DisenCTR: Dynamic Graph-based Disentangled Representation for Click-Through Rate Prediction

Yifan Wang
School of Computer Science,
Peking University
Beijing, China
yifanwang@pku.edu.cn

Yifang Qin
Department of Computer Science,
School of EECS, Peking University
Beijing, China
qinyifang@pku.edu.cn

Fang Sun
Department of Computer Science,
School of EECS, Peking University
Beijing, China
fts@pku.edu.cn

Bo Zhang
Meituan
Beijing, China
zhangbo37@meituan.com

Xuyang Hou
Meituan
Beijing, China
houxuyang@meituan.com

Ke Hu
Meituan
Beijing, China
huke05@meituan.com

Jia Cheng
Meituan
Beijing, China
jia.cheng.sh@meituan.com

Jun Lei
Meituan
Beijing, China
leijun@meituan.com

Ming Zhang*
School of Computer Science,
Peking University
Beijing, China
mzhang_cs@pku.edu.cn

ABSTRACT

Click-through rate (CTR) prediction plays a critical role in recommender systems and other applications. Recently, modeling user behavior sequences attracts much attention and brings great improvements in the CTR field. Many existing works utilize attention mechanism or recurrent neural networks to exploit user interest from the sequence, but fail to recognize the simple truth that a user's real-time interests are inherently diverse and fluid. In this paper, we propose **DisenCTR**, a novel dynamic graph-based disentangled representation framework for CTR prediction. The key novelty of our method compared with existing approaches is to model evolving diverse interests of users. Specifically, we construct a time-evolving user-item interaction graph induced by historical interactions. And based on the rich dynamics supplied by the graph, we propose a disentangled graph representation module to extract diverse user interests. We further exploit the fluidity of user interests and model the temporal effect of historical behaviors using Mixture of Hawkes Process. Extensive experiments on three real-world datasets demonstrate the superior performance of our method comparing to state-of-the-art approaches.

CCS CONCEPTS

• **Information systems → Recommender systems**.

*Ming Zhang is the corresponding author.

SIGIR '22, July 11–15, 2022, Madrid, Spain.

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531851

KEYWORDS

CTR Prediction, Disentangled Representation Learning, Graph Neural Networks

ACM Reference Format:
Yifan Wang, Yifang Qin, Fang Sun, Bo Zhang, Xuyang Hou, Ke Hu, Jia Cheng, Jun Lei, and Ming Zhang. 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 '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3477495.3531851

1 INTRODUCTION

Click-through rate (CTR) prediction plays a vital role in numerous information retrieval (IR) scenarios like recommendation, online advertising and web search [3, 4]. It aims to predict the probability that a user will click on a specific item. Since the quality of CTR not only influences the overall revenue of the whole platform, but also directly affects user experience and satisfaction, it is prevailing to understand user's real-time intentions in the CTR prediction task.

Users in online platforms always interact with items chronologically. Therefore, the modeling of users' historical interaction sequence in CTR prediction has drawn the attention of both academic and industrial communities [5, 14, 23–25]. Sequential methods such as DIN [25] use attention-based methods to capture relative interests from the user behavior sequence with regard to candidate items. DIEN [24] further utilizes a two-layer recurrent neural network structure (RNN) to capture the evolution of user interest. DSIN [5] leverages users' multiple historical sessions and employs a self-attention layer as well as RNN structure to model users' inter- and intra-session interests. However, when historical interactions are sparse, these methods have difficulty in capturing the users' real intentions. A user would be anchored to a certain behavioral pattern that is extant in his/her sparse interaction history, without exploring his/her latent interests, which can be more complex.

Graphs have been used to alleviate data sparsity issue in recommendation. To better capture the collaborative signal, graph neural networks (GNNs) [7, 10, 15] have been applied in the constructed user-item graph. NGCF [16] and LightGCN [9] use neighborhood embedding propagation to exploit high-order user-item relations on graph-structured data. As for CTR models, Fi-GNN [13] represents multi-field features as graph and explicitly models relations among features via a GNN framework. GIN [12] constructs a co-occurrence item graph and adopts multi-layer graph diffusion to mine user intention. DE-GNN [6] leverages both feature graph and user-item interaction graph to alleviate feature and behavioral sparsity. Despite the great success made by these GNN-based methods, they cannot effectively capture users' real-time interests, especially when users' interests are diverse and evolving with time.

