02 (Last.FM), 0.02 (MovieLens-1M), and 0.01 (E-Commerce), respectively in Accuracy. In addition, we conduct ablation studies to verify whether each component can boost the model performance. We observe 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 component integrates multi-source heterogeneous information, providing synergistic effects for a holistic understanding of user preferences. Third, DH-GCN demonstrates significant superiority over existing models in terms of AUC, Accuracy, and F1, underscoring the effectiveness of our model in capturing heterogeneous information and dynamic interactions.

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

2. Related Work

2.1 CTR Prediction

CTR prediction has emerged as a pivotal area of research, attracting extensive studies aimed at enhancing its accuracy and effectiveness. Among traditional methods, logistic regression models rely on feature engineering to improve accuracy [2, 17], which can be easily implemented due to simplicity, but consider only the first-order interaction and have limited representation power [18]. Factorization Machines (FM) [19-21] are proficient in capturing feature interactions efficiently; however, they are unable to provide an adequate understanding of intricate patterns [22]. Deep Neural Networks (DNNs) provide significant advancements over traditional methods and can capture high-order feature interactions automatically [23-25], as exemplified by models such as Wide&Deep [26], DeepFM [27], and DCN [28, 29]. For further information, refer to a detailed review on CTR prediction by Yang and Zhai [2].

Within the scope of CTR prediction, the relationship between users and items can be depicted as a heterogeneous bipartite graph, where nodes denote users and items, and edges denote interactions. GNNs are well-suited to handle graph structures and can effectively model complex relationships. GNNs propagate information along links in the graph and aggregate neighbor information to update the node representation [30, 31]. Recent research (e.g., [32, 33]) augment the user-item graph with additional information such as user demographics, item characteristics, or contextual situations to improve the performance of GNNs in CTR prediction. Li et al. [18] design a graph structure to represent features in multi-field categories, enabling modeling the complex feature interactions flexibly and explicitly. Ouyang et al. [34] build connections between ads based on their features and propose three specific Graph Meta Embedding (GME) models from different perspectives to distill information. Zhai et al. [31] 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.

To capture the time-varying interactions and the dynamic dependencies among features, researchers consider the evolution of user behaviors [35-37]. Zhou et al. [38] design a local activation unit to model latent user interests underlying concrete behaviors, enabling adaptive learning of user interest representations. Their approach assigns higher weights to historical behaviors more relevant to a specific ad. Zhou et al. [10] point out that most previous works neglect modeling latent interests behind concrete behaviors. They incorporate GRU augmented with an attention-based update gate to capture the evolving user interests from historical behaviors. He et al. [39] apply contrastive learning to decouple different types of item-level behavior sequences, extracting discrepancy and consistency characteristics among various behaviors. Additionally, to address the emerging recommendation scenario, some research (e.g., [40-42]) focus on trigger-induced recommendation and model dynamic changes of user instant interests. Some other studies observe the insufficiency of fixed item embeddings and focus on modeling dynamic item characteristics [43]. For example, Zhang et al. [44] handle the problem of sparse user behaviors and dynamically capture the characteristics of the item with interacting users and timestamps. Similar to us, Wang et al. [45] construct a time-evolving graph of user's sequential behaviors and dig into the user's real-time interests, but they only model the dynamics from the user's view.

2.2 Heterogeneous Information Used in CTR Prediction

Many researchers have introduced heterogeneous information into CTR prediction, which has proven effective in improving performance. The heterogeneous networks differ in the types of entities and relations, with the main presentations of user-item interactions, social networks, content-item networks, and knowledge graphs [46].

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 modeled by algorithms like neural networks [47, 48]. For example, Zhang et al. [44] utilize Transformer to learn the item's multiple aspects, representing the candidate items through a sequence of interacting users and timestamps. Lyu et al. [49] propose a Multi Classifier CTR prediction model to model users' inherent click tendency and preference towards various items.

Besides the single user-item interactions, hybrid heterogeneous information networks are also commonly employed in CTR prediction. The combination of the networks leverages diverse relations and information sources. For instance, Zhang et al. [5] design a Multi-Interactive Layer, which simultaneously considers user-item interactions and context information to model fine-grained features. Yang et al. [50] 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.

Recent studies have gone a further step by incorporating knowledge grph as side information. Among the various heterogeneous networks, knowledge graph provides a much more comprehensive and sophisticated semantic representation [51]. For example, Wang et al. [52] design knowledge-aware CNN for news CTR prediction, which uses TransD [53] to generate knowledge graph embeddings and realize the alignment and combination of word-level and knowledge-level information. However, this approach ignores high-order connections between entities. On the contrary, Feng et al. [54] bridge relational paths on the knowledge graph and apply Bi-LSTM [55] to encode each path, modeling multiplex relations 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. [56] construct ripple sets on item knowledge graph to propagate user preferences, but the relations are weakly personalized. Wang et al. [57] characterize personal neighborhood information and achieve high-order connection modeling on the knowledge graph under the GNNs framework. Subsequently, some research (e.g., [58, 59]) perform high-order propagation within the heterogeneous graph structure utilizing the potential of contrastive learning, which contrasts positive pairs against negative pairs from multiple views.

Besides heterogeneous information, dynamic interactions are another valuable source for feature augmentation. Peng et al. [15] utilize graph networks with correlating relations and apply Bi-LSTM to capture evolving user interests. Li et al. [14] mine knowledge-enhanced paths and generate path embeddings using SEP2Vec, enabling learning user sequences of interest.

Existing methods rarely fuse multi-source heterogeneous information and fail to handle complicated interactive patterns by independent learning. When capturing the dynamics of the user-item interactions, they focus on the single use-side representations over the entire timeline. Distinct from these methods, our work incorporates multi-source heterogeneous networks and performs graph-to-graph learning for more expressive and comprehensive modeling. To discover both short- and long-term user-item dependency, we construct multi-granularity time-sliced user-item graphs.

3. Dynamic Heterogeneous Graph Convolutional Networks

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

3.1 Model Architecture

Figure 1 shows the overall architecture of DH-GCN, which consists of four main modules. The first component, graph construction constructs a user-item graph, knowledge graph, and user-user graph. The second component, graph representation learning employs hierarchical multi-head self-attention [60] and GCN to learn enriched 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 RWKV. The third component, heterogeneous information fusion fuses heterogeneous user and item representations through the convolution operations. The final component outputs the predicted click probability.

