Learning Common Interests for Cold-Start Group Recommendation

Yu-Ting Cheng
National Taiwan Normal University, Taiwan
kirstin.yt.cheng@gmail.com

Pin-Hsin Hsiao
National Taiwan University, Taiwan
henry1233815@gmail.com

Chiou-Shann Fuh
National Taiwan University, Taiwan
fuh@csie.ntu.edu.tw

Pu-Jen Cheng
National Taiwan University, Taiwan
pjcheng@csie.ntu.edu.tw

Abstract

Previous studies on recommender systems have primarily focused on learning implicit preferences from individual user behaviors or enhancing recommendation performance by identifying similar users. However, in real-life scenarios, group decision-making is often required, such as when a group of friends decides which movie to watch together. Thus, discovering common interests has become a key research issue in group recommendation.

The most straightforward approach to group recommendation is to model the past joint behaviors of a user group. Nevertheless, this method fails to handle newly formed groups with no historical interactions. To address this limitation, we apply Graph Convolution Networks to capture high-order structural features within the user–item interaction graph, thereby uncovering the potential common interests of cold-start groups. Experimental evaluations on three real-world datasets demonstrate the feasibility and effectiveness of the proposed method.

Keywords: Discovery of Common Interests, Cold-Start Groups, Group Recommendation

1 Introduction

Recommender systems have become an essential component of modern digital experiences, assisting users in exploring products and potential social connections by analyzing their behaviors and preferences. For example, platforms such as Amazon and TripAdvisor provide personalized product and hotel suggestions based on user interactions and reviews.

Despite the impressive success of existing recommender systems in delivering personalized recommendations, they often overlook group decision-making scenarios, such as a group of friends choosing a movie to watch together or deciding on a restaurant for dining. Our work aims to bridge this gap by uncovering common interests within user groups, particularly for *cold-start groups* (a set of users who come together for the first time and for whom the system has no prior record of collective interactions or shared history). This capability not only enables recommendations that align with collective group preferences but also opens new possibilities for collaborative content creation, such as co-writing a script.

Prior research on group recommendation (Berkovsky, 2010; Baltrunas, 2010; Amer-Yahia, 2009) has primarily targeted *persistent groups*, in which members are fixed and have interacted multiple times as a group. In contrast, *cold-start group recommendation* poses a greater challenge, since *ephemeral groups* typically lack prior interactions or shared histories. In such cases, balancing individual preferences with group dynamics to produce recommendations that satisfy all members is a highly non-trivial problem.

Recent group recommendation advances (Sajjadi Ghaemmaghami, 2021) include attention-based aggregation over persistent groups (AGREE) (Cao, 2018) and multi-view modeling for occasional groups (GAME) (He, 2020a). For *ephemeral* groups without joint history, GroupIM (Sankar, 2020) maximizes