Actually, in real-world applications, different users are distinct in terms of their interests and the same user may also be interested in various kinds of items due to diverse interests [1, 11, 17, 18, 20, 21]. For instance, when a user clicks on an ink cartridge during weekdays, and browses for clothing at weekends, he/she is probably displaying the different aspects of his/her interests. Moreover, these different aspects of a user's interests keep evolving, as we would not be surprised to see the user's taste for clothes change to catch up with the season's trends. Therefore, capturing the dynamics of these diverse interests is important for user behavior modeling.

To tackle the above challenges, we borrow the idea of disentangled representation learning [19, 20] and propose a novel dynamic graph-based disentangled representation framework for CTR prediction (**DisenCTR**), which models evolving diverse interests of users. Specifically, we construct a novel time-evolving user-item interaction graph to capture dynamic sequential behaviors of users. Instead of compressing diverse user interests into one single vector, our model learns disentangled user representations that reflect diverse interests of users in the constructed graphs and calculates CTR with Mixture of Hawkes Process (MHP), which has great merits in capturing both the temporal excitation effects of historical behaviors and the multiple interests within the behavior sequence.

We summarize our main contributions as follows:

- We construct a novel dynamic sequential user-item graph to capture evolving user interests. To the best of our knowledge, this work is the first dynamic-graph-based multi-interest CTR prediction framework.
- We propose to extract multiple user interests with disentangled representation learning on graphs, and model the multi-aspect temporal excitation effects of past behaviors using Mixture of Hawkes Process.
- We conduct extensive experiments on three real-world CTR benchmarks for CTR prediction. Our model consistently outperforms strong baselines, achieving state-of-the-art performance on all three benchmark datasets.

2 THE PROPOSED MODEL

2.1 Problem Formulation

In CTR prediction, let $\mathcal{U} = \{u_1, \ldots, u_M\}$ denote the set of M users, and $\mathcal{V} = \{v_1, \ldots, v_N\}$ denote the set of N items. The historical user-item interactions can be defined as $\mathcal{E} = \{(u, v, \tau)\}$, where each tuple represents a user u interacting with an item v associated with a timestamp $\tau \in \mathbb{R}^+$. Each user has a sequence of historical behaviors $H_u = [h_1^u, h_2^u, \ldots, h_T^u]$, in which h_i^u stands for the i-th behavior record of user u sorted by timestamp. The goal of CTR prediction is to predict the probability for target user u to click target item v given u's historical behaviors, formulated as: $\hat{y}_{uv} = f(u, v|H_u; \theta)$, where f is the learned function with parameters θ.

2.2 Dynamic Sequential Graph Construction

Given the target user u and his/her historical behavior sequence H_u, we construct a time-evolving sequential graph up to current time t, which combines the multi-hop connectivity in graphs as well as fine-grained temporal dependency in sequences and is constructed using breadth-first search. Specifically, suppose nodes at (l)-hop are all users (vice versa for items), and the set of interactions with neighboring item nodes at $(l+1)$-hop can be defined as:

$$\mathcal{G}_{u,t}^{(l+1)} = \{(u', v, \tau)|\tau < t', (u', v, \tau) \in \mathcal{E}, (\cdot, u', t') \in \mathcal{G}_{u,t}^{(l)}\}, \quad (1)$$

where $\mathcal{G}_{u,t}^{(0)}, \mathcal{G}_{u,t}^{(1)}$ are initialized as $\{(u, t)\}$ and u's interactions with H_u respectively, $\tau < t'$ is considered to make sure interaction time t' of centroid node u' at (l)-hop is posterior to its neighbors' at $(l + l)$-hop. We set $0 \leq l \leq L$ to collect L hops neighborhood of u. Combining all depths of neighbors, we construct the dynamic sequential graph $\mathcal{G}_{u,t} = \{\mathcal{G}_{u,t}^{(0)}, \mathcal{G}_{u,t}^{(1)}, \ldots, \mathcal{G}_{u,t}^{(L)}\}$, which represents all the evolving high-order interactions with u up to time t.

