Knowledge-Adaptive Contrastive Learning for Recommendation

Hao Wang
Beijing University of Posts and Telecommunications
Beijing, China

Yao Xu
Researcher
Beijing, China

Cheng Yang*
Chuan Shi
Beijing University of Posts and Telecommunications
Beijing, China

Xin Li
Researcher
Beijing, China

Ning Guo
Researcher
Beijing, China

Zhiyuan Liu
Tsinghua University
Beijing, China

Code: https://github.com/wsdmanonymous/KACL

WSDM 2023

Reported by Nengqiang Xiang

Introduction

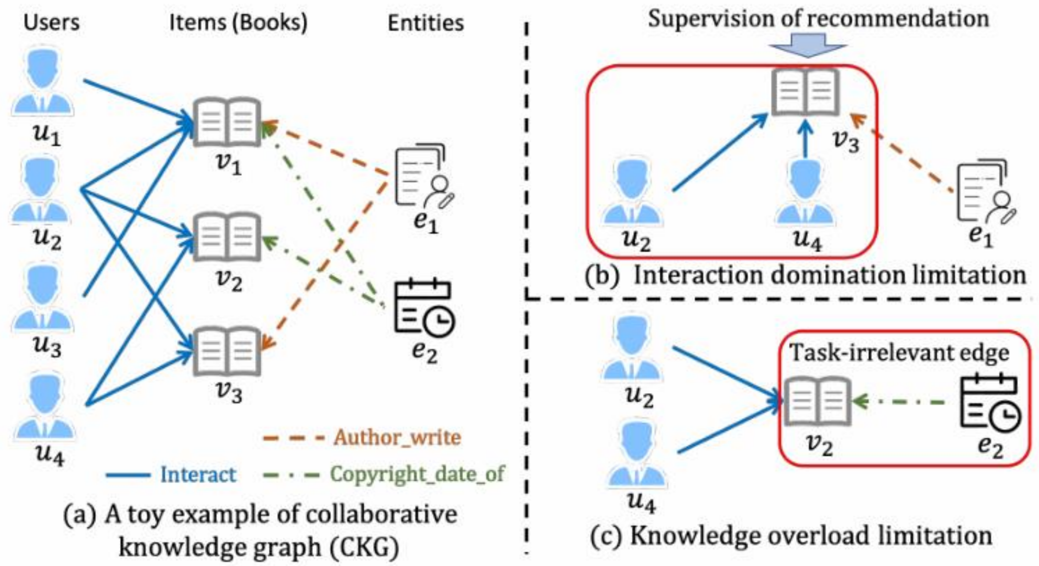


Figure 1: An illustration of two limitations in the state-of-the-art paradigm for KG-based recommendation (*i.e.*, GNN methods based on CKG modeling).

There are two limitations in the existing methods:

(1) Interaction domination: the supervision signal of user-item interaction will dominate the model training, and thus the information of KG is barely encoded in learned item representations;

(2) Knowledge overload: KG contains much recommendation-irrelevant information, and such noise would be enlarged during the message aggregation of GNNs.

In this paper, the author proposes a novel algorithm named Knowledge-Adaptive Contrastive Learning(KACL) to address these challenges.

