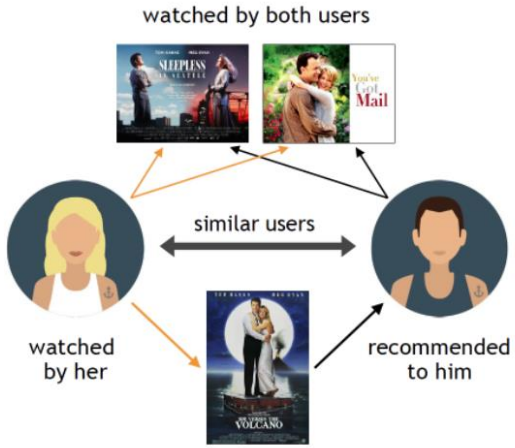
Are Graph Augmentations Necessary? Simple Contrastive Learning for Recommendation

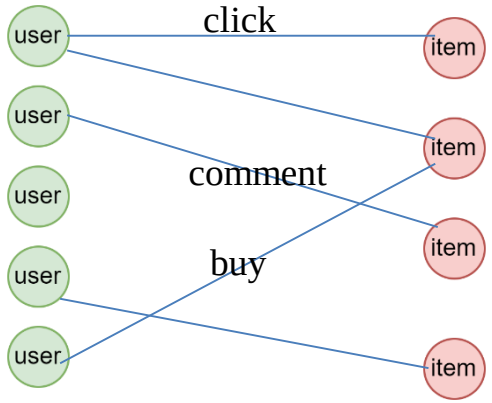
(SIGIR, 2022)

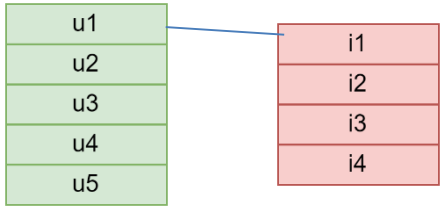
paper, code

Recommendation System (Collaborative Filtering)

Things of a kind come together.





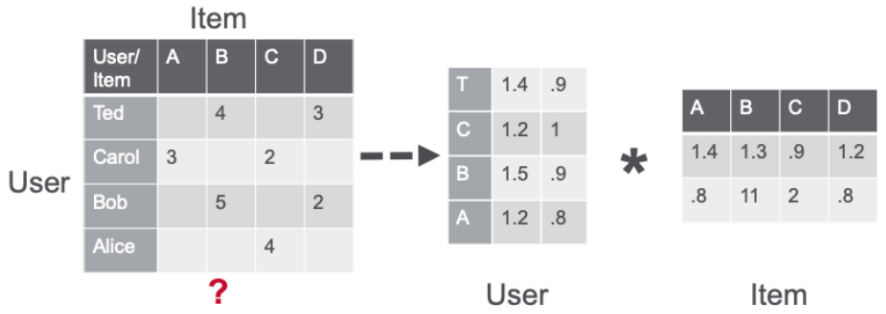


$$y = e_{u1}^{T} . e_{i1} \uparrow$$

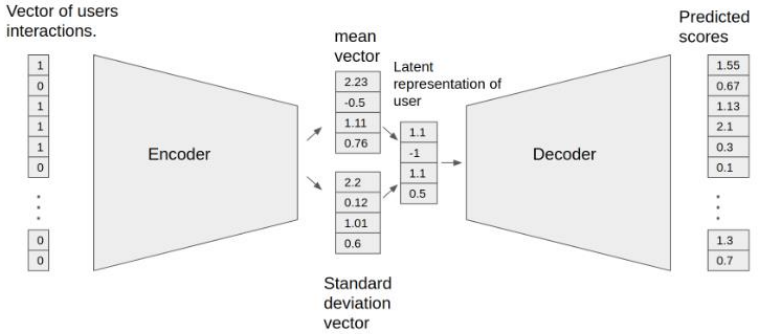
Recommendation System (Collaborative Filtering)

◆ **CF Encoder**

(1) Matrix factorization(MF)-based methods.



(2)Autoencoder-based methods.



(3) GNN-based methods.

◆ **Loss Functions**

(1)Binary cross-entropy (BCE)

(2)Mean square error (MSE)

(3)Bayesian personalized ranking (BPR): $L_{bpr} = -\log(\sigma(e_u^T e_i - e_u^T e_j))$

Graph base CF

Let $\boldsymbol{U}$ and $\boldsymbol{I}$ be the set of users and items respectively.

Let $\boldsymbol{R} = \{\boldsymbol{r_{ui}} = 1 \; or \; 0 \; | u \in \boldsymbol{U}, i \in \boldsymbol{I}\}$ be the interactions matrix.

