Artificial Intelligence

Self-Supervised Hypergraph Transformer for Recommender Systems

Lianghao Xia
University of Hong Kong
Hong Kong, China
aka_xia@foxmail.com

Chao Huang*
University of Hong Kong
Hong Kong, China
chaohuang75@gmail.com

Chuxu Zhang
Brandeis University
Waltham, USA
chuxuzhang@brandeis.edu

—KDD 2022

Code: https://github.com/akaxlh/SHT.

gesis Leibniz-Institut für Sozialwissenschaften

Forschungszentrum · Research Center L3S

Reported by Yabo Yin

Artificial

1.Introduction

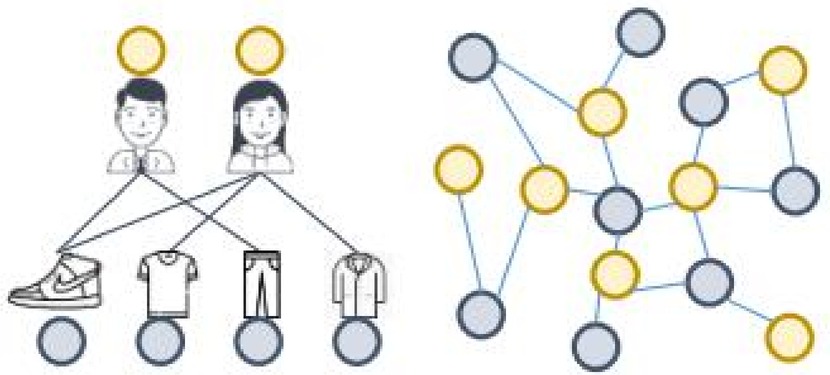
2.Method

3.Experiments

Introduction

For example, users may click their uninterested products due to the over-recommendation of popular items(noise issue).

Data sparsity and skewed distaribution issue still stand in the way of effective user- item interaction modeling.



User-Item Interaction Graph

Method

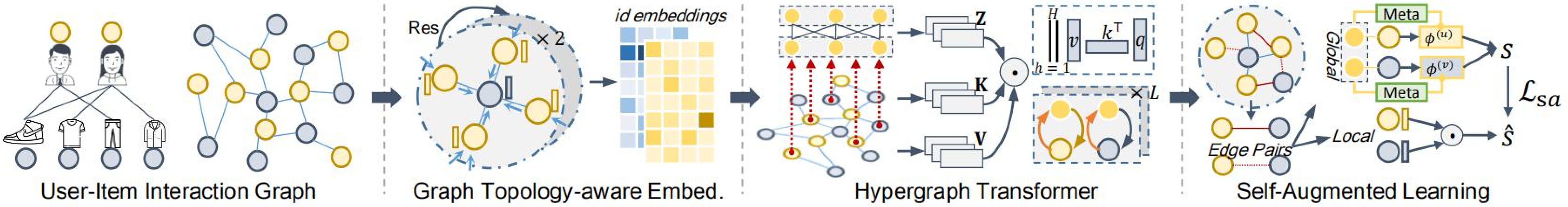
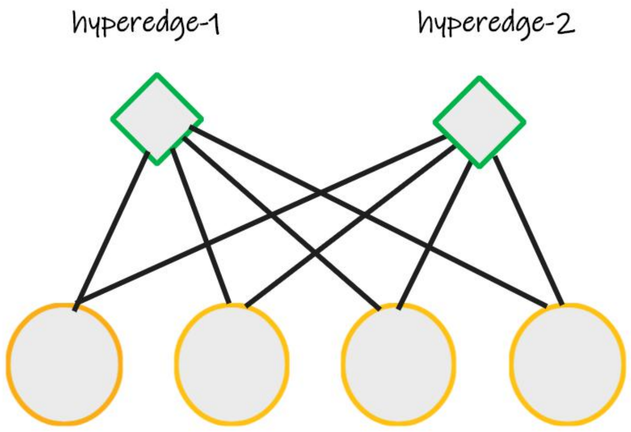


Figure 1: Overall framework of the proposed SHT model.