Method

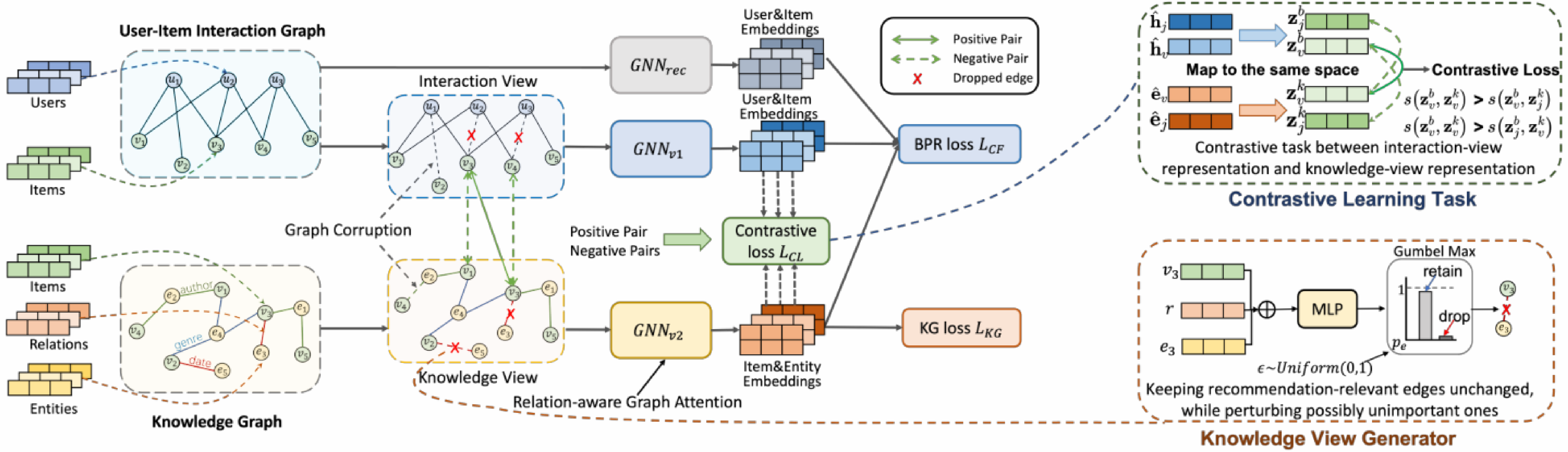
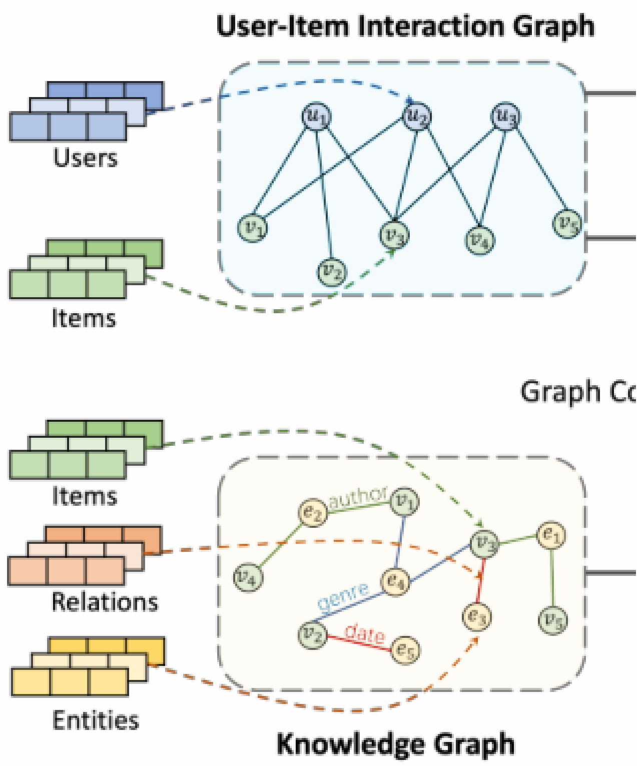


Figure 2: The model architecture of our proposed Knowledge-Adaptive Contrastive Learning (KACL).

Method



PROBLEM FORMULATION:

User-Item Interaction Graph:

$$\mathcal{G}_b = \{(u, y_{uv}, v)\}$$

Knowledge Graph

$$\mathcal{G}_k = \{(h, r, t)\}$$

Classical GNN-based Recommender:

$$\alpha_{ij} = \frac{\exp(LeakyReLU(a_b^T[W_b\mathbf{h}_i||W_b\mathbf{h}_j]))}{\sum_{k\in\mathcal{N}_i}\exp(LeakyReLU(a_b^T[W_b\mathbf{h}_i||W_b\mathbf{h}_k]))}, \tag{1}$$