2.3 Dynamic Graph Disentangling Layer

To capture the diverse interests of users from historical user behaviors, we propose the dynamic graph disentangling layer for the constructed graph. As shown in Figure 1, there are three components in the layer. By stacking L layers, we can disentangle user/item representation into different components and aggregate multiple hops of neighbor information to enrich the corresponding components, which are further used for CTR prediction.

2.3.1 Disentangled Embedding Transformation. Since different interests always represent distinct semantics, we first transform features of user/item into disentangled representation. Specifically, for each user/item node in the graph, we would like its representation to be segmented into K components representing different interests, i.e., $e_u = \{c_{u,1}, c_{u,2}, \ldots, c_{u,K}\}$. To achieve this goal, we project the feature of node $x_{u/v}$ into K latent spaces to make each component extract distinct semantics from the node feature, take user u for example (vice versa for item v):

$$c_{u,k} = \sigma(W_k \cdot x_u), \quad (2)$$

where $W_k \in \mathbb{R}^{M \times \frac{d}{K}}$ is the weight of the k-th component, d is the total representation dimension, and σ is the activation function.

2.3.2 Disentangled Embedding Propagation. For different components of the representation, we define a set of scoring matrices $S = \{S_k|\forall k \in \{1, \ldots, K\}\}$ for K different latent components, where each entry $S_k(u, v)$ denotes the intention of user u to item v under interest k. We uniformly initialize each scoring matrix as $S_k^0(u, v) = 1$ and the weight can be iteratively updated.

At iteration $i \in \{1, \ldots, I\}$, we perform embedding propagation under each component. For target user/item node in $\mathcal{G}_{u,t}$, we aggregate embeddings from its neighbors to update its k-th component

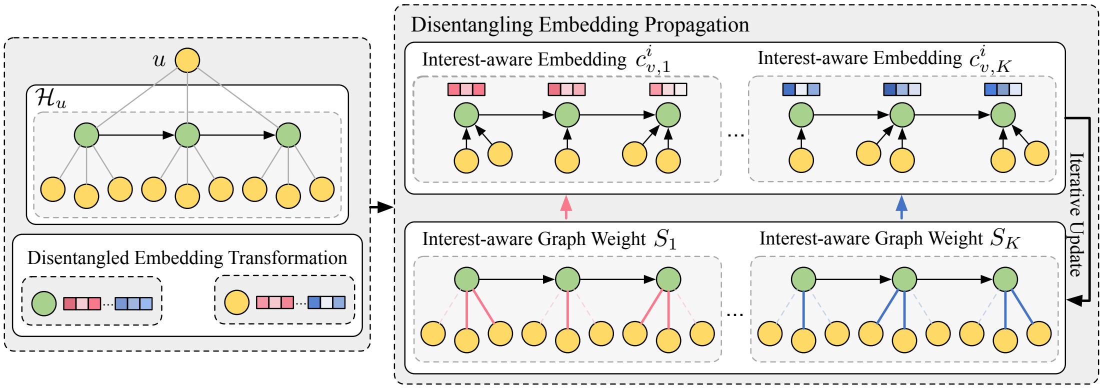


Figure 1: Illustration of dynamic graph disentangling layer.

representation. We first calculate the intention distribution over all interests by normalizing S via the softmax function:

$$\tilde{S}^i_k = \frac{\exp S^i_k(u,v)}{\sum_{k'=1}^{K} \exp S^i_{k'}(u,v)}. \tag{3}$$