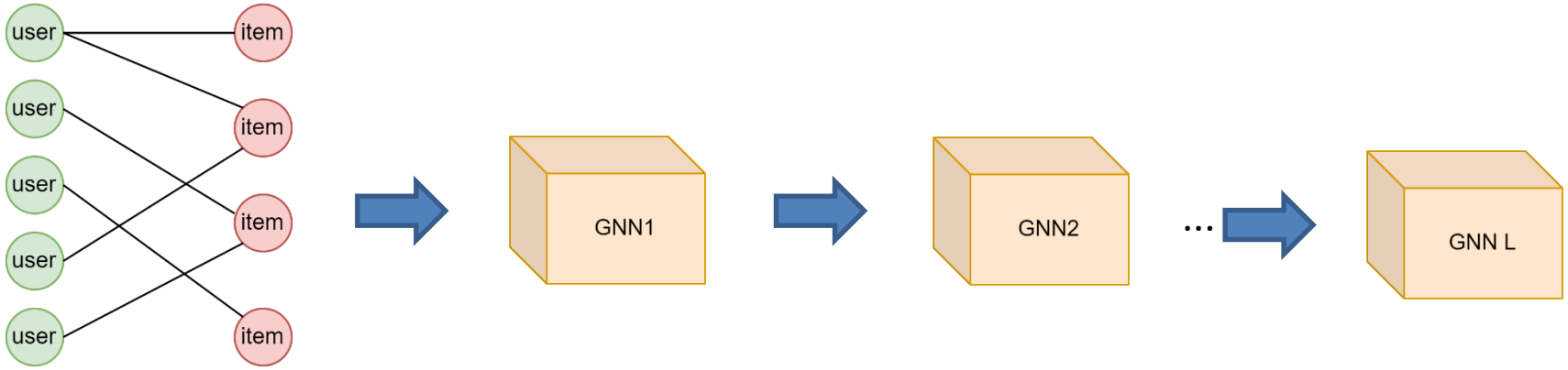
Construct a bipartite graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$, where $\boldsymbol{V} = \boldsymbol{U} \cup \boldsymbol{I}$, $\boldsymbol{E} = \boldsymbol{R}^{\sim}$

$$\boldsymbol{e}^{(l)} = \boldsymbol{H}\left(\boldsymbol{e}^{(l-1)}, G\right)$$

$$\boldsymbol{e}_u = f_{readout}(\{\boldsymbol{e}_u^{(l)} | l = [0, \dots, L]\}), \quad \hat{y}_{ui} = \boldsymbol{e}_u^T \boldsymbol{e}_i$$

$$\mathcal{L}_{main} = \sum_{(u,i,j) \in R} -log\sigma(\hat{y}_{ui} - \hat{y}_{uj})$$

Graph based CF (LightGCN-2020)



(e_u^0, e_i^0) (e_u^1, e_i^1) (e_u^2, e_i^2) (e_u^L, e_i^L)

concatenation (i.e., $\mathbf{e}_u^* = \mathbf{e}_u^{(0)} \| \cdots \| \mathbf{e}_u^{(L)}$) OR sum (i.e., $\mathbf{e}_u^* = \mathbf{e}_u^{(0)} + \cdots + \mathbf{e}_u^{(L)}$) $\hat{y}_{ui} = \mathbf{e}_u^T \mathbf{e}_i$,

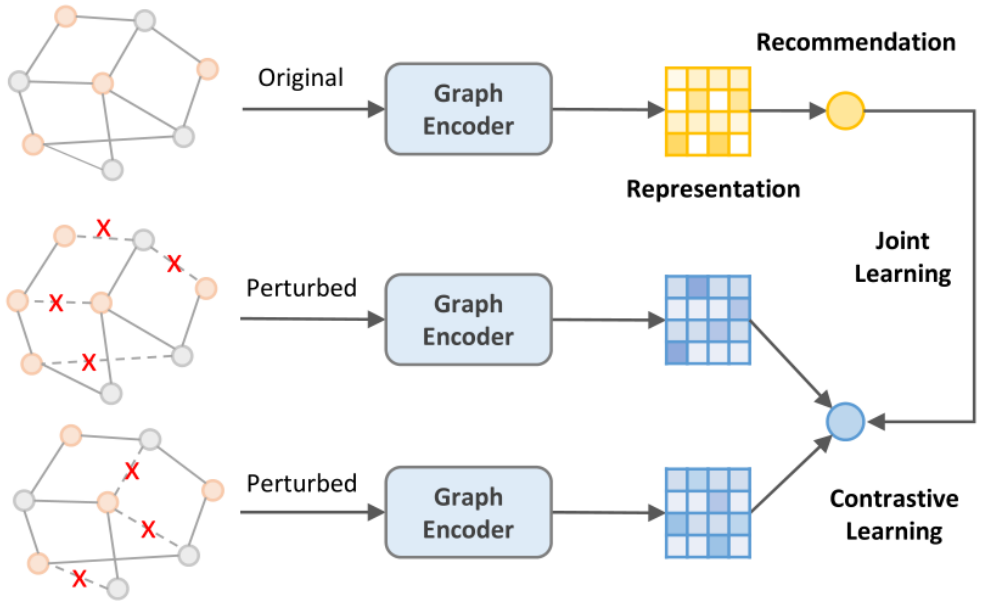
$$\mathbf{e}_u^{(k+1)} = \mathrm{AGG}(\mathbf{e}_u^{(k)}, \{\mathbf{e}_i^{(k)} : i \in \mathcal{N}_u\}).$$

$$\mathbf{e}_u^{(k+1)} = \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}} \mathbf{e}_i^{(k)},$$

$$\mathbf{e}_i^{(k+1)} = \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}} \mathbf{e}_u^{(k)}.$$

$$\mathbf{A} = \begin{pmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{R}^T & \mathbf{0} \end{pmatrix}, \quad \mathbf{E}^{(k+1)} = (\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}})\mathbf{E}^{(k)}$$