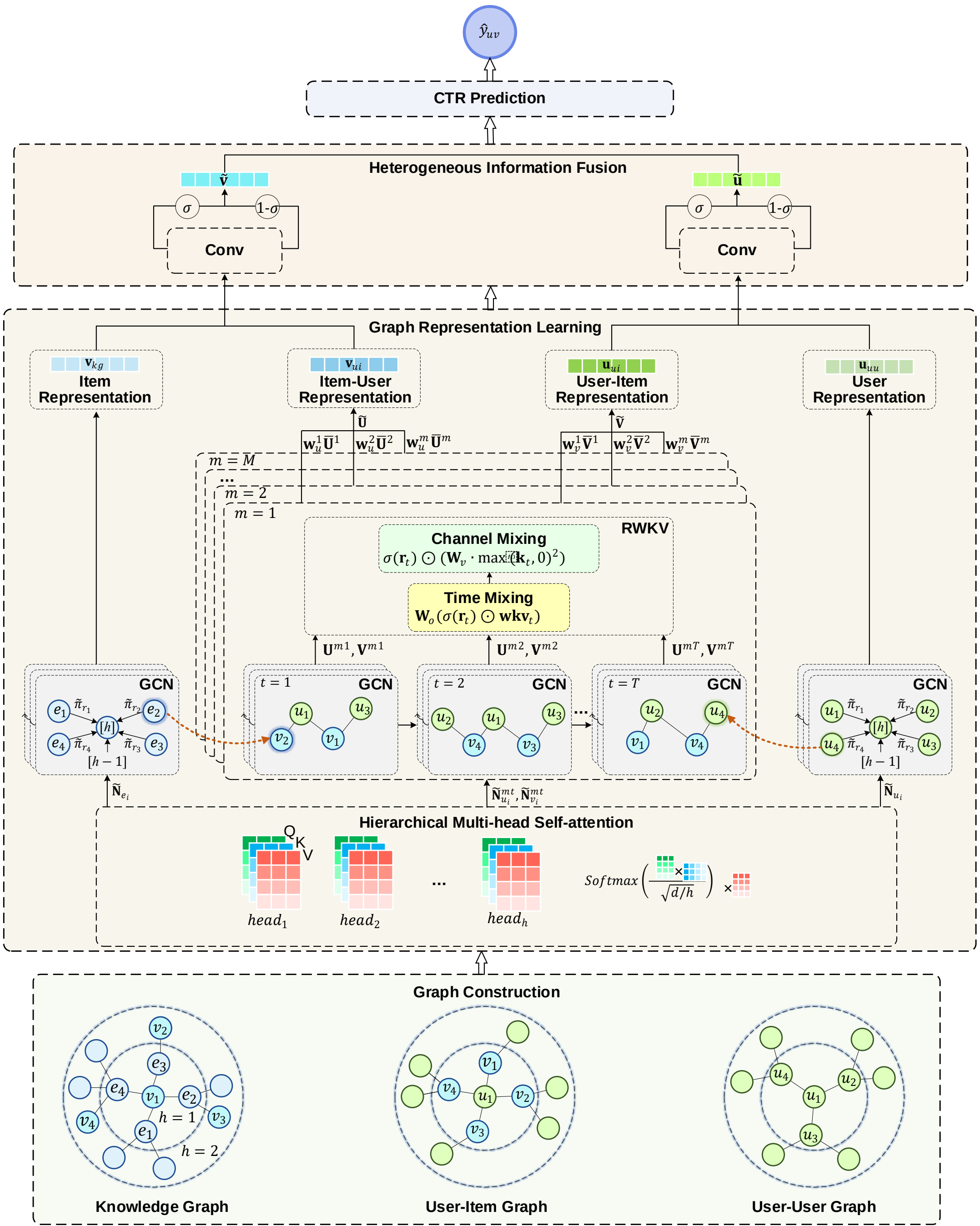


Figure 1. Architecture of DH-GCN.

3.2 Graph Construction

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

User-item graph. In a typical sequential recommendation scenario, a temporal bipartite graph $\mathcal{G}_{\mathrm{uv}}$ 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 $\mathcal{U} = \{u_1, u_2, \dots, u_{|\mathcal{U}|}\}$ and a set of items is defined as $\mathcal{V} = \{v_1, v_2, \dots, v_{|\mathcal{V}|}\}$, 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 user-item graph $\mathcal{G}_{\mathrm{uv}}$. We slice the holistic timeline into T intervals, where a certain interval length represents a 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-sliced interactions $\mathcal{T}^{mt}$ 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.

Knowledge graph. The knowledge graph $\mathcal{G}_{kg}$ is composed of massive (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 (Titanic, film.actor, Leonardo DiCaprio) expresses the fact that "Leonardo DiCaprio" is the actor in the film "Titanic". 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, $\mathcal{V} \subseteq \mathcal{E}$. 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 $\mathcal{G}_{\mathrm{uu}}$ to learn user preference similarity. If user 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 $\mathrm{w}_{\mathrm{ij}} = \frac{|S^{u_i} \cap S^{u_j}|}{|S^{u_j}|}$, where S^{u_i} and S^{u_j} denote the clicked items set of the user u_i and u_j, respectively.

Given the user-item graph $\mathcal{G}_{uv}$, knowledge graph $\mathcal{G}_{kg}$, and user-user graph $\mathcal{G}_{uu}$, we aim to learn a prediction function $\hat{y}_{uv} = \mathcal{F}(u, v, \mathcal{G}_{uv}, \mathcal{G}_{kg}, \mathcal{G}_{uu}, \mathcal{T}; \Theta)$, where $\hat{y}_{uv}$ denotes the 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.

Table 1: Notations and Descriptions.

Notations	Descriptions
u	User
v	Item
$\mathcal{U} = \{u_1, u_2, \dots, u_{\lvert\mathcal{U}\rvert}\}$	A set of users with the number of $\lvert\mathcal{U}\rvert$
$\mathcal{V} = \{v_1, v_2, \dots, v_{\lvert\mathcal{V}\rvert}\}$	A set of items with the number of $\lvert\mathcal{V}\rvert$
$Y = \{y_{uv} \mid u \in \mathcal{U}, v \in \mathcal{V}\}$	The user-item interaction matrix
$\mathcal{G}_{\mathrm{uv}}$	User-item interaction graph
$\mathcal{G}_{kg}$	Knowledge graph
$\mathcal{G}_{\mathrm{uu}}$	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)	Knowledge 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.
S^{u_i}	The clicked item set of the user u_i
$\mathrm{w_{ij}}$	The weight from u_i to u_j in $\mathcal{G}_{\mathrm{uu}}$
$\hat{y}_{uv}$	Predicted user u's click probability to item v
$\mathcal{F}$	Click-through rate prediction function
Θ	Parameters of prediction function $\mathcal{F}$
d	The embedding dimension
$\mathbf{u}_0$, $\mathbf{v}_0$	Initialized embedding
$\mathbf{t}_u$, $\mathbf{t}_v$	Temporal embedding
$\mathbf{p}_u$, $\mathbf{p}_v$	Positional encoding
$\mathbf{W}^O$,$\mathbf{W}_i^Q$,$\mathbf{W}_i^K$,$\mathbf{W}_i^V$	Parameter matrices of the projection transformation in multi-head self-attention
$\mathcal{N}(e_i)$	Direct neighbors of the entity e_i
$\widetilde{\mathbf{N}}_{e_i}$, $\widetilde{\mathbf{N}}_{u_i}$, $\widetilde{\mathbf{N}}_{u_i}^{mt}$	Attentive neighbor matrices generated from $\mathcal{G}_{kg}$, $\mathcal{G}_{\mathrm{uu}}$, $\mathcal{G}_{\mathrm{uv}}$
$\mathcal{T}$	The time-sliced interactions with m granularities
$\mathcal{T}^{mt}$	The t-th time-sliced interactions under the granularity m
$\mathcal{G}^{mt}$	The user-item bipartite graph corresponding to $\mathcal{T}^{mt}$

$MultiHead(\cdot)$	Multi-head self-attention
$g(\cdot)$	The function to calculate the score between the user u and the relation r
π_r^u	The score between the user u and the relation r
$\mathbf{e}_i$	Representation of e_i
$\mathbf{e}_{\mathcal{N}(e_i)}^u$	Aggregation information of $\mathcal{N}(e_i)$
h	Number of propagation layers in GCN
$f(\cdot)$	Aggregator
$\mathbf{v}_{kg}$	Item representation generated from $\mathcal{G}_{kg}$
γ	Nonlinear activation function such as ReLU
$\mathbf{W}$, $\mathbf{W}_1$, $\mathbf{W}_2$	Parameter matrices in the aggregator
$\|\|$	Concatenation
$\odot$	Element-wise product
$\pi_{r_{u_i,u_j}}$	The score between the user u_i and u_j generated from $\mathcal{G}_{\mathrm{uu}}$
$\mathbf{u}_{\mathcal{N}(u_i)}$	Neighbor information of the user u_i generated from $\mathcal{G}_{\mathrm{uu}}$
$\mathbf{u}_{uu}$	User representation generated from $\mathcal{G}_{uu}$
$\pi_{r_{u_i^{mt},v_i^{mt}}}$	The score between the user u_i and interacted item v_i
$\mathbf{u}_{\mathcal{N}(u_i^{mt})}$	Neighbor information of the 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
$\mathbf{wkv}_t$	The attention score in RWKV
$\boldsymbol{W}_r$, $\boldsymbol{W}_k$, $\boldsymbol{W}_v$, $\mathbf{W}_o$	Trainable weight matrices in RWKV
$\sigma(\mathbf{r}_t)$	The receptance of past information in RWKV
Θ_U^m, Θ_V^m	Trainable model parameters in RWKV
$\mathbf{u}_{ui}$	User representation generated from $\mathcal{G}_{uv}$
$\mathbf{v}_{ui}$	Item representation generated from $\mathcal{G}_{uv}$
$\sigma(\cdot)$	The sigmoid function
$\tilde{\mathbf{u}}$	The final user representation
$\tilde{\mathbf{v}}$	The final item representation
$\mathcal{L}$	The complete loss function
$\mathcal{J}$	Cross-entropy loss
P	Negative uniformed sampling distribution
N^u	The number of each user's negative samples
λ	$L2$ regularization coefficient

3.3 Graph Embedding

We first map a user or an item to a low-dimensional embedding. Each embedding consists of three types of embeddings: initial embedding, temporal embedding, and positional encoding. Initial embedding is generated randomly. We adopt temporal embedding in [61] 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 proposed by [60] 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 as follows. The embeddings of entities in the knowledge graph and users in the user-user graph are also initialized randomly in the same way.