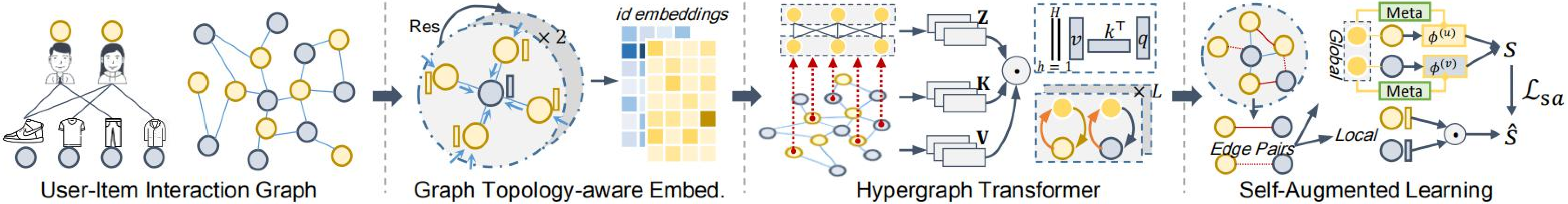


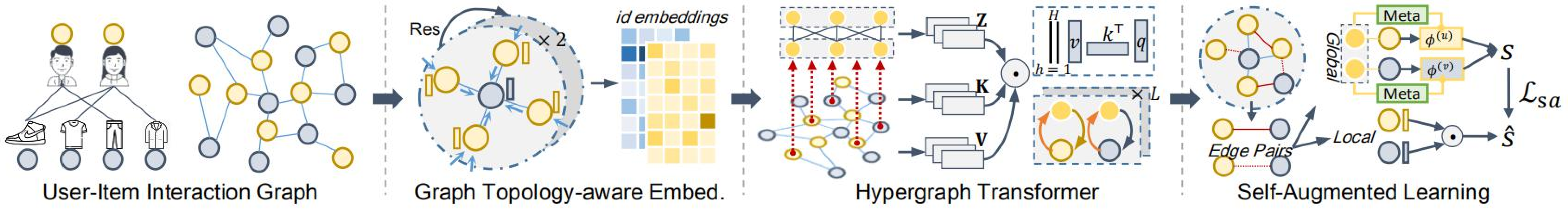
Figure 1: Overall framework of the proposed SHT model.

For user u_i and item v_j, embedding vectors $\mathbf{e}_i, \mathbf{e}_j \in \mathbb{R}^d$ $\quad \mathbf{E}^{(u)} \in \mathbb{R}^{I\times d}, \mathbf{E}^{(v)} \in \mathbb{R}^{J\times d}$

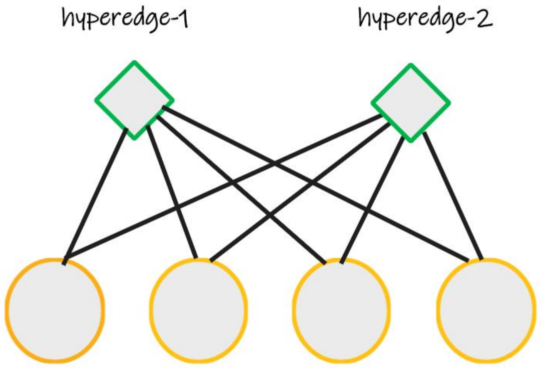
$$\bar{\mathbf{E}}^{(u)} = \mathrm{GCN}^2(\mathbf{E}^{(v)}, \mathcal{G}) = \bar{\mathcal{A}} \cdot \bar{\mathcal{A}}^\top \mathbf{E}^{(u)} + \bar{\mathcal{A}} \cdot \mathbf{E}^{(v)} \qquad (1) \qquad \bar{\mathbf{E}}^{(u)} \in \mathbb{R}^{I\times d}$$

$$\bar{\mathcal{A}}_{i,j} = \mathcal{A}_{i,j} / (\mathbf{D}_i^{(u)1/2} \mathbf{D}_j^{(v)1/2}) \qquad \bar{\mathcal{A}} \in \mathbb{R}^{I\times J}$$

$$\tilde{\mathbf{e}}_i = \mathbf{e}_i + \bar{\mathbf{e}}_i; \qquad \tilde{\mathbf{e}}_j = \mathbf{e}_j + \bar{\mathbf{e}}_j \qquad (2)$$



Node-to-Hyperedge Propagation.



$$\tilde{\mathbf{z}}_k = \Big\Vert_{h=1}^{H} \bar{\mathbf{z}}_{k,h}; \quad \bar{\mathbf{z}}_{k,h} = \sum_{i=1}^{I} \mathbf{v}_{i,h}\mathbf{k}_{i,h}^{\top}\mathbf{q}_{k,h} \qquad (3)$$

$\bar{\mathbf{z}}_{k,h} \in \mathbb{R}^{d/H}$

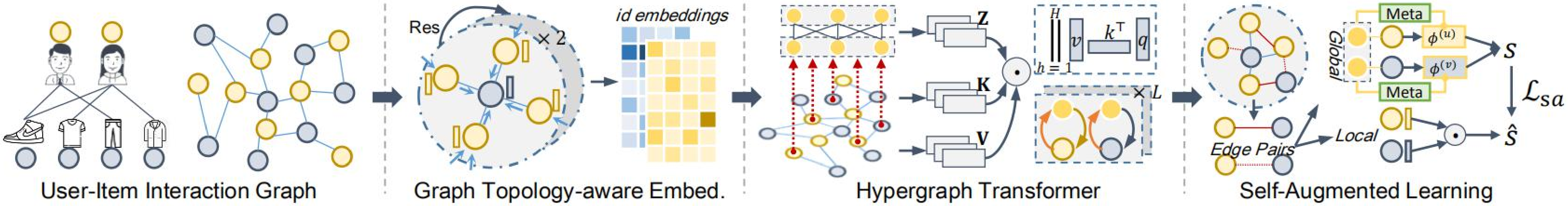
$\tilde{\mathbf{z}}_k \in \mathbf{R}^d$ denotes the embedding for the k-th hyperedge.

