Neighbour Interaction based Click-Through Rate Prediction via Graph-masked Transformer

Erxue Min*
erxue.min@gmail.com
National Centre for Text Mining,
Department of Computer Science,
The University of Manchester
United Kingdom

Yu Rong†
Tingyang Xu
Yatao Bian
yu.rong@hotmail.com
tingyangxu@tencent.com
yatao.bian@gmail.com
Tencent AI Lab
China

Da Luo
Kangyi Lin
lodaluo@tencent.com
plancklin@tencent.com
Weixin Open Platform, Tencent
China

Junzhou Huang
jzhuang@uta.edu
Department of Computer Science and Engineering, University of Texas at Arlington
United States

Sophia Ananiadou
Sophia.Ananiadou@manchester.ac.uk
National Centre for Text Mining,
Department of Computer Science,
The University of Manchester
United Kingdom

Peilin Zhao
masonzhao@tencent.com
Tencent AI Lab
China

SIGIR 2022

Reported by Nengqiang Xiang

Introduction

Existing methods often limited by the recommender system's direct exposure and inactive interactions, and thus fail to mine all potential user interests.

To tackle these problems, we propose Neighbor Interaction based CTR prediction (NI-CTR), which considers this task under a Heterogeneous Information Network (HIN) setting.

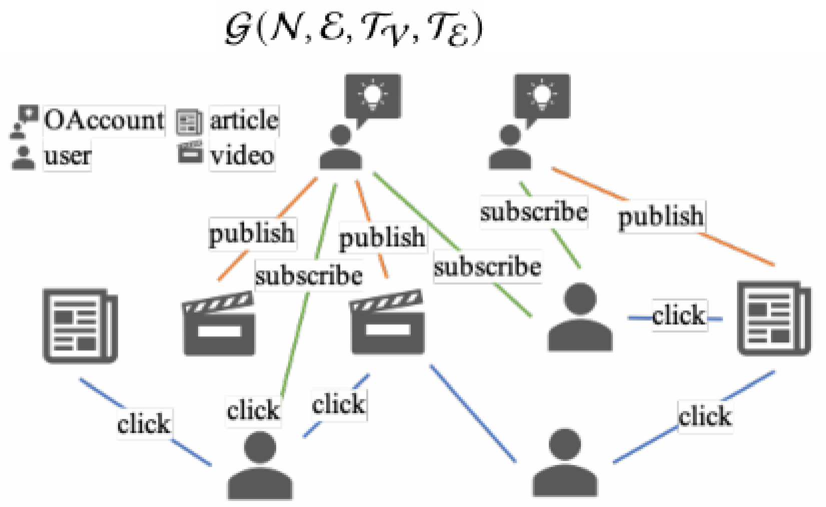


Figure 1: An illustration of the constructed HIN. It contains four kinds of nodes (OAccount, article, user and video) and three kinds of edges (click, publish and subscribe).