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

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

3.4 Graph Representation Learning

We first introduce hierarchical multi-head self-attention to distinctively prioritize neighboring nodes. The learned importance weights are then utilized by GCN and RWKV to learn heterogeneous and dynamic representations.

3.4.1 Hierarchical Multi-head Self-attention

Given the knowledge graph $\mathcal{G}_{kg}$, user-user graph $\mathcal{G}_{\mathrm{uu}}$, and user-item graph $\mathcal{G}_{\mathrm{uv}}$, we introduce hierarchical multi-head self-attention which characterizes the importance of neighbors with the same connectivity order. The attention computation is a scaled dot-product computation between a query ($\mathbf{Q}$) and a set of key ($\mathbf{K}$) -value ($\mathbf{V}$) pairs as follows.

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

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:

$$\begin{cases} MultiHead(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = Concat(head_1, \dots, head_h)\mathbf{W}^O \\ head_i = Attention\left(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V\right) \end{cases}, \tag{3}$$

where $\mathbf{W}^O$,$\mathbf{W}_i^Q$,$\mathbf{W}_i^K$,$\mathbf{W}_i^V$ are parameter matrices in the projection transformation.

For knowledge graph, which contains fruitful connections and provides side information for items [59, 62], it is imperative to differentiate the importance of adjacent entities in describing and characterizing user preferences [63, 64]. The user-user graph considers similar behaviors indicating shared interests among users [65]. Obviously, neighbor 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 the neighbors in each graph through multi-head self-attention.

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

where $\mathbf{N}_{e_i}$ and $\widetilde{\mathbf{N}}_{e_i}$ denote neighbor matrices of the 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 the user u_i in $\mathcal{G}_{uu}$ before and after the update, respectively; $\mathbf{N}_{u_i}^{mt}$ and $\widetilde{\mathbf{N}}_{u_i}^{mt}$ denote neighbor matrices of the user u_i in $\mathcal{G}^{mt}$ before and after the update, respectively and it is analogous for the item side.

4.3.2 Dynamic Heterogeneous Graph Convolutional Networks

In this section, we introduce dynamic heterogeneous graph convolutional networks for representation learning on each graph, 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 component 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 [57] to capture the 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.

For a specific user u, the user's personal preferences should be differentiated. For example, in the purchase scenario, one user may value the brand of the product more, while another user may pay more attention to the style of the product. We define a function $g(\cdot)$ such as inner product to calculate the relevance score between the user u and the relation r.

$$\pi_r^u = g(\mathbf{u}, \mathbf{r}) \tag{5}$$

As a result, the relation-aware attention mechanism has the capability to focus more on the relevant neighboring nodes [66]. Subsequently, neighborhood information is aggregated with different weights:

$$\begin{cases} \mathbf{e}_{\mathcal{N}(e_i)}^u = \sum_{e_j \in \mathcal{N}(e_i)} \tilde{\pi}_{r_{e_i,e_j}}^u \mathbf{e}_j \\ \tilde{\pi}_{r_{e_i,e_j}}^u = \frac{exp(\pi_{r_{e_i,e_j}}^u)}{\sum_{e_j \in \mathcal{N}(e_i)} exp(\pi_{r_{e_i,e_j}}^u)} \end{cases}, \tag{6}$$

where $\mathbf{e}_i$ denotes the representation of the entity e_i, $\mathbf{e}_{\mathcal{N}(e_i)}^u$ denotes aggregation information of $\mathcal{N}(e_i)$, and $\tilde{\pi}_{r_{v,e}}^u$ is from the function $g(\cdot)$ with normalization.

To achieve the item representation with high-order structural and semantic information, we update the entity representation $\mathbf{e}_i$ with its neighboring representation $\mathbf{e}_{\mathcal{N}(e_i)}^u$. The h-hop propagation process can be formally expressed as

$$\mathbf{e}_i[h] = f\left(\mathbf{e}_i[h-1], \mathbf{e}_{\mathcal{N}(e_i)}^u[h-1]\right). \tag{7}$$

We evaluate the three aggregators in Section 4.5.1.

Sum aggregator [67] sums the entity representation and its neighboring representation before nonlinear transformation.

$$f_{sum} = \gamma\left(\mathbf{W}\left(\mathbf{e}_i + \mathbf{e}^u_{\mathcal{N}(e_i)}\right)\right), \tag{8}$$

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

Concat aggregator [68] concatenates the entity representation and its neighboring representation, followed by nonlinear transformation.

$$f_{concat} = \gamma(\mathbf{W}(\mathbf{e}_i || \mathbf{e}^u_{\mathcal{N}(e_i)})), \tag{9}$$

where $||$ is the concatenation operation.

Bi-interaction [69] aggregator considers a combination of a single interaction between two representations, applying both summation and element-wise product operators.

$$f_{Bi-interaction} = \gamma\left(\mathbf{W}_1\left(\mathbf{e}_i + \mathbf{e}^u_{\mathcal{N}(e_i)}\right)\right) + \gamma\left(\mathbf{W}_2\left(\mathbf{e}_i \odot \mathbf{e}^u_{\mathcal{N}(e_i)}\right)\right), \tag{10}$$

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

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

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.

$$\begin{cases} \mathbf{u}_{\mathcal{N}(u_i)} = \sum_{u_j \in \mathcal{N}(u_i)} \tilde{\pi}_{r_{u_i,u_j}} \mathbf{u}_j \\ \tilde{\pi}_{r_{u_i,u_j}} = \frac{exp(\pi_{r_{u_i,u_j}})}{\sum_{u_j \in \mathcal{N}(u_i)} exp(\pi_{r_{u_i,u_j}})}, \\ \pi_{r_{u_i,u_j}} = g\left(\mathbf{u}_i, w_{ij}\mathbf{r}_{u_i,u_j}\right) \end{cases} \tag{11}$$

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

Then, through propagation with aggregators, preferences from similar users are incorporated.

$$\mathbf{u}_i[h] = f\left(\mathbf{u}_i[h-1], \mathbf{u}_{N(u_i)}[h-1]\right), \tag{12}$$

where the final aggregation layer generates the learned user representation $\mathbf{u}_{uu}$ in the user-user graph.

Time-sliced user-item graph representation learning. It is worth noting that temporal dependency between users and items underlying dynamic interactions cannot be overlooked. Therefore, we design 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 interval.

$$\begin{cases} \mathbf{u}_{\mathcal{N}(u_i^{mt})} = \sum_{v_i^{mt} \in \mathcal{N}(u_i^{mt})} \tilde{\pi}_{r_{u_i^{mt}, v_i^{mt}}} \mathbf{v}_i^{mt} \\ \tilde{\pi}_{r_{u_i^{mt}, v_i^{mt}}} = \frac{exp(\pi_{r_{u_i^{mt}, v_i^{mt}}})}{\sum_{v_i^{mt} \in \mathcal{N}(u_i^{mt})} exp(\pi_{r_{u_i^{mt}, v_i^{mt}}})} \\ \pi_{r_{u_i^{mt}, v_i^{mt}}} = g(\mathbf{u}_i^{mt}, \mathbf{v}_i^{mt}) \end{cases}, \tag{13}$$

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

Then we aggregate user-item interaction information.

$$\mathbf{u}_i^{mt}[h] = f\left(\mathbf{u}_i^{mt}[h-1], \mathbf{u}_{\mathcal{N}(u_i^{mt})}[h-1]\right) \tag{14}$$

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 novel Receptance Weighted Key Value (RWKV) model 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, which can be formulated as follows.