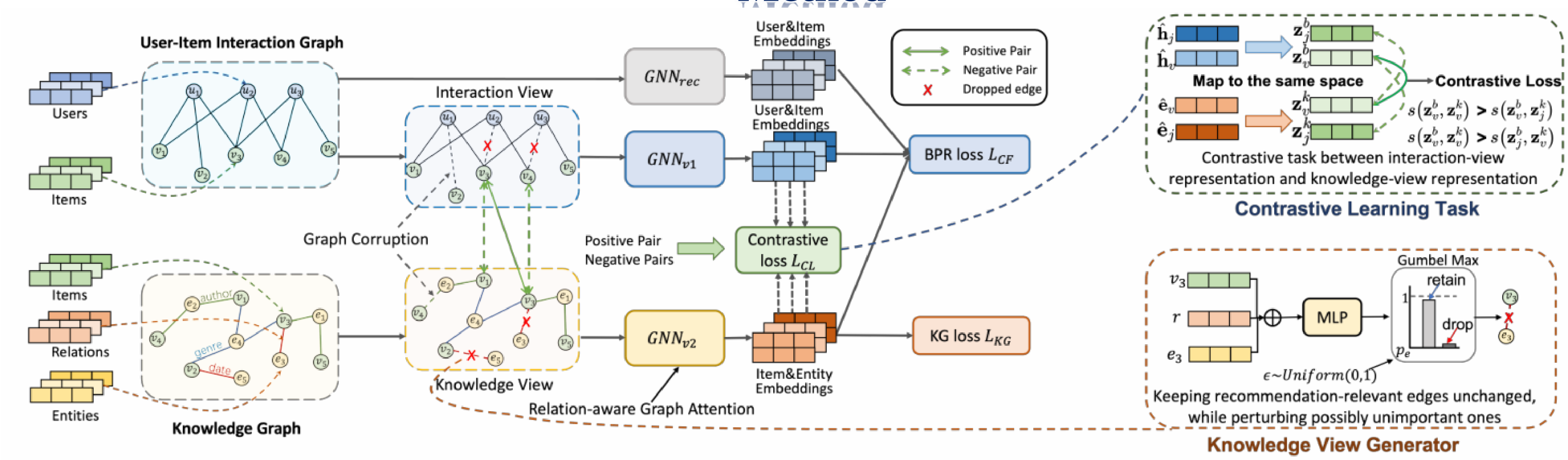
$$\mathbf{h}_{\mathcal{N}_i} = \sum_{j\in\mathcal{N}_i}\alpha_{ij}\mathbf{h}_j$$

$$\mathbf{h}_i^{(l)} = \sigma(W_b^{(l)}(\mathbf{h}_i^{(l-1)} + \mathbf{h}_{\mathcal{N}_i}^{(l-1)})), \tag{2}$$