$$\mathbf{q}_{k,h} = \mathbf{Z}_{k,p_{h-1}:p_h}; \quad \mathbf{k}_{i,h} = \mathbf{K}_{p_{h-1}:p_h,:}\tilde{\mathbf{e}}_i; \quad \mathbf{v}_{i,h} = \mathbf{V}_{p_{h-1}:p_h,:}\tilde{\mathbf{e}}_i \qquad (4)$$

$\mathbf{Z} \in \mathbb{R}^{K\times d}$ $\quad \mathbf{K},\mathbf{V} \in \mathbb{R}^{d\times d}$ $\quad \mathbf{q}_{k,h}, \mathbf{k}_{i,h}, \mathbf{v}_{i,h} \in \mathbb{R}^{d/H}$

$$\hat{\mathbf{Z}} = \mathrm{HHGN}^2(\tilde{\mathbf{Z}}); \quad \mathrm{HHGN}(\mathbf{X}) = \sigma(\mathcal{H}\cdot\mathbf{X}+\mathbf{X}) \qquad (5)$$

$\hat{\mathbf{Z}}, \tilde{\mathbf{Z}} \in \mathbb{R}^{K\times d}$ $\quad \hat{\mathbf{z}}, \tilde{\mathbf{z}} \in \mathbb{R}^d$,



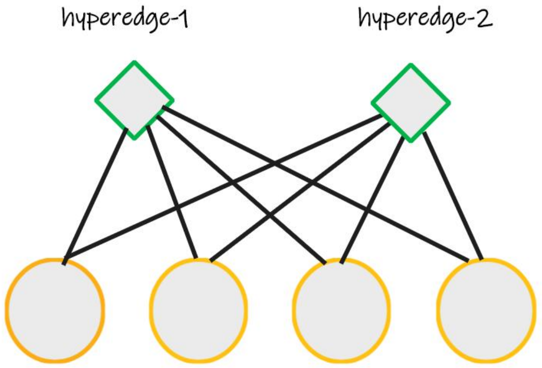
Hyperedge-to-Node Propagation

$$\tilde{\mathbf{e}}'_i = \Big\Vert_{h=1}^{H} \bar{\mathbf{e}}'_{i,h}; \quad \bar{\mathbf{e}}'_{i,h} = \sum_{k=1}^{K} \mathbf{v}'_{k,h}\mathbf{k}'^{\top}_{k,h}\mathbf{q}'_{i,h} \tag{6}$$

$$\tilde{\mathbf{e}}'_i \in \mathbb{R}^d$$

$$\mathbf{q}'_{i,h} = \mathbf{k}_{i,h}; \quad \mathbf{k}'_{k,h} = \mathbf{q}_{k,h}; \quad \mathbf{v}'_{k,h} = \mathbf{V}_{p_{h-1}:p_h,:}\hat{\mathbf{z}}_k \tag{7}$$

$$\mathbf{q}'_{i,h}, \mathbf{k}'_{k,h}, \mathbf{v}'_{k,h} \in \mathbb{R}^{d/H}$$



Iterative Hypergraph Propagation

$$\tilde{\mathbf{E}}_l = \mathrm{HyperTrans}(\tilde{\mathbf{E}}_{l-1}); \quad \hat{\mathbf{E}} = \sum_{l=1}^{L} \tilde{\mathbf{E}}_l \tag{8}$$