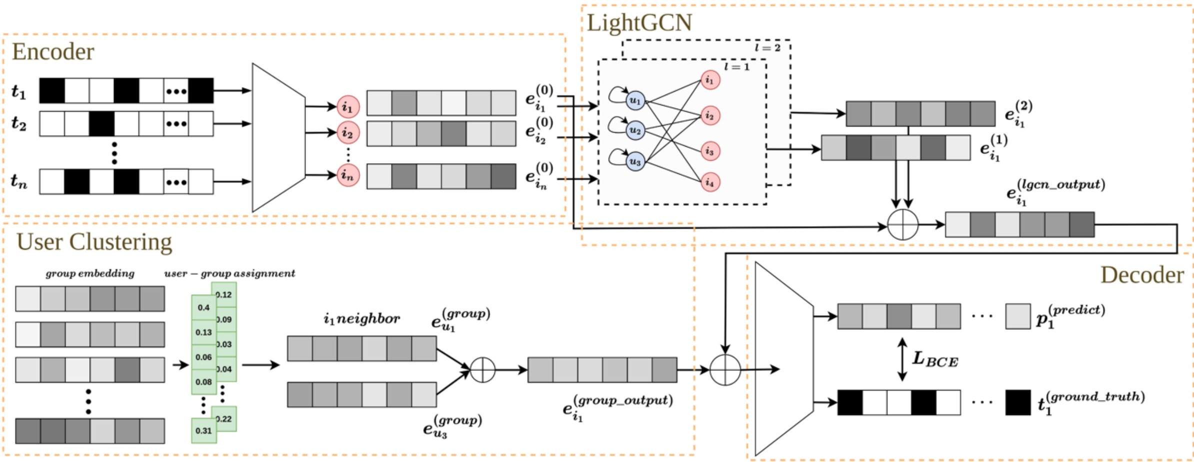

Figure 1: The COIN architecture illustrating training on item i_1 for user group $\{u_1, u_3\}$.

mutual information between group/user/item representations. These works generally lack interpretability and do not uncover the latent semantics behind group members' shared interests. Disentangled recommendation methods (Ma, 2019) are capable of learning factorized representations to capture latent semantics in user–item interaction data, but their focus remains on individual user behavior rather than on cold-start groups.

In this paper, we propose the **COmmon INtereset model (COIN)** to discover potential common interests in cold-start user groups. We leverage Graph Convolution Networks (GCNs) to capture high-order relations in user–item interactions and construct a *virtual item* for a cold-start group. This virtual item represents the most suitable recommendation for a given group, and by incorporating its tag attributes as auxiliary data (Liu, 2020), COIN can effectively reflect the group's potential common interests. The COIN model consists of four main components: (1) a **tag encoder** that transforms sparse item tag attributes into dense vectors; (2) **LightGCN** (He, 2020b), which captures high-order user–item interactions; (3) a **user clustering module** that models user–group–tag level preferences; and (4) a **tag decoder** that reconstructs tag-level semantics from the learned dense representations. Extensive experiments have been conducted on three real-world datasets, and the results demonstrate the feasibility and effectiveness of the proposed method.

2 The Proposed COIN Model

As shown in Fig. 1, the COIN model consists of four main components working together to uncover group-level common interests. First, the *Encoder* transforms sparse item tag attributes into dense embeddings, providing compact semantic representations for items. Next, *LightGCN* captures high-order relations in the user–item interaction graph, refining embeddings through graph propagation. Meanwhile, the *User Clustering* component models user–group–tag preferences by softly assigning users to latent groups and generating group-aware item embeddings. Finally, the Decoder combines the outputs from LightGCN and clustering, reconstructs tag semantics, and predicts the common interests of cold-start groups, optimized via a binary cross-entropy loss.

2.1 Problem Formulation

Let U denote the set of users, I the set of items, and T the set of tag attributes. Each item i is associated with a multi-hot vector t_i representing its tag attributes. We define the set of user–item interactions as $R^+ = \{(u, i) \mid u \in U, i \in I\}$. Given a *cold-start user group* S (a subset of users), the objective is to learn a function that predicts the top-k tag attributes representing the common interests of this cold-start group.

2.2 Tag Attribute Encoder

Each item tag attribute $t_i \in R^T$ is represented as a high-dimensional multi-hot vector. Since directly training with such sparse vectors is impractical, we employ a two-layer Multi-Layer Perceptron (MLP) as the encoder. The encoder projects each sparse tag vector into a dense embedding space, yielding an initial item embedding $e_i^{(0)} \in R^d$ for subsequent model training.

For users, we construct an embedding look-up table, where each column represents a user embedding $e_u^{(0)} \in R^d$. These user embeddings, together with the encoded item embeddings, serve as the foundation for later components of the COIN model.

The encoding process can be expressed as:

$$e_i^{(0)} = W_2 \cdot \mathrm{ReLU}(W_1 t_i),$$

$$E_u = [e_{u_1}^{(0)}, \ldots, e_{u_N}^{(0)}],$$

where W_1 and W_2 are trainable weight matrices, and E_u denotes the collection of user embeddings.

2.3 LightGCN

After obtaining the initial embeddings of users and items from the encoder, the next step is to capture their potential common interests. When multiple users interact with the same item, it indicates they may share latent preferences reflected in the item's tag attributes.