Method

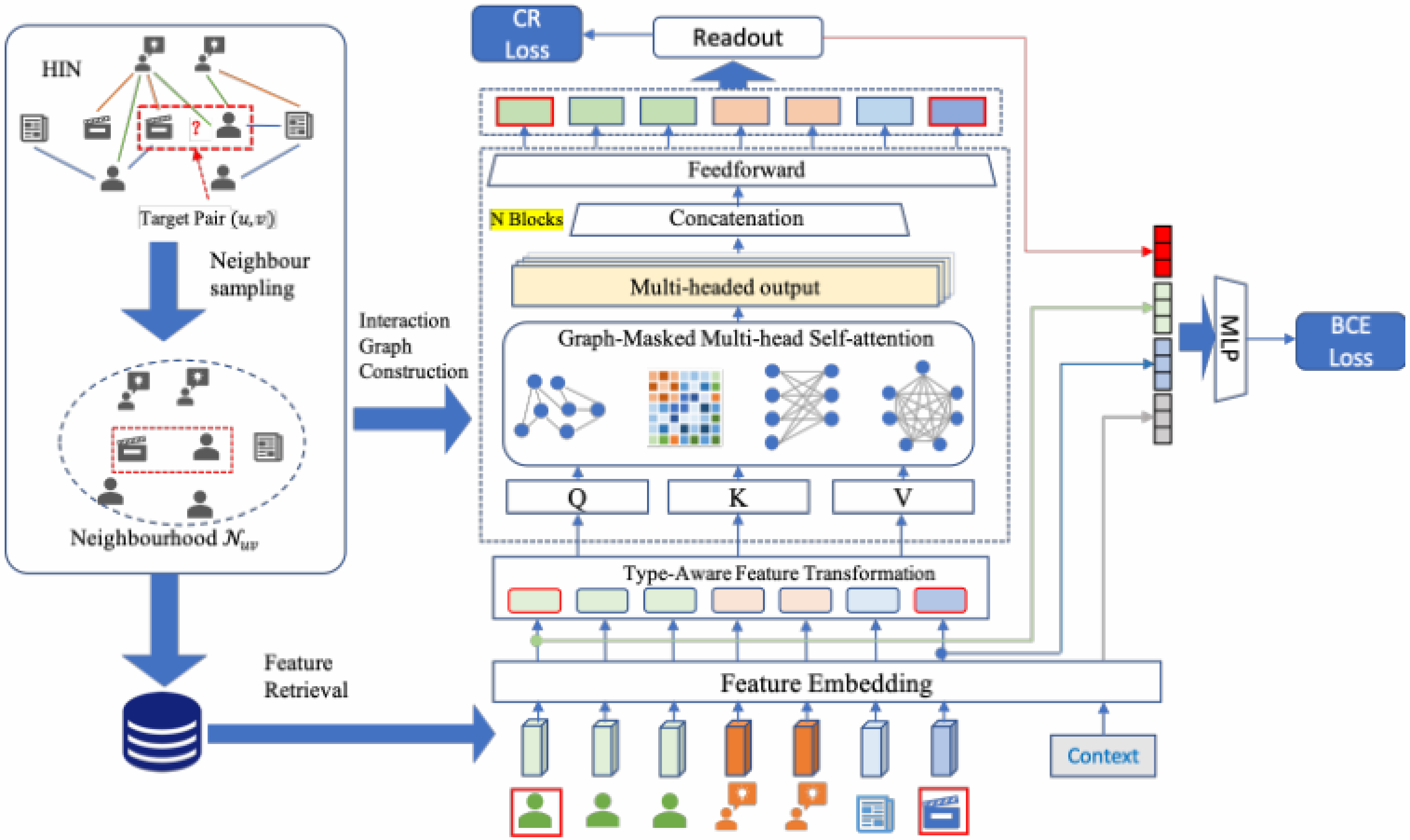
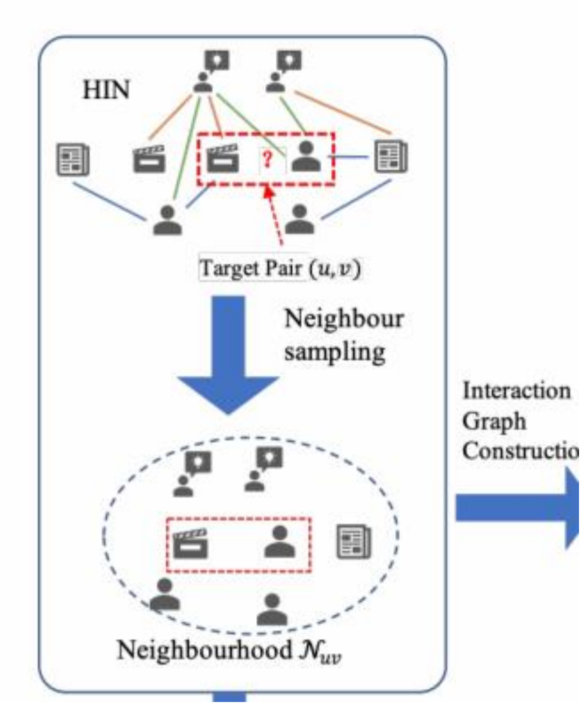


Figure 2: Overview of NI-CTR. Given a target user-item pair, we first perform neighbour sampling in the HIN to obtain associated neighbours. Then we retrieve the corresponding entity features and construct interaction graphs based on the neighbours. After that, we apply a Graph-Masked Transformer to encode both the feature information and topological information. A binary cross-entropy loss and a consistency regularization loss are combined to optimize the network.

