



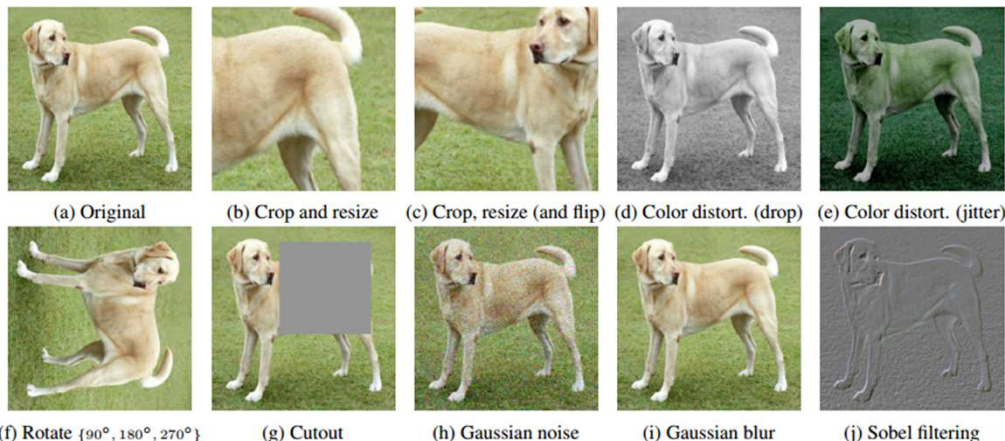
Graph Contrastive Learning with Generative Adversarial Network

Cheng Wu wuc22@mails.tsinghua.edu.cn Tsinghua University Beijing, China	Chaokun Wang* chaokun@tsinghua.edu.cn Tsinghua University Beijing, China	Jingcao Xu xjc20@mails.tsinghua.edu.cn Tsinghua University Beijing, China	Ziyang Liu liuzy21@mails.tsinghua.edu.cn Tsinghua University Beijing, China
Kai Zheng zhengkai@kuaishou.com Kuaishou Inc. Beijing, China	Xiaowei Wang wangxiaowei03@kuaishou.com Kuaishou Inc. Beijing, China	Yang Song yangsong@kuaishou.com Kuaishou Inc. Beijing, China	Kun Gai gai.kun@qq.com Unaffiliated Beijing, China

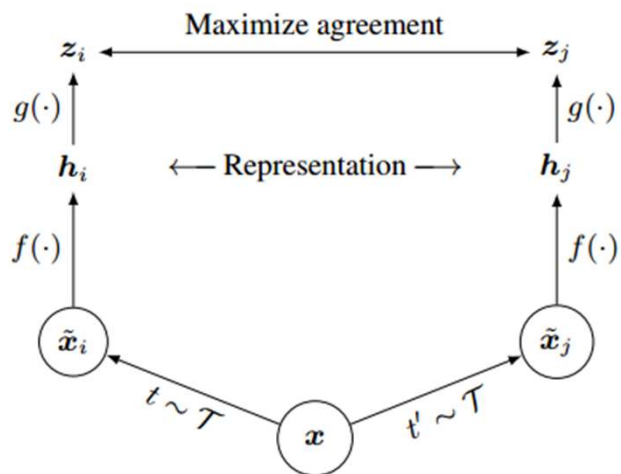
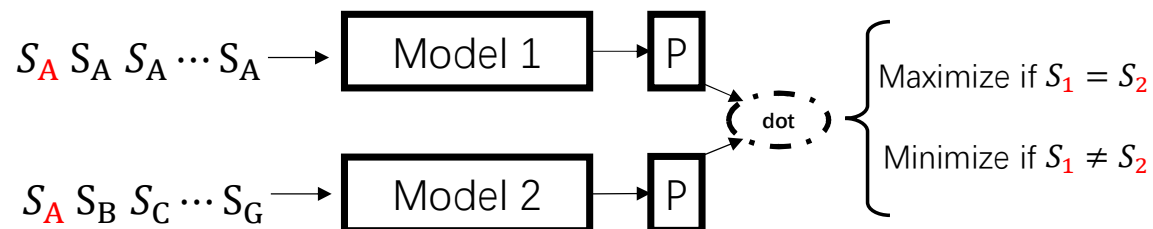
Introduction



SimCLR (ICML 2020)