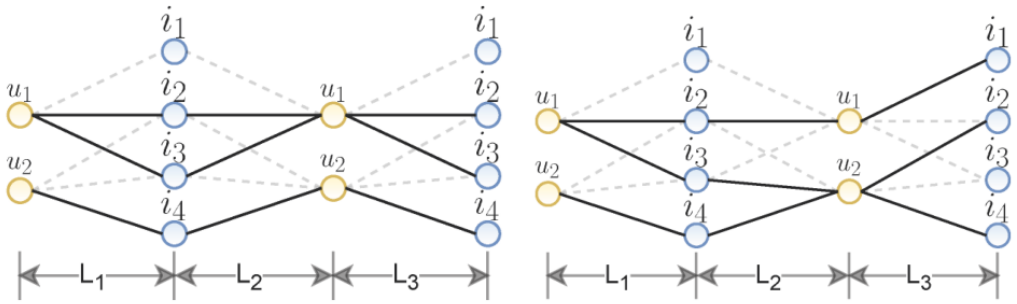
Contrastive Learning for Graph based CF (SGL-2021)



BPR Loss: $\mathcal{L}_{rec} = -\log(\sigma(\mathbf{e}_u^\top \mathbf{e}_i - \mathbf{e}_u^\top \mathbf{e}_j)),$

CL Loss: $\mathcal{L}_{cl} = \sum_{i \in \mathcal{B}} -\log \frac{\exp(\mathbf{z}_i'^\top \mathbf{z}_i'' / \tau)}{\sum_{j \in \mathcal{B}} \exp(\mathbf{z}_i'^\top \mathbf{z}_j'' / \tau)},$

Joint Learning: $\mathcal{L}_{joint} = \mathcal{L}_{rec} + \lambda \mathcal{L}_{cl}$



(a) Edge Dropout (b) Random Walk

Node Dropout (ND). $s_1(\mathcal{G}) = (\mathbf{M}' \odot \mathcal{V}, \mathcal{E}), \quad s_2(\mathcal{G}) = (\mathbf{M}'' \odot \mathcal{V}, \mathcal{E}),$

Edge Dropout (ED). $s_1(\mathcal{G}) = (\mathcal{V}, \mathbf{M}_1 \odot \mathcal{E}), \quad s_2(\mathcal{G}) = (\mathcal{V}, \mathbf{M}_2 \odot \mathcal{E}),$

Random Walk (RW). $s_1(\mathcal{G}) = (\mathcal{V}, \mathbf{M}_1^{(l)} \odot \mathcal{E}), \quad s_2(\mathcal{G}) = (\mathcal{V}, \mathbf{M}_2^{(l)} \odot \mathcal{E}),$

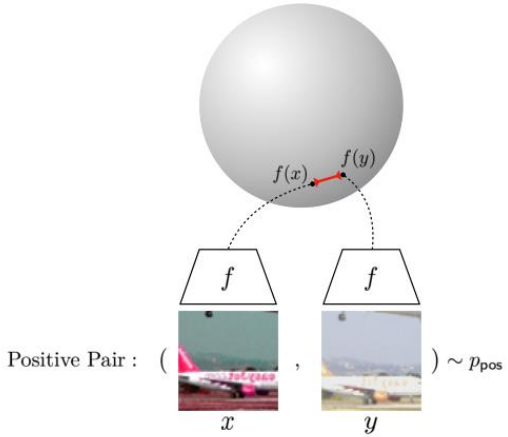
Part1: Is Graph Aug necessary for Recommendation

$$\mathcal{L}_{cl} = \sum_{i \in \mathcal{B}} -\log \frac{\exp(\mathbf{z}_i'^{\top} \mathbf{z}_i'' / \tau)}{\sum_{j \in \mathcal{B}} \exp(\mathbf{z}_i'^{\top} \mathbf{z}_j'' / \tau)}, \quad \Rightarrow \quad \mathcal{L}_{cl} = \sum_{i \in \mathcal{B}} -\log \frac{\exp(1/\tau)}{\sum_{j \in \mathcal{B}} \exp(\mathbf{z}_i^{\top} \mathbf{z}_j / \tau)}. \tag{4}$$

