# Tiger: Transferable Interest Graph Embedding for Domain-Level Zero-Shot Recommendation

Jianhuan Zhuo*
zhuojianhuan@iie.ac.cn
Institute of Information Engineering, Chinese Academy of Sciences & School of Cyber Security, University of Chinese Academy of Sciences
Beijing, China

Jianxun Lian
jianxun.lian@outlook.com
Microsoft Research Asia
Beijing, China

Lanling Xu
xulanling_sherry@163.com
Gaoling School of Artificial Intelligence, Renmin University of China
Beijing, China

Ming Gong
Linjun Shou
Daxin Jiang
migon@microsoft.com
lisho@microsoft.com
djiang@microsoft.com
Microsoft STC Asia
Beijing, China

Xing Xie
xingx@microsoft.com
Microsoft Research Asia
Beijing, China

Yinliang Yue†
yueyinliang@iie.ac.cn
Institute of Information Engineering, Chinese Academy of Sciences & School of Cyber Security, University of Chinese Academy of Sciences
Beijing, China

(CIKM-2022) https://github.com/JianhuanZhuo/Tiger-Code-and-Dataset-for-CIKM2022

gesis Leibniz-Institut für Sozialwissenschaften

Forschungszentrum · Research Center L3S

**Reported by Ke Gan**

1. Introduction
2. Approach
3. Experiments

# Introduction

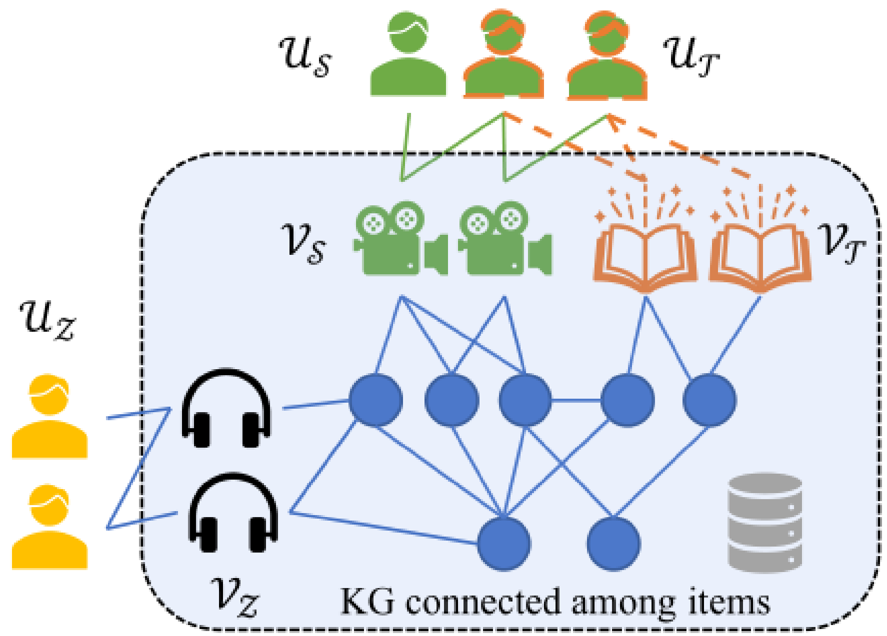



**Figure 1: Interest graph: connecting isolated user-item datasets with an knowledge graph**

connect isolated collaborative filtering datasets with a knowledge graph tailored to recommendations

# Introduction

Table 1: A comparison of DZSR with related tasks.

| Task | interactions used to train | |
|---|---|---|
| | source domain | target domain |
| In-domain Rec. [30, 34, 39] | × | ✓ |
| Cold-start Rec. [1, 26, 33] | ✓ | partial |
| Cross-domain Rec. [25, 51] | ✓ | ✓ |
| Domain-level ZSR [this paper] | ✓ | × |

$$\mathcal{D} \in \{\mathcal{T}, \mathcal{S}\}$$

$$\mathcal{U}_{\mathcal{T}} \subset \mathcal{U}_{\mathcal{S}},$$

$$\mathcal{V}_{\mathcal{T}} \cap \mathcal{V}_{\mathcal{S}} = \emptyset.$$

$$\mathcal{P}_{\mathcal{D}}^{+} = \{(u,v)|y(u,v) = 1, u \in \mathcal{U}, v \in \mathcal{V}_{\mathcal{D}}\} \subset \mathcal{I}_{\mathcal{D}},$$

$$\mathcal{P}_{\mathcal{D}}^{-} = \{(u,v)|y(u,v) = 0, u \in \mathcal{U}, v \in \mathcal{V}_{\mathcal{D}}\} \subset \mathcal{I}_{\mathcal{D}}.$$

$$\mathcal{H}_{u}^{\mathcal{D}} = \{v|y(u,v) = 1, v \in \mathcal{V}_{\mathcal{D}}\}$$