$$\begin{cases} \mathbf{r}_t = \mathbf{W}_r(\boldsymbol{\mu}_r \mathbf{u}^{mt} + (1 - \boldsymbol{\mu}_r)\mathbf{u}^{m(t-1)})) \\ \mathbf{k}_t = \mathbf{W}_k(\boldsymbol{\mu}_k \mathbf{u}^{mt} + (1 - \boldsymbol{\mu}_k)\mathbf{u}^{m(t-1)})) \\ \mathbf{v}_t = \mathbf{W}_v(\boldsymbol{\mu}_v \mathbf{u}^{mt} + (1 - \boldsymbol{\mu}_v)\mathbf{u}^{m(t-1)})) \\ \mathbf{wkv}_t = \frac{\sum_{i=1}^{t-1} e^{-(t-1-i)\mathbf{w} + \mathbf{k}_i} \mathbf{v}_i + e^{\mathbf{x} + \mathbf{k}_t} \mathbf{v}_t}{\sum_{i=1}^{t-1} e^{-(t-1-i)\mathbf{w} + \mathbf{k}_i} + e^{\mathbf{x} + \mathbf{k}_t}} \\ \mathbf{o}_t = \mathbf{W}_o(\sigma(\mathbf{r}_t) \odot \mathbf{wkv}_t) \end{cases}, \tag{15}$$

where $\mathbf{k}_t$ and $\mathbf{v}_t$ play the role of $\mathbf{Q}$ and $\mathbf{K}$ in traditional attention, respectively; $\mathbf{wkv}_t$ represents the attention score; $\mathbf{w}$ and $\mathbf{x}$ are two time decay vectors; $\sigma(\mathbf{r}_t)$ means the receptance of past information; $\boldsymbol{W}_r$, $\boldsymbol{W}_k$, $\boldsymbol{W}_v$ 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 follows.

$$\begin{cases} \mathbf{r}_t = \mathbf{W}_r(\boldsymbol{\mu}_r \mathbf{u}^{mt} + (1-\boldsymbol{\mu}_r)\mathbf{u}^{m(t-1)})) \\ \mathbf{k}_t = \mathbf{W}_k(\boldsymbol{\mu}_k \mathbf{u}^{mt} + (1-\boldsymbol{\mu}_k)\mathbf{u}^{m(t-1)})) \\ \mathbf{o}_t = \sigma(\mathbf{r}_t) \odot (\mathbf{W}_v \cdot \max(\mathbf{k}_t, 0)^2) \end{cases} \tag{16}$$

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

$$\overline{\mathbf{U}}^m = RWKV(\mathbf{U}^m; \Theta_U^m)|_{T+1},\ \overline{\mathbf{V}}^m = RWKV(\mathbf{V}^m; \Theta_V^m)|_{T+1}, \tag{17}$$

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. These adaptive weights are learned from 1×1 convolution layers $Conv(\cdot)$ followed by the softmax operation [70].

$$\begin{cases} \widetilde{\mathbf{U}} = \sum_{m=1}^{M} \mathbf{w}_u^m \overline{\mathbf{U}}^m \\ [\mathbf{w}_u^1; \mathbf{w}_u^2; \cdots; \mathbf{w}_u^m] = Softmax\big(Conv(\overline{\mathbf{U}})\big) \\ \widetilde{\mathbf{V}} = \sum_{m=1}^{M} \mathbf{w}_v^m \overline{\mathbf{V}}^m \\ [\mathbf{w}_v^1; \mathbf{w}_v^2; \cdots; \mathbf{w}_{vl}^m] = Softmax(Conv(\overline{\mathbf{V}})) \end{cases} \tag{18}$$

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 $\mathbf{u}_{ui}$ and item's representation $\mathbf{v}_{ui}$ from the user-item interaction graph.

3.5 Heterogeneous Representation Fusion

Having obtained the user-side and item-side representations from heterogeneous networks, the ensuing task is how to trade off the contributions from different perspectives [54, 71]. To further inspire the strength of DH-GCN, we implement a sophisticated fusion through a learnable convolutional gate as follows.

$$\begin{cases}\tilde{\mathbf{u}} = \sigma(Conv(\mathbf{u}_{uu} + \mathbf{u}_{ui})) \odot \mathbf{u}_{uu} + (1 - \sigma(Conv(\mathbf{u}_{uu} + \mathbf{u}_{ui}))) \odot \mathbf{u}_{ui} \\ \tilde{\mathbf{v}} = \sigma(Conv(\mathbf{v}_{kg} + \mathbf{v}_{ui})) \odot \mathbf{v}_{kg} + (1 - \sigma(Conv(\mathbf{v}_{kg} + \mathbf{v}_{ui}))) \odot \mathbf{v}_{ui}\end{cases} \tag{19}$$

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 [72, 73]. 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 $\tilde{\mathbf{u}}$ and item representation $\tilde{\mathbf{v}}$.

3.6 CTR Prediction

DH-GCN generates user and item representations learned from the knowledge graph, user-user graph, and time-sliced user-item graphs. Finally, the sigmoid function predicts the click probability of the user clicking the item.

$$\hat{y}_{uv} = \sigma(\tilde{\mathbf{u}}, \tilde{\mathbf{v}}) \tag{20}$$

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

$$\mathcal{L} = \sum_{u \in \mathcal{U}} \left(\sum_{v: y_{uv}=1} \mathcal{J}(y_{uv}, \hat{y}_{uv}) - \sum_{i=1}^{N^u} \mathbb{E}_{v_i \sim P(v_i)} \mathcal{J}\left(y_{uv_i}, \hat{y}_{uv_i}\right) \right) + \lambda \parallel \mathcal{F} \parallel_2^2, \tag{21}$$

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 $L2$ regularization.

4. Experiments

In this section, we present an experimental evaluation on three public datasets to compare DH-GCN with ten state-of-the-art methods.

4.1 Datasets

We utilize three datasets from different scenarios to measure our method's capability.

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".

We filter out items that lack aligned entities in the knowledge graph. 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. As for Last.FM and MovieLens-1M, we employ the item knowledge graphs released by Wang et al. [57]. 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.

Table 2: Basic statistics of the datasets.

Statistics		Last.FM	MovieLens-1M	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

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 [74] combines the strengths of matrix factorization with personalized ranking,

[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

learning by maximizing posterior probabilities derived from Bayesian analysis. It can effectively predict users' preferences.

CKE [75] 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 [56] exploits the knowledge graph as side information and simulates user preference propagation in the knowledge graph similar to water ripples.

KGAT [69] investigates the combination of knowledge graph and user-item graph. It employs the attention mechanism to discriminate the importance of neighboring entities during propagation.

KGCN [57] captures users' potential interests, recursively propagating high-order relations. It aggregates and incorporates neighborhood information with bias in the knowledge graph.

DIN [38] overcomes the limitations of a fixed-length user interest representation by adaptively learning representations based on historical behaviors, tailored to each candidate item. This approach enhances the model's expressive ability, capturing nuances and relevancies specific to different items.

DIEN [10] 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 [76] is an efficient end-to-end method for capturing long-term user behaviors. By sampling from various hash functions, it creates hash signatures for the target and user behavior items, directly capturing user interest through shared hash signatures.

DisenCTR [45] introduces a novel approach to 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 [77] 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: AUC (Area Under ROC Curve), ACC (Accuracy), and F1 Score (F1). AUC represents the area under the Receiver Operating Characteristic (ROC) 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 instances 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 $L2$ 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, $\cdots$,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. We summarize the following observations.

(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 heterogeneous information interactions and dynamic representations. 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 unsatisfactorily performance. This is in conformity with our expectations, as it utilizes only limited information.

(3) Among the baselines with heterogeneous information, KGAT performs worst, probably because it does not properly utilize the higher-order interaction information.

(4) From an overall perspective, methods that take into account historical user interactions tend to perform better, suggesting the rationality of exploring dynamics underlying the 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.

Table 3: Model comparison in CTR prediction.

