Group Identification via Transitional Hypergraph Convolution with Cross-view Self-supervised Learning

Mingdai Yang
University of Illinois at Chicago
Chicago, USA
myang72@uic.edu

Zhiwei Liu
Salesforce AI Research
Palo Alto, USA
zhiweiliu@salesforce.com

Liangwei Yang
lyang84@uic.edu
University of Illinois at Chicago
Chicago, USA

Xiaolong Liu
Chen Wang
xliu262@uic.edu
cwang266@uic.edu
University of Illinois at Chicago
Chicago, USA

Hao Peng*
School of Cyber Science and Technology, Beihang University, Beijing, China
Yunnan Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology, Kunming, China
penghao@buaa.edu.cn

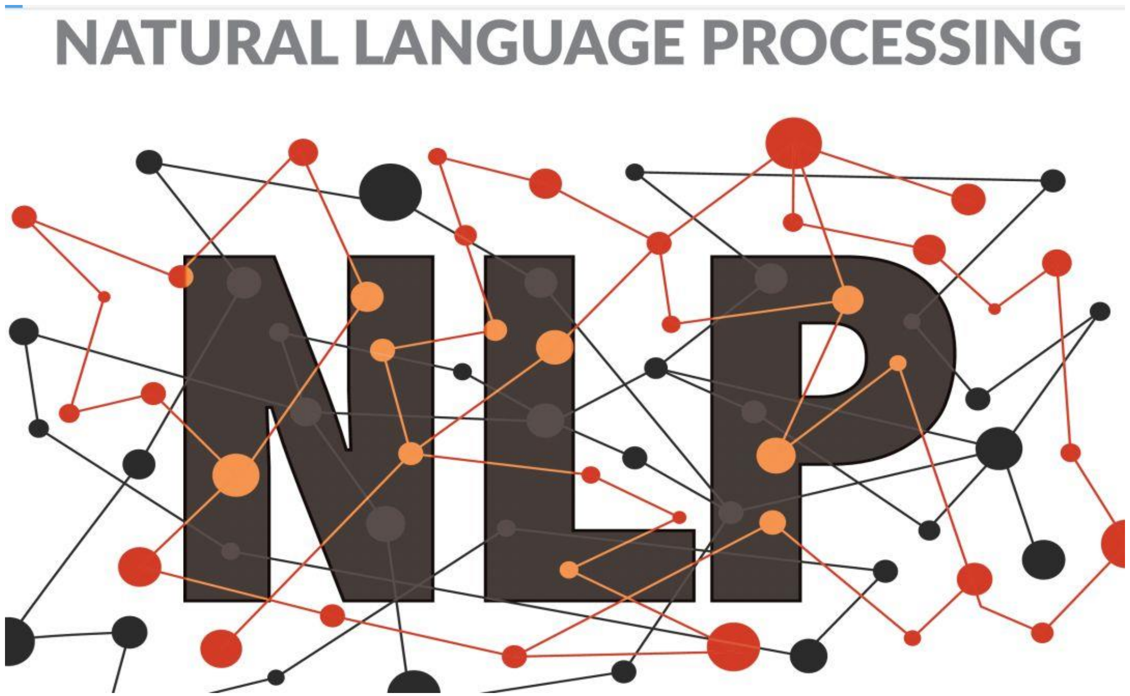
Philip S. Yu
University of Illinois at Chicago
Chicago, USA
psyu@uic.edu

code: https://github.com/mdyfrank/GTGS

2023. 9. 28 • ChongQing

2023_CIKM

Reported by Junhao Cao



1.Introduction

2.Method

3.Experiments

Introduction

existing hypergraph convolution methods directly aggregate the information from neighbors connected by hyperedges, while neglecting intrinsic information of hyperedges.