$$\mathcal{L}_{CF}(u, v^+, v^-) = -\log\sigma(y(u, v^+) - y(u, v^-)). \tag{3}$$

$$y(u, v) = \mathbf{h}_u{}^T\mathbf{h}_v$$

Method



Knowledge-Adaptive Contrastive Learning：

Adaptive Data Augmentation

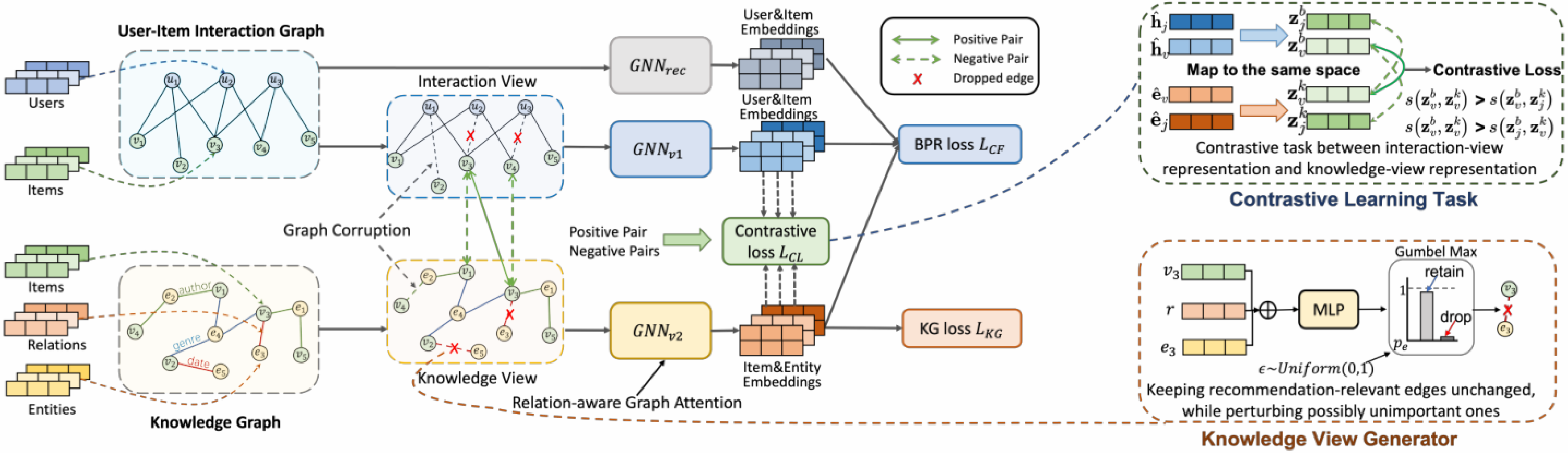
$$w_e^b = MLP_b([\mathbf{h}_u^{(0)} || \mathbf{h}_v^{(0)}]), \quad (4)$$

$$p_e^b = \sigma((\log(\epsilon) - \log(1-\epsilon) + w_e^b)/\tau_b) \quad (5)$$

Relation-aware Graph Attention for Node Encoding

$$\alpha_{ht} = \frac{\exp(LeakyReLU(a_k^T [W_k \mathbf{e}_h || W_r \mathbf{m}_{r(\langle h,t \rangle)} || W_k \mathbf{e}_t]))}{\sum_{j \in \mathcal{N}_h} \exp(LeakyReLU(a_k^T [W_k \mathbf{e}_h || W_r \mathbf{m}_{r(\langle h,j \rangle)} || W_k \mathbf{e}_j]))}, \quad (7)$$

Method



Contrastive Learning Task

$$(\hat{\mathbf{h}}_v, \hat{\mathbf{e}}_v) \longrightarrow (\mathbf{z}_v^b, \mathbf{z}_v^k)$$

$$\mathcal{L}_{CL}(v) = -\log \frac{\exp(s(\mathbf{z}_v^b, \mathbf{z}_v^k)/\tau_{cl})}{\sum_{j \in \mathbb{N} \cup \{v\}} \exp(s(\mathbf{z}_v^b, \mathbf{z}_j^k)/\tau_{cl}) + \exp(s(\mathbf{z}_j^b, \mathbf{z}_v^k)/\tau_{cl})} \quad (8)$$

Model Prediction and Optimization:

$$y(u, v) = (\mathbf{h}_u || \hat{\mathbf{h}}_u || \hat{\mathbf{e}}_u)^T (\mathbf{h}_v || \hat{\mathbf{h}}_v || \hat{\mathbf{e}}_v). \quad (9)$$

$$f(h, r, t) = \hat{\mathbf{e}}_h^T R_r \hat{\mathbf{e}}_t, , \quad (10)$$

$$\mathcal{L}_{KG}(h, r, t^+, t^-) = -\log \sigma(f(h, r, t^-) - f(h, r, t^+)), \quad (11)$$

$$\mathcal{L} = \mathcal{L}_{CF} + \lambda_1 \mathcal{L}_{CL} + \lambda_2 \mathcal{L}_{KG}, \quad (12)$$

Experiments

Table 1: The statistics of datasets.