# Approach

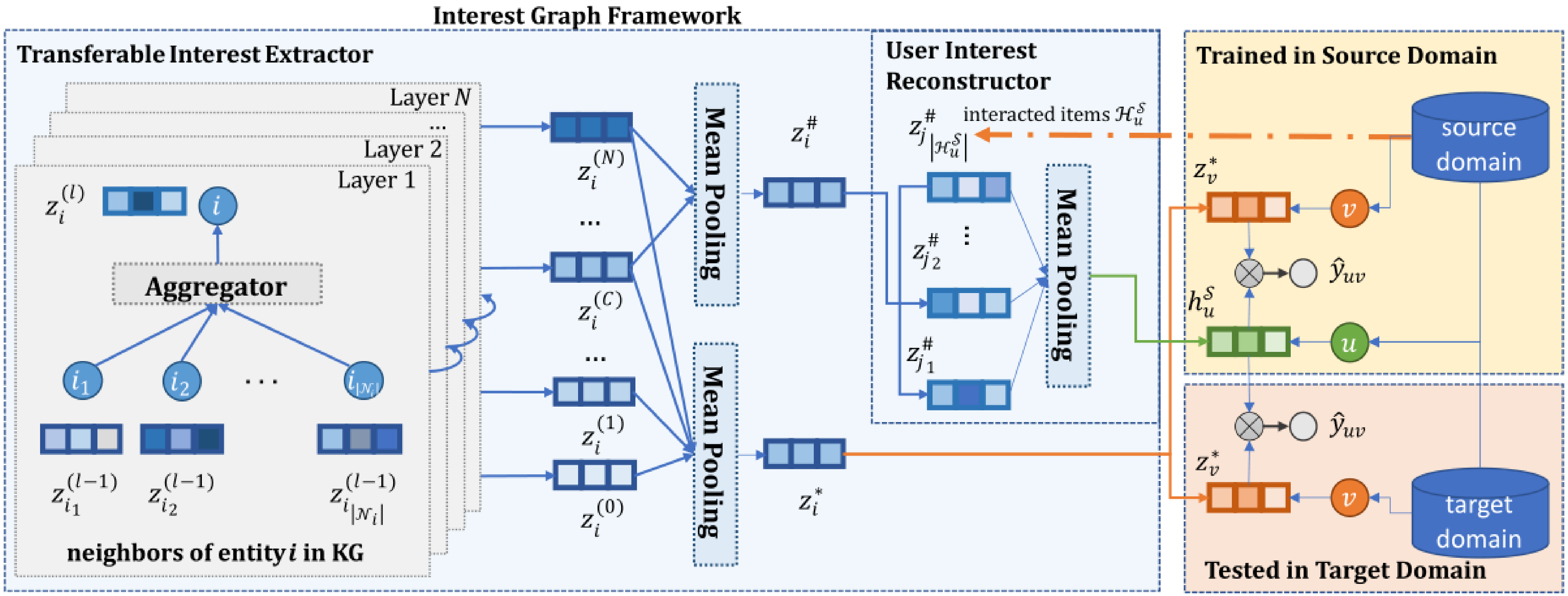