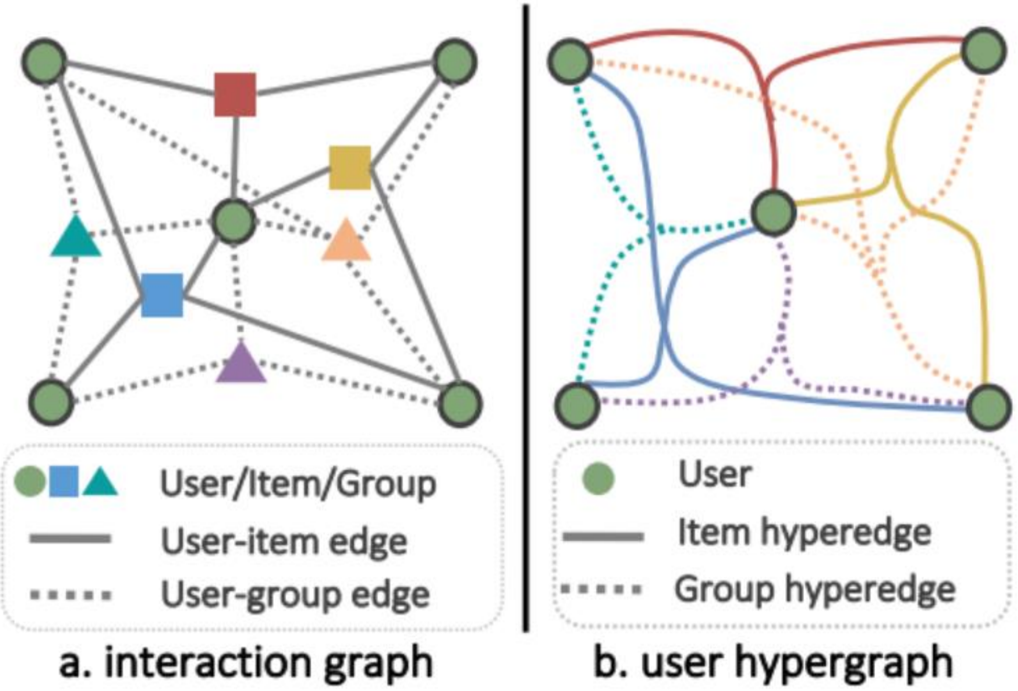


Figure 1: Toy example of how to construct a GI hypergraph from user-group-item interaction graph.