We leverage graph convolution network-based solutions, particularly LightGCN, which has proven highly effective for various recommendation tasks. The encoder's initial embeddings serve as the input. LightGCN propagates information across the user–item interaction graph:

$$e_u^{(l+1)} = \sum_{i \in N_u} \frac{1}{\sqrt{|N_u|}\sqrt{|N_i|}} e_i^{(l)},$$

$$e_i^{(l+1)} = \sum_{u \in N_i} \frac{1}{\sqrt{|N_u|}\sqrt{|N_i|}} e_u^{(l)},$$

where N_u denotes the set of items interacted with by user u, and N_i denotes the set of users interacting with item i.

However, high-order propagation may cause the problem of over-smoothing, where user embeddings lose their unique semantics and become dominated by item embeddings. To mitigate this issue, we introduce a residual connection for users, which preserves user-specific information:

$$e_u^{(l+1)} = e_u^{(l)} + \sum_{i \in N_u} \frac{1}{\sqrt{|N_u|}\sqrt{|N_i|}} e_i^{(l)},$$

$$e_i^{(l+1)} = \sum_{u \in N_i} \frac{1}{\sqrt{|N_u|}\sqrt{|N_i|}} e_u^{(l)}.$$

The discussion of the training–inference gap for items is deferred to Sections 2.6 and 2.7.

For efficiency in implementation and training, we rewrite the propagation rule in matrix form. Let $R \in R^{N \times M}$ denote the user–item interaction matrix, where $R_{ui}=1$ if user u has interacted with item i, and 0 otherwise. By adding user self-loops, the adjacency matrix is defined as:

$$A = \begin{pmatrix} I & R \\ R^\top & 0 \end{pmatrix},$$

with degree matrix D. Let $E^{(l)} \in R^{(N+M) \times d}$ be the embeddings at layer l. The propagation becomes:

$$E^{(l+1)} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}} E^{(l)}.$$

After L layers of propagation, we aggregate embeddings from all of the layers (including the encoder's initial embeddings) by averaging, to retain semantic information learned at each stage:

$$e_i^{(\mathrm{lgcn\_output})} = \frac{1}{L+1} \sum_{l=0}^{L} e_i^{(l)}.$$

This ensures that the final item embeddings preserve both high-order relational knowledge and the semantic features from earlier layers.

2.4 User Clustering

We assume the existence of $|G|$ latent user groups, each capturing abstract and complex group–tag preferences. A user may belong to multiple groups simultaneously, and thus can be represented as a combination of these group memberships. Since no external resources are available, we adopt a simple soft clustering approach that is directly learned from user embeddings.

Formally, let $S \in R^{N\times|G|}$ denote the user–group assignment matrix, where $S_{i,j}$ corresponds to the probability of user u_i belonging to group g_j. To obtain this, we apply a linear projection $W_{proj} \in R^{d\times|G|}$ to the user embeddings, followed by a softmax function to ensure each row forms a valid probability distribution:

$$S = \mathrm{softmax}(E_u W_{\mathrm{proj}}).$$

Next, we maintain a group embedding look-up table, where each column represents a latent embedding $e_g \in R^d$. Rather than learning direct group–tag preferences (which would be computationally prohibitive given the high dimensionality of tag attributes), we instead learn a latent vector for each group. Combining the user–group assignment matrix with the group embeddings, we derive user representations in the group space:

$$E_g = [e_{g_1}, \ldots, e_{g_{|G|}}],$$
$$E_{\mathrm{user\_group}} = SE_g.$$

Finally, an item embedding is represented by averaging over the group-based user embeddings of its neighboring users:

$$e_i^{(\mathrm{group\_output})} = \frac{1}{|N_i|} \sum_{u \in N_i} e_u^{(\mathrm{user\_group})}.$$

Through this design, the model captures group-level user preferences and mitigates the over-smoothing issue encountered in LightGCN for low-degree users.

