



# GIFT4Rec: An Effective Side Information Fusion Technique Apply to Graph Neural Network for Cold-Start Recommendation

Tran-Ngoc-Linh Nguyen<sup>1,2</sup>, Chi-Dung Vu<sup>1,2(✉)</sup>, Hoang-Ngan Le<sup>1</sup>, Anh-Dung Hoang<sup>1</sup>, Xuan Hieu Phan<sup>2</sup>, Quang Thuy Ha<sup>2</sup>, Hoang Quynh Le<sup>2</sup>, and Mai-Vu Tran<sup>2</sup>

<sup>1</sup> Data Analytics Center - Viettel Telecom, Viettel Group, Hanoi, Vietnam  
{linhntn3,dungvc2,nganlh,dungha7}@viettel.com.vn

<sup>2</sup> University of Engineering and Technology - Vietnam National University, Hanoi, Vietnam  
{hieupx,thuyhq,lhquynh,vutm}@vnu.edu.vn

**Abstract.** Recommendation systems are highly interested in technology companies nowadays. The businesses are constantly growing users and products, causing the number of users and items to continuously increase over time, to very large numbers, this leads to the cold-start problem. Recommending purely cold-start users is a long-standing and fundamental challenge in the recommendation systems where systems are unable to recommend relevant items to the users due to unavailability of adequate information about them. To solve this problem, extensive studies have been carried out using the side information techniques (user information, item information, ...). However, we argue that this work will affect the user/product group that had a lot of interaction, using this side information can reduce the performance of the model when just focusing on learning based on the side information. In this paper, we propose a combination of global and local side **I**nformation **F**usion **T**echniques based on attention algorithm applied to **G**raph neural network-based models for cold-start users recommendation, and we call this architecture **GIFT4Rec**.

**Keywords:** recommendation system · cold-start problem · meta learning · side information fusion · graph neural network

## 1 Introduction

Personalization is the topic of investment with high returns in recent years. Two typical collaborative filtering (CF) algorithms for the recommendation problem are matrix factorization [1] and two-head DNNs [4, 5]. While recent studies focus on the accuracy of the recommended system in the lab and achieve positive results such as BiVAE [2], VASP [3], ... We find these methods facing difficulties in deploying on the product environment because as the number of users increases, the model cannot recommend for new users and new items.

Cold-start recommendation is a challenge in recommendation systems where the system needs to make recommendations for new users or new items that have little to no historical interaction data. In other words, it refers to the situation where a recommendation system is presented with a user or item that it has never encountered before. The cold-start problem can occur in two scenarios: new user cold-start, for example, when a new user signs up for a service, there is no historical data available for the system to use to make personalized recommendations for them and new item cold-start, when a new item is added to the system, there is little or no data available about the item's characteristics and how users might interact with it. To address the cold-start problem, recommendation systems can use various techniques such as transfer learning, cross-domain, information fusion. Transfer learning approach involves leveraging knowledge learned from other domains or tasks to improve recommendation accuracy in cold-start scenarios. Transfer learning can be effective in situations where there is limited data available for the target domain or task.

Side information fusion is a technique used in cold-start recommendation systems to address the problem of limited data by incorporating additional information about users and items. The technique involves using side information or auxiliary data, such as demographic information, social network information, or item attributes, to improve the accuracy of recommendations for new users or items. The side information fusion technique works by combining the user-item interaction data with the side information to build a more comprehensive user-item model. This combined model can then be used to generate recommendations for new users or items by leveraging the information in the side information. For example, in a movie recommendation system, side information such as demographics, movie genres, directors, and actors can be used to improve recommendations for new movies or users. By incorporating this information into the recommendation model, the system can make more accurate predictions about which movies a new user might like based on their preferences for specific genres or actors. The side information fusion technique can be applied using various machine learning methods such as matrix factorization, or graph-based approaches. It is particularly useful for cold-start scenarios, where the system lacks sufficient data to make accurate recommendations, and can improve the overall performance of the recommendation system.