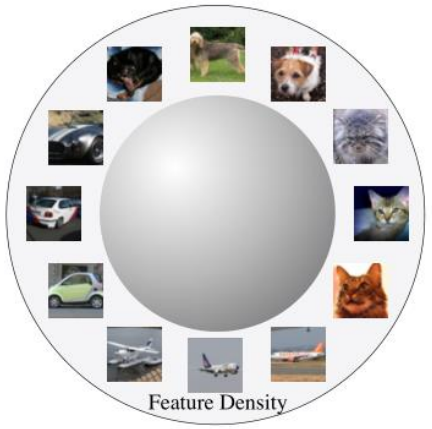
Method	Yelp2018		Amazon-Book	
	Recall@20	NDCG@20	Recall@20	NDCG@20
LightGCN	0.0639	0.0525	0.0410	0.0318
SGL-ND	0.0644	0.0528	0.0440	0.0346
SGL-ED	**0.0675**	**0.0555**	**0.0478**	**0.0379**
SGL-RW	0.0667	0.0547	0.0457	0.0356
SGL-WA	0.0671	0.0550	0.0466	0.0373
CL Only	0.0245	0.0190	0.0314	0.0258

InfoNCE Loss Influences More

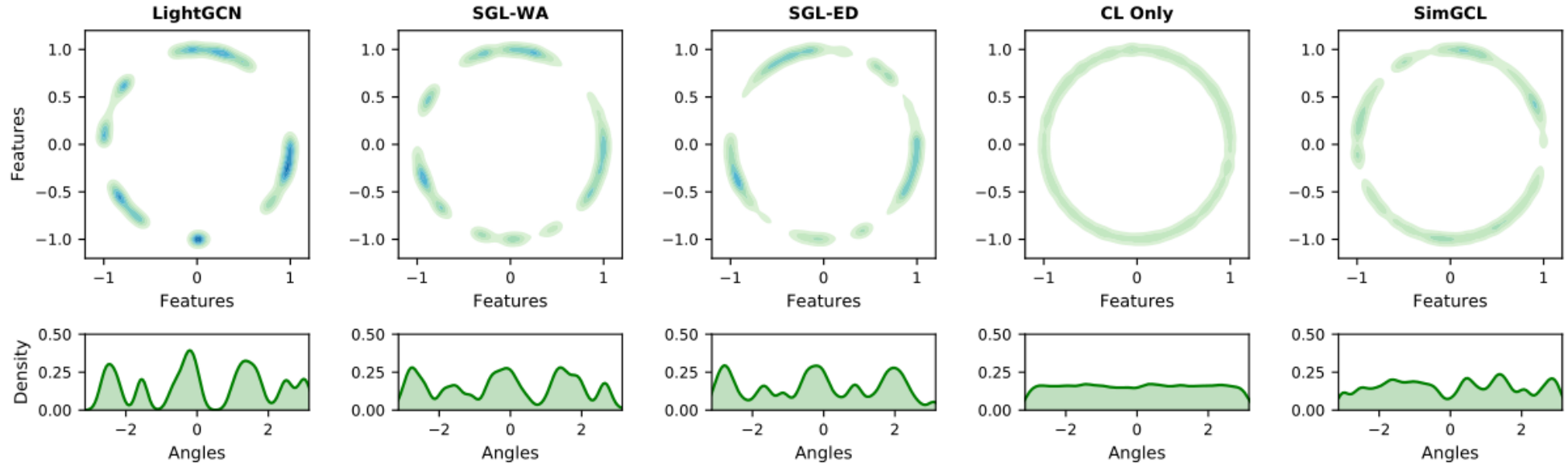
CL Makes Alignment and Uniformity



Alignment: Similar samples have similar features.
(Figure inspired by Tian et al. (2019).)



Uniformity: Preserve maximal information.



(a) Distribution of item representations learned from the dataset of Yelp2018.

Part2: A Simple Graph CL For Recommendation

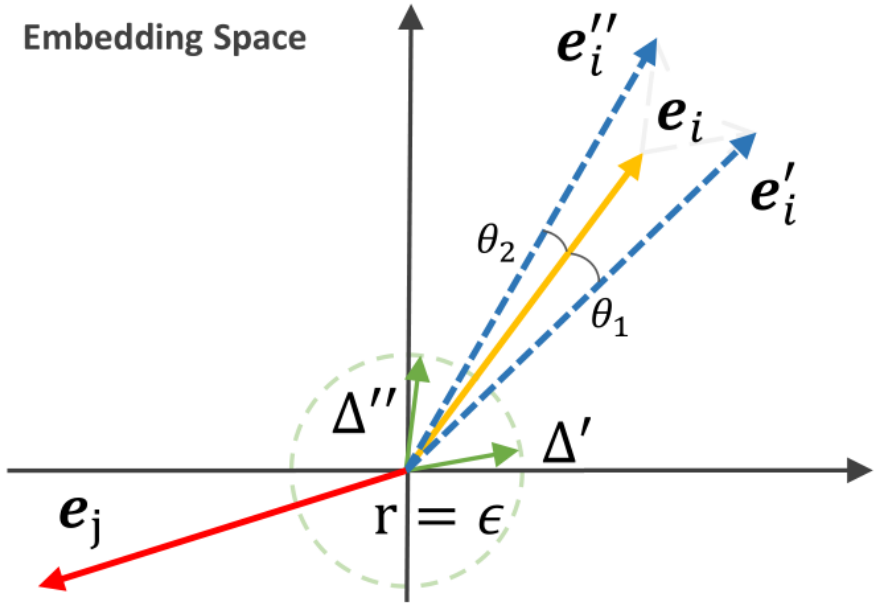
- **Add uniform noise to the embedding space.**