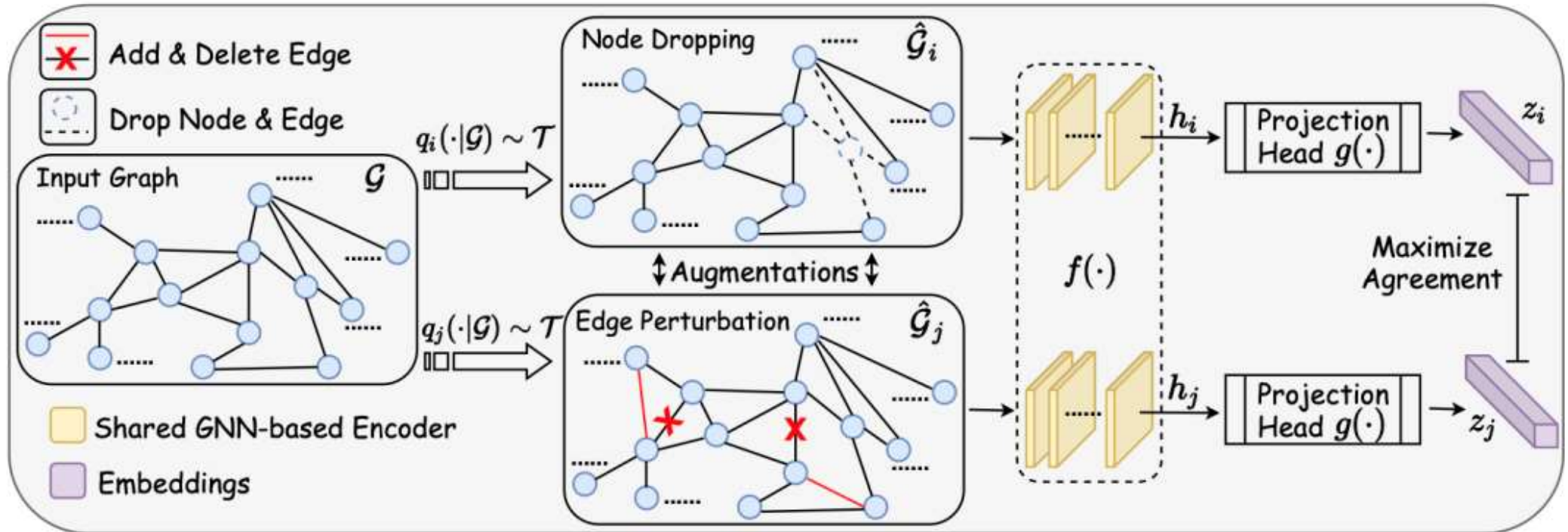
BERT-CT (ICLR 2021)



$$z = f_1(s_1)^T \cdot f_2(s_2)$$

$$\mathcal{L}(z, s_1, s_2) = \begin{cases} -\log \sigma(z) & \text{if } s_1 = s_2 \\ -\log \sigma(1 - z) & \text{if } s_1 \neq s_2 \end{cases}$$

Introduction



$$L_{gcl} = \sum_{i=0}^M -\log \frac{\exp(\sin(z'_i, z''_i)/\tau)}{\sum_{j=0}^M \exp(\sin(z'_i, z''_j)/\tau)}$$

Preliminaries

$$G = (\mathcal{V}, \xi)$$

$$x_i \in R^d \mid i \in \mathcal{V}$$

$$A \in R^{|\mathcal{V}| \times |\mathcal{V}|}$$

$$A_{i,j} = \begin{cases} 1, & \text{if } (v_i, v_j) \in \xi \\ 0, & \text{otherwise} \end{cases}$$