Method



Problem Definition:

users: $\boldsymbol{u} = \{u_1, u_2, \ldots u_M\}$

items: $I = \{v_1, v_2, \ldots v_N\}$

user-item interactions: $\boldsymbol{Y} \in \mathbb{R}^{M \times N}$

Neighbour Sampling in HIN:

GHNSampling iteratively samples a list of nodes for a target node r from one hop to further. Let $\{s_k\}_{k=1}^{|\mathcal{T}_V|}$ denotes the budget sampling sizes for each node type, C_k denotes the neighbours of type k we have already sampled. GHNSampling greedily retrieves nodes from 1-hop to further until meeting the budget. In l-th hop, we retrieve all the neighbours of nodes in $(l-1)$-th hop as $\mathcal{B}^l$, with $\mathcal{B}_k^l \subset \mathcal{B}^l$ as retrieved nodes of type k. For node $t \in \mathcal{B}^l$, we calculate the number of nodes it connects in the sampled node set C as $f_t = |\{s|(s,t) \in \mathcal{E}, s \in C\}|$. If $|\mathcal{B}_k^l| > s_k - |C_k|$, we sample $s_k - |C_k|$ nodes from $\mathcal{B}_k^l$ with the probability proportional to f_t. We iteratively run the steps until budges of all node types are met.

Method

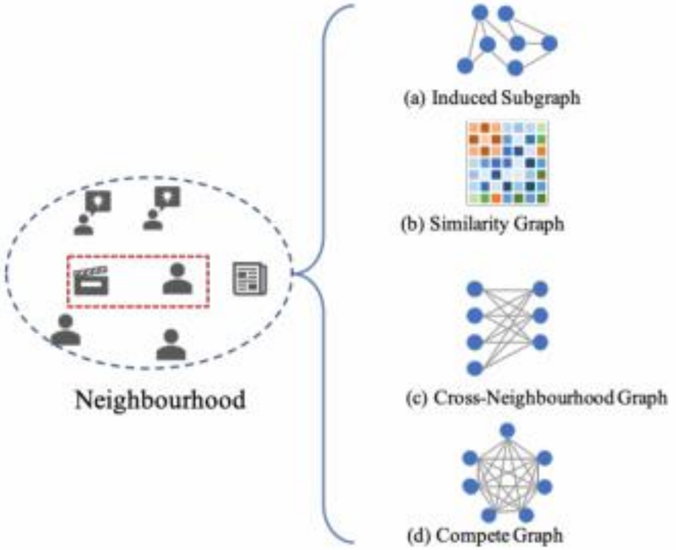


Figure 3: Four types of interaction graphs for neighbourhood modeling, which contain natural interactions, feature similarities, cross-neighbourhood interactions and all pair-wise interactions.

Construction of Local Interaction Graphs:

Induced Subgraph G^{I}_{uv}

Similarity Subgraph G^{S}_{uv}

$$\mathrm{sim}(i,j) = \frac{\mathbf{f}_i[g(t(i),t(j))] \cdot \mathbf{f}_j[g(t(j),t(i))]}{\|\mathbf{f}_i[g(t(i),t(j))]\| \cdot \|\mathbf{f}_j[g(t(j),t(i))]\|},$$