$$\mathbf{e}_i' = \mathbf{e}_i + \Delta_i', \quad \mathbf{e}_i'' = \mathbf{e}_i + \Delta_i''$$

where noise vectors Δ_i' and Δ_i'' are subject to $\|\Delta\|_2 = \epsilon$

$$\Delta = \bar{\Delta} \odot \text{sign}(\mathbf{e}_i), \ \bar{\Delta} \in \mathbb{R}^d \sim U(0,1).$$

$$\mathbf{E}' = \frac{1}{L}\big((\tilde{\mathbf{A}}\mathbf{E}^{(0)} + \Delta^{(1)}) + (\tilde{\mathbf{A}}(\tilde{\mathbf{A}}\mathbf{E}^{(0)} + \Delta^{(1)}) + \Delta^{(2)})) + \ldots + (\tilde{\mathbf{A}}^L\mathbf{E}^{(0)} + \tilde{\mathbf{A}}^{L-1}\Delta^{(1)} + \ldots + \tilde{\mathbf{A}}\Delta^{(L-1)} + \Delta^{(L)})\big)$$



Part2: A Simple Graph CL For Recommendation

- **Verify the uniform property.**

$$\mathcal{L}_{\text{uniform}}(f) = \log \mathop{\mathbb{E}}_{\substack{u,v \overset{i.i.d}{\sim} p_{\text{node}}}} e^{-2\|f(u)-f(v)\|_2^2}.$$

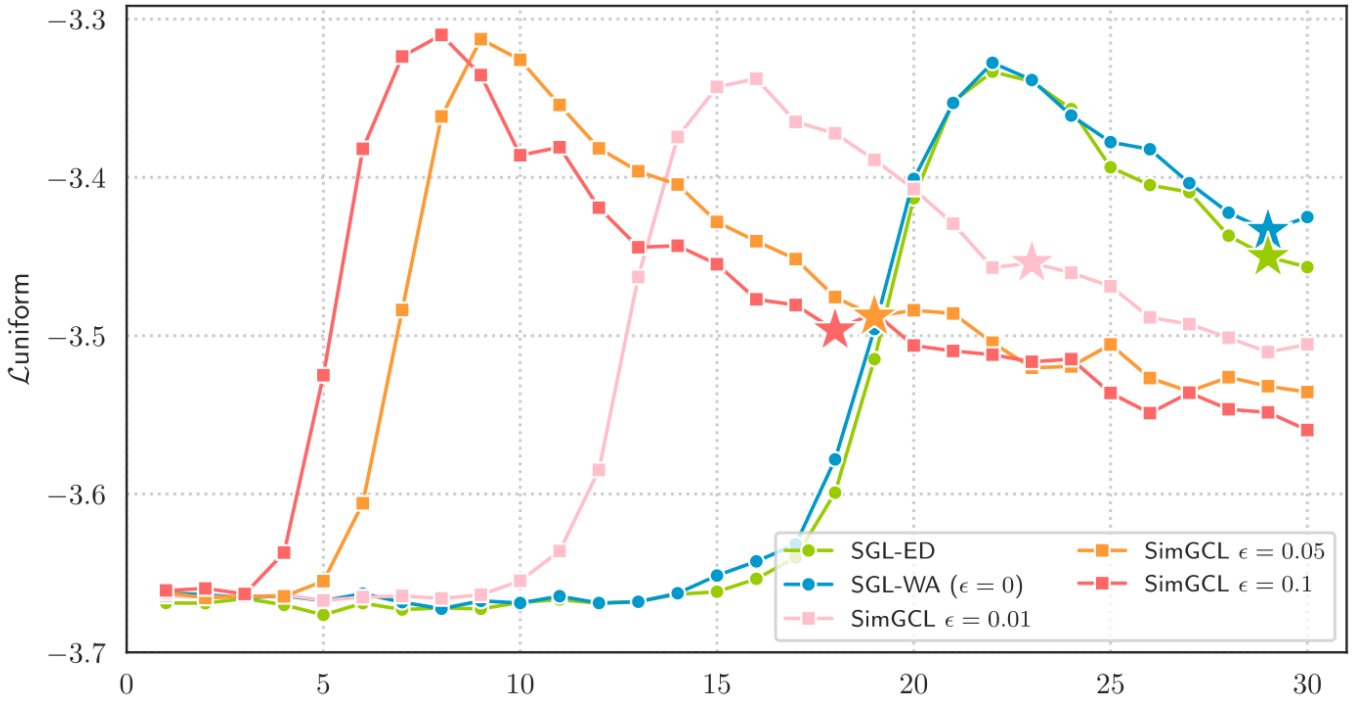


Figure 4: Trends of uniformity. The star indicates the epoch where the best recommendation performance is reached. Lower $\mathcal{L}_{\text{uniform}}$ numbers are better.