Method

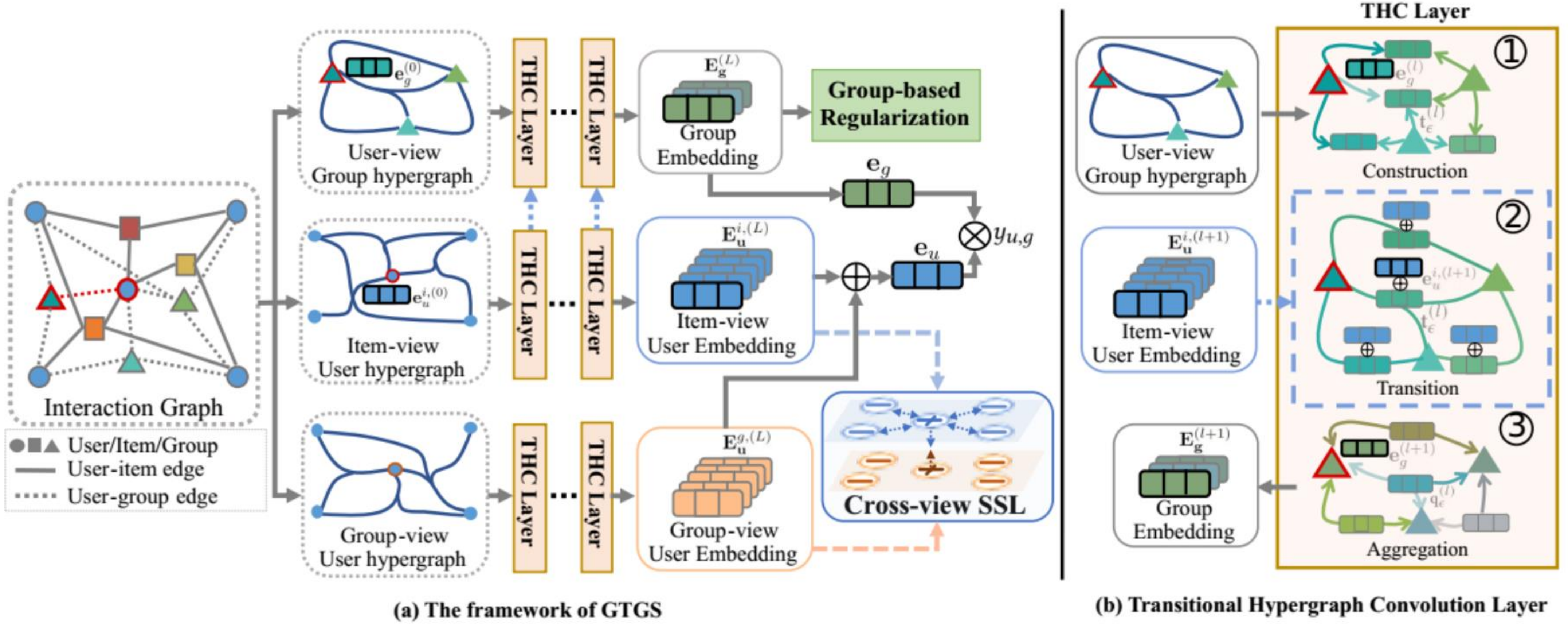
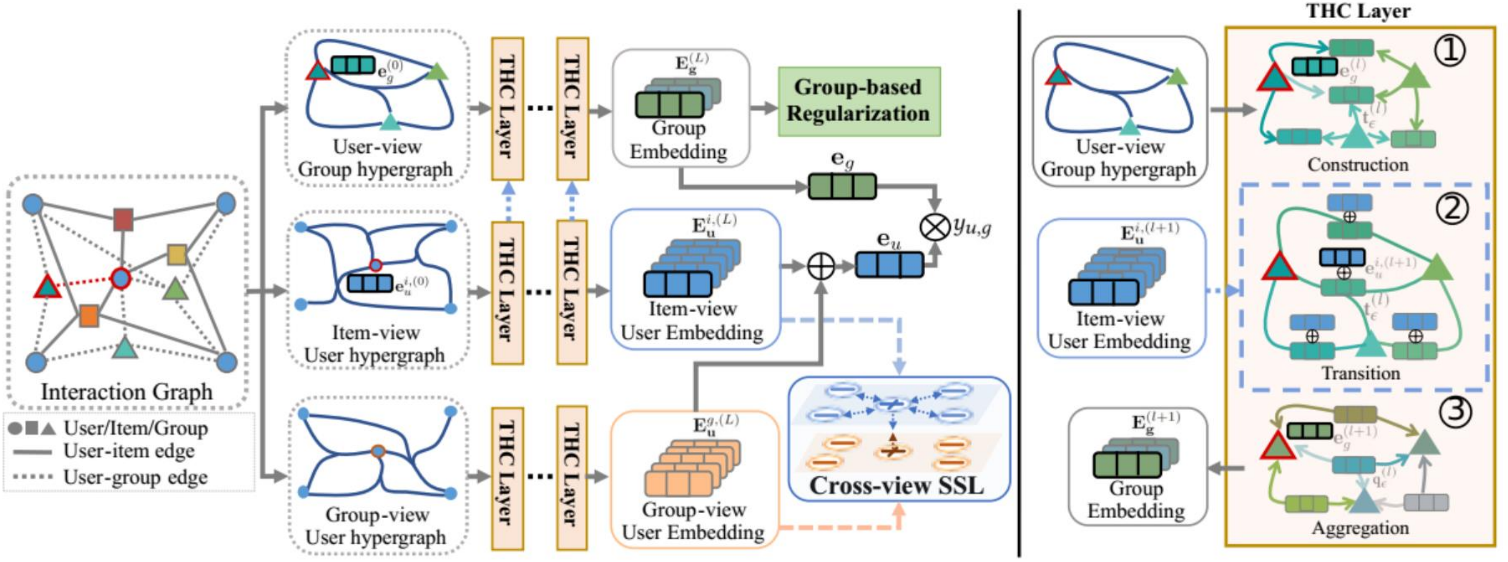


Figure 2: (a) The overall framework of GTGS. First, we construct three hypergraphs and apply THC layers to them. Next, we conduct group-based regularization on the output group embeddings, and employ cross-view SSL to optimize item-view and group-view user embeddings. At last, the inner product of group embedding and item-view user embedding is calculated for prediction; (b) the illustration of the Transitional Hypergraph Convolution (THC) layer.