2.5 Tag Attribute Decoder

To generate the final item representation, we apply a linear combination of the outputs from LightGCN and the user clustering module. Specifically, given hyperparameter α, the final item embedding is computed as:

$$e_i^{(\mathrm{final})} = \alpha \cdot e_i^{(\mathrm{lgcn\_output})} + (1-\alpha) \cdot e_i^{(\mathrm{group\_output})}.$$

This embedding is then passed through a decoder, which mirrors the structure of the encoder. The decoder transforms the dense item embedding back into the tag attribute space, thereby reconstructing semantic features. Finally, we apply a *sigmoid* activation to produce probabilities for each tag attribute:

$$p_i = \sigma\big(W_4 \cdot \mathrm{ReLU}(W_3 e_i^{(\mathrm{final})})\big),$$

where W_3 and W_4 are trainable weight matrices. These probabilities are compared against the ground-truth tag attributes using binary cross-entropy loss, ensuring that the learned item embeddings preserve interpretable semantics aligned with observed tag data.

2.6 Model Training

During training, the COIN model jointly learns (1) user embeddings, (2) the encoder for projecting item tag attributes into a latent semantic space, (3) group embeddings with user–group assignments to capture group-level preferences, and (4) the decoder for reconstructing item tag attributes to reveal common interests.

To simulate cold-start groups, we adopt a masking strategy in which certain items are randomly removed, along with all co-interacted items from the same subset of users, thereby forming a common interest set. For instance, if users u_1 and u_3 interacted with items i_4 and i_7, both items are removed to define their shared interest during testing.

The model then predicts tag attribute distributions for items, which are optimized using binary cross-entropy (BCE) loss:

$$\mathcal{L}_{BCE} = \sum_{k=0}^{|T|} -\Big(t_{ik}\log(p_{ik}) + (1-t_{ik})\log(1-p_{ik})\Big),$$

Table 1: Statistics of the datasets

	# User	# Item	# Density	# Tag Number	# Avg Tag per item	# Group Avg User
citeulike-a	5551	16,980	0.00217	46,390	14.09	5.43
citeulike-t	7947	25,975	0.00065	52,946	11.19	2.04
yelp	15844	19042	0.00140	889	5.01	2.85

where t_{ik} is the ground-truth value of the k-th tag attribute for item i, and p_{ik} is the predicted probability. This ensures that reconstructed attributes closely align with true tag labels, guaranteeing that the learned embeddings preserve sufficient semantic information for decoding back into the tag attribute space.

2.7 Model Inference

During inference, the goal is to predict the probability distribution over tag attributes that represent the common interests of a given cold-start user group. As in training, the encoder first transforms item tag attributes into embeddings, producing both item and user representations.

To handle cold-start groups, we introduce a *virtual item* into the user–item interaction graph. This virtual item is connected to all users in the input group, enabling propagation through both LightGCN and the user clustering module. Unlike training, where the graph structure remains fixed, the inference graph is dynamically constructed for each cold-start group. The virtual item's embedding thus encodes the aggregated preferences of the group.

Finally, the virtual item embedding is decoded back into the tag attribute space, yielding probabilities for each tag. The *top-k tag attributes* are selected as the predicted common interests of the cold-start user group.

3 Experiments

In this section, we compare the proposed COIN model against several baselines and provide an empirical analysis of each model component.

3.1 Experimental Settings

We evaluate the COIN model on three publicly available real-world datasets: Citeulike-a, Citeulike-t (Wang, 2013), and Yelp, with their statistics summarized in Table 1. The Citeulike-a and Citeulike-t datasets are collected from *CiteULike*, an online platform where users share and manage academic papers. Each paper includes metadata such as title, abstract, and tag attributes, and in our experiments we utilize the user–paper interactions along with paper tag attributes to predict group-level common interests. The Yelp dataset is a subset of Yelp's business data, containing user–business interactions, reviews, check-ins, and location tag attributes. For our task, we specifically use the user–location interactions and location tag attributes to infer common interests within user groups.