This distribution serves as an attention score, which illustrates which interest is the main driving force behind a specific user interaction (u, v, τ). Then, we update target node's k-th component with the weighted sum of its neighborhood's corresponding components. Take user u as an example:

$$c^i_{u,k} = \sum_{v \in \mathcal{N}_u} \frac{\tilde{S}^i_k(u,v)}{\sqrt{D^t_k(u) \cdot D^t_k(v)}} \cdot c^{i-1}_{v,k}, \tag{4}$$

where $c^i_{u,k}$ is the refined k-th component, and $c^0_{u,k}$ is set to $c_{u,k}$. Because different nodes can have varying numbers of neighbors, we normalize the component against the degree of user u, $D^t_k(u) = \sum_{v' \in \mathcal{N}_u} \tilde{S}^i_k(u,v')$, and the degree of item v, $D^t_k(v) = \sum_{u' \in \mathcal{N}_v} \tilde{S}^i_k(u',v)$. $\mathcal{N}_u$ and $\mathcal{N}_v$ are the neighbors of u and v respectively.

2.3.3 *Iterative Intention Update.*

Intuitively, for the target user, historical items driven by the same interest tend to have similar representations under the corresponding component, which could further magnify the click preference between them. We hence iteratively adjust the scoring matrices based on the refined components:

$$S^{i+1}_k(u,v) = S^i_k(u,v) + {c^i_{u,k}}^\top \tanh(c^i_{v,k}), \tag{5}$$

where ${c^i_{u,k}}^\top \tanh(c^i_{v,k})$ measures the affinity between refined $c^i_{u,k}$ and $c^i_{v,k}$, and tanh is a nonlinear activation function.

Convergence Analysis. The essence of iterative propagation is an inference mechanism that enriches the k-th component of target node with the most relevant neighbors. Thus, it is equivalent to an expectation maximization (EM) algorithm for the mixture model. Let $Z = \{c_{u,k}\}_{k=1}^{K}$ be the enriched representation of target user u, $C = \{c_{v,k} | \forall v \in \mathcal{N}_u\}_{k=1}^{K}$. The EM algorithm maximizes $p(C;Z) = \sum_S p(C,S;Z)$, and the log-likelihood is formulated as:

$$\ln P(C;Z) = \sum_S q(S) \ln \frac{P(C,S;Z)}{q(S)} + \sum_S q(S) \ln \frac{q(S)}{p(S|C;Z)}, \tag{6}$$

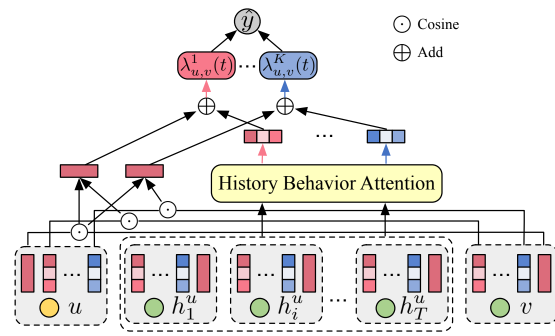


Figure 2: Illustration of MHP based CTR prediction.

where the first term is the evidence lower bound ELBO$(C,S;Z)$ and the second term is the Kullback-Leibler (KL) divergence with non-negative value. Our model alternatively updates $q(S)$ and Z by a) setting optimal $q(S)$ to $p(S|C;Z)$ so that ELBO$(C,S;Z) = \ln P(C;Z)$ for current C and Z in E step and b) maximizing ELBO w.r.t. Z in M step. Thus, for i-th iteration, $\ln P(C;Z^{(i)}) \geq \text{ELBO}(C;S^{(i)},Z^{(i)}) \geq \text{ELBO}(C;S^{(i)},Z^{(i-1)}) = \ln P(C;Z^{(i-1)})$, which improves the log-likelihood monotonically until the algorithm converges.

2.4 Mixture of Hawkes Based CTR Prediction

Hawkes process is a typical temporal point process in modeling the temporal decay effect of historical behaviors [2, 22, 26]. As shown in Figure 2, the clicking event can be driven by K latent interests. Hence, we construct the base intensity under interest k upon the last layer output of the disentangling module as $\mu_{u,v}\gamma^k_{u,v}$:

$$\begin{aligned} &\mu_{u,v} = \mathcal{F}(\tilde{e}_u \cdot \tilde{e}_v), \\ &\gamma^k_{u,v} = \text{Softplus}\left(\mathcal{F}(c^I_{u,k}, c_{v,k})\right) = \frac{1}{\beta}\log\left(1 + \exp(\beta \cdot \mathcal{F}(c^I_{u,k}, c_{v,k}))\right) \end{aligned} \tag{7}$$