$$p_{i,j} = \hat{\mathbf{e}}_i^{(u)\top}\hat{\mathbf{e}}_j^{(v)}$$

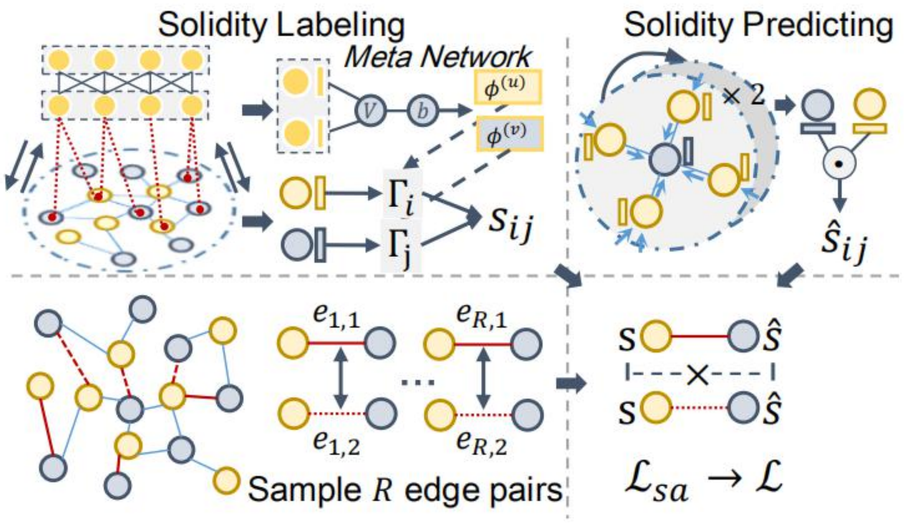
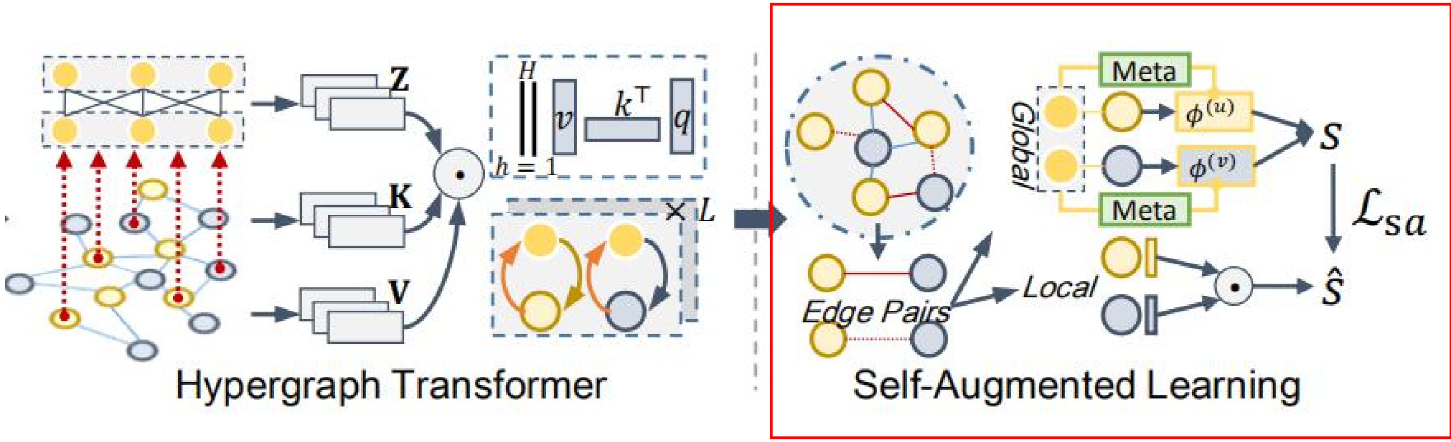


Figure 2: Workflow of the self-augmented learning.



$$\Gamma_i = \phi^{(u)}\left(\Big\Vert_{h=1}^{H} \mathbf{k}_{i,h} \right); \quad \Gamma_j = \phi^{(v)}\left(\Big\Vert_{h=1}^{H} \mathbf{k}_{j,h} \right) \tag{9}$$

$$\phi(\mathbf{x}; \mathbf{Z}) = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b}); \quad \mathbf{W} = \mathbf{V}_1 \bar{\mathbf{z}} + \mathbf{W}_0; \quad \mathbf{b} = \mathbf{V}_2 \bar{\mathbf{z}} + \mathbf{b}_0 \tag{10}$$

$\bar{\mathbf{z}} \in \mathbb{R}^d$

$\bar{\mathbf{z}} = \sum_{k=1}^{K} \mathbf{z}_k / K$

$\mathbf{V}_1 \in \mathbb{R}^{d\times d\times d}, \mathbf{W}_0 \in \mathbb{R}^{d\times d}, \mathbf{V}_2 \in \mathbb{R}^{d\times d}, \mathbf{b}_0 \in \mathbb{R}^d$

$$s_{i,j} = \mathrm{sigm}(\mathbf{d}^\top \cdot \sigma(\mathbf{T} \cdot [\Gamma_i; \Gamma_j] + \Gamma_i + \Gamma_j + \mathbf{c})) \tag{11}$$

$\mathbf{d} \in \mathbb{R}^d, \mathbf{T} \in \mathbb{R}^{d\times 2d}, \mathbf{c} \in \mathbb{R}^d$