Method



(a) The framework of GTGS

(b) Transitional Hypergraph Convolution Layer

Figure 2: (a) The overall framework of GTGS. First, we construct three hypergraphs and apply THC layers to them. Next, we conduct group-based regularization on the output group embeddings, and employ cross-view SSL to optimize item-view and group-view user embeddings. At last, the inner product of group embedding and item-view user embedding is calculated for prediction; (b) the illustration of the Transitional Hypergraph Convolution (THC) layer.

$$\mathbf{t}_\epsilon^{(l)} = \frac{1}{|\mathcal{N}_\epsilon|}\sum_{g'\in\mathcal{N}_\epsilon}\mathbf{e}_{g'}^{(l)}, \tag{1}$$

$$\mathbf{q}_\epsilon^{(l)} = \text{Transition}(\mathbf{t}_\epsilon^{(l)}, \gamma\mathbf{c}_\epsilon^{(l)}), \tag{2}$$

$$\mathbf{e}_g^{(l+1)} = \frac{1}{|\mathcal{N}_g|}\sum_{\epsilon\in\mathcal{N}_g}\mathbf{q}_\epsilon^{(l)}, \tag{3}$$

$$\begin{aligned}\mathbf{E}_{\mathrm{g}}^{(l+1)} &= \mathrm{THC}(\mathbf{E}_{\mathrm{g}}^{(l)}, \mathbf{H}, \gamma\mathbf{C}_\epsilon^{(l)}) \\ &= \mathbf{D}^{-1}\mathbf{H}\cdot\text{Transition}(\mathbf{B}^{-1}\mathbf{H}^\top\mathbf{E}_{\mathrm{g}}^{(l)}, \gamma\mathbf{C}^{(l)}),\end{aligned} \tag{4}$$

$$\begin{aligned}\mathbf{E}_{\mathbf{u}}^{i,(L)} &= \mathrm{THC}^L(\mathbf{E}_{\mathbf{u}}^{i,(0)}, \mathbf{U}_i, 0), \\ \mathbf{E}_{\mathbf{u}}^{g,(L)} &= \mathrm{THC}^L(\mathbf{E}_{\mathbf{u}}^{g,(0)}, \mathbf{U}_g, 0),\end{aligned} \tag{5}$$

$$\mathbf{E}_u = \beta\mathbf{E}_{\mathbf{u}}^i + (1-\beta)\mathbf{E}_{\mathbf{u}}^g, \tag{6}$$