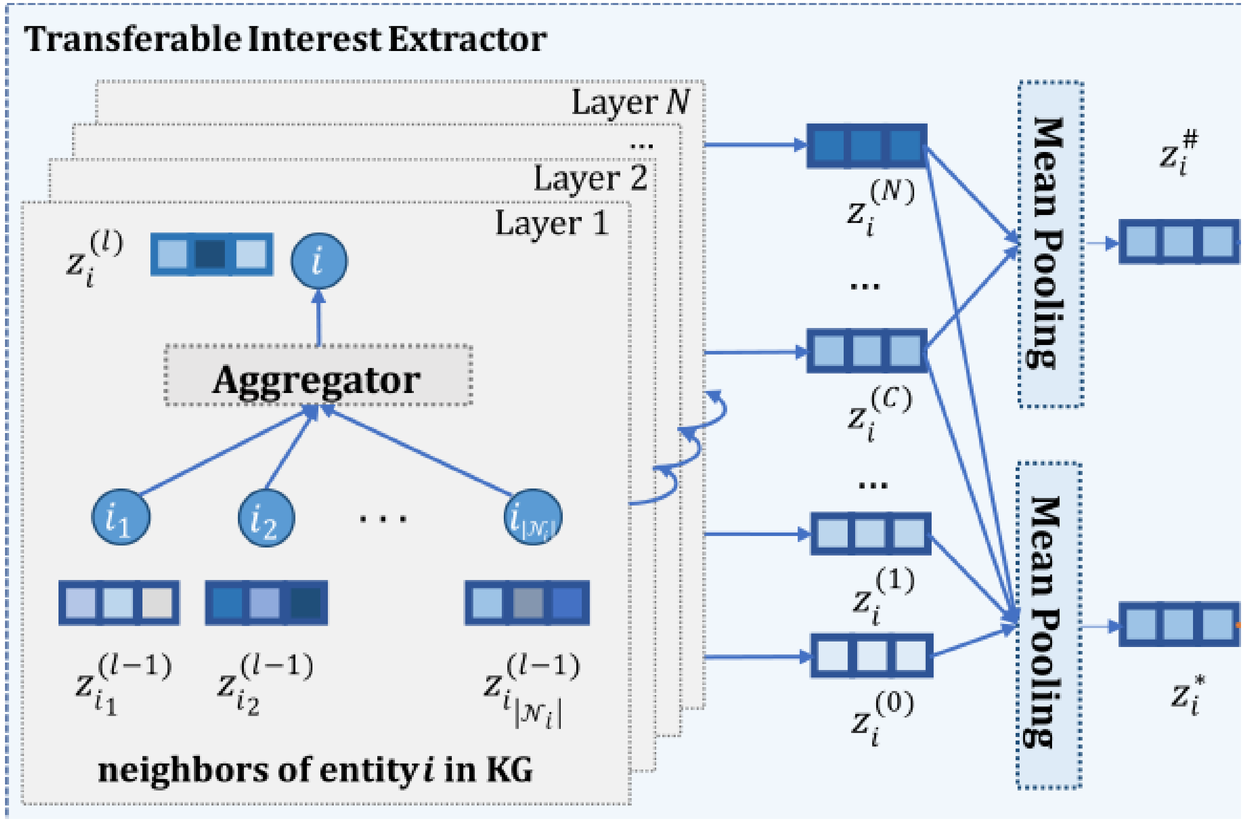