Some recent studies [14, 15] have shown that the side information fusion is a useful technique for cold-start recommendation systems, there are some disadvantages and limitations to consider:

First, side information can introduce bias into the recommendation model if the side information is biased or incomplete. For example, if the side information is based on user demographics, it may lead to recommendations that are biased towards a particular group or stereotype.

Second, incorporating side information can increase the risk of overfitting, where the model becomes too closely tailored to the training data and performs poorly on new, unseen data. This can happen if the side information is too closely aligned with the training data or if the model is too complex.

In this paper, we try to solve these two non-optimal points, and propose a new architecture that can effectively be implemented in the production environment. Our main contributions include:

- We propose a new technique using side information to learn cold-start users interest and recommend more suitable items for them.
- We propose a new attention-based technique that can control and estimate the priority of each user’s information used from different sources to capture their interest for an unbiased and fairness recommendation system.
- We propose a meta learning techniques that can decrease the risk of overfitting, model can learn generally on the unseen data.

## 2 Related Works

### 2.1 Cold-Start User Problems

The main issue of the cold start problem is non-availability of information required for making recommendations. In such cases, the only There are two popular methods to address this problem:

First is Cross-Domain Recommendation technique, which uses users behavior of source domain to predict their interests at target domain. Ye Bi et al. [9] and Cheng Zhao et al. [8] both map users behavior embedding of source domain to target domain via MLP layers. However, there aren’t always more than one domain sharing the same users in reality.

Second is the side information fusion method. This method is more stable than the first one due to the fact that the side information always exists. DropoutNet [12] aims to maintain recommendation accuracy on non-cold start users while improving model performance on cold-start users, by combining all side information with users interactions to learn to reconstruct output from model just using users interaction. Beside, this model also randomly choosing some data just using side information of users or items to learn to reconstruct, that increases the affection of side information to model output which is very suitable for cold-start recommendation. However, this technique is not designed to control and estimate the affection of side information to each user, which may harm to model performance. To address this limitation of DropoutNet, we propose a new technique called AttentionDropoutNet improving model performance in all users type as active users, warm-start users and cold-start users simultaneously.

### 2.2 Meta-learning

Meta-learning, also called learning-to-learn, aims to train a model that can rapidly adapt to a new task which is not used during the training with a few examples Meta-learning can be classified into three types: metric-based, memory-based, and optimization-based meta-learning. Previous research as Manqing

Dong et al. [10] or Ye Bi et al. [11] are about applying optimization-based meta-learning to recommendation system that provide a more quickly and efficiently new data learning method for better cold-start recommendation. Inspired by that, we create a new metric-based meta-learning method for an unbiased and fairness recommendation system [13] and the rapidly changing of users preference better learning.

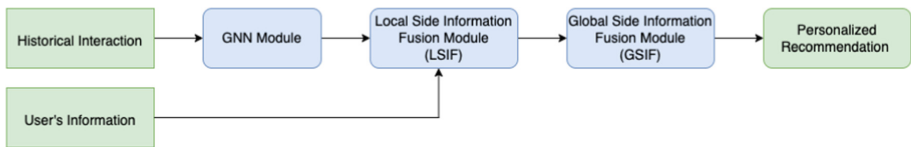
### 2.3 Graph Neural Network

Graph Neural Network [6] also known as GNN is a deep learning model applied to graph structure for many different problems. GNN learns the higher representation by aggregating their neighbor nodes information and learn them jointly with downstream task as node classification, link prediction or graph classification ...

In recent years, there are many methods applying GNN to recommendation systems by treating interactions between users and items like graph structure. In this graph structure, users and items are defined as nodes, each interaction between them is defined as an edge. GNN aims to learn the relation between each node via available links displaying in graph structure to predict possible relation of each two nodes not displaying of graph for different recommendation tasks. Rex Ying et al. [18] proposed a combination of GCN and hard-negative sampling method for similar items recommendation. Besides, Xiang Wang et al. [7] are about each node neighbors weight learning via their relation that is suitable for recommendation system.