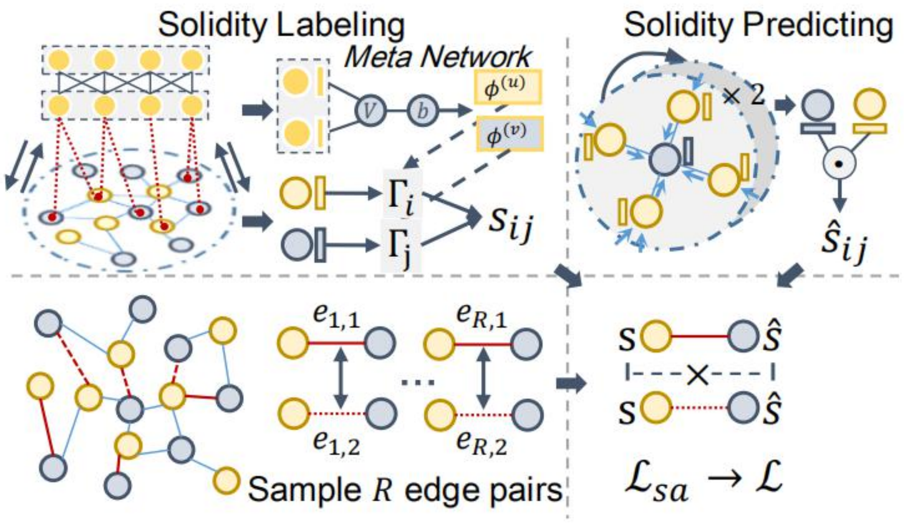
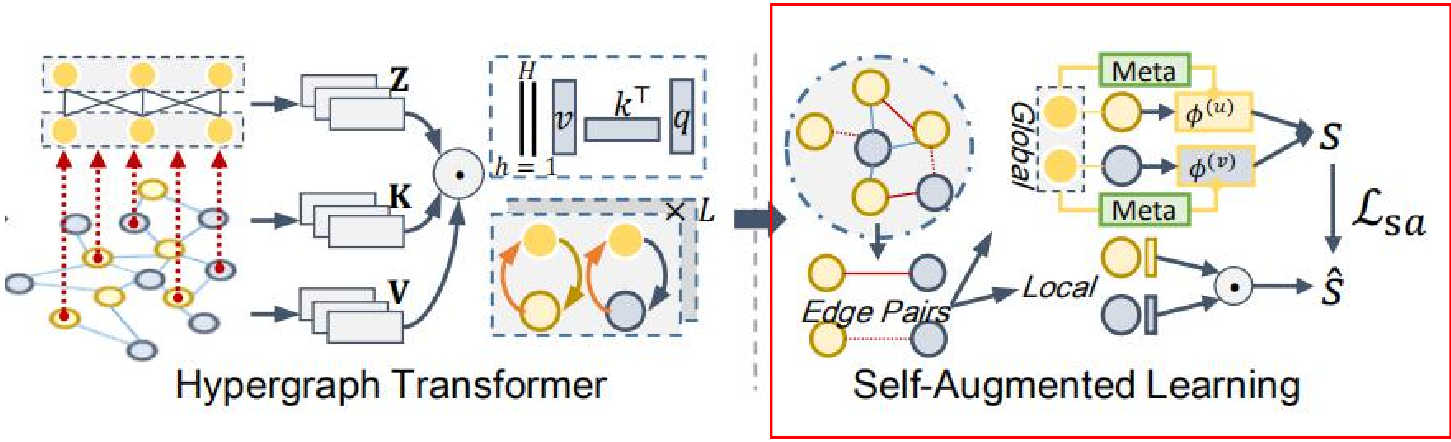


Figure 2: Workflow of the self-augmented learning.



training labels. In particular, R pairs of edges $\{(e_{1,1}, e_{1,2}),...,(e_{R,1}, e_{R,2})\}$ from the observed edges in $\mathcal{G}$ are sampled, and SHT gives predictions on the solidity using the topology-aware embeddings. The

$$\mathcal{L}_{sa} = \sum_{r=1}^{R} \max(0, 1 - (\hat{s}_{u_{r,1},v_{r,1}} - \hat{s}_{u_{r,2},v_{r,2}})(s_{u_{r,1},v_{r,1}} - s_{u_{r,2},v_{r,2}}));$$
$$\hat{s}_{u_{r,1},v_{r,1}} = \mathbf{e}_{u_{r,1}}^{\top} \mathbf{e}_{v_{r,1}}; \qquad \hat{s}_{u_{r,2},v_{r,2}} = \mathbf{e}_{u_{r,2}}^{\top} \mathbf{e}_{v_{r,2}} \tag{12}$$

together with the self-augmented ranking task. Specifically, R' positive edges (observed in $\mathcal{G}$) and R' negative edges (not observed in $\mathcal{G}$) are sampled $\{(e_{1,1}, e_{1,2}), (e_{2,1}, e_{2,2})..., (e_{R',1}, e_{R',2})\}$, where $e_{r,1}$ and $e_{r,2}$ are individual positive and negative sample, respectively.

$$\mathcal{L} = \sum_{r=1}^{R'} \max(0, 1 - (p_{u_{r,1},v_{r,1}} - p_{u_{r,2},v_{r,2}})) + \lambda_1 \mathcal{L}_{\text{sa}} + \lambda_2 \|\Theta\|_{\text{F}}^2 \tag{13}$$

Experiments

Table 1: Statistical information of the experimental datasets.

Stat.	Yelp	Gowalla	Tmall
# Users	29601	50821	47939
# Items	24734	24734	41390
# Interactions	1517326	1069128	2357450
Density	2.1×10^{-3}	4.0×10^{-4}	1.2×10^{-3}

Table 2: Performance comparison on Yelp, MovieLens, Amazon datasets in terms of *Recall* and *NDCG*.