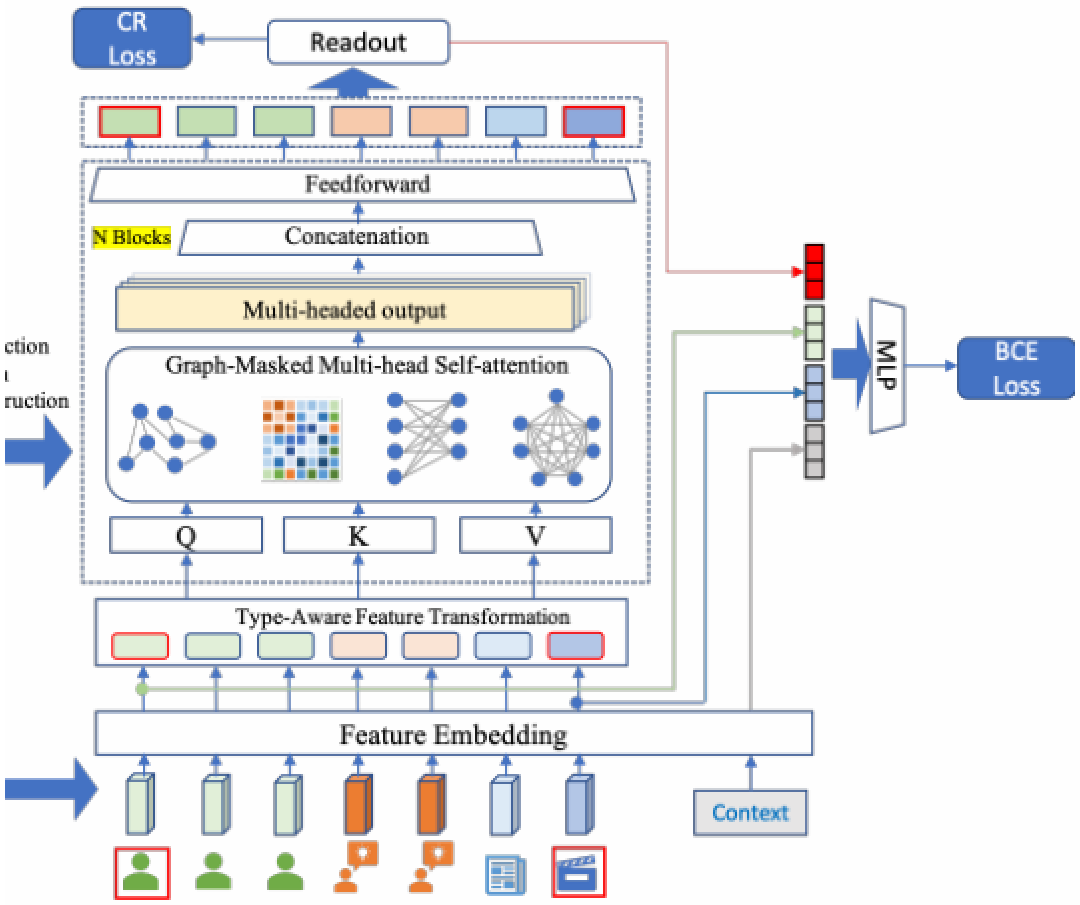
Cross Neighbourhood Subgraph G^{C}_{uv}

$$\mathcal{G}^{C}_{uv} = \{(s,t) | s \in \mathcal{N}_u, t \in \mathcal{N}_v\}$$

$$\mathcal{N}_u \cap \mathcal{N}_v = \emptyset$$

Complete Subgraph G^{P}_{uv}

Method



Heterogeneous Node Feature Transformation layer:

$$\mathbf{x}_i = \mathbf{W}_i \mathbf{f}_i$$

User or item: $[\mathbf{f}_1, \cdots, \mathbf{f}_k]$

$$\mathbf{h}_i = \text{Linear}^{t(i)}(\mathbf{x}_i). \quad (1)$$

Graph-masked Multi-head Self-attention:

$$e_{ij} = \frac{(\mathbf{Q}\mathbf{h}_i)^\top (\mathbf{K}\mathbf{h}_i)}{\sqrt{d}},$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{n} \exp(e_{ik})}, \quad (2)$$

$$\mathbf{z}_i = \sum_{j=1}^{n} \alpha_{ij}(\mathbf{V}\mathbf{h}_i),$$

$$e_{ij} = f_m\left(\frac{(\mathbf{Q}\mathbf{h}_i)^\top (\mathbf{K}\mathbf{h}_i)}{\sqrt{d}}, \mathbf{M}_{ij}\right), \quad (3)$$