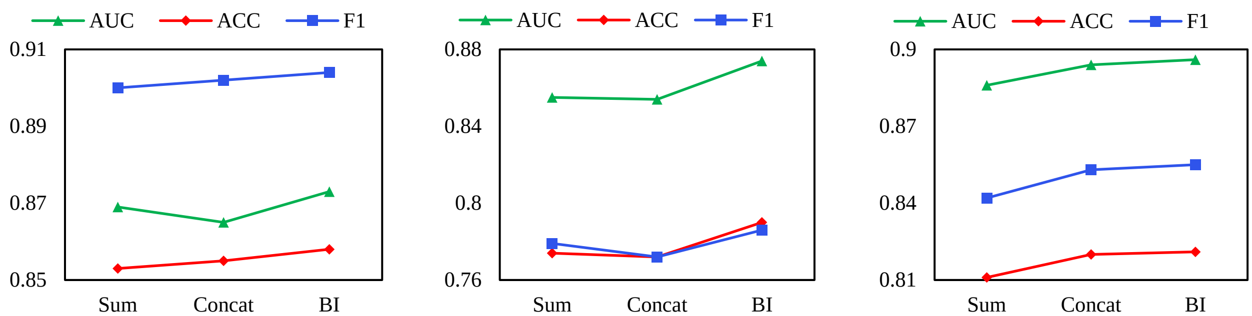
Model	Last.FM			MovieLens-1M			E-Commerce		
	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1
BPR-MF	0.820	0.756	0.833	0.819	0.757	0.755	0.852	0.784	0.821
CKE	0.855	0.792	0.821	0.849	0.765	0.757	0.881	0.799	0.836
RippleNet	0.842	0.792	0.818	0.848	0.762	0.747	0.876	0.810	0.837
KGAT	0.858	0.837	0.889	0.773	0.690	0.717	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

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 Figure 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.

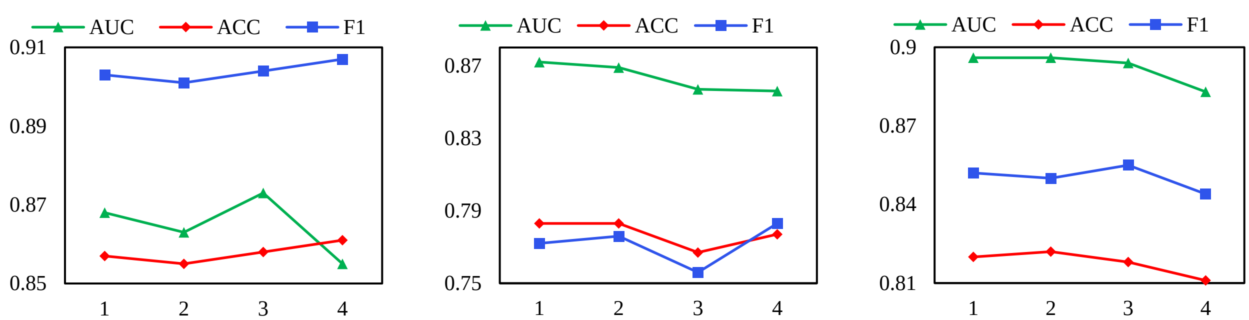


(a) Last.FM (b) MovieLens-1M (c) E-Commerce

Figure 2. Influences of aggregators.

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 (Figure 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.

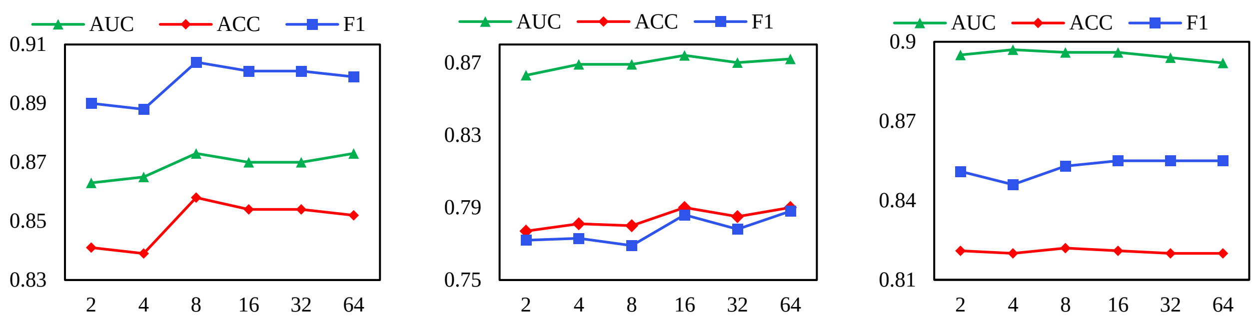


(a) Last.FM (b) MovieLens-1M (c) E-Commerce

Figure 3. Influences of the number of GCN layers.

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 to 8 or 16 in most cases as shown in Figure 4. This illustrates that increasing the neighbor size appropriately provides sufficient information for exploring latent connections and extending users' potential interests.



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

Figure 4. Influences of the neighbor size.

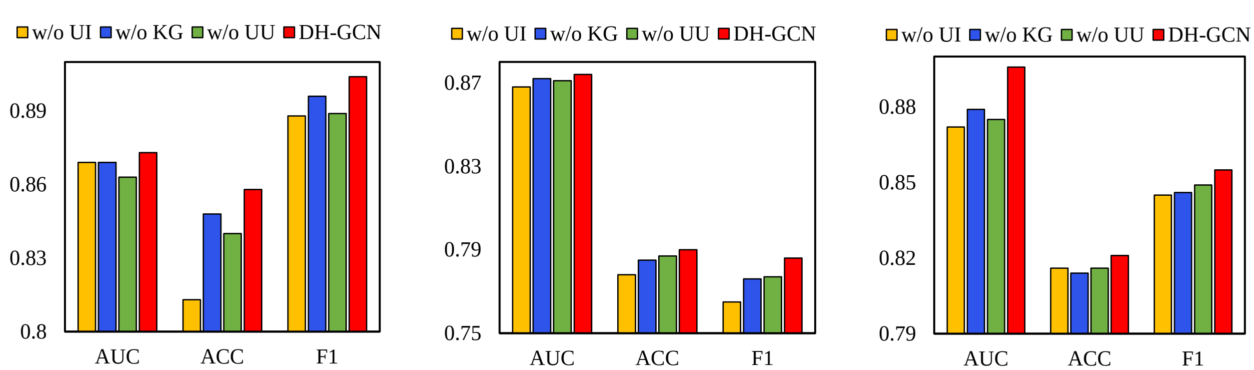
4.6 Ablation Study

To get deep insights into the design rationality, we clarify how different key components influence the performance, including the impacts of heterogeneous information, graph representation learning, and multi-granularity time-sliced user-item interaction 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 UI” denotes the absence of historical user-item interactions, “w/o KG” removes the external knowledge about items, and “w/o UU” means not injecting the user-user graph with similar preferences.

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

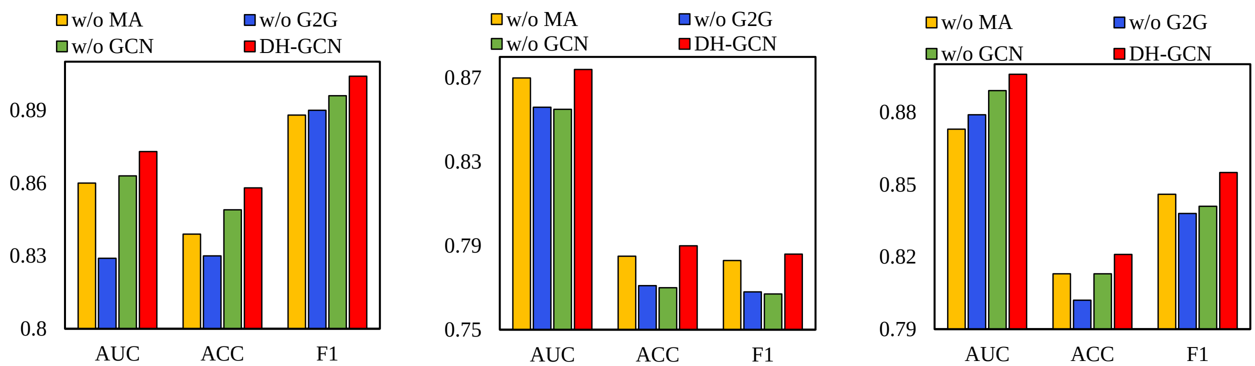


(a) Last.FM (b) MovieLens-1M (c) E-Commerce

Figure 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 graph. “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 component with isolated individual graph learning. “w/o GCN” means replacing GCN with simple mean operations to aggregate neighbors.