Method

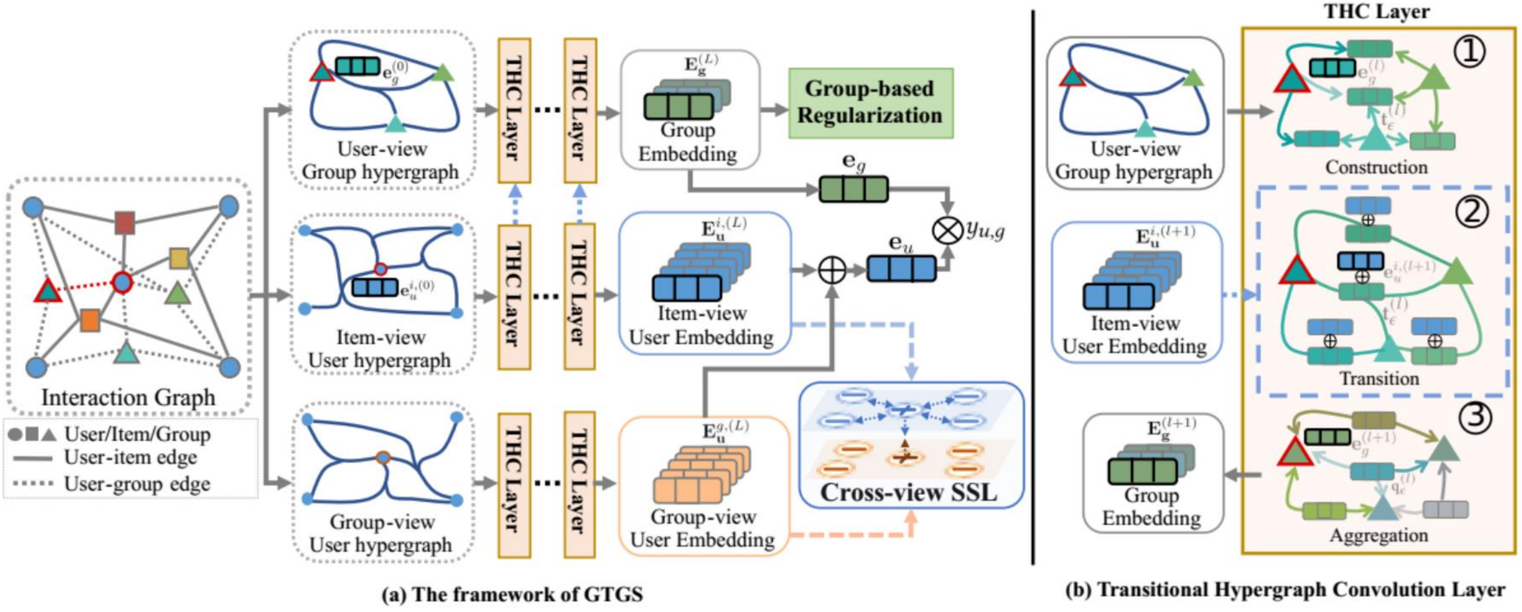


Figure 2: (a) The overall framework of GTGS. First, we construct three hypergraphs and apply THC layers to them. Next, we conduct group-based regularization on the output group embeddings, and employ cross-view SSL to optimize item-view and group-view user embeddings. At last, the inner product of group embedding and item-view user embedding is calculated for prediction; (b) the illustration of the Transitional Hypergraph Convolution (THC) layer.

$$\mathcal{L}_{cssl}^{user} = \sum_{u=0}^{|\mathcal{U}|} -\log \frac{\exp(\mathrm{sim}(\mathbf{e}_u^i, \mathbf{e}_u^g)/\tau_\mathbf{u})}{\sum_{v=0}^{|\mathcal{U}|} \exp(\mathrm{sim}(\mathbf{e}_u^i, \mathbf{e}_v^g))/\tau_\mathbf{u})}, \quad (7)$$

$$\mathcal{L}_{reg}^{group} = \sum_{g=0}^{|\mathcal{G}|} -\log \frac{\exp(1/\tau_\mathbf{g})}{\sum_{k=0}^{|\mathcal{G}|} \exp(\mathrm{sim}(\mathbf{e}_g, \mathbf{e}_k)/\tau_\mathbf{g})}, \quad (8)$$

$$y_{u,g} = \mathbf{e}_u \cdot \mathbf{e}_g, \quad (9)$$

$$\mathcal{L}_{bpr} = \sum -\log \sigma(\hat{y}_{u,g} - \hat{y}_{u,g'}), \quad (10)$$

$$\mathcal{L} = \mathcal{L}_{bpr} + \lambda(\mathcal{L}_{cssl}^{user} + \mathcal{L}_{reg}^{group}) + \lambda_\Theta \|\Theta\|_2^2, \quad (11)$$

Table 1: The statistics of datasets.

Dataset	Steam	Beibei	Weeplaces
# users	19,608	11,487	8,550
# groups	46,587	4,035	8,535
# items	3,951	13,814	22,357
# user-group edges	105,271	20,972	16,529
# user-item edges	1,209,979	105,210	152,258
Avg. # groups/user	5.37	1.83	1.93
Avg. # users/group	2.26	5.2	1.94
Avg. # items/user	61.71	9.16	18.44
Avg. # users/item	306.25	7.62	6.81

Experiment

Table 2: Performance comparison on three datasets.

Dataset	Steam				Beibei				Weeplaces			
Metric	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20
LGCN	0.0938	0.1236	0.0734	0.0803	0.1042	0.1329	0.0741	0.0821	0.1176	0.1506	0.0705	0.0790
SGL	0.1290	0.1565	0.0879	0.0856	0.1084	0.1377	0.0779	0.0883	0.1192	0.1486	0.0707	0.0784
ENMF	0.1075	0.1386	0.0774	0.0868	0.1154	0.1560	0.0826	0.0959	0.1310	0.1720	0.0734	0.0854
HGNN	0.0398	0.0515	0.0407	0.0463	0.0945	0.1279	0.0864	0.1070	0.0967	0.1367	0.0468	0.0605
HCCF	0.0765	0.0890	0.0698	0.0771	0.0860	0.1236	0.0586	0.0812	0.0818	0.1200	0.0366	0.0505
DHCF	0.1747	0.2082	0.1549	0.1724	0.0828	0.1261	0.0629	0.0876	0.1080	0.1466	0.0633	0.0780
LGCN+	0.1019	0.2619	0.0804	0.1745	0.0925	0.1574	0.0698	0.1007	0.1668	0.2490	0.0824	0.1065
GAT	0.1249	0.1329	0.1155	0.1204	0.0860	0.1701	0.1040	0.1142	0.0350	0.0716	0.0175	0.0298
HGNN+	0.1752	0.1753	0.1985	0.1986	0.1349	0.2003	0.1064	0.1410	0.1500	0.2337	0.0723	0.0956
GTGS	**0.2347**	**0.2903**	**0.2075**	**0.2335**	**0.2125**	**0.2970**	**0.1630**	**0.1931**	**0.2061**	**0.3090**	**0.0994**	**0.1292**
Improv.	33.96%	10.84%	4.53%	17.57%	57.52%	48.28%	53.20%	36.95%	23.56%	24.10%	20.63%	21.31%