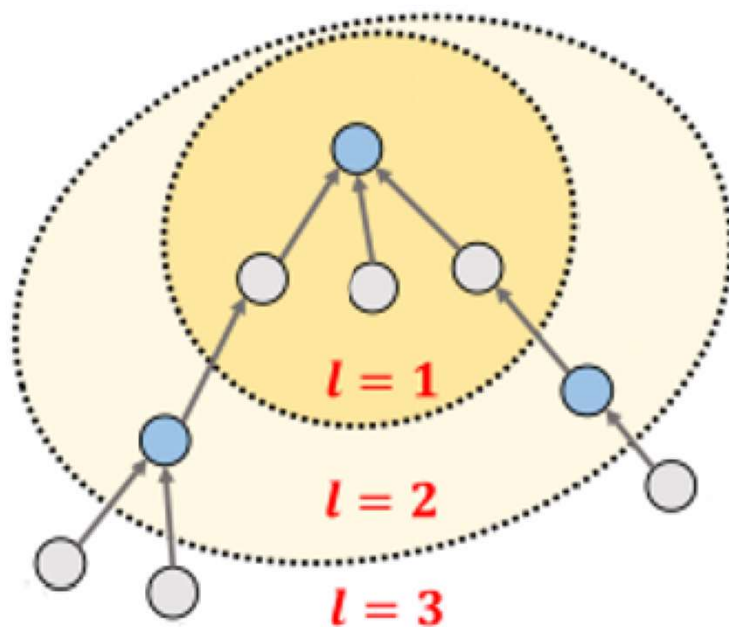


南京航空航天大学

Nanjing University of Aeronautics and Astronautics

Graph Representation Learning

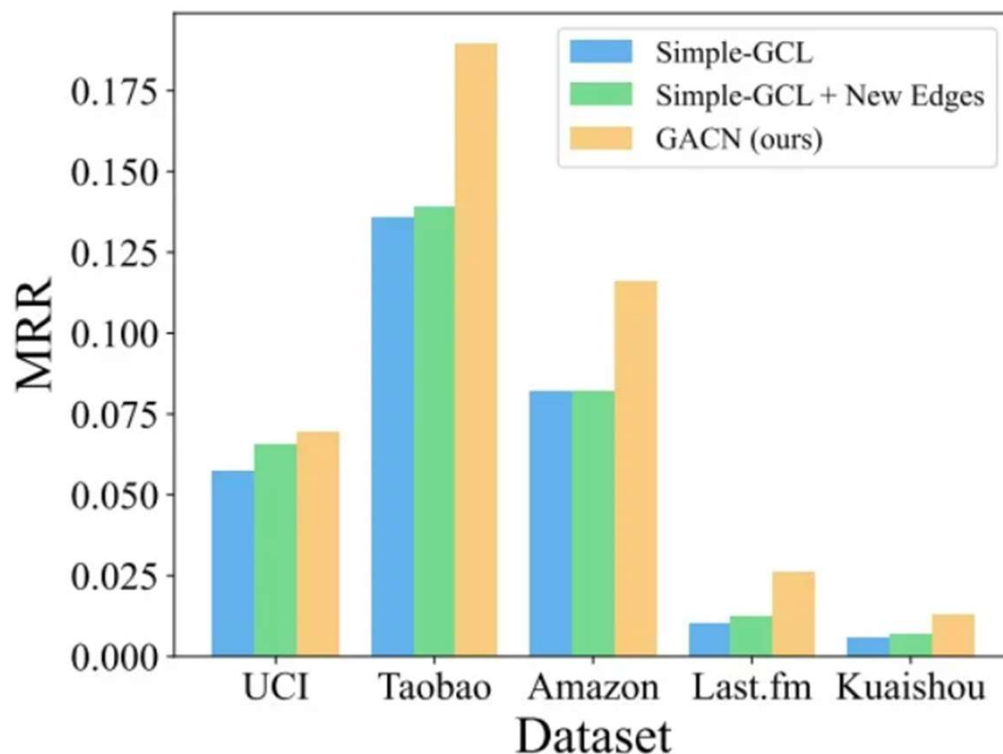
Target: $x_i \in R^d \mid i \in \mathcal{V}$