where $\tilde{e}_u, \tilde{e}_v$ are the identity embedding of target user and item, $\mathcal{F}$ is the cosine similarity. Softplus is applied to preserve intensity monotonicity. Combining the historical behavior influence, the conditional intensity function of current user-item interaction (u, v, t) under interest k is defined as:

$$\lambda^k_{u,v}(t) = \mu_{u,v}\gamma^k_{u,v} + \sum_{h \in \mathcal{H}_u} \alpha^k_{v,h}\gamma^k_{v,h}\mathcal{J}(t-\tau), \tag{8}$$

Table 1: **Descriptive statistics of our three datasets.**

Dataset	#User	#Item	Interactions	Avg.SeqLen
Amazon	192,403	63,001	1,689,188	8.78
MovieLens-1M	6,040	3,706	1,000,209	158.00
Meituan	294,230	90,642	2,031,071	11.57

where $\mathcal{J}(t-\tau) = \exp(-\mathcal{D}_u(t-\tau))$ is the kernel function that models the time decay effect of historical behaviors on the current interaction, $\mathcal{D}_u$ is the trainable function of each user, and $\alpha^k_{v,h}$ is the attention score of each historical effect $\gamma^k_{v,h}$ under interest k:

$$\alpha^k_{v,h} = \frac{\exp\left(\mathcal{F}(\tilde{e}_v, \tilde{e}_h)\gamma^k_{v,h}\right)}{\sum_{h'\in\mathcal{H}_u}\exp\left(\mathcal{F}(\tilde{e}_v, \tilde{e}_{h'})\gamma^k_{v,h'}\right)}, \quad (9)$$
$$\gamma^k_{v,h} = \text{Softplus}\left(\mathcal{F}(c_{v,k}, c^I_{h,k})\right),$$

where $\tilde{e}_h$ is the identity embedding of historical item. We combine the conditional intensity of K different interests to derive the CTR prediction score $\hat{y} = \sigma(\frac{1}{K}\sum_{k=1}^{K}\lambda^k_{u,v}(t))$. Given the real label $u \in \{0,1\}$ and the predicted CTR score, we adopt binary cross-entropy for the training process, formulated as:

$$\mathcal{L} = -\sum_{(u,v,t)} y\log\hat{y} + (1-y)\log(1-\hat{y}). \quad (10)$$

3 EXPERIMENT

3.1 Experimental Setup

We evaluate the proposed method on three datasets collected from real-world platforms, namely, **Amazon**[1] [8], **MovieLens-1M**[2] and **Meituan**. For MovieLens-1M, we mark samples with ratings no less than 3 as positive; for Amazon and Meituan, we keep the clicked samples as positive samples. The statistics of the datasets are summarized in Table 1. We sort each user's interactions in chronological order and reserve the last interaction for evaluation, while the remaining interactions are used for training. The evaluation set is then split into halves randomly as the validation set and the test set, respectively.

To demonstrate the effectiveness of our model, we compare DisenCTR with three classes of methods: (A) Sequential-based methods, including DIN [25] and DIEN [24]; (B) Multi-interest methods, including MIND [11], ComiRec-DR [1] and ComiRec-SA [1]; (C) GNN-based methods, including NGCF [16] and LightGCN [9]. We adopt AUC and Logloss as performance metrics, which are two of the most widely used metrics for CTR prediction. The embedding size is fixed to 128 for all models. For DisenCTR, we set $L = 2, \beta = 1$ and K is tuned amongst $\{1, 2, 4, 8, 16\}$. For baseline methods, we apply a grid search for optimal hyper-parameters.

3.2 Performance Comparison

The performances of all compared methods are summarized in Table 2. Multi-interest methods outperform most sequential-based

[1] http://jmcauley.ucsd.edu/data/amazon/
[2] https://grouplens.org/datasets/movielens/

Table 2: The performance of DisenCTR (Ours) and other baseline methods over three datasets.