Results

- SimSGL VS SGL

Method		Douban-Book		Yelp2018		Amazon-Book	
		Recall	NDCG	Recall	NDCG	Recall	NDCG
1-Layer	LightGCN	0.1288	0.1081	0.0631	0.0515	0.0384	0.0298
	SGL-ND	0.1619 (+25.7%)	0.1448 (+34.0%)	0.0643 (+1.9%)	0.0529 (+2.7%)	0.0432 (+12.5%)	0.0334 (+12.1%)
	SGL-ED	0.1658 (+28.7%)	0.1491 (+37.9%)	0.0637 (+1.0%)	0.0526 (+2.1%)	0.0451 (+17.4%)	0.0353 (+18.5%)
	SGL-RW	0.1658 (+28.7%)	0.1491 (+37.9%)	0.0637 (+1.0%)	0.0526 (+2.1%)	0.0451 (+17.4%)	0.0353 (+18.5%)
	SGL-WA	0.1628 (+26.4%)	0.1454 (+34.5%)	0.0628 (-0.4%)	0.0525 (+1.9%)	0.0403 (+4.9%)	0.0320 (+7.4%)
	SimGCL	**0.1720 (+33.5%)**	**0.1519 (+40.5%)**	**0.0689 (+9.2%)**	**0.0572 (+11.1%)**	**0.0453 (+18.0%)**	**0.0358 (+20.1%)**
2-Layer	LightGCN	0.1485	0.1272	0.0622	0.0504	0.0411	0.0315
	SGL-ND	0.1622 (+9.2%)	0.1434 (+12.7%)	0.0658 (+5.8%)	0.0538 (+6.7%)	0.0427 (+3.9%)	0.0335 (+6.3%)
	SGL-ED	0.1721 (+15.9%)	0.1525 (+19.9%)	0.0668 (+7.4%)	0.0549 (+8.9%)	0.0468 (+13.9%)	0.0371 (+17.8%)
	SGL-RW	0.1710 (+15.2%)	0.1516 (+19.2%)	0.0644 (+3.5%)	0.0530 (+5.2%)	0.0453 (+10.2%)	0.0358 (+13.7%)
	SGL-WA	0.1687 (+13.6%)	0.1501 (+18.0%)	0.0653 (+5.0%)	0.0544 (+7.9%)	0.0453 (+10.2%)	0.0358 (+13.7%)
	SimGCL	**0.1770 (+19.2%)**	**0.1582 (+24.4%)**	**0.0719 (+15.6%)**	**0.0601 (+19.2%)**	**0.0507 (+23.4%)**	**0.0405 (+28.6%)**
3-Layer	LightGCN	0.1392	0.1188	0.0639	0.0525	0.0410	0.0318
	SGL-ND	0.1626 (+16.8%)	0.1450 (+22.1%)	0.0644 (+0.8%)	0.0528 (+0.6%)	0.0440 (+7.3%)	0.0346 (+8.8%)
	SGL-ED	0.1732 (+24.4%)	0.1551 (+30.6%)	0.0675 (+5.6%)	0.0555 (+5.7%)	0.0478 (+16.6%)	0.0379 (+19.2%)
	SGL-RW	0.1730 (+24.3%)	0.1546 (+30.1%)	0.0667 (+4.4%)	0.0547 (+4.2%)	0.0457 (+11.5%)	0.0356 (+12.0%)
	SGL-WA	0.1705 (+22.5%)	0.1525 (+28.4%)	0.0671 (+5.0%)	0.0550 (+4.8%)	0.0466 (+13.7%)	0.0373 (+18.4%)
	SimGCL	**0.1772 (+27.3%)**	**0.1583 (+33.2%)**	**0.0721 (+12.8%)**	**0.0601 (+14.5%)**	**0.0515 (+25.6%)**	**0.0414 (+30.2%)**

Results

- SimSGL VS Other SOTA

Method	Douban-Book		Yelp2018		Amazon-Book	
	Recall	NDCG	Recall	NDCG	Recal	NDCG
LightGCN	0.1485	0.1272	0.0639	0.0525	0.0411	0.0315
Mult-VAE	0.1310	0.1103	0.0584	0.0450	0.0407	0.0315
DNN+SSL	0.1366	0.1148	0.0483	0.0382	0.0438	0.0337
BUIR	0.1533	0.1317	0.0578	0.0461	0.0423	0.0326
MixGCF	0.1731	0.1552	0.0713	0.0589	0.0485	0.0378
SimGCL	**0.1772**	**0.1583**	**0.0721**	**0.0601**	**0.0515**	**0.0410**

- Running Time

Method	Douban-Book Time (s)	Yelp2018 Time (s)	Amazon-Book Time (s)
LightGCN	3.6	13.6	41.5
SGL-WA	4.4 (1.2x)	16.3 (1.2x)	47.0 (1.1x)
SGL-ED	13.3 (3.7x)	62.3 (4.6x)	235.3 (5.7x)
SimGCL	6.1 (1.7x)	27.9 (2.1x)	98.4 (2.4x)