$$x_i^{k+1} = \sum_{j \in N_i} \frac{1}{\sqrt{|N_i||N_j|}} x_j^k$$

$$x_i = \sum_{k=0}^K x_i^k$$

Motivation

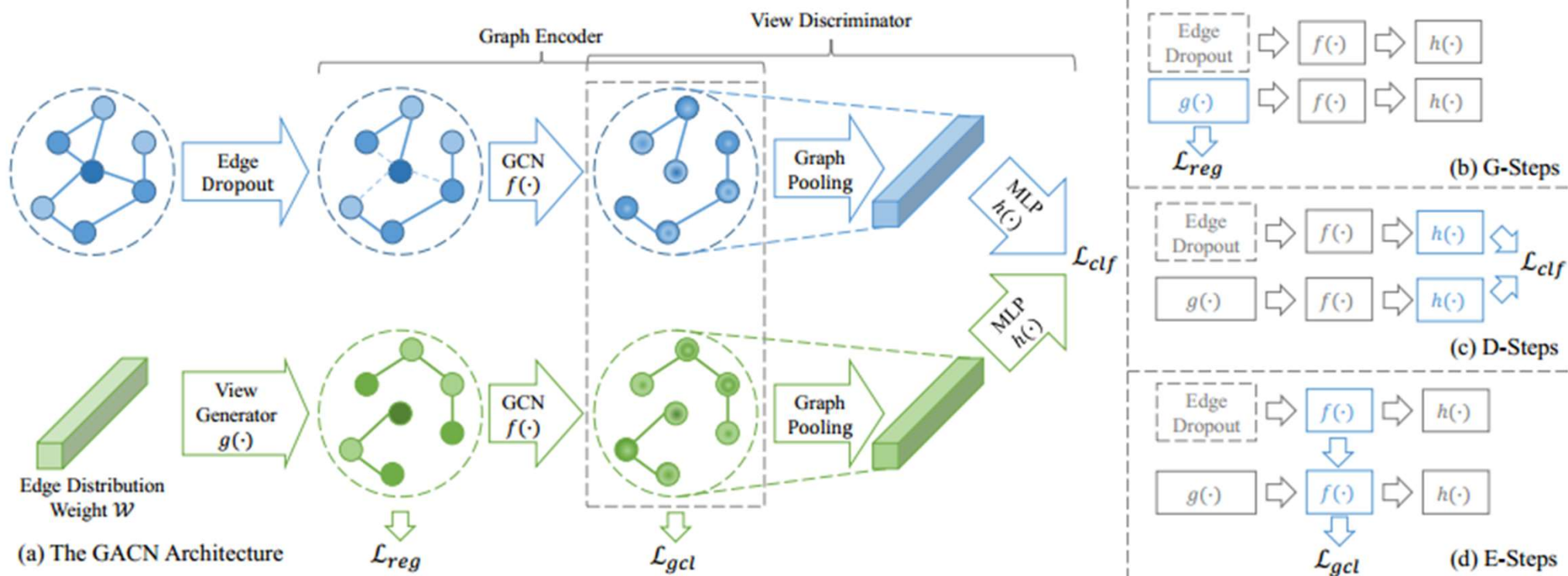


平均结果中的排序倒数，表示待推荐的项目是否放在了用户更显眼的位置，强调“顺序性”

$$MRR = \frac{1}{N} \sum_{i=1}^N \frac{1}{p_i}$$

现有的视图生成策略忽视了图的演变和分布。直观地说，图是由节点和边依次创建而成，观察到的图是这个过程的快照。因此，未连接的节点对有可能在未来形成新的边。

Solution



Solution



View Generator

$$G = (v, \xi)$$

$$\xi' = \begin{bmatrix} P_{(1,1)} & \cdots & P_{(v,1)} \\ \vdots & \ddots & \vdots \\ P_{(1,v)} & \cdots & P_{(v,v)} \end{bmatrix}, P_{(i,j)} \sim \text{Bernoulli}(W_{(i,j)}),$$

$$P = \text{sigmod}\left(\frac{W - X_g}{\tau_g}\right), X_g \sim U(0,1), \tau_g \sim \text{parameter}(0,1)$$

$$G' = (v, \xi') = G(W)$$

Edge Count Loss:

$$L_{\text{cnt}} = \left| \lambda_g \cdot |\xi| - \sum_{i,j} P_{(i,j)} \right|$$

New Edge Loss:

$$L_{\text{new}} = \sum_{i,j} (1 - A_{(i,j)}) \cdot P_{(i,j)}$$

Solution

View Discriminator

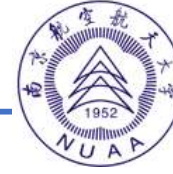
$$d_v = f(G) \quad G_g \text{ 和 } G_p$$

$$p_G = m\left(\frac{1}{|V|} \sum_{v \in V} d_v, \text{MaxPool}(\{d_v | v \in V\})\right)$$

$$L_{clf} = -y_G \cdot \log(p_G) - (1 - y_G) \cdot \log(1 - p_G) = -\log(p_G)$$

$$\mathcal{L}_{gcl} = - \sum_{v \in \mathcal{V}} \log \frac{\exp\left(\frac{d_v^p \top d_v^g}{\tau_f}\right)}{\sum_{u \in \mathcal{V}} \exp\left(\frac{d_u^p \top d_u^g}{\tau_f}\right)}$$

$$\mathcal{L}_{bpr} = - \frac{1}{|O|} \sum_{(i,j,k) \in O} \log \sigma(d_i^\top d_j - d_i^\top d_k)$$



Graph Encoder

Algorithm 1: GACN Framework.