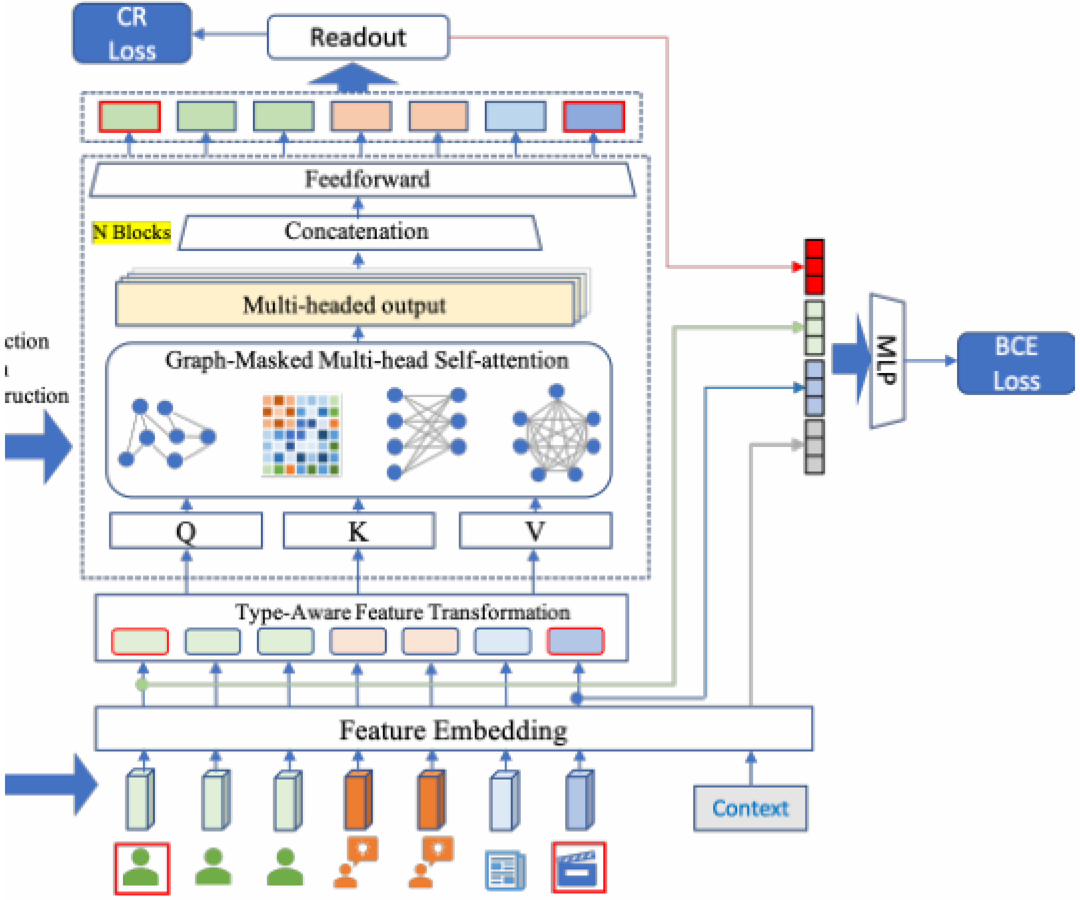
$$f_m(x, \lambda) = \begin{cases} \lambda x & \lambda \neq 0 \\ -\infty & \lambda = 0. \end{cases} \quad (4)$$