Experiment

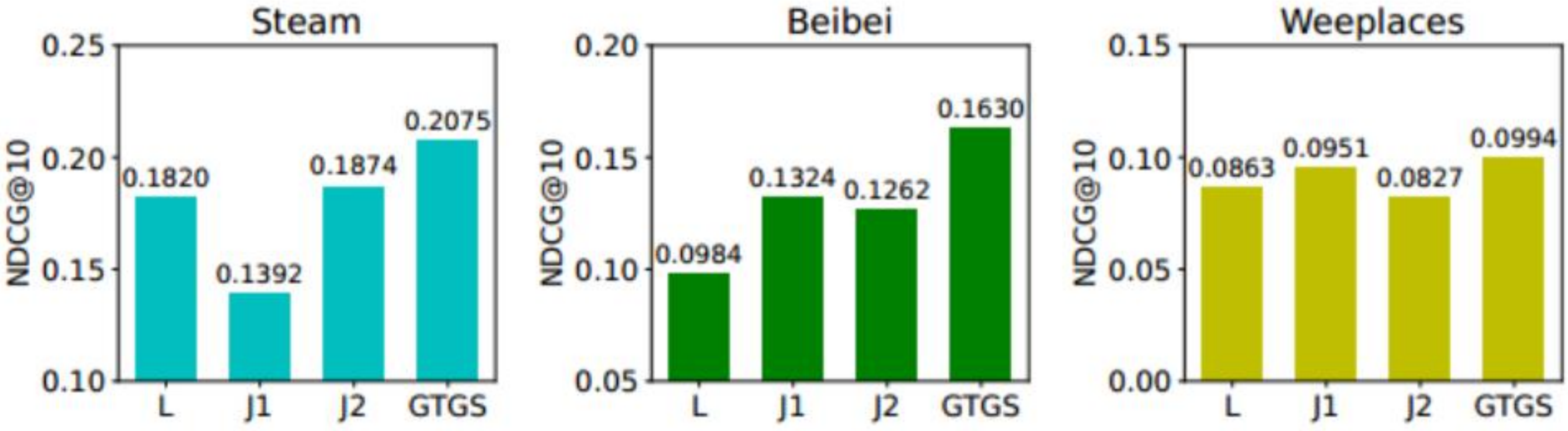


Figure 3: Performance of GTGS w.r.t. different hypergraph constructions.

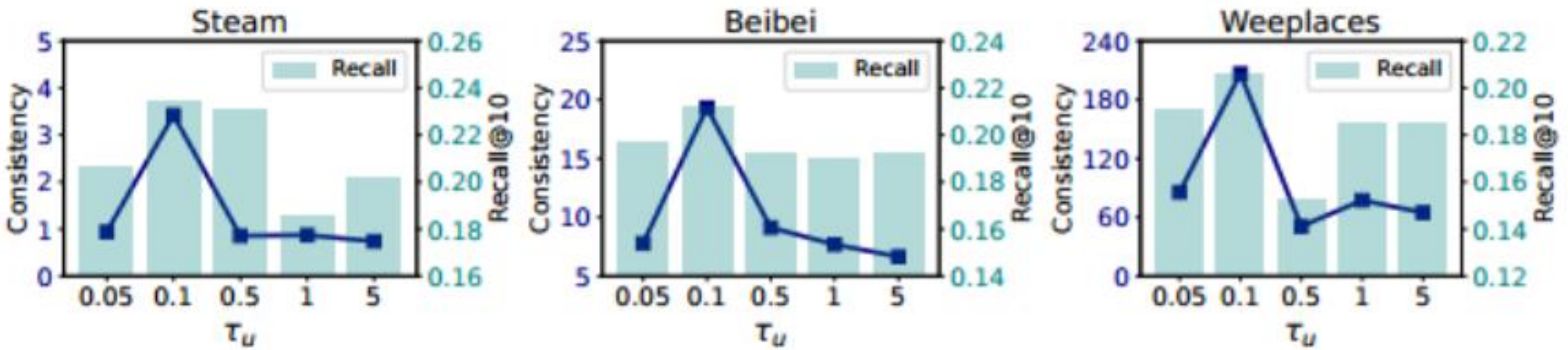


Figure 5: Average consistency of user embeddings w.r.t CSSL temperature τ_u. Histograms denote performances in Recall.