Model	Amazon		MovieLens-1M		Meituan	
	AUC	Logloss	AUC	Logloss	AUC	Logloss
DIN	0.7488	0.5888	0.8500	0.5175	0.7245	0.6123
DIEN	0.7478	0.5686	0.9041	0.4639	0.7937	0.5617
MIND	0.7726	0.5612	0.8273	0.5074	0.7486	0.5746
ComiRec-DR	0.7256	0.6919	0.8073	0.6921	0.6935	0.6376
ComiRec-SA	0.7792	0.5785	0.8590	0.4704	0.7512	0.5992
NGCF	0.7763	0.5633	0.8910	0.4340	0.7875	0.5840
LightGCN	0.7719	0.5746	0.8631	0.4917	0.7500	0.6254
Ours	**0.8123**	**0.5326**	**0.9085**	**0.3841**	**0.8042**	**0.5311**

methods, demonstrating the potency of disentangled representation to extract diverse user interests for CTR prediction. GNN-based methods, which introduce embedding propagation to capture high-order collaborative signals, also achieve better performance than sequential-based methods. DisenCTR combines the strength of all the compared methods. It captures the evolving multi-interest of users and performs consistently better than all baseline methods on all three datasets. The significant improvement also indicates the positive effect on achieving better disentangled representations with the constructed dynamic sequential graph for CTR prediction.

3.3 Analysis of DisenCTR

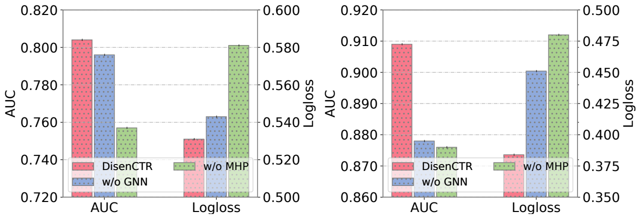
In this section, we perform ablation studies to show the necessity of the graph structure and point-process modeling. We also verify the effectiveness of the proposed dynamic graph disentangling layer.

3.3.1 Ablation study. We conduct an ablation study to verify the effectiveness of DisenCTR. As shown in Figure 3, performance suffers when we set the number of disentangling layers to 0 (w/o GNN) or replace the Mixture of Hawkes Process with multilayer perceptron (w/o MHP), especially w/o MHP. The results indicate that time-evolving sequential graph and the MHP are both indispensable for modeling dynamic user interests: the former provides rich user dynamics, while the latter captures the temporal effect of historical behaviors from the graph.

3.3.2 Effectiveness of Disentangled Representation. To investigate whether DisenCTR can benefit from disentangled representation, we study its performance with varying number of latent components K in dynamic graph disentangling layer in the range of $\{1, 2, 4, 8, 16\}$. Figure 4 summarizes the experimental results. When $K = 1$, the model degenerates into an entangled-based model with poor performance. DisenCTR achieves optimal performance at $K = 2$ in Meituan and $K = 4$ in MovieLens-1M, which indicates that disentangled representation better captures users' diverse interests.

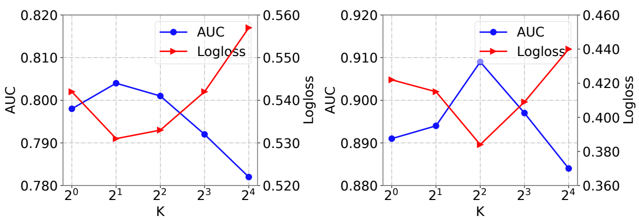
4 CONCLUSION

The inherent diversity and fluidity of real-time user interests pose a huge challenge to CTR prediction. In this paper, we exploit diverse aspects of users' evolving real-time interests, and propose



(a) Meituan **(b) MovieLens-1M**

Figure 3: Ablation study of the DisenCTR, *w/o* means we remove corresponding module from the original DisenCTR.