$$z_i = \text{FFN}(\mathbf{W}^O \text{Concat}(\mathbf{z}_i^1, \cdots, \mathbf{z}_i^H)), \quad (5)$$

$$\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{|\mathcal{N}_{uv}|}\}.$$

$$\mathbf{g}_{uv} = \text{Readout}(\mathbf{Z}), \quad (6)$$

Method



Classification and Optimization:

$$\mathbf{z}^{o} = \mathrm{Concat}(\mathbf{g}_{uv}, \mathbf{x}_{u}, \mathbf{x}_{v}, \mathbf{C}). \quad (7)$$

$$\hat{y}_{uv} = \sigma(f_{mlp}(\mathbf{z}^{o}, \theta)) \quad (8)$$

$$\mathcal{L}_{\mathrm{BCE}} = \frac{1}{S} \sum_{<u,v>\in\mathcal{D}} \sum_{s=1}^{S} (y_{uv}\log\hat{y}_{uv}^{s} + (1-y_{uv})\log(1-\hat{y}_{uv}^{s})), \quad (9)$$

$$\mathcal{L}_{\mathrm{CR}} = \frac{1}{S} \sum_{<u,v>\in\mathcal{D}} \sum_{s=1}^{S} \frac{1}{d_g} ||\hat{\mathbf{g}}_{uv}^{s} - \bar{\mathbf{g}}_{uv}^{s}|| \quad (10)$$

$$\bar{\mathbf{g}}_{uv}^{s} = \frac{1}{S} \sum_{s=1}^{S} \hat{\mathbf{g}}_{uv}^{s}$$

$$\mathcal{L} = \mathcal{L}_{\mathrm{BCE}} + \gamma \mathcal{L}_{\mathrm{CR}} \quad (11)$$

Experiments

Table 1: Statistics of the WeChat HIN

Node type	Count	Fields	Features[a]
User	728M	75	147572
OAcc	369K	95	187323
Article	74M	26	148284
Video	846K	23	134758

Edge Type	Count	Ave Src Deg	Ave Dst Deg
user-video	998M	1.35	1167.08
user-article	11.3B	15.53	151.25
user-OAcc	33.9B	46.67	9726.76
OAcc-video	50M	1.43	5.91
OAcc-article	74.7M	21.39	1.0

[a]Here we do not count in any entity (user/OAcc/article/video) ids, which would be extremely large.

Experiments

Table 2: Results on offline datasets

Category	Model	WC_FULL		WC_SMALL		Tmall	
		AUC	Logloss	AUC	Logloss	AUC	Logloss
FI	DeepFM	0.7009	0.2379	0.7022	0.2365	0.9012	0.1999
	xDeepFM	0.7021	0.2370	0.7042	0.2354	0.9023	0.1978
UIM	DIN	0.7042	0.2345	0.7073	0.2320	0.9034	0.1954
	DIEN	0.7043	0.2347	0.7069	0.2334	0.9045	0.1943
	DMR	0.7098	0.2280	0.7089	0.2310	0.9065	0.1926
GNN	GraphSAGE	0.7032	0.2366	0.7056	0.2378	0.9234	0.1789
	GAT	0.7130	0.2214	0.7145	0.2210	0.9245	0.1776
	RGCN	0.7078	0.2289	0.7101	0.2265	0.9201	0.1801
	HAN	0.7015	0.2378	0.7041	0.2399	0.9180	0.1823
	NIRec	0.7149	0.2200	0.7167	0.2197	0.9246	0.1775
Transformer	Transformer	0.7200	0.2174	0.7260	0.2075	0.9339	0.1700
	Graph-Trans	0.7201	0.2175	0.7277	0.2063	0.9321	0.1715
	Graph-BERT	0.7211	0.2165	0.7290	0.2054	0.9345	0.1693
	GMT	**0.7290**	**0.2103**	**0.7360**	**0.2014**	**0.9410**	**0.1603**