Experiment

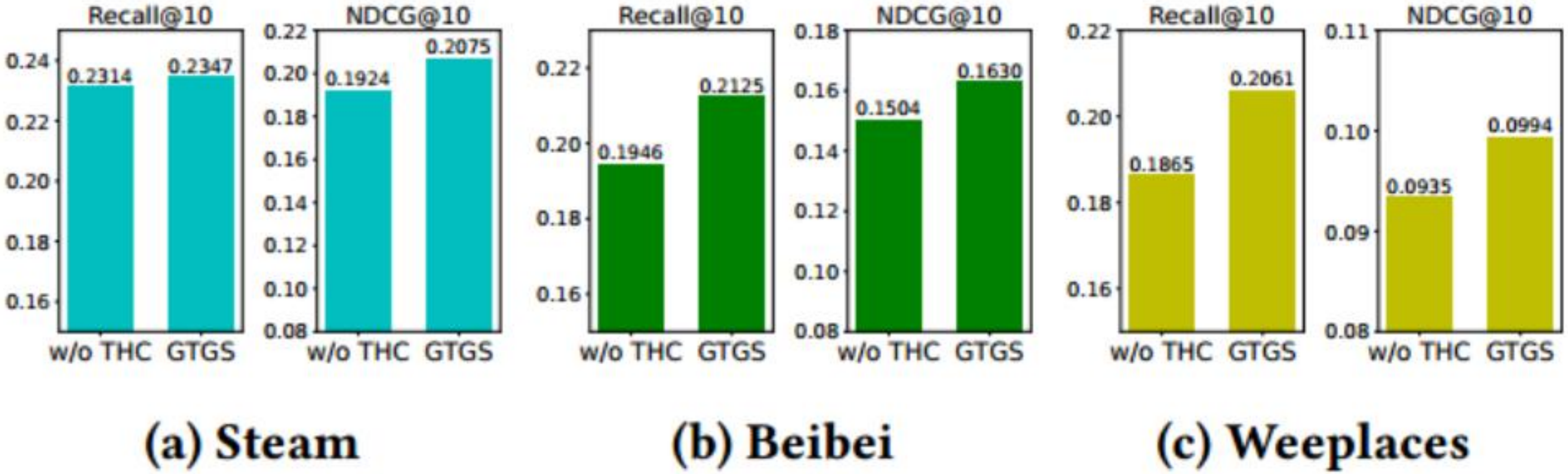


Figure 4: Performance of GTGS with and without THC.

Table 3: Ablation study on self-supervised learning settings.

Dataset	Steam		Beibei		Weeplaces	
Metric	R@10	N@10	R@10	N@10	R@10	N@10
w/o SSL	0.2214	0.1829	0.1841	0.1470	0.1821	0.0931
w/o user	0.2242	0.2037	0.1865	0.1367	0.1831	0.0939
w/o group	0.2325	0.1888	0.1900	0.1482	0.1964	0.0989
GTGS	**0.2347**	**0.2075**	**0.2028**	**0.1526**	**0.2061**	**0.0994**