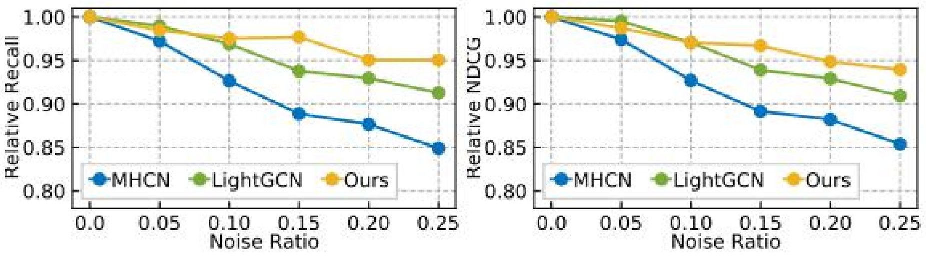
Data	Metric	BiasMF	NCF	AutoR	GCMC	PinSage	NGCF	STGCN	LightGCN	GCCF	DGCF	HyRec	DHCF	MHCN	SLRec	SGL	*SHT*	p-val.
Yelp	Recall@20	0.0190	0.0252	0.0259	0.0266	0.0345	0.0294	0.0309	0.0482	0.0462	0.0466	0.0472	0.0449	0.0503	0.0476	0.0526	**0.0651**	$9.3e^{-7}$
	NDCG@20	0.0161	0.0202	0.0210	0.0251	0.0288	0.0243	0.0262	0.0409	0.0398	0.0395	0.0395	0.0381	0.0424	0.0398	0.0444	**0.0546**	$9.1e^{-8}$
	Recall@40	0.0371	0.0487	0.0504	0.0585	0.0599	0.0522	0.0504	0.0803	0.0760	0.0774	0.0791	0.0751	0.0826	0.0821	0.0869	**0.1091**	$4.1e^{-7}$
	NDCG@40	0.0227	0.0289	0.0301	0.0373	0.0385	0.0330	0.0332	0.0527	0.0508	0.0511	0.0522	0.0493	0.0544	0.0541	0.0571	**0.0709**	$2.2e^{-7}$
Gowalla	Recall@20	0.0196	0.0171	0.0239	0.0301	0.0576	0.0552	0.0369	0.0985	0.0951	0.0944	0.0901	0.0931	0.0955	0.0925	0.1030	**0.1232**	$5.3e^{-7}$
	NDCG@20	0.0105	0.0106	0.0132	0.0181	0.0373	0.0298	0.0217	0.0593	0.0535	0.0522	0.0498	0.0505	0.0574	0.0581	0.0623	**0.0731**	$6.3e^{-7}$
	Recall@40	0.0346	0.0216	0.0343	0.0427	0.0892	0.0810	0.0542	0.1431	0.1392	0.1401	0.1306	0.1356	0.1393	0.1305	0.1500	**0.1804**	$1.5e^{-7}$
	NDCG@40	0.0145	0.0118	0.0160	0.0212	0.0417	0.0367	0.0262	0.0710	0.0684	0.0671	0.0669	0.0660	0.0689	0.0680	0.0746	**0.0881**	$3.2e^{-7}$
Tmall	Recall@20	0.0103	0.0082	0.0103	0.0103	0.0202	0.0180	0.0146	0.0225	0.0209	0.0235	0.0233	0.0156	0.0203	0.0191	0.0268	**0.0387**	$4.3e^{-9}$
	NDCG@20	0.0072	0.0059	0.0072	0.0072	0.0136	0.0123	0.0105	0.0154	0.0141	0.0163	0.0160	0.0108	0.0139	0.0133	0.0183	**0.0262**	$4.9e^{-9}$
	Recall@40	0.0170	0.0140	0.0174	0.0159	0.0345	0.0310	0.0245	0.0378	0.0356	0.0394	0.0350	0.0261	0.0340	0.0301	0.0446	**0.0645**	$4.0e^{-9}$
	NDCG@40	0.0095	0.0079	0.0097	0.0086	0.0186	0.0168	0.0140	0.0208	0.0196	0.0218	0.0199	0.0145	0.0188	0.0171	0.0246	**0.0352**	$3.5e^{-9}$

Experiments

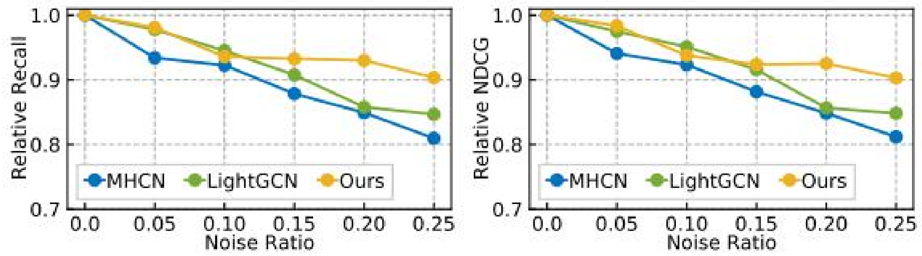
Table 3: Ablation study on key components of SHT.

Category	Data Variants	Yelp Recall	Yelp NDCG	Gowalla Recall	Gowalla NDCG	Tmall Recall	Tmall NDCG
Local	-Pos	0.0423	0.0352	0.0816	0.0487	0.0218	0.0247
Global	-Trans	0.0603	0.0504	0.0999	0.0608	0.0321	0.0206
	-DeepH	0.0645	0.0540	0.1089	0.0634	0.0347	0.0234
	-HighH	0.0598	0.0497	0.1091	0.0646	0.0336	0.0227
	-Hyper	0.0401	0.0346	0.0879	0.0531	0.0209	0.0144
SAL	-Meta	0.0615	0.0526	0.1108	0.0717	0.0375	0.0255
	-SAL	0.0602	0.0519	0.1099	0.0699	0.0363	0.0251
SHT		0.0651	0.0546	0.1232	0.0731	0.0387	0.0262