Figure 2: An overview of the proposed Tiger model

# Approach




$$\Theta^* = \arg\min_{\Theta} \mathcal{L}(y(\mathcal{S}), \mathcal{S}; \mathcal{G}) \tag{1}$$

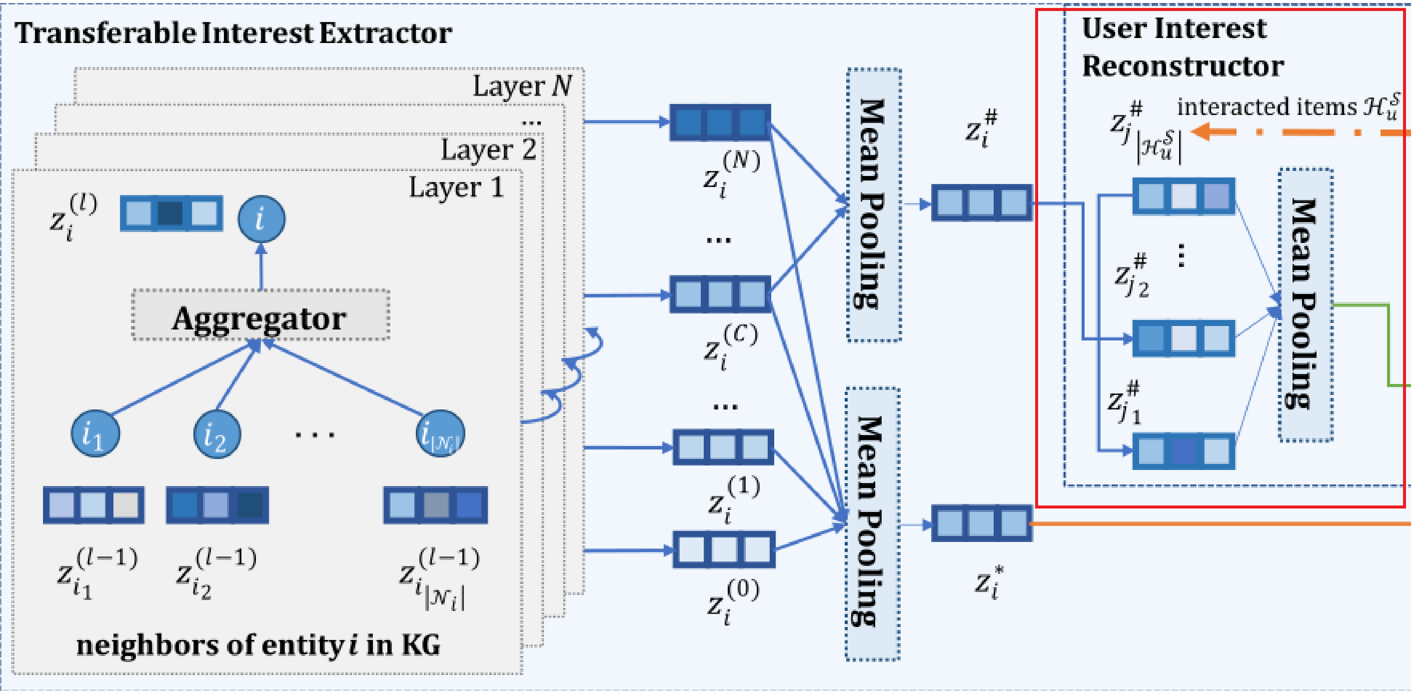
$$\mathbf{z}_i^{(0)} = \mathbf{e}_i \tag{2}$$

$$\mathbf{z}_i^{(l)} = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \mathbf{z}_j^{(l-1)} \tag{3}$$