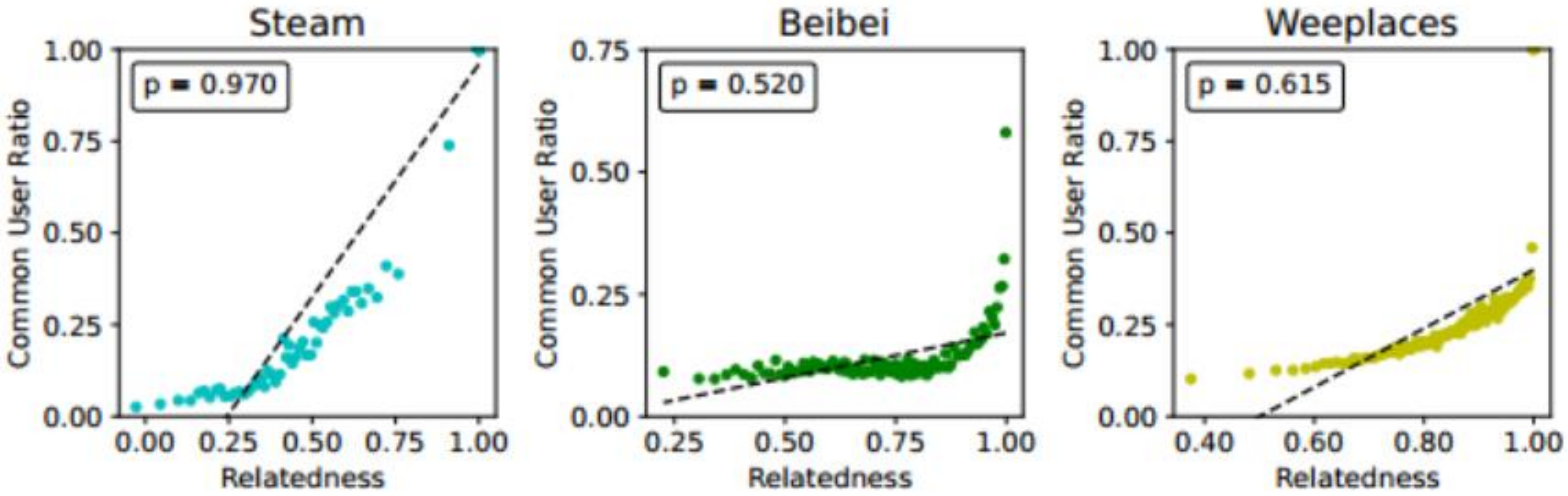


Figure 6: Common user ratio w.r.t relatedness of group pairs based on group embeddings.

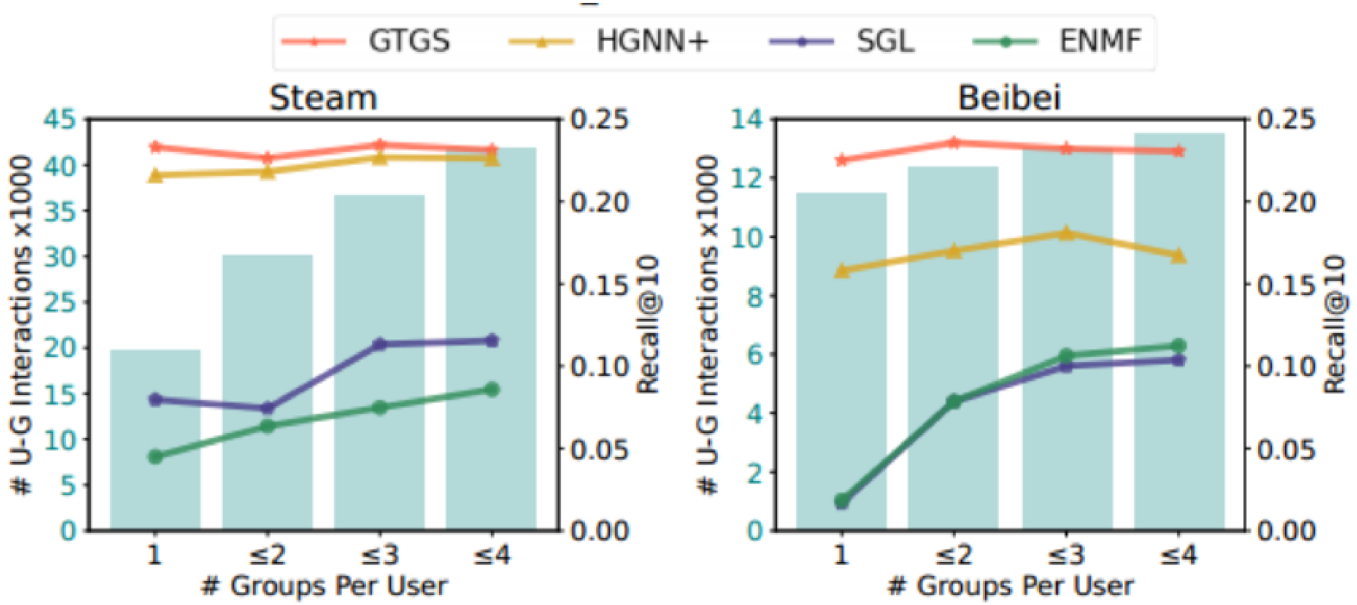


Figure 7: Cold-start performance of different methods. The background histograms denote the number of user-group interactions left in the training set with different thresholds. The solid lines indicate the performances of different methods with respect to different thresholds.

Thank you!