Dataset	Amazon-Book	LastFM	Movielens
# Users	70,679	23,566	37,385
# Items	24,915	48,123	6,182
# Interactions	846,434	3,034,763	539,300
# Entities	113,487	106,389	24,536
# Relations	39	9	20
# Triplets	2,557,746	464,567	237,155

Table 2: Experimental results of recall@20 and ndcg@20 in top-K recommendation.

Model	Amazon-Book		LastFM		Movielens	
	recall	ndcg	recall	ndcg	recall	ndcg
BPR-MF	0.1321	0.0682	0.0715	0.0637	0.4052	0.2609
CKE	0.1352	0.0699	0.0746	0.0652	0.4106	0.2669
KGCN	0.1464	0.0769	0.0819	0.0705	0.4237	0.2753
KGNNLS	0.1448	0.0759	0.0806	0.0695	0.4218	0.2741
KGAT	0.1507	0.0802	0.0877	0.0749	0.4532	0.3007
CKAN	0.1467	0.0702	0.0812	0.0690	0.4314	0.2891
KGPL	0.1503	0.0785	0.0896	0.0751	0.4417	0.2864
DSKReG	0.1551	0.0863	0.0924	0.0816	0.4589	0.3017
KGIN	0.1631	0.0881	0.0967	0.0847	0.4661	0.3120
CKER	0.1619	0.0863	0.0951	0.0832	-	-
KGCL	0.1569	0.0833	0.0899	0.0793	0.4516	0.2967
KACL	**0.1657**	**0.0915**	**0.1133**	**0.0989**	**0.4752**	**0.3278**
%Improv	1.35%	3.86%	17.18%	16.77%	1.95%	5.06%