Following the training and inference procedures described in Sections 2.6 and 2.7, we randomly mask certain items for testing. Items co-interacted by the same user subset are grouped together, and their averaged tag attributes are treated as the ground-truth common interests. For evaluation, we adopt three top-k ranking metrics (Krichene, 2020): Recall@20, F1@20, and NDCG@20. The predicted tag attributes of the virtual item are ranked and compared against the ground-truth top-k tag attributes.

We compare the COIN model against several baselines. The **Intersection** method is a naïve approach that defines a group's common interest as the intersection of all tag attributes associated with items interacted by group members. The **Probability** method adopts a simple probabilistic strategy, where each user's tag preference is calculated by multiplying the probability of the user interacting with an item and the tag probability of that item, and the group's common interest is then derived by combining the tag preferences of all group members. **MAGREE** is a modified version of the AGREE model (Cao, 2018), in which a

Table 2: Performance comparison

	citeulike-a			citeulike-t			yelp		
	Recall	F1	NDCG	Recall	F1	NDCG	Recall	F1	NDCG
Intersection	0.0002	0.0004	0.0019	0.0001	0.0003	0.0048	0.0015	0.0014	0.0013
Probability	0.0724	0.0987	0.2030	0.0336	0.0550	0.1825	0.2825	0.2436	0.4022
UAP	0.0570	0.0897	0.0234	0.0322	0.0600	0.3510	0.2597	0.2262	0.3640
MAGREE	0.0628	0.0978	0.2526	0.0355	0.0646	**0.3822**	0.2911	0.2454	0.3813
AE+GCN	0.0125	0.0184	0.0401	0.0018	0.0033	0.0166	0.0379	0.0314	0.0283
AE+LightGCN	0.0971	0.1379	0.2896	0.0617	0.1028	0.3431	0.2990	0.2477	0.4127
COIN	**0.1097**	**0.1517**	**0.3104**	**0.0646**	**0.1077**	0.3644	**0.3540**	**0.2930**	**0.4735**
Improvement	12.97 %	10.00 %	7.18 %	4.70 %	4.76 %	-4.88 %	8.36%	6.33%	6.32%

UAP: User Attribute Prediction Model MAGREE: Modified AGREE

tag attribute prediction layer is added to better suit our task. The **User Attribute Prediction Model (UAP)** trains only user embeddings and the decoder; item embeddings are generated during training by averaging the embeddings of interacting users and decoded to reconstruct tag attributes, while during inference a virtual item embedding is formed by averaging the embeddings of the input user set and then decoded to predict tag attributes. **AE+LightGCN** employs an encoder for item embeddings together with a user embedding lookup table, and applies LightGCN with residual connections to capture high-order semantics in the user–item graph, with the resulting embeddings decoded into tag attributes. Finally, **AE+GCN** (Kipf, 2017) serves as a variant using graph convolution networks, where each layer includes a linear transformation and activation function after graph propagation; this distinguishes it from LightGCN but also introduces potential mismatches between training and inference due to structural differences in the graphs.

3.2 Results and Discussion

The results are shown in Table 2, where "Improvement" indicates the relative gain of COIN over the second-best one (underlined).

The **Intersection** method, though intuitive, performs poorly since it only considers tag overlaps from user–item interactions, ignoring latent semantics and failing to generalize. The **User Attribute Prediction Model (UAP)** underperforms as well, since it does not learn item embeddings directly from tag attributes and relies on averaging user embeddings, which limits its ability to capture high-order semantics. **MAGREE** performs better than Intersection and UAP due to its attention mechanism, but still falls short of COIN because it does not explicitly model high-order relations in the user–item graph.

AE+GCN performs even worse than UAP: while it leverages an encoder and high-order interactions, its reliance on transformations and activations creates mismatches between training and inference, leading to degraded performance. **AE+LightGCN** achieves stronger results by learning item embeddings through an encoder and exploiting high-order relations without transformation mismatches; however, its performance drops on low-degree interactions where embeddings are prone to over-smoothing. By contrast, the **COIN** model consistently outperforms all baselines. Through tag-aware encoding, residual-enhanced LightGCN propagation, soft clustering to address over-smoothing, and effective decoding into tag attributes, COIN captures both high-order relations and latent semantics, leading to more accurate identification of common interests within groups.