(a) Meituan **(b) MovieLens-1M**

Figure 4: Performance w.r.t. different number of latent components for disentangled representation.

a dynamic graph-based disentangled representation framework for CTR prediction (DisenCTR). Specifically, we construct a novel dynamic sequential user-item graph and propose a disentangled representation learning module on the graph to extract diverse user interests. We further exploit the fluidity of user interests with Mixture of Hawkes Process (MHP), which captures the multi-aspect temporal effects of past behaviors. Experiments show the effectiveness of our proposed model and the important role of disentangled representation in CTR prediction.

ACKNOWLEDGMENTS

This research is partially supported by National Key Research and Development Program of China with Grant No. 2018AAA0101902, the National Natural Science Foundation of China (NSFC Grant 62106008 & 62006004) as well as Meituan.

REFERENCES

[1] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable multi-interest framework for recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2942–2951.

[2] Yutian Chang, Guannan Liu, Yuan Zuo, and Junjie Wu. 2021. Multi-Aspect Temporal Network Embedding: A Mixture of Hawkes Process View. *arXiv preprint arXiv:2105.08566* (2021).

[3] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.

[4] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.

[5] Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. 2019. Deep session interest network for click-through rate prediction. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 2301–2307.

[6] Wei Guo, Rong Su, Renhao Tan, Huifeng Guo, Yingxue Zhang, Zhirong Liu, Ruiming Tang, and Xiuqiang He. 2021. Dual Graph enhanced Embedding Neural Network for CTR Prediction. In *Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 496–504.

[7] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in neural information processing systems*. 1024–1034.

[8] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*. 507–517.

[9] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.

[10] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations*.

[11] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2615–2623.

[12] Feng Li, Zhenrui Chen, Pengjie Wang, Yi Ren, Di Zhang, and Xiaoyu Zhu. 2019. Graph intention network for click-through rate prediction in sponsored search. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 961–964.

[13] Zekun Li, Zeyu Cui, Shu Wu, Xiaoyu Zhang, and Liang Wang. 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*. 539–548.

[14] Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Practice on long sequential user behavior modeling for click-through rate prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2671–2679.

[15] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *Proceedings of the 6th International Conference on Learning Representations*.

[16] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 165–174.

[17] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled graph collaborative filtering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1001–1010.

[18] Yifan Wang, Yongkang Li, Shuai Li, Weiping Song, Jiangke Fan, Shan Gao, Ling Ma, Bing Cheng, Xunliang Cai, Sheng Wang, and Ming Zhang. 2022. Deep Graph Mutual Learning for Cross-domain Recommendation. In *27th International Conference on Database Systems for Advanced Applications*.

[19] Yifan Wang, Yiping Song, Shuai Li, Chaoran Cheng, Wei Ju, Ming Zhang, and Sheng Wang. 2022. DisenCite: Graph-based Disentangled Representation Learning for Context-specific Citation Generation. In *36th AAAI Conference on Artificial Intelligence*.

[20] Yifan Wang, Suyao Tang, Yuntong Lei, Weiping Song, Sheng Wang, and Ming Zhang. 2020. Disenhan: Disentangled heterogeneous graph attention network for recommendation. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*. 1605–1614.

[21] Zhibo Xiao, Luwei Yang, Wen Jiang, Yi Wei, Yi Hu, and Hao Wang. 2020. Deep multi-interest network for click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*. 2265–2268.

[22] Shuang-Hong Yang and Hongyuan Zha. 2013. Mixture of mutually exciting processes for viral diffusion. In *Proceedings of the 30th International Conference on Machine Learning*. 1–9.

[23] Chang Zhou, Jinze Bai, Junshuai Song, Xiaofei Liu, Zhengchao Zhao, Xiusi Chen, and Jun Gao. 2018. Atrank: An attention-based user behavior modeling framework for recommendation. In *32nd AAAI Conference on Artificial Intelligence*. 4564–4571.

[24] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep interest evolution network for click-through rate prediction. In *33rd AAAI Conference on Artificial Intelligence*. 5941–5948.

[25] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1059–1068.

[26] Yuan Zuo, Guannan Liu, Hao Lin, Jia Guo, Xiaoqian Hu, and Junjie Wu. 2018. Embedding temporal network via neighborhood formation. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 2857–2866.