## 3 Proposal Model

In this section, we will first give an overview about the proposed model, then detail each model component



**Fig. 1.** Overall GIFT4Rec architecture

The architecture of the proposal model is shown in Fig. 1. The model consists of two components: Graph neural network (GNN) module, our global and local side information fusion module. The GNN module learns and extracts the characteristics of the user's behavior and the item's representation. The global and local side information fusion module builds a way to integrate side information into the user's embedding vector, which is the output of GNN module.

Given the items catalog  $V = \{v_1, v_2, \dots, v_p\}$  with  $p$  items. For a sample user  $u_i$ ,  $i \in \{1, 2, \dots, N\}$  with side information vector  $X_{info_i}$ , we have a set of interacted items  $S_i = \{s_{i1}, s_{i2}, s_{i3}, \dots, s_{iq}; s_{ij} \in T, q \leq p\}$ .

The GNN module is shown in Fig. 2. A graph is represented as  $G = (U, V)$ , which is defined as  $\{(u_i, s_{ij}, v_j) | u_i \in U, v_j \in V\}$ , where  $U$  and  $V$  separately denote the user and item sets, and a link  $s_{ij} = 1$  indicates that there is an observed interaction between user  $u_i$  and item  $v_j$ , otherwise  $v_{ij} = 0$ . The neighborhood of a node is denoted as  $N(\cdot)$ . Given the graph data, the main idea of GNN is to iteratively aggregate feature information from neighbors and integrate the aggregated information with the current central node representation during the propagation process [19, 20]. From the perspective of network architecture, GNN stacks multiple propagation layers, which consist of the aggregation and update operations. The formulation of propagation is

$$\text{Aggregation: } n^{(\ell)} = \text{Aggregator}_{\ell}(\{h_u^{\ell}, \forall u \in N(\cdot)\})$$

$$\text{Update: } h^{(\ell+1)} = \text{Update}_{\ell}(h^{(\ell)}, n^{(\ell)})$$

Where  $h_{u_i}^{(\ell)}$  denotes the representation of user  $u_i$  and  $h_{v_j}^{(\ell)}$  denotes the representation of item  $v_j$  at  $\ell^{th}$  layer, and  $\text{Aggregator}_{\ell}$  and  $\text{Update}_{\ell}$  represent the function of aggregation operation and update operation at  $\ell^{th}$  layer, respectively. In the aggregation step, existing works either treat each neighbor equally with the mean-pooling operation [21, 22], or differentiate the importance of neighbors with the attention mechanism [23]. In the update step, the representation of the central node and the aggregated neighborhood will be integrated into the updated representation of the central node. After training, the GNN model  $G$  will perform interaction embedding to build a vector  $X_{u_i} \in R^{1 \times D}$  - the behaviors embedding of user  $i$  and a vector  $X_{i_j}$  - the representation of item  $v_{i_j}$ :

$$X_{u_i}, X_{i_j} \leftarrow G(s_i)$$

The combination of  $X_{u_i}$  and  $X_{info_{u_i}}$  via our Weight Generated module in Fig. 3 for the last representation of user  $i$  are defined as  $X_{final_{u_i}}$ . Then the final score between  $u_i$  and  $i_j$  is computed as:

$$y_{u_i, i_j} = \text{softmax}(X_{final_{u_i}} \cdot X_{i_j})$$

We feed the final score to our Cross Entropy loss function defined as  $L_{CF}$ , which is computed as:

$$L_{CF} = \sum_{u_i, i_j, i_{jneg}} [\log y_{u_i, i_j} + \log(1 - y_{u_i, i_{jneg}})]$$

After learning the relation of each user and item having interactions used via  $L_{CF}$ , we use a new technique that make our model learning user representations more efficiently called Global Side Information Fusion also know as GSIF. In GSIF module,  $X_{final_{u_i}}$  and  $X_i$  are generated from the GNN module. Then all parameters are all frozen except the Weight Generated module in local side information fusion module and global side information module. Finally,  $a_u$  are generated from local side information fusion module then feed to global side information fusion module along with  $X_{final_{u_i}}$  and  $X_i$

### 3.1 General Side Information Module

We propose two side information techniques that support each other by observing each user with from different angles. Those methods aims to control and estimate the impact of each information to each user to combine them efficiently for fairness and unbiased recommendation that focus not only in any source of information, which can't always contain information related to user interest. The first one forces Weight Generated module to learn via optimize  $L_{CF}$ , the remaining technique provide this module a general knowledge via indirectly observing unseen interactions. These two modules shared parameters that generate weights for each user side information and behaviors called Weight Generated.