3.3 Ablation Study

To evaluate the contribution of each component in the COIN model, we conduct an ablation study on two key modules: the residual connection in LightGCN and user clustering. The results are shown in Table 3.

Table 3: Ablation study

	citeulike-a			citeulike-t			yelp		
	Recall	F1	NDCG	Recall	F1	NDCG	Recall	F1	NDCG
COIN	**0.1097**	**0.1517**	**0.3104**	**0.0646**	**0.1077**	**0.3644**	**0.3546**	**0.2900**	**0.4735**
w/o residual	0.0876	0.1263	0.2677	0.0585	0.0986	0.3495	0.3446	0.2881	0.4698
w/o clustering	0.1009	0.1428	0.2942	0.0635	0.1055	0.3498	0.3179	0.2588	0.4318

Table 4: Impact of clustering over low-degree users

	citeulike-a			citeulike-t			yelp		
	Recall	F1	NDCG	Recall	F1	NDCG	Recall	F1	NDCG
w/o clustering	0.0944	0.1017	0.2380	0.0709	0.0865	0.2724	0.3104	0.2107	0.4013
with clustering	0.1108	0.1145	0.2698	0.0751	0.0901	0.2829	0.3674	0.2530	0.4643
Improvement	17.37%	13.36%	12.58%	5.92%	4.16%	3.85%	18.36%	20.07%	15.69%

Table 5: Impact of clustering over high-degree users

	citeulike-a			citeulike-t			yelp		
	Recall	F1	NDCG	Recall	F1	NDCG	Recall	F1	NDCG
w/o clustering	0.0991	0.1214	0.3553	0.0366	0.0573	0.5322	0.3027	0.2357	0.4469
with clustering	0.1061	0.1304	0.3775	0.0372	0.0579	0.5606	0.3333	0.2613	0.4878
Improvement	7.06%	7.04%	6.27%	1.63%	1.04%	5.33%	10.10%	10.86%	9.15%

Table 6: Impact of layer number

	citeulike-a			citeulike-t			yelp		
	Recall	F1	NDCG	Recall	F1	NDCG	Recall	F1	NDCG
COIN-1	0.0492	0.0782	0.2156	0.0494	0.0048	0.0508	0.0507	0.0521	0.0814
COIN-2	**0.1097**	**0.1517**	**0.3104**	0.0646	0.1077	**0.3644**	**0.3540**	**0.2930**	**0.4735**
COIN-3	0.0984	0.1382	0.2777	0.0636	0.1064	0.3643	0.3240	0.2668	0.4396
COIN-4	0.0997	0.1399	0.2828	**0.0665**	**0.1082**	0.3545	0.3078	0.2634	0.4388

The residual connection improves performance by directly preserving user-specific information during propagation. Without it, user embeddings tend to collapse into overly similar representations dominated by neighboring items, leading to a loss of individual semantics. As illustrated in Table 3, removing the residual connection reduces Recall and NDCG across all datasets, confirming that retaining user individuality is crucial for accurate recommendation.

The user clustering module also plays an important role. By softly assigning users to multiple latent groups, clustering captures higher-level group semantics and alleviates sparsity in low-degree interactions. This effect is evident in Table 3, where removing clustering leads to a noticeable performance drop, particularly for datasets with many low-degree users (e.g., CiteULike-t). Clustering enables the model to infer preferences for sparse users by leveraging patterns shared with similar users, thereby mitigating cold-start and over-smoothing issues.

We also investigate the impact of user clustering on groups with different interaction degrees. To this end, users are divided into low-degree and high-degree categories based on whether their number of interactions falls below or above the median interaction count in the dataset. The results, presented in Tables 4 and 5, show that user clustering improves performance in both categories but provides a more substantial gain for low-degree users. This demonstrates that clustering is particularly effective in alleviating the over-smoothing problem in sparse interaction scenarios, as it allows low-degree users to leverage shared group-level semantics to compensate for limited individual interactions.