$$\mathbf{z}_i^{*} = \frac{1}{N+1} \sum_{j=0}^{N} \mathbf{z}_i^{j} \tag{4}$$

$$\mathbf{z}_i^{\#} = \frac{1}{N-C+1} \sum_{j=C}^{N} \mathbf{z}_i^{j} \tag{5}$$

# Approach

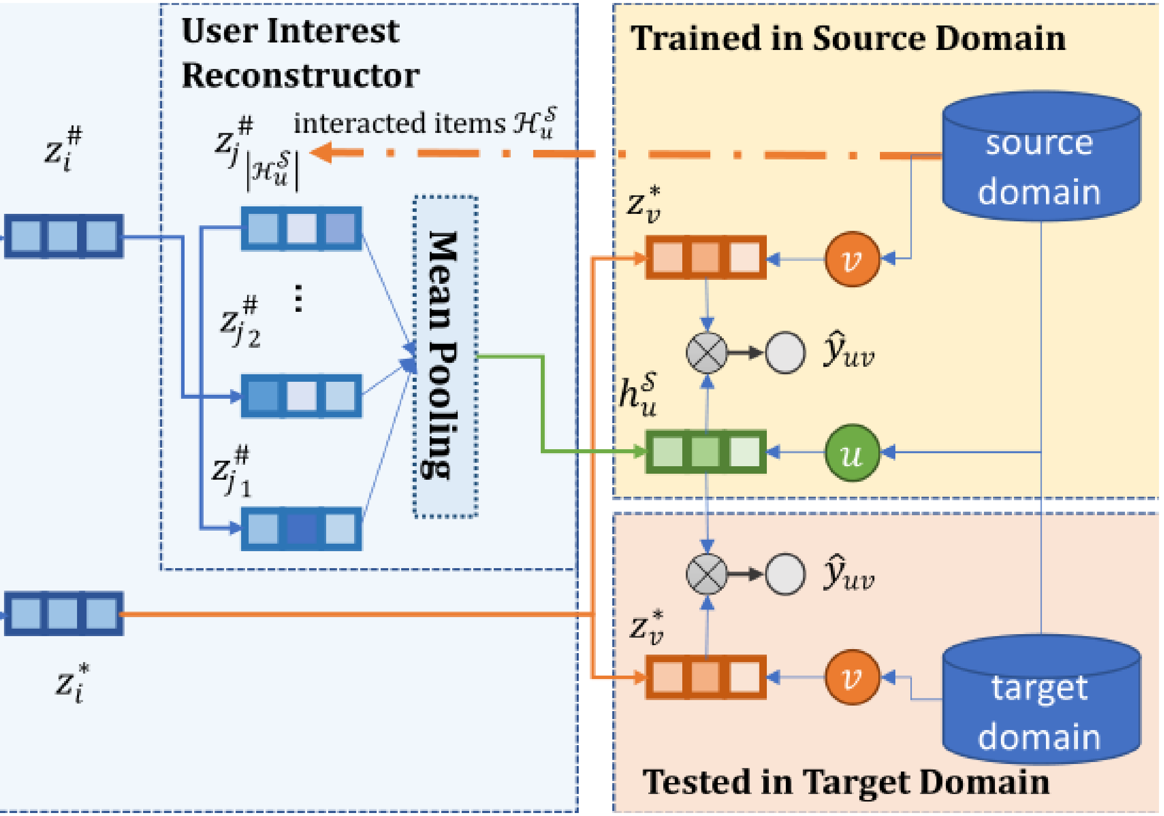



$$\mathbf{z}_v^* = \mathbf{z}_i^* \text{ and } \mathbf{z}_v^\# = \mathbf{z}_i^\#.$$

$$\mathbf{h}_u = \mathbf{e}_u \tag{6}$$

$$\mathbf{h}_u^S = \frac{1}{|\mathcal{H}_u^S|} \sum_{v \in \mathcal{H}_u^S} \mathbf{z}_v^\# \tag{7}$$

# Approach




$$\hat{y}_{uv} = \mathbf{z}_v^{*\top}\mathbf{h}_u^S \tag{8}$$

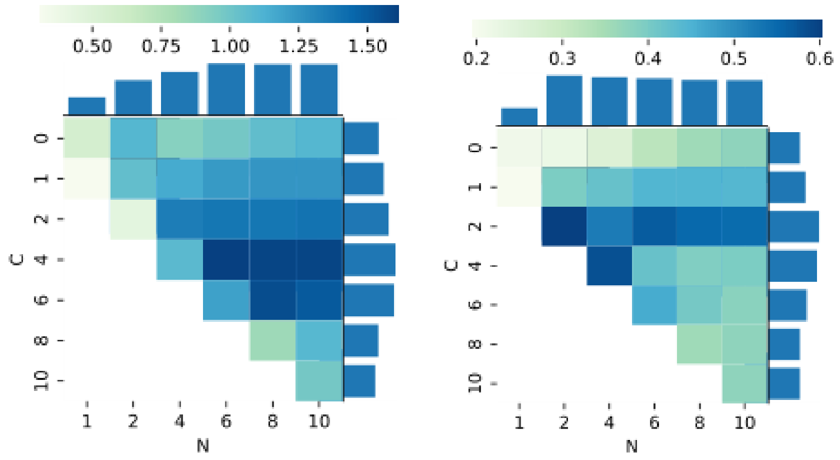
$$\mathcal{L} = -\sum_{u\in\mathcal{U}}\sum_{v\in\mathcal{H}_u^S}\sum_{v'\notin\mathcal{H}_u^S}\ln\sigma(\hat{y}_{uv}-\hat{y}_{uv'}) \tag{9}$$

# Experiment

**Table 3: Performance comparison. The higher value of all measures means the better performance. The best zero-shot result is highlighted in bold and the runner-up is underlined, the same below. * indicates the oracle result.**