Experiments

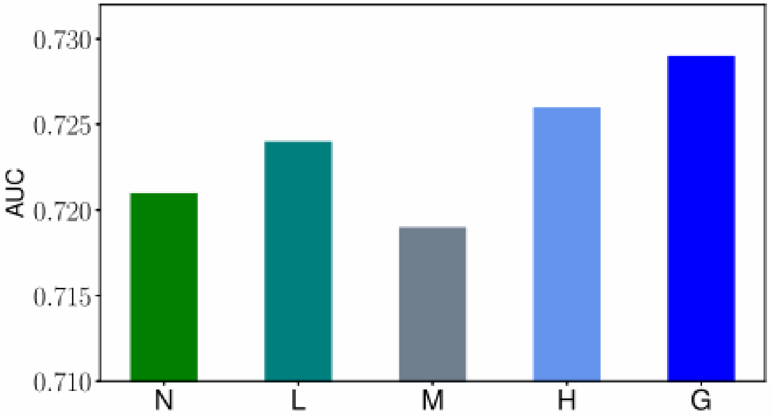


Figure 4: Results of graph sampling methods. N: Node-wise sampling; L: Layer-wise sampling; M: Metapath sampling; H: HGSampling; G: GHSampling.

Table 3: Ablation results of each module on WC_FULL dataset

Modules					WC_FULL	
$\mathcal{G}_{uv}^{I}$	$\mathcal{G}_{uv}^{S}$	$\mathcal{G}_{uv}^{C}$	$\mathcal{G}_{uv}^{P}$	CR Loss	AUC	Logloss
✓	✓	✓	✓	✓	**0.7290**	**0.2103**
✓				✓	0.7180	0.2193
	✓			✓	0.7179	0.2193
		✓		✓	0.7211	0.2154
			✓	✓	0.7203	0.2157
	✓	✓	✓	✓	0.7243	0.2132
✓		✓	✓	✓	0.7252	0.2126
✓	✓		✓	✓	0.7237	0.2149
✓	✓	✓		✓	0.7263	0.2123
✓	✓	✓	✓		0.7274	0.2116
					0.7200	0.2174

Experiments

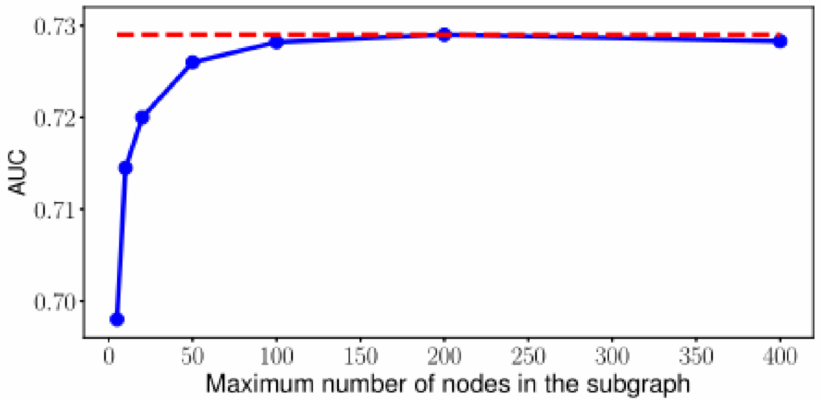


Figure 5: Results of different maximum numbers of sampled nodes in the subgraph.

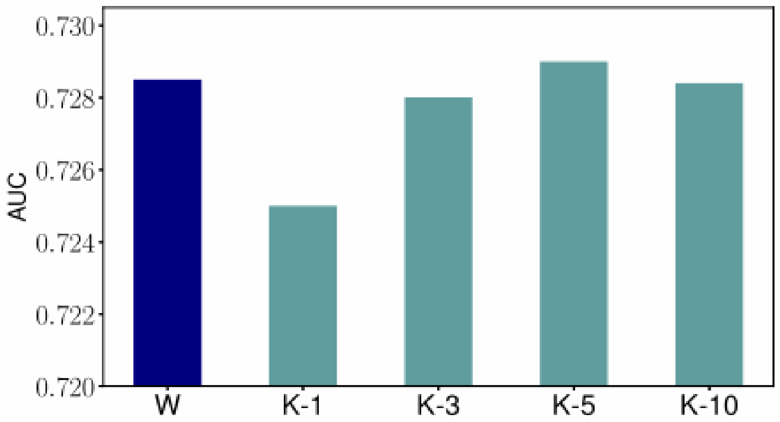


Figure 6: Results of different similarity graphs, where W denotes weighted similarity graph, and K-n denotes k-NN similarity graph with $k = n$.