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

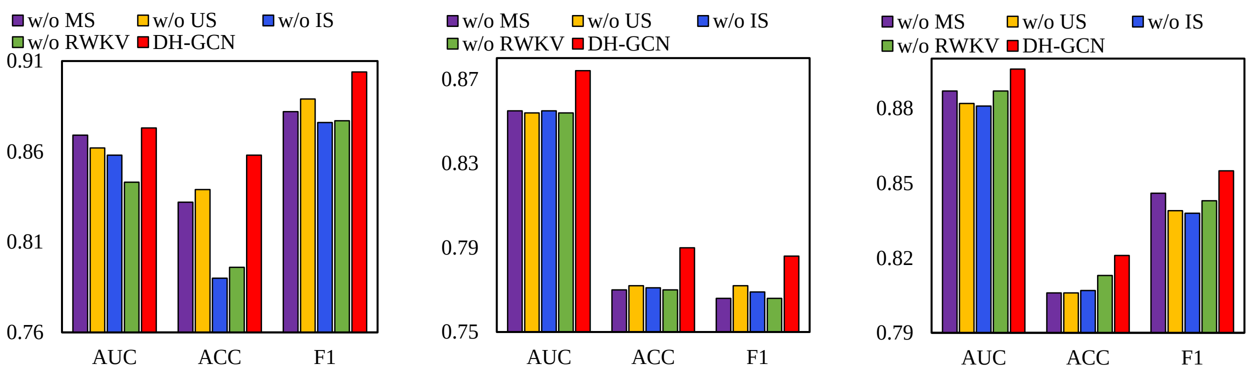
Figure 6. Influences of graph representation learning.

From Figure 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.

4.6.3 Influences of Multi-granularity Time-sliced User-item Interaction 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 user-side dynamic user-item interactions, “w/o IS” removes the counterpart item-side dynamic user-item interactions, “w/o RWKV” means not utilizing RWKV to capture sequential patterns and replacing it with the mean operations on the representations generated from each time slices.

From Figure 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 sides. Third, “w/o RWKV” validates that correlating sequential representations can boost performance. The possible reason is that RWKV discovers the dependency underlying the user-item interactions.



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

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

5. Conclusion and Future Work

In this paper, we propose Dynamic Heterogeneous Graph Convolutional Networks (DH-GCN), which integrates multi-source heterogeneous information and dynamic historical user-item interactions for CTR prediction. We devise a graph-to-graph learning component and provide a sharing neighborhood space to distill complex correlations across multi-source heterogeneous information networks. Moreover, we introduce a multi-granularity time-sliced user-item graph representation learning method to capture structural and temporal dependency simultaneously underlying dynamic user-item interactions. Experimental evaluation conducted on three real-world datasets demonstrates the superior performance of DH-GCN and verifies the contributions of its core modules. We also notice that dynamics of the item side and multi-granularity time-sliced user-item graphs are helpful for capturing varying user preferences.

In the future research, we intend to delve into real-time updates of knowledge graph, ensuring that prediction models are continuously informed by the latest semantics. For example, it is crucial to capture the real-time dynamics of news to make timely recommendations. Moreover, it is also worthwhile exploring multi-modal heterogeneous information for CTR prediction. Traditional heterogeneous information sources typically rely on textual data, but with the increasing availability of multimedia data (e.g., images, videos, audio), there is a growing interest in incorporating multi-modal features to extract richer information and perform high-quality representation learning. Furthermore, inspired by the achievements of self-supervised pre-training in natural language processing and computer vision, data augmentations that randomly omit certain items from the user-item interaction data to generate supervision signals have the potential to enhance the model's predictive capability and adaptability.

Declarations

Ethics approval and consent to participate

Not applicable

Consent for publication

All authors have read and agreed to submit version of the manuscript.

Availability of data and material

The datasets will be provided on request.

Competing interests

The authors declare that they have no conflicts of interest.

Funding

This research is supported by the National Natural Science Foundation of China (72172092, 72171093) and the Fundamental Research Funds for the Central Universities (2019114032).

Authors' Contributions

All authors contributed to the study conception and design. Conceptualization: Ying Jin, Yanwu Yang, Baojun Ma; Methodology: 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.

Acknowledgements

The authors would like to thank the anonymous referees for their comments and suggestions.

References

1. Yang, Y., Zhang, C., Zhao, K., and 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), 103388.
2. Yang, Y. and Zhai, P. (2022). Click-Through Rate Prediction in Online Advertising: A Literature Review. Information Processing & Management, 59(2), 102853.
3. Ling, X., Deng, W., Gu, C., Zhou, H., Li, C., and Sun, F. (2017). Model Ensemble for Click Prediction in Bing Search Ads. In Proceedings of the 26th International Conference on World Wide Web Companion (WWW 2017), Perth, Australia. International World Wide Web Conferences Steering Committee, 689–698.
4. Fan, S., Zhu, J., Han, X., Shi, C., Hu, L., Ma, B., and Li, Y. (2019). Metapath-guided Heterogeneous Graph Neural Network for Intent Recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2019), Anchorage, AK, USA. Association for Computing Machinery, New York, NY, USA, 2478–2486.
5. Zhang, K., Qian, H., Cui, Q., Liu, Q., Li, L., Zhou, J., Ma, J., and Chen, E. (2021). Multi-Interactive Attention Network for Fine-grained Feature Learning in CTR Prediction. In Proceedings of the 14th ACM International Conference on Web Search and Data Mining (WSDM 2021), Israel. Association for Computing Machinery, New York, NY, USA, 984–992.
6. Wang, Z., Lin, G., Tan, H., Chen, Q., and Liu, X. (2020). CKAN: Collaborative Knowledge-aware Attentive Network for Recommender Systems. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2020), China. Association for Computing Machinery, New York, NY, USA, 219–228.
7. Wang, Y., Zhang, R., Yang, Q., Zhou, Q., Zhang, S., Fan, Y., Huang, L., Li, K., and 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), 103682.
8. Chen, C., Cai, F., Hu, X., Chen, W., and Chen, H. (2021). HHGN: A Hierarchical Reasoning-based Heterogeneous Graph Neural Network for Fact Verification. Information Processing & Management, 58(5), 102659.
9. Xia, L., Huang, C., Xu, Y., Dai, P., Zhang, X., Yang, H., Pei, J., and Bo, L. (2021). Knowledge-Enhanced Hierarchical Graph Transformer Network for Multi-Behavior Recommendation. In Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI 2021). AAAI Press,

Washington, DC, USA, 4486-4493.

10. Zhou, G., Mou, N., Fan, Y., Pi, Q., Bian, W., Zhou, C., Zhu, X., and Gai, K. (2019). Deep Interest Evolution Network for Click-Through Rate Prediction. In 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. AAAI Press, Washington, DC, USA, Article 729.
11. Chen, Z., Zhang, W., Yan, J., Wang, G., and Wang, J. (2021). Learning Dual Dynamic Representations on Time-Sliced User-Item Interaction Graphs for Sequential Recommendation. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM 2021). Association for Computing Machinery, New York, NY, USA, 231-240.
12. Yang, Y., Huang, C., Xia, L., Liang, Y., Yu, Y., and Li, C. (2022). Multi-Behavior Hypergraph-Enhanced Transformer for Sequential Recommendation. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2022), Washington DC, USA. Association for Computing Machinery, New York, NY, USA, 2263–2274.
13. Gao, H., Kong, D., Lu, M., Bai, X., and Yang, J. (2018). Attention Convolutional Neural Network for Advertiser-level Click-through Rate Forecasting. In Proceedings of the 2018 World Wide Web Conference (WWW 2018), Lyon, France. International World Wide Web Conferences Steering Committee, 1855–1864.
14. Li, Y., Guo, X., Lin, W., Zhong, M., Li, Q., Liu, Z., Zhong, W., and 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.
15. Peng, W., Cheng, J., Wang, Z., Zhao, M., and Wu, X. (2023). DS-KGAT: A Deep Session GAT with Knowledge Enhancement for CTR Prediction. In 2023 IEEE 3rd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA 2023), Chongqing, China, 1590-1594.
16. Peng, B., Alcaide, E., Anthony, Q. G., Albalak, A., Arcadinho, S., Cao, H., Cheng, X., Chung, M., Grella, M., Kranthikiran, G., et al. (2023). RWKV: Reinventing RNNs for the Transformer Era. In Findings of the Association for Computational Linguistics: EMNLP 2023 (EMNLP 2023), Singapore. Association for Computational Linguistics, Stroudsburg, PA, USA, 14048-14077.
17. Kumar, R., Naik, S. M., Naik, V. D., Shiralli, S., Sunil, V. G., and Husain, M. (2015). Predicting Clicks: CTR Estimation of Advertisements using Logistic Regression Classifier. In 2015 IEEE International Advance Computing Conference (IACC 2015), Banglore, India, 1134-1138.
18. Li, Z., Cui, Z., Wu, S., Zhang, X., and Wang, L. (2019). Fi-GNN: Modeling Feature Interactions via Graph Neural Networks for CTR Prediction. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM 2019).