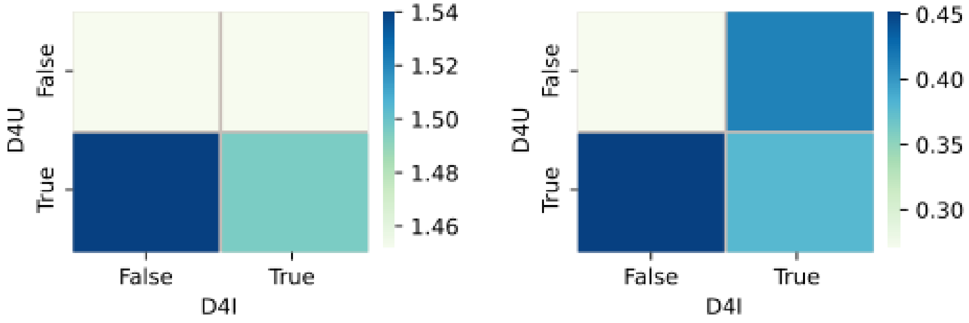
| Model | Source | Target | H@10 | N@10 | H@100 | N@100 |
|---|---|---|---|---|---|---|
| Random | - | AM | 0.0620 | 0.0282 | 0.6211 | 0.1300 |
| BPR (Oracle) | - | AM | 2.8440* | 1.4066* | 14.1040* | 3.5393* |
| NLP-based | AB | AM | 0.1307 | 0.0488 | 1.2900 | 0.2586 |
| TransE | AB | AM | 0.3203 | 0.1580 | 1.4858 | 0.3719 |
| KGCN | AB | AM | 0.5368 | 0.2491 | 3.6032 | 0.8155 |
| Tiger (UserAsEmb) | AB | AM | 0.7711 | <u>0.4198</u> | 4.7242 | 1.1510 |
| Tiger (normal) | AB | AM | <u>0.9312</u> | 0.3751 | **7.3072** | <u>1.5401</u> |
| Tiger (+ out domain) | ML+LFM+AB | AM | **1.0854** | **0.7484** | <u>7.1886</u> | **1.8811** |
| Random | - | AB | 0.0211 | 0.0096 | 0.2111 | 0.0442 |
| BPR (Oracle) | - | AB | 0.7859* | 0.4051* | 3.9472* | 1.0014* |
| NLP-based | AM | AB | 0.0505 | 0.0202 | 0.4580 | 0.0948 |
| TransE | AM | AB | 0.0623 | 0.0293 | 0.3915 | 0.0913 |
| KGCN | AM | AB | 0.0860 | 0.0487 | 0.7117 | 0.1657 |
| Tiger (UserAsEmb) | AM | AB | 0.2343 | 0.1185 | 1.2604 | 0.3100 |
| Tiger (normal) | AM | AB | <u>0.3055</u> | <u>0.1370</u> | <u>1.9692</u> | <u>0.4519</u> |
| Tiger (+ out domain) | ML+LFM+AM | AB | **0.5872** | **0.3392** | **2.5178** | **0.5659** |

# Experiment




(a) AmazonMovie (b) AmazonBook

**Figure 3: Performance comparison between different settings. The darker means the better.**



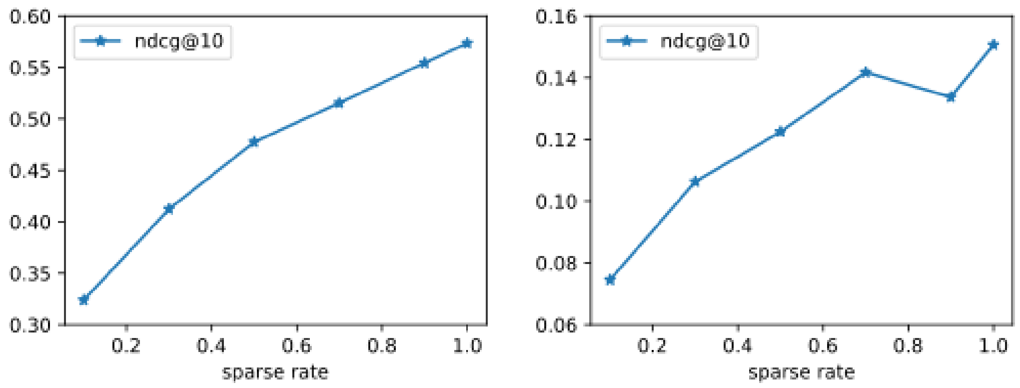

(a) AmazonMovie (b) AmazonBook

**Figure 4: Result comparison of decomposed representations**

# Experiment

$$\Theta^* = \arg\min_{\Theta} \mathcal{L}(y(\mathcal{S}+\mathcal{Z}), \mathcal{S}+\mathcal{Z}; \mathcal{G}) \qquad (10)$$

$$\Theta^* = \arg\min_{\Theta} \mathcal{L}(y(\mathcal{Z}), \mathcal{Z}; \mathcal{G}) \qquad (11)$$