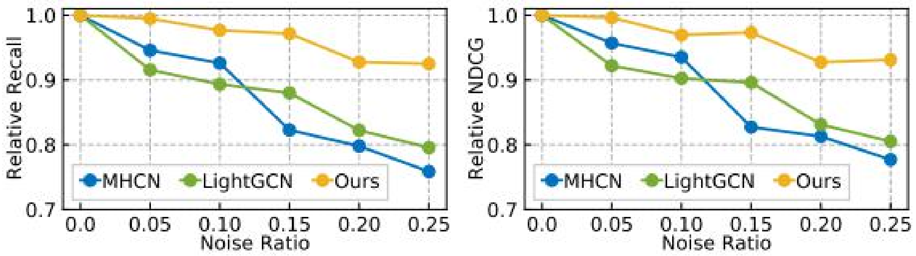
Experiments



(a) Yelp data

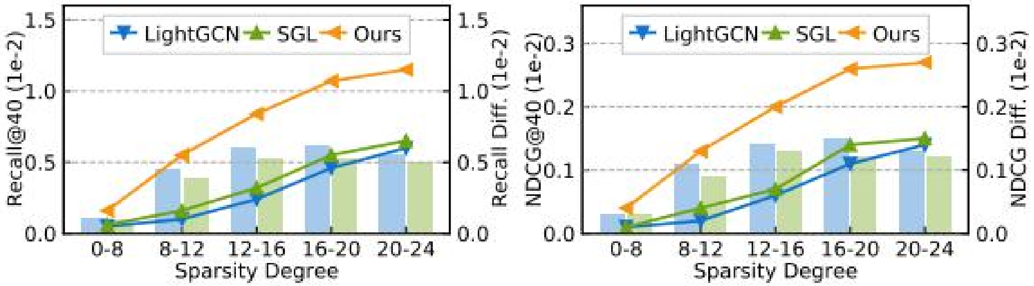


(b) Gowalla data

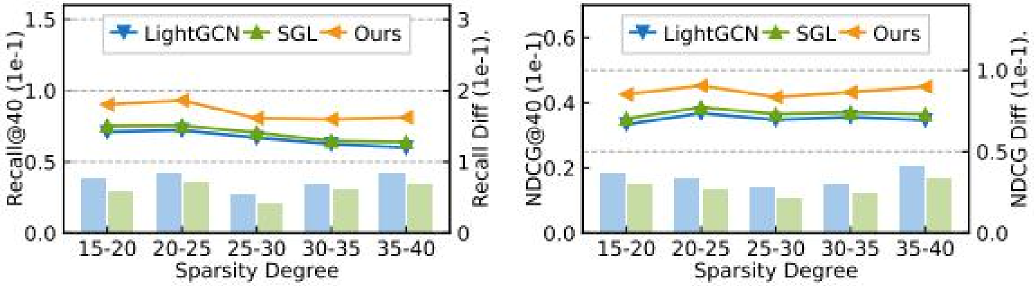


(c) Tmall data

Figure 3: Relative performance degradation *w.r.t* noise ratio.



(a) Performance *w.r.t.* item interaction numbers



(b) Performance *w.r.t.* user interaction numbers

Figure 4: Performance *w.r.t.* different data sparsity degrees on Gowalla data. Lines present Recall@40 and NDCG@40 values, and bars shows performance differences between baselines and our SHT with corresponding colors.

Experiments

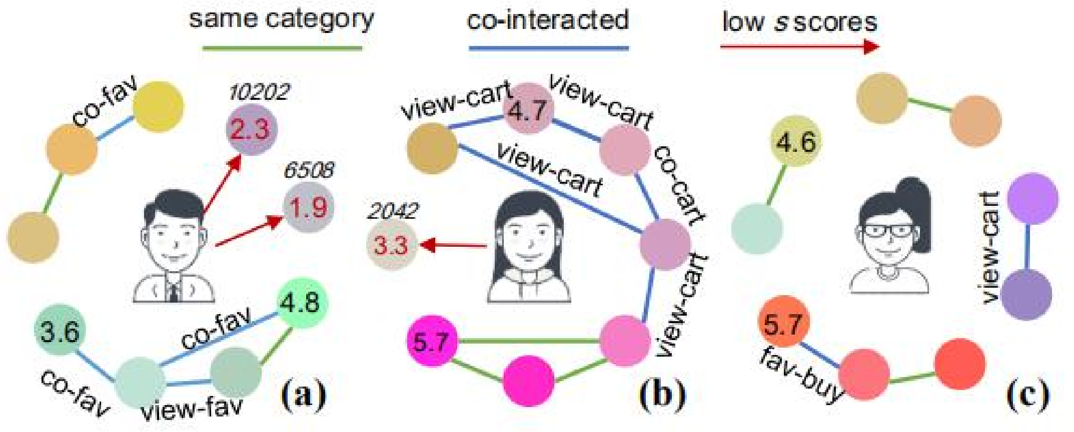


Figure 5: Case study on inferring implicit item-wise relations and discriminating potential noise edges. Circles denote items interacted by the centric users, and their learned embeddings are visualized with colors. Implicit item-wise relations not utilized during model training are presented by green and blue lines. The type of co-interactions are also labeled (*e.g.*, *view-cart* denotes viewed and added-to-cart by same users). Also, the inferred solidity scores s are shown on the circles, where red values are anomalously low scores indicating noisy edges.