Experiments

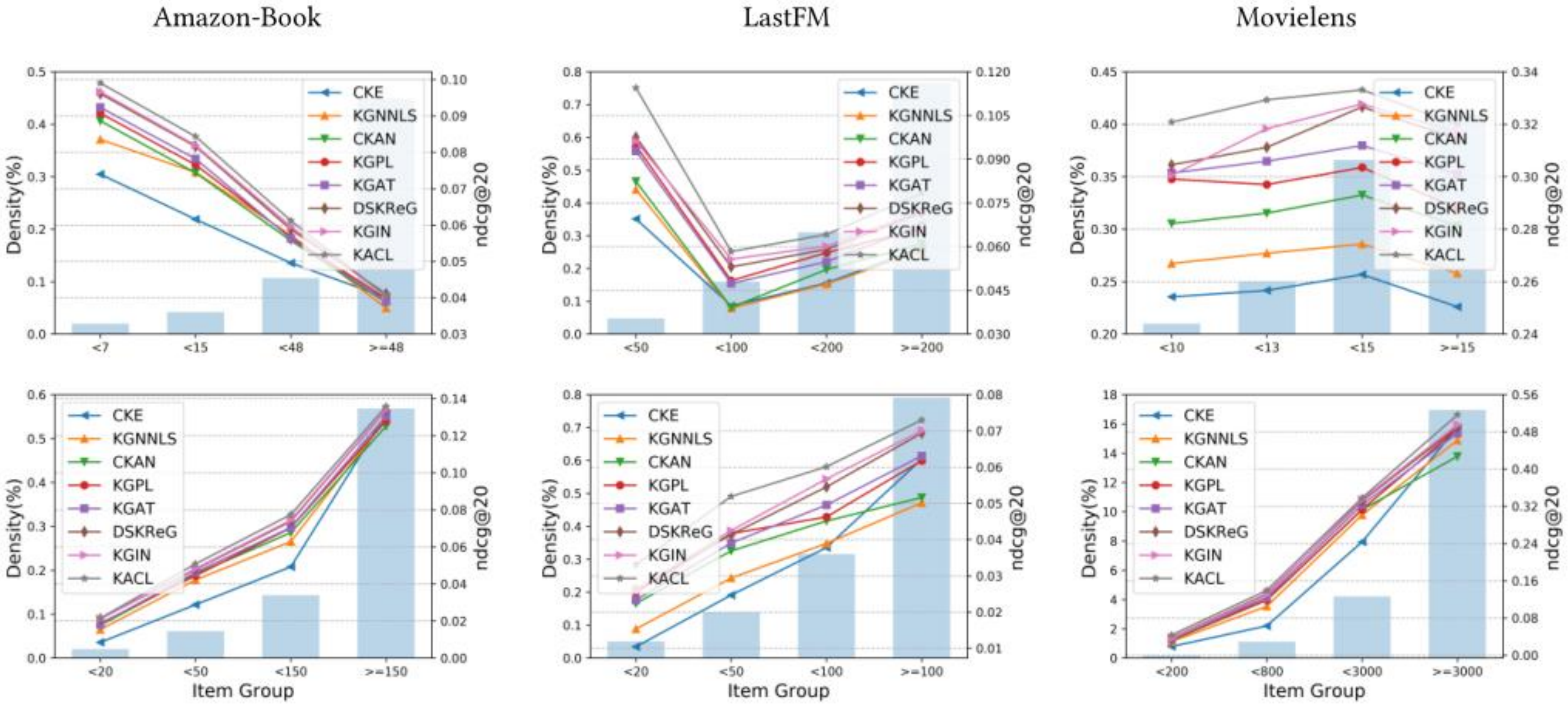
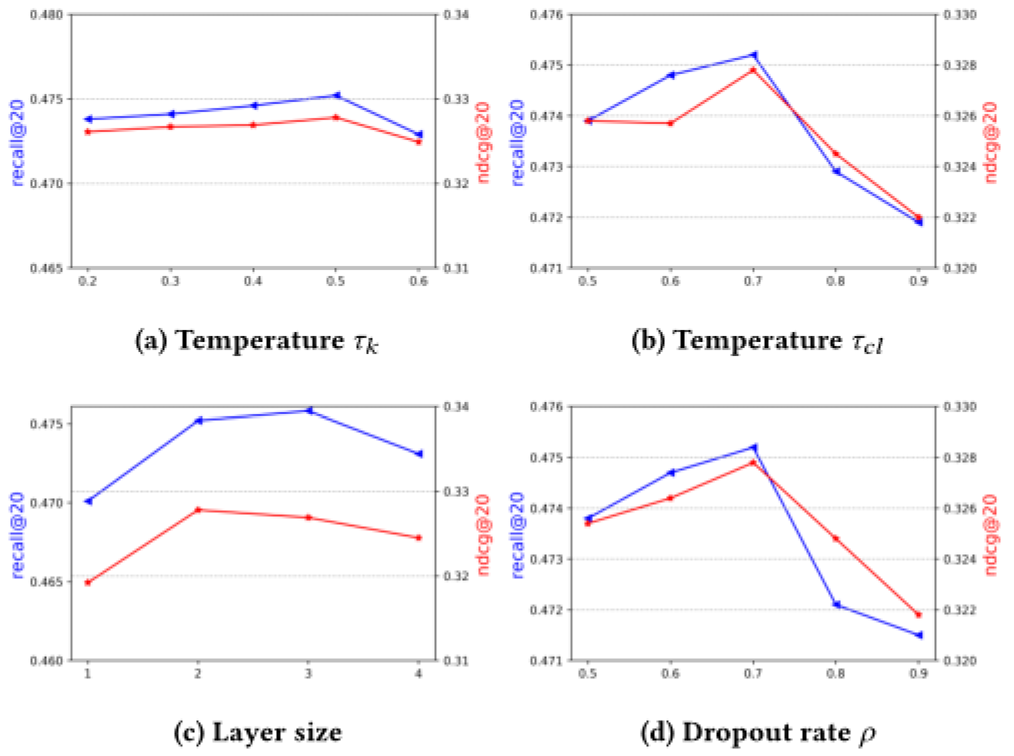


Figure 3: Performance comparison over user/item groups with different sparsity. The background histograms indicate the density of each group. Each line presents the ndcg@20 performance of the corresponding method. Due to space limitation, we omit the results of KGCN which have a similar trend to that of KGNNLS. Best viewed in color.