Association for Computing Machinery, New York, NY, USA, 539-548.

19. Juan, Y., Zhuang, Y., Chin, W.-S., and Lin, C.-J. (2016). Field-aware Factorization Machines for CTR Prediction. In Proceedings of the 10th ACM Conference on Recommender Systems (RecSys 2016). Association for Computing Machinery, New York, NY, USA, 43-50.
20. Xiao, J., Ye, H., He, X., Zhang, H., Wu, F., and Chua, T.-S. (2017). Attentional Factorization Machines: Learning the Weight of Feature Interactions via Attention Networks. In Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI 2017). AAAI Press, Washington, DC, USA, 3119-3125.
21. Tao, Z., Wang, X., He, X., Huang, X., and Chua, T.-S. (2020). HoAFM: A High-order Attentive Factorization Machine for CTR Prediction. Information Processing & Management, 57(6), 102076.
22. Song, W., Shi, C., Xiao, Z., Duan, Z., Xu, Y., Zhang, M., and Tang, J. (2019). AutoInt: Automatic Feature Interaction Learning via Self-Attentive Neural Networks. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM 2019). Association for Computing Machinery, New York, NY, USA, 1161-1170.
23. He, J., Mei, G., Xing, F., Yang, X., Bao, Y., and Yan, W. (2020). DADNN: Multi-Scene CTR Prediction via Domain-Aware Deep Neural Network. arXiv:2011.11938.
24. Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, HI, USA. IEEE Computer Society, 2261-2269.
25. Guo, X., Lin, W., Li, Y., Liu, Z., Yang, L., Zhao, S., and Zhu, Z. (2020). DKEN: Deep Knowledge-Enhanced Network for Recommender Systems. Information Sciences, 540, 263-277.
26. Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., and Ispir, M. (2016). Wide & Deep Learning for Recommender Systems. In Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS 2016). Association for Computing Machinery, New York, NY, USA, 7-10.
27. Guo, H., Tang, R., Ye, Y., Li, Z., and He, X. (2017). DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI 2017), Melbourne, Australia. AAAI Press, Washington, DC, USA, 1725-1731.
28. Wang, R., Fu, B., Fu, G., and Wang, M. (2017). Deep & Cross Network for Ad Click Predictions. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2017). Association for Computing Machinery, New York, NY, USA, 1-7.
29. Wang, F., Gu, H., Li, D., Lu, T., Zhang, P., and Gu, N. (2023). Towards Deeper, Lighter and Interpretable Cross Network for CTR Prediction. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM 2023). Association for Computing Machinery, New York, NY, USA, 2523–2533.

30. Song, C., Shu, K., and Wu, B. (2021). Temporally Evolving Graph Neural Network for Fake News Detection. Information Processing & Management, 58(6), 102712.

31. Zhai, P., Yang, Y., and Zhang, C. (2023). Causality-based CTR Prediction using Graph Neural Networks. Information Processing & Management, 60(1), 103137.

32. Li, F., Yan, B., Long, Q., Wang, P., Lin, W., Xu, J., and Zheng, B. (2021). Explicit Semantic Cross Feature Learning via Pre-trained Graph Neural Networks for CTR Prediction. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021). Association for Computing Machinery, New York, NY, USA, 2161-2165.

33. Guo, W., Su, R., Tan, R., Guo, H., Zhang, Y., Liu, Z., Tang, R., and He, X. (2021). Dual Graph Enhanced Embedding Neural Network for CTR Prediction. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2021). Association for Computing Machinery, New York, NY, USA, 496-504.

34. Ouyang, W., Zhang, X., Ren, S., Li, L., Zhang, K., Luo, J., Liu, Z., and Du, Y. (2021). Learning Graph Meta Embeddings for Cold-Start Ads in Click-Through Rate Prediction. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021). Association for Computing Machinery, New York, NY, USA, 1157-1166.

35. Li, C., Liu, Z., Wu, M., Xu, Y., Zhao, H., Huang, P., Kang, G., Chen, Q., Li, W., and Lee, D. L. (2019). Multi-Interest Network with Dynamic Routing for Recommendation at Tmall. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM 2019), Beijing, China. Association for Computing Machinery, New York, NY, USA, 2615–2623.

36. Feng, Y., Lv, F., Shen, W., Wang, M., Sun, F., Zhu, Y., and Yang, K. (2019). Deep Session Interest Network for Click-Through Rate Prediction. In Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI 2019), Macao, China. AAAI Press, Washington, DC, USA, 2301–2307.

37. Chen, T., Yin, H., Nguyen, Q. V. H., Peng, W. C., Li, X., and Zhou, X. (2020). Sequence-Aware Factorization Machines for Temporal Predictive Analytics. In 2020 IEEE 36th International Conference on Data Engineering (ICDE 2020), Dallas, TX, USA, 1405-1416.

38. Zhou, G., Zhu, X., Song, C., Fan, Y., Zhu, H., Ma, X., Yan, Y., Jin, J., Li, H., and Gai, K. (2018). Deep Interest Network for Click-Through Rate Prediction. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2018), London, United Kingdom. Association for Computing Machinery, New York, NY, USA, 1059–1068.

39. He, T., Li, K., Chen, S., Wang, H., Liu, Q., Wang, X., and Wang, D. (2023). DMBIN: A Dual Multi-behavior Interest Network for Click-Through Rate Prediction via Contrastive Learning. In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2023). Association for Computing Machinery,

New York, NY, USA, 1366–1375.

40. Xia, Y., Cao, Y., Hu, S., Liu, T., and Lu, L. (2023). Deep Intention-Aware Network for Click-Through Rate Prediction. In Companion Proceedings of the ACM Web Conference 2023 (WWW 2023). Association for Computing Machinery, New York, NY, USA, 533–537.
41. Xiao, Z., Yang, L., Zhang, T., Jiang, W., Ning, W., and Yang, Y. (2024). Deep Evolutional Instant Interest Network for CTR Prediction in Trigger-Induced Recommendation. In Proceedings of the 17th ACM International Conference on Web Search and Data Mining (WSDM 2024). Association for Computing Machinery, New York, NY, USA, 846–854.
42. Shen, Q., Wen, H., Tao, W., Zhang, J., Lv, F., Chen, Z., and Li, Z. (2022). Deep Interest Highlight Network for Click-Through Rate Prediction in Trigger-Induced Recommendation. In Proceedings of the ACM Web Conference 2022 (WWW 2022). Association for Computing Machinery, New York, NY, USA, 422–430.
43. Li, X., Wang, C., Tong, B., Tan, J., Zeng, X., and Zhuang, T. (2020). Deep Time-Aware Item Evolution Network for Click-Through Rate Prediction. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM 2020), Virtual Event, Ireland. Association for Computing Machinery, New York, NY, USA, 785–794.
44. Zhang, J., Lin, F., Yang, C., and Wang, W. (2022). Deep Multi-Representational Item Network for CTR Prediction. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2022), Madrid, Spain. Association for Computing Machinery, New York, NY, USA, 2277–2281.
45. Wang, Y., Qin, Y., Sun, F., Zhang, B., Hou, X., Hu, K., Cheng, J., Lei, J., and Zhang, M. (2022). DisenCTR: Dynamic Graph-based Disentangled Representation for Click-Through Rate Prediction. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2022), Madrid, Spain. Association for Computing Machinery, New York, NY, USA, 2314–2318.
46. Liu, J., Shi, C., Yang, C., Lu, Z., and Philip, S. Y. (2022). A Survey on Heterogeneous Information Network Based Recommender Systems: Concepts, Methods, Applications and Resources. AI Open, 3, 40-57.
47. Wu, C., Wu, F., Lyu, L., Huang, Y., and Xie, X. (2022). FedCTR: Federated Native Ad CTR Prediction with Cross-platform User Behavior Data. ACM Transactions on Intelligent Systems and Technology, 13(4).
48. Chen, X., Tang, Q., Hu, K., Xu, Y., Qiu, S., Cheng, J., and Lei, J. (2022). Hybrid CNN Based Attention with Category Prior for User Image Behavior Modeling. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2022), Madrid, Spain. Association for Computing Machinery, New York, NY, USA, 2336–2340.
49. Lyu, S., Cai, H., Zhang, C., Ling, S., Shen, Y., Zeng, X., Gu, J., Zhang, G., and Zhang, H. (2022). See Clicks Differently: Modeling User Clicking Alternatively with Multi Classifiers for CTR Prediction. In Proceedings of the 31st ACM International Conference on Information &