(a) AmazonMovie (b) AmazonBook

**Figure 5: Performance under different sparse levels of KG**

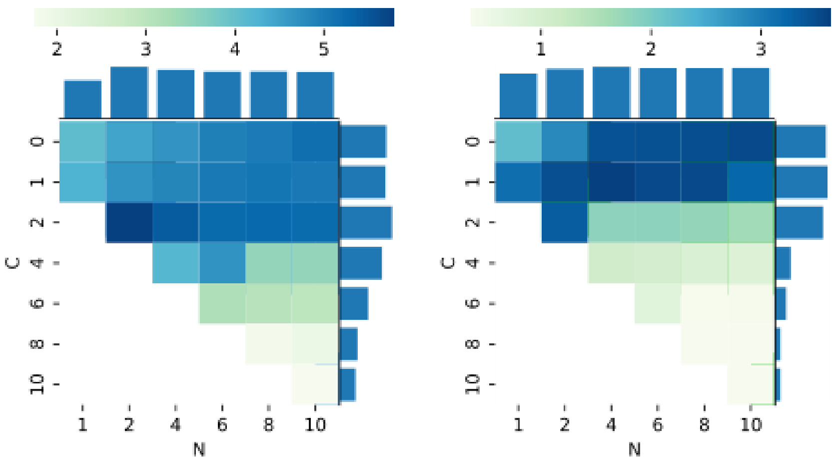
**Table 4: Performance comparison of Tiger trained on different datasets**

| Training | Target | H@10 | N@10 | H@100 | N@100 |
|---|---|---|---|---|---|
| AB | AM | 0.9312 | 0.3751 | 7.3072 | 1.5401 |
| ML | AM | 0.6139 | 0.2391 | **7.3577** | 1.4408 |
| LFM | AM | 0.9253 | 0.4784 | 6.3968 | 1.4474 |
| ML+AB | AM | **1.1210** | 0.5418 | 6.7438 | 1.6184 |
| LFM+AB | AM | 1.1121 | 0.3882 | 7.2598 | 1.4984 |
| ML+LFM+AB | AM | 1.0854 | **0.7484** | 7.1886 | **1.8811** |
| AM | AB | 0.3055 | 0.1370 | 1.9692 | 0.4519 |
| ML | AB | 0.3203 | 0.1456 | 1.7527 | 0.4121 |
| LFM | AB | 0.1275 | 0.0397 | 2.1501 | 0.4275 |
| ML+AM | AB | 0.5397 | 0.3013 | 2.0848 | **0.6007** |
| LFM+AM | AB | 0.4938 | 0.3211 | 1.6770 | 0.5434 |
| ML+LFM+AM | AB | **0.5872** | **0.3392** | **2.5178** | 0.5659 |

# Experiment

Table 5: Performance comparison in the subsequent optimization

| Model | Target | H@10 | N@10 | H@100 | N@100 |
|---|---|---|---|---|---|
| **Random** | AM | 0.0620 | 0.0282 | 0.6211 | 0.1300 |
| **BPR** | AM | 2.8440 | 1.4066 | 14.1040 | 3.5393 |
| **KGCN** | AM | 4.2912 | 2.1425 | 20.1097 | 5.1583 |
| **Tiger** | AM | **5.0445** | **2.5381** | **21.9781** | **5.7812** |
| **Random** | AB | 0.0211 | 0.0096 | 0.2111 | 0.0442 |
| **BPR** | AB | 0.7859 | 0.4051 | 3.9472 | 1.0014 |
| **KGCN** | AB | 3.3007 | 1.8283 | 12.0196 | 3.5146 |
| **Tiger** | AB | **4.8221** | **2.5774** | **15.3203** | **4.6344** |




(a) AmazonMovie (b) AmazonBook

Figure 6: Performance with different settings of GCN layer for the normal recommendation

# Thank you!