Input: graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, dimension of embedding s , hyper-parameters $\tau_g, \tau_f, \lambda_g, \lambda_{cnt}, \lambda_{new}, \lambda_{gcl}, \lambda_{bpr}$
Output: node representations $\{d_v | v \in \mathcal{V}\}$

- 1 Initialize $\Theta_g, \Theta_h, \Theta_f$;
- 2 **while** GACN not converge **do**
- 3 **for** G -Steps **do**
- 4 Sample X_g and calculate P according to Eq. (4);
- 5 Compute $p_{\mathcal{G}_g}$ using P according to Eq. (9), (10) and (11);
- 6 Update Θ_g according to Eq. (7) and (17);
- 7 **end**
- 8 **for** D -Steps **do**
- 9 Generate and label $\vec{\mathcal{G}}_p, \vec{\mathcal{G}}_g$;
- 10 Encode $G \in \vec{\mathcal{G}}_p \cup \vec{\mathcal{G}}_g$ using Eq. (9), (10) and (11);
- 11 Update Θ_h according to Eq. (12);
- 12 **end**
- 13 **for** E -Steps **do**
- 14 Generate $\mathcal{G}_p, \mathcal{G}_g$ for Eq. (14) and sample O for Eq. (15);
- 15 Update Θ_f according to Eq. (16);
- 16 **end**
- 17 **end**
- 18 $\{d_v | v \in \mathcal{V}\} = f(\mathcal{G})$;
- 19 **return** $\{d_v | v \in \mathcal{V}\}$.

Experiment



Table 3: The experimental results of node classification. The best results are illustrated in bold and the number underlined is the runner-up.

Dataset Metric	Cora			Citeseer		
	P	R	F1	P	R	F1
DeepWalk	0.7753	0.7012	0.7292	0.5579	0.4962	0.4869
LINE	0.7873	0.6970	0.7281	0.5992	0.4437	0.4413
node2vec	0.7744	0.7352	0.7516	0.4717	0.4581	0.4552
LightGCN	0.7615	0.7342	0.7453	0.4434	0.4513	0.4361
Simple-GCL	0.8491	0.8154	0.8287	0.7019	0.6930	0.6946
DGI	0.8320	0.8129	0.8212	0.6427	0.6357	0.6278
GraphCL	0.7993	0.7500	0.7689	0.6547	0.6156	0.6112
GRACE	0.8546	<u>0.8377</u>	0.8445	0.7232	0.6963	0.6948
SGL	0.8029	0.7769	0.7887	0.7177	0.7045	0.7063
GraphGAN	0.4195	0.2177	0.1745	0.3148	0.2994	0.2696
AD-GCL	0.4176	0.3672	0.3800	0.2842	0.2809	0.2763
GraphMAE	<u>0.8667</u>	0.8287	<u>0.8447</u>	<u>0.7278</u>	<u>0.7048</u>	<u>0.7064</u>
GACN	0.8705	0.8545	0.8614	0.7311	0.7187	0.7212

Think



What is a good graph representation?

The Mutual Information Maximization Principle.

The target of GCL

$$\text{InfoMax: } \max_f I(G, f(G))$$

Graph Information Bottleneck

Supervised learning

$$\text{GIB: } \max_f I(f(G), Y) - \beta I(G, f(G))$$

Self-supervised learning

$$\min_T \max_f I(f(G), f(G')) = I(f(G), f(t(G))), t \sim T$$

Think



Dataset	Metric	NGCF	LightGCN	SGL	NCL	SimGCL	XsimGCL	LightGCL	ours	Improv.
Yelp	recall@10	0.0584	0.0627	0.0845	0.0847	0.0851	<u>0.0853</u>	0.0596	0.0892	4.57%
	recall@20	0.096	0.0996	0.1279	0.1308	<u>0.1313</u>	0.1302	0.0992	0.1337	1.83%
	ndcg@10	0.0444	0.0492	<u>0.0681</u>	0.068	0.0671	0.0678	0.0596	0.0728	7.06%
	ndcg@20	0.0568	0.0612	0.0822	<u>0.0831</u>	0.082	0.0823	0.0728	0.0871	4.81%
Gowalla	recall@10	0.0974	0.1124	0.1348	0.1323	0.133	<u>0.1354</u>	0.116	0.1384	2.22%
	recall@20	0.142	0.1616	0.1928	0.1931	0.1879	<u>0.1902</u>	0.1695	0.1941	2.05%
	ndcg@10	0.0776	0.0898	0.1058	0.1064	0.1079	<u>0.1102</u>	0.0827	0.1131	2.63%
	ndcg@20	0.0915	0.105	0.124	0.1256	0.1249	<u>0.127</u>	0.1234	0.1301	2.44%
Amazon	recall@10	0.059	0.0633	0.0869	0.0824	0.0901	<u>0.0923</u>	0.0566	0.0945	2.38%
	recall@20	0.0931	0.0983	0.1316	0.1211	0.1344	<u>0.1357</u>	0.0842	0.1382	1.84%
	ndcg@10	0.045	0.0498	0.0689	0.0643	0.0713	<u>0.0732</u>	0.0524	0.0763	4.23%
	ndcg@20	0.0562	0.0612	0.0834	0.0798	0.0856	<u>0.0872</u>	0.0657	0.0903	3.56%