Knowledge Management (CIKM 2022), Atlanta, GA, USA. Association for Computing Machinery, New York, NY, USA, 4299–4303.

50. Yang, X., Deng, T., Tan, W., Tao, X., Zhang, J., Qin, S., and Ding, Z. (2019). Learning Compositional, Visual and Relational Representations for CTR Prediction in Sponsored Search. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM 2019), Beijing, China. Association for Computing Machinery, New York, NY, USA, 2851–2859.
51. Zhu, Z., Zhang, D., Li, L., Li, K., Qi, J., Wang, W., Zhang, G., and Liu, P. (2023). Knowledge-guided Multi-granularity GCN for ABSA. Information Processing & Management, 60(2), 103223.
52. Wang, H., Zhang, F., Xie, X., and Guo, M. (2018). DKN: Deep Knowledge-Aware Network for News Recommendation. In Proceedings of the 2018 World Wide Web Conference (WWW 2018). International World Wide Web Conferences Steering, Committee, Republic and Canton of Geneva, Switzerland, 1835-1844.
53. Ji, G., He, S., Xu, L., Liu, K., and Zhao, J. (2015). Knowledge Graph Embedding via Dynamic Mapping Matrix. In 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). Association for Computational Linguistics, Stroudsburg, PA, USA, 687-696.
54. Feng, Y., Lv, F., Hu, B., Sun, F., Kuang, K., Liu, Y., Liu, Q., and Ou, W. (2020). MTBRN: Multiplex Target-Behavior Relation Enhanced Network for Click-Through Rate Prediction. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM 2020). Association for Computing Machinery, New York, NY, USA, 2421-2428.
55. Graves, A.and Schmidhuber, J. (2005). Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures. Neural networks, 18(5-6), 602-610.
56. Wang, H., Zhang, F., Wang, J., Zhao, M., Li, W., Xie, X., and Guo, M. (2018). RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM 2018). Association for Computing Machinery, New York, NY, USA, 417-426.
57. Wang, H., Zhao, M., Xie, X., Li, W., and Guo, M. (2019). Knowledge Graph Convolutional Networks for Recommender Systems. In Proceedings of the 2019 World Wide Web Conference (WWW 2019), San Francisco, CA, USA. Association for Computing Machinery, New York, NY, USA, 3307-3313.
58. Ong, R. K., Qiu, W., and Khong, A. W. H. (2023). Quad-Tier Entity Fusion Contrastive Representation Learning for Knowledge Aware Recommendation System. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM 2023), Birmingham, United Kingdom. Association for Computing Machinery, New York, NY,

USA, 1949–1959.

59. Zou, D., Wei, W., Mao, X.-L., Wang, Z., Qiu, M., Zhu, F., and Cao, X. (2022). Multi-level Cross-view Contrastive Learning for Knowledge-aware Recommender System. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2022), Madrid, Spain. Association for Computing Machinery, New York, NY, USA, 1358–1368.
60. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is All You Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, California, USA. Curran Associates Inc., Red Hook, NY, USA, 6000–6010.
61. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. (2021). Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI 2021). AAAI Press, Washington, DC, USA, 11106-11115.
62. Cao, X., Shi, Y., Yu, H., Wang, J., Wang, X., Yan, Z., and Chen, Z. (2021). DEKR: Description Enhanced Knowledge Graph for Machine Learning Method Recommendation. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021). Association for Computing Machinery, New York, NY, USA, 203-212.
63. Tang, X., Wang, T., Yang, H., and Song, H. (2019). AKUPM: Attention-enhanced Knowledge-aware User Preference Model for Recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2019). Association for Computing Machinery, New York, NY, USA, 1891-1899.
64. Huang, W., Wu, J., Song, W., and Wang, Z. (2022). Cross Attention Fusion for Knowledge Graph Optimized Recommendation. Applied Intelligence, 52(9), 10297-10306.
65. Zeng, D., Liu, Y., Yan, P., and Yang, Y. (2021). Location-Aware Real-Time Recommender Systems for Brick-and-Mortar Retailers. INFORMS J. on Computing, 33(4), 1608–1623.
66. Gao, Y., Li, Y.-F., Lin, Y., Gao, H., and Khan, L. (2020). Deep Learning on Knowledge Graph for Recommender System: A Survey. arXiv:2004.00387.
67. Kipf, T. N.and Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. In 5th International Conference on Learning Representations (ICLR 2017), Toulon, France, 11305-11312.
68. Hamilton, W. L., Ying, R., and Leskovec, J. (2017). Inductive Representation Learning on Large Graphs. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, California, USA. Curran Associates Inc., Red Hook, NY, USA, 1025–1035.
69. Wang, X., He, X., Cao, Y., Liu, M., and Chua, T.-S. (2019). KGAT: Knowledge Graph Attention Network for Recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2019). Association for Computing Machinery, New York, NY, USA, 950-958.

70. Liu, S., Huang, D., and Wang, Y. (2019). Learning Spatial Fusion for Single-Shot Object Detection. arXiv:1911.09516.

71. Chen, Q., Zhao, H., Li, W., Huang, P., and Ou, W. (2019). Behavior Sequence Transformer for E-commerce Recommendation in Alibaba. In Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data (DLP-KDD 2019). Association for Computing Machinery, New York, NY, USA, 1-4.

72. Liu, Y., Li, B., Zang, Y., Li, A., and Yin, H. (2021). A Knowledge-Aware Recommender with Attention-Enhanced Dynamic Convolutional Network. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM 2021), Virtual Event, Queensland, Australia. Association for Computing Machinery, New York, NY, USA, 1079–1088.

73. Wang, C., Zhu, Y., Liu, H., Ma, W., Zang, T., and Yu, J. (2021). Enhancing User Interest Modeling with Knowledge-Enriched Itemsets for Sequential Recommendation. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM 2021). Association for Computing Machinery, New York, NY, USA, 1889-1898.

74. Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2009). BPR: Bayesian Personalized Ranking from Implicit Feedback. In Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI 2009), Montreal, Quebec, Canada. AUAI Press, Arlington, VA, USA, 452–461.

75. Zhang, F., Yuan, N. J., Lian, D., Xie, X., and Ma, W.-Y. (2016). Collaborative Knowledge Base Embedding for Recommender Systems. In Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2016). Association for Computing Machinery, New York, NY, USA, 353-362.

76. Cao, Y., Zhou, X., Feng, J., Huang, P., Xiao, Y., Chen, D., and Chen, S. (2022). Sampling is All You Need on Modeling Long-Term User Behaviors for CTR Prediction. In Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM 2022), Atlanta, GA, USA. Association for Computing Machinery, New York, NY, USA, 2974–2983.

77. Mao, K., Zhu, J., Su, L., Cai, G., Li, Y., and Dong, Z. (2023). FinalMLP: An Enhanced Two-Stream MLP Model for CTR Prediction. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2023). AAAI Press, Washington, DC, USA, 4552-4560.