Experiments

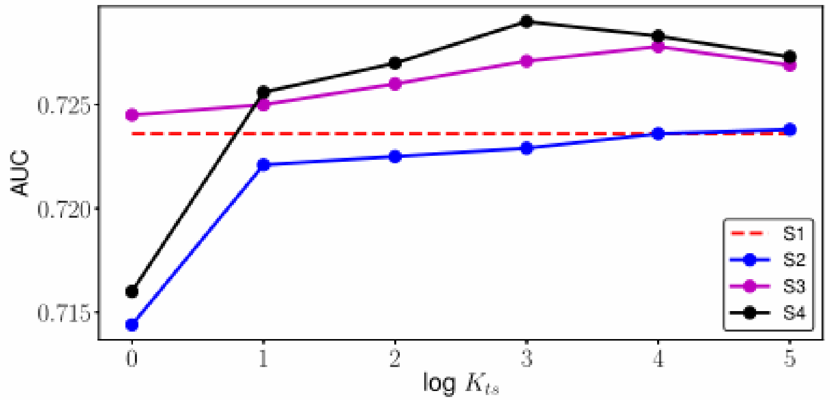


Figure 7: Results of different feature exploitation strategies with varied threshold value K_{ts}.

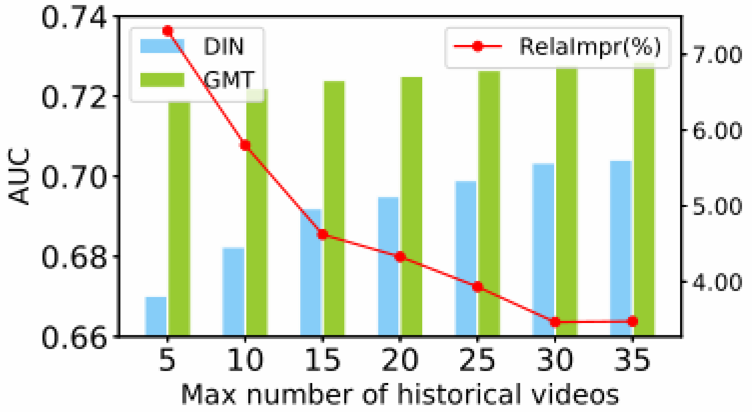


Figure 8: Cold-start analysis result.

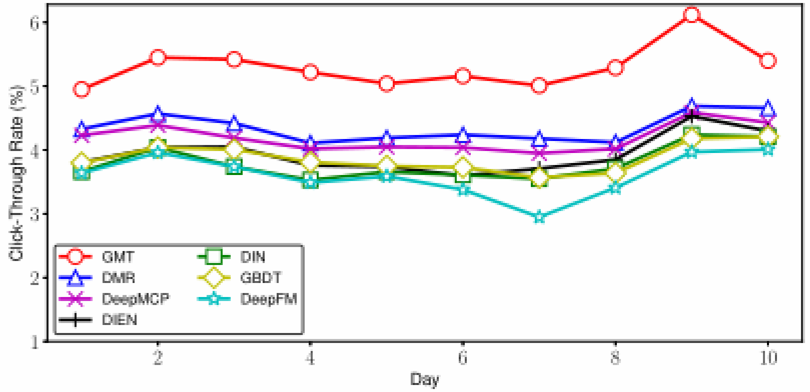


Figure 9: Results from Online A/B test during 10 consecutive days. The red curve is our method.

Thanks