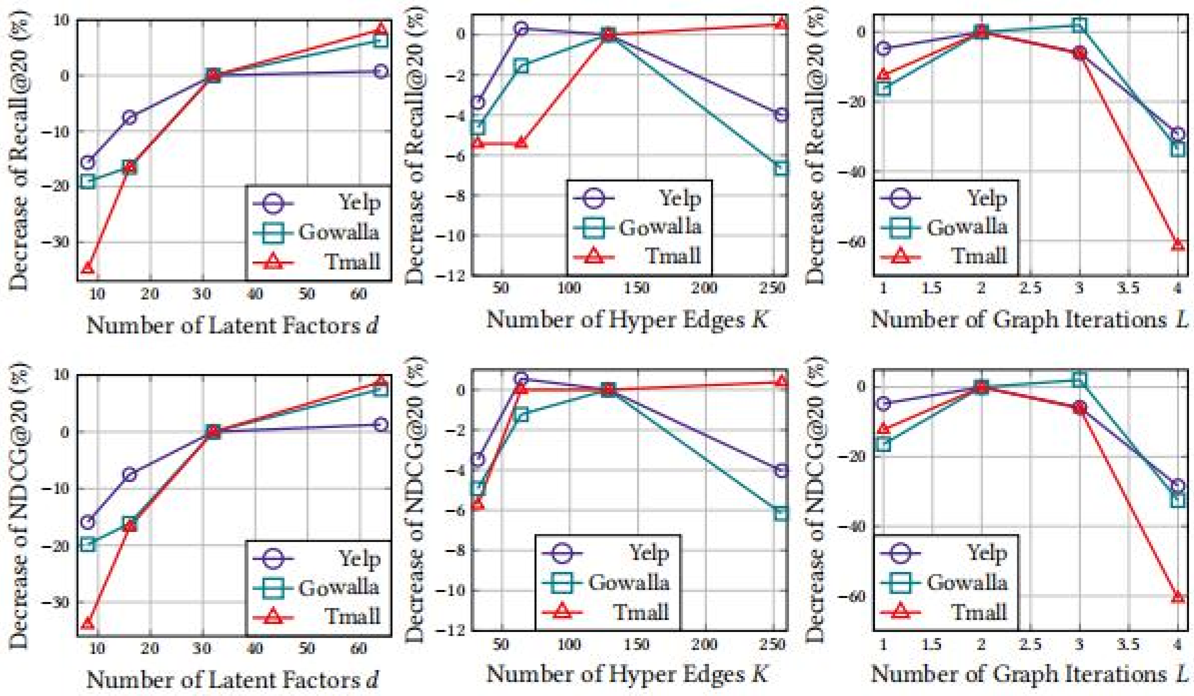


Figure 6: Hyperparameter study of the SHT.

Experiments

Algorithm 1: Learning Process of SHT Framework

Input: user-item interaction graph $\mathcal{G}$, number of graph layers L, number of edges to sample R, R', maximum epoch number E, learning rate η

Output: trained parameters in $\mathbf{\Theta}$

Initialize all parameters in $\mathbf{\Theta}$

for $e = 1$ *to* E **do**

  Draw a mini-batch $\mathbf{U}$ from all users $\{1, 2, ..., I\}$

  Calculate the graph topology-aware embeddings $\bar{\mathbf{E}}$

  Generate input embeddings $\tilde{\mathbf{E}}_0$ for hypergraph transformer

  for $l = 1$ *to* L **do**

    Conduct node-to-hyperedge propagation to obtain $\tilde{\mathbf{Z}}^{(u)}, \tilde{\mathbf{Z}}^{(v)}$ for both users and items

    Conduct hierarchical hyperedge feature transformation for $\hat{\mathbf{Z}}^{(u)}, \hat{\mathbf{Z}}^{(v)}$

    Propagate information from hyperedges back to user/item nodes to obtain $\tilde{\mathbf{E}}_l^{(u)}, \tilde{\mathbf{E}}_l^{(v)}$

  end

  Aggregate the iteratively propagated embeddings to get $\hat{\mathbf{E}}$

  Sample R edge pairs for self-augmented learning

  Acquire the user/item transformation function $\phi^{(u)}$ and $\phi^{(v)}$ with the meta network

  Conduct user/item embedding transformations using $\phi(\cdot)$ to get $\mathbf{\Gamma}^{(u)}, \mathbf{\Gamma}^{(v)}$

  Calculate the solidity score s for the R edge pairs

  Calculate the solidity predictions $\hat{s}$ for the R edge pairs

  Compute loss $\mathcal{L}_{sa}$ for self-augmented learning according to Eq 12

  Sample R' edge pairs for the main task

  Calculate the pair-wise marginal loss $\mathcal{L}$ according to Eq 13

  for *each parameter* $\theta \in \mathbf{\Theta}$ **do**

    $\theta = \theta - \eta \cdot \partial\mathcal{L}/\partial\theta$

  end

end

return all parameters $\mathbf{\Theta}$