Experiments

Table 3: Comparison of KACL and its ablated variants. The improvements of KACL over all variants are significant (0.05 level paired t-test).

Model	Amazon-Book		LastFM		Movielens	
	recall	ndcg	recall	ndcg	recall	ndcg
w/o CL	0.1563	0.0802	0.0912	0.0808	0.4614	0.3094
w/o Ada	0.1614	0.0883	0.1091	0.0946	0.4719	0.3227
w/o KG	0.1584	0.0827	0.1026	0.0912	0.4695	0.3196
KACL	0.1657*	0.0915*	0.1133*	0.0989*	0.4752*	0.3278*



(a) Temperature τ_k

(b) Temperature τ_{cl}

(c) Layer size

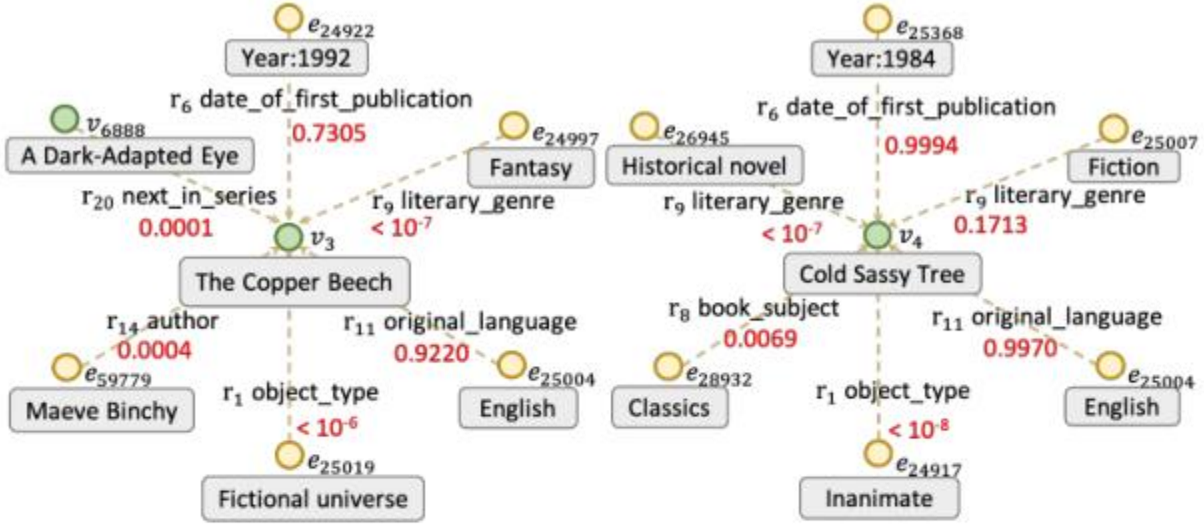
(d) Dropout rate ρ

Figure 4: Hyper-parameter sensitivity analysis on Movielens.

Experiments

Table 4: Top-5 relations with the lowest/highest learned dropping ratios on Amazon-Book dataset.

Id	Relation	Dropping ratio	Triplets
1	object_type	$< 10^{-4}$	666,473
8	book_subject	$< 10^{-4}$	39,180
9	literary_genre	$< 10^{-4}$	50,388
14	author	$< 10^{-4}$	37,255
17	book_character	$< 10^{-4}$	10,182
2	copyright_date	0.853	11,657
7	notable_types	0.852	106,637
6	date_of_first_publication	0.810	17,298
11	original_language	0.797	13,916
12	is_reviewed	0.601	4,752



(a) Book "The Copper Beech" **(b) Book "Cold Sassy Tree"**

Figure 5: Case study of learned edge dropping probabilities on Amazon-Book dataset.