3.4 Impact of Layer Number

COIN benefits from light propagation, which enables it to capture high-order interactions within the user–item graph. To evaluate the effect of different propagation depths, we tested configurations with 1, 2, 3, and 4 layers, and the results are summarized in Table 6. The findings show that COIN-2 and COIN-3 achieve the strongest performance across most metrics. In contrast, COIN-1, which does not include propagation, performs significantly worse, indicating that modeling high-order interactions is essential for learning meaningful common interests. However, when the number of propagation layers becomes too large, as in COIN-4, the model suffers from over-smoothing, where embeddings lose their distinctiveness and fail to capture nuanced user–item semantics. This suggests that a moderate depth, specifically two or three layers, strikes the best balance between leveraging high-order relationships and preserving semantic richness in the embeddings.

4 Conclusions

In this paper, we propose the COIN model for discovering common interests in cold-start user groups. Unlike prior methods focused on persistent groups, COIN tackles the problem of cold-start group recommendation by leveraging item tag attributes and high-order semantics captured through LightGCN and user clustering. Experiments on Citeulike-a, Citeulike-t, and Yelp datasets demonstrate that COIN consistently outperforms baseline methods. Ablation and sensitivity analyses confirm that residual connections reduce over-smoothing, clustering enhances low-degree user modeling, and decoding provides interpretable tag-level explanations.

Future work could extend the model by incorporating temporal dynamics to capture evolving group preferences and integrating multimodal signals such as reviews or images for richer semantics.

References

Amer-Yahia, S., Roy, S. B., Chawlat, A., Das, G., & Yu, C. 2009. Group recommendation: Semantics and efficiency. In *Proc. of the VLDB Endowment*.

Baltrunas, L., Makcinskas, T., & Ricci, F. 2010. Group recommendations with rank aggregation and collaborative filtering. In *Proc. of ACM Conference on Recommender Systems* (pp. 119-126).

Berkovsky, S., & Freyne, J. 2010. Group-based recipe recommendations: analysis of data aggregation strategies. In *Proc. of ACM Conference on Recommender Systems* (pp. 111-118).

Cao, D., He, X., Miao, L., An, Y., Yang, C., & Hong, R. 2018. Attentive group recommendation. In *Proc. of ACM SIGIR Conference* (pp. 645-654).

He, Z., Chow, C. Y., & Zhang, J. D. 2020a. GAME: Learning graphical and attentive multi-view embeddings for occasional group recommendation. In *Proc. of ACM SIGIR Conference* (pp. 649-658).

He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., & Wang, M. 2020b. LightGCN: Simplifying and powering graph convolution network for recommendation. In *Proc. of ACM SIGIR Conference* (pp. 639-648).

Kipf, T. N. 2017. Semi-supervised classification with graph convolutional networks. In *Proc. of* International Conference on Learning Representations.

Krichene, W., & Rendle, S. 2020. On sampled metrics for item recommendation. In *Proc. of ACM SIGKDD Conference* (pp. 1748-1757).

Liu, Q., Xie, R., Chen, L., Liu, S., Tu, K., Cui, P., ... & Lin, L. 2020. Graph neural network for tag ranking in tag-enhanced video recommendation. In *Proc. of ACM CIKM Conference* (pp. 2613-2620).

Ma, J., Zhou, C., Cui, P., Yang, H., & Zhu, W. 2019. Learning disentangled representations for recommendation. *Advances in Neural Information Processing Systems*, *32*.

Sajjadi Ghaemmaghami, S., & Salehi-Abari, A. 2021. DeepGroup: Group recommendation with implicit feedback. In *Proc. of ACM CIKM Conference* (pp. 3408-3412).

Sankar, A., Wu, Y., Wu, Y., Zhang, W., Yang, H., & Sundaram, H. 2020. GroupIM: A mutual information maximization framework for neural group recommendation. In *Proc. of ACM SIGIR Conference* (pp. 1279-1288).

Wang, H., Chen, B., & Li, W. J. 2013. Collaborative topic regression with social regularization for tag recommendation. In Proc. of *IJCAI* Conference (pp. 2719-2725).