Ablation Study

Different Kind of Noise

Method	Douban-Book		Yelp2018		Amazon-Book	
	Recall	NDCG	Recall	NDCG	Recal	NDCG
LightGCN	0.1485	0.1272	0.0639	0.0525	0.0411	0.0315
SimGCL$_a$	0.1561	0.1379	0.0604	0.0505	0.0455	0.0358
SimGCL$_p$	0.1751	0.1565	0.0708	0.0593	0.0514	0.0409
SimGCL$_g$	**0.1773**	**0.1586**	0.0718	0.0599	0.0511	0.0408
SimGCL$_u$	0.1772	0.1583	**0.0721**	**0.0601**	**0.0515**	**0.0414**

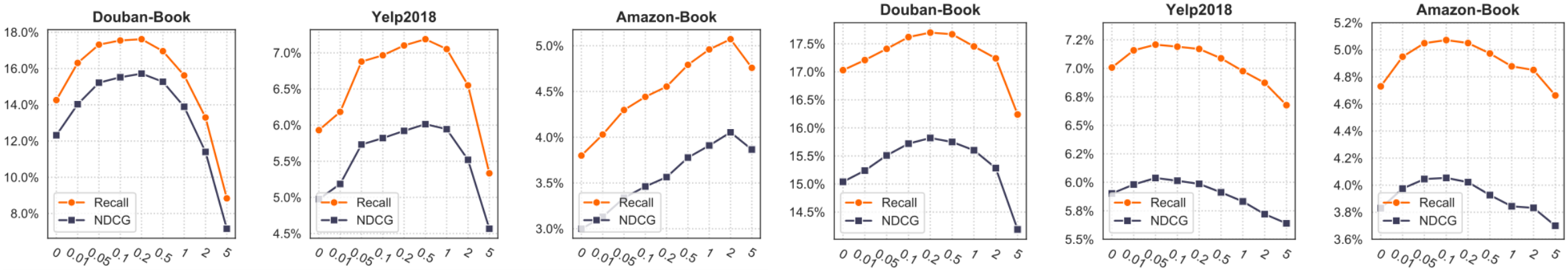
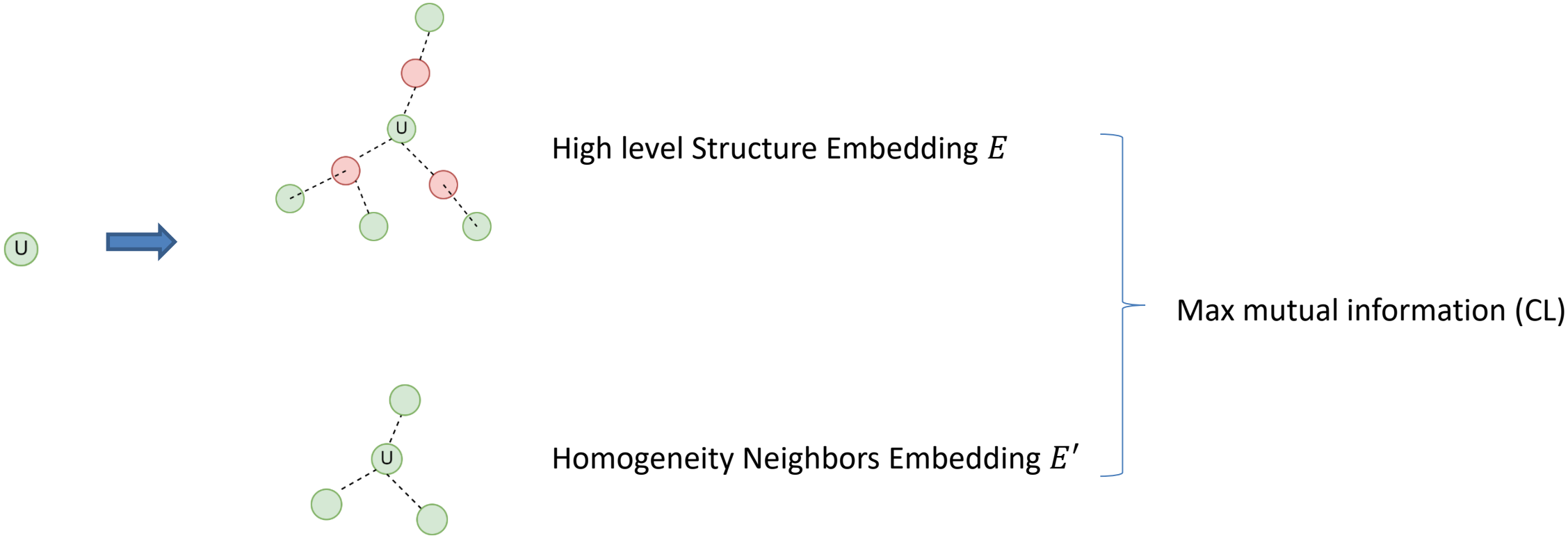


Figure 8: Influence of the magnitude λ of CL.

Figure 9: Influence of the noise magnitude ϵ.

Recent Work


U
U
U
High level Structure Embedding E
Homogeneity Neighbors Embedding E'
Max mutual information (CL)

Recent Work

$E^0 \in R^{(|U|+|V|)*d}$

Initial embedding (parameter to be learned)

$A = \begin{vmatrix} 0 & R \\ R^{\wedge} & 0 \end{vmatrix}, \qquad A^{\sim} = D^{-1/2} A D^{-1/2}$

Adjacency matrix

$E' = E^2 + E^4 + \cdots = A^{\sim}(A^{\sim}E^0) + \cdots$

User/item Homogeneity Neighbors Embedding

$$E = E^0 + E^1 + \cdots + E^L$$
$$= E^0 + A^{\sim}E^0 + A^{\sim}(A^{\sim}E^0) + A^{\sim}\left(A^{\sim}(A^{\sim}E^0)\right) + \cdots$$

LightGCN Encoder High Level Embedding

$$\sum_{u \in U} -log \frac{\exp(\cos(e'_u, e_u)/\tau)}{\sum_{j \in U} exp(\cos(e'_u, e_j)/\tau)}$$

CL Loss

Denoise part

$$w_{ui} = (\cos(e'_u, e_i) + 1)/\ 2$$

$$w = w * (w > \theta)$$

$$A = w\ A$$

Momentum update:

$$w^t = w^{t-1} * \beta + w\ * (1 - \beta)$$

Algorithm1
INPUT: A, GCN encoder, embedding E^0
1. For e in epochs:
2. Calculate denoise weight $\boldsymbol{w}$
3. Let $\boldsymbol{A} = \boldsymbol{w}\ \boldsymbol{A}$
4. $\boldsymbol{E},\ \boldsymbol{E'} = \boldsymbol{GCN}(\boldsymbol{E^0}, \boldsymbol{A})$
5. Calculate $\boldsymbol{L_{bpr}}\ loss$ with $\boldsymbol{E}$
6. Calculate $\boldsymbol{L_{cl}}$ loss with $\boldsymbol{E}$ and $\boldsymbol{E'}$
7. $\boldsymbol{Loss} = \boldsymbol{L_{bpr}} + \boldsymbol{\lambda}\ * \boldsymbol{L_{cl}}$

Result

ML-1M

Methods	clean	epochs	R@10	R@20	R@50	N@10	N@20	N@50	valid
lightgcn	clean	1000(10)	0.1822	0.2748	0.4382	0.2569	0.2659	0.3109	0.2053
SGL	clean	1000(10)	0.1851	0.2777	0.441	0.2598	0.2689	0.3147	0.2091
NCL	clean	1000(10)	0.1987	0.2976	0.463	0.2775	0.2876	0.3331	0.2204
SimSGL	clean	1000(10)	0.2011	0.3019	0.4693	0.2779	0.2889	0.3356	0.2233
Demo_v1	clean	1000(10)	**0.203**	**0.3035**	**0.47**	**0.2809**	**0.2912**	**0.3372**	**0.226**
lightgcn	noise	1000(10)	0.1676	0.2552	0.4123	0.2457	0.2526	0.295	0.1953
SGL	noise	1000(10)	0.1784	0.2683	0.4261	0.2553	0.2629	0.3061	0.2066
NCL	noise	1000(10)	0.1925	0.2895	0.4506	0.2712	0.2805	0.3244	0.2199
SimSGL	noise	1000(30)	0.1721	0.2576	0.4103	0.2425	0.2496	0.2921	0.1988
Demo_v2	noise	1000(10)	**0.1971**	**0.2908**	**0.4514**	**0.2739**	**0.2818**	**0.3257**	**0.2193**

Result

Yelp

Methods	clean	epochs	R@10	R@20	R@50	N@10	N@20	N@50	valid
lightgcn	clean	1000(10)	0.0733	0.1153	0.2018	0.0572	0.0709	0.0953	0.0572
SGL	clean	1000(10)	0.0866	0.1305	0.218	0.0703	0.0844	0.109	0.0691
NCL	clean	1000(10)	**0.0907**	0.1347	0.2177	**0.073**	**0.0872**	0.1107	**0.0711**
SimSGL	clean	1000(10)	0.0877	**0.1365**	**0.2277**	0.0697	0.0855	**0.1122**	0.0675
Demo_v1	clean	1000(10)	0.0886	0.1327	0.2166	0.0717	0.086	0.1097	0.0704
SGL	noise	1000(10)	0.0783	0.1214	0.2056	0.0632	0.0771	0.1008	0.0629
NCL	noise	1000(10)	0.0779	0.1182	0.1964	0.0623	0.0754	0.0975	0.0618
SimSGL	noise	1000(30)	0.079	0.1242	0.2142	0.0622	0.0769	0.1021	0.0616
Demo_v2	noise	1000(10)	**0.0845**	**0.1272**	0.2102	**0.0688**	**0.0825**	**0.1059**	**0.0677**

Thanks