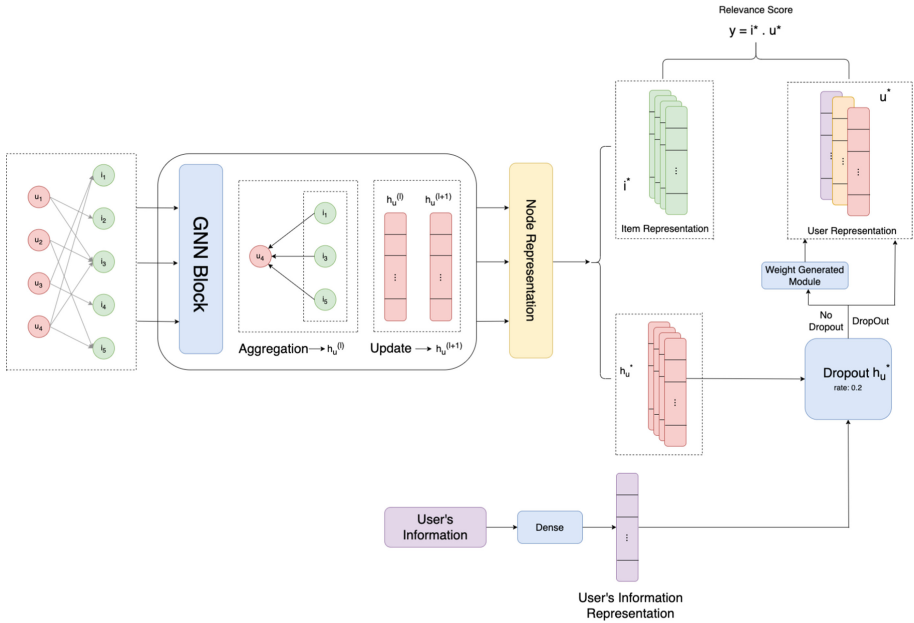
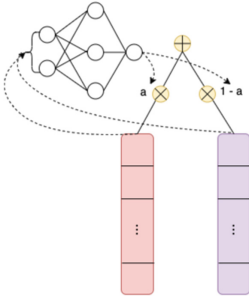


Fig. 2. Local Side Information Fusion module architecture

**Local Side Information Fusion Module.** We proposed a new technique called Attention DropoutNet also known as ADN that combining the technique used in [12] with out Weigh Generated module controlling side information and behavior of each users to better learning. Our module concatenate  $X_{u_i}$  and  $X_{info_{u_i}}$  via the last dim as the input of module called  $X_{concat_{u_i}}$

$$X_{concat_{u_i}} = concat([X_{u_i}, X_{info_{u_i}}])$$

We apply the MLP model to our Weight Generated module. We feed  $X_{concat_{u_i}}$  to Weight Generated module using a Sigmoid activation function in the last layer to get  $a_{u_i}$



**Fig. 3.** Weight Generated Module

The last representation of user  $i$ :

$$X_{final_{u_i}} = a_{u_i} \cdot X_{u_i} + (1 - a_{u_i}) \cdot X_{info_{u_i}}$$

That is how we estimate the impact of each information to user  $i$  and combine them to control the representation. Beside that, we use a technique that sample a random value from a uniform distribution over  $[0, 1)$  for each data when training. If that value less than the limit we set, the last representation would just be computed as side information embedding to force our model learning to use more side information of each user to predict their interest:

$$X_{final_{u_i}} = X_{info_{u_i}}.$$

During inference, the cold-start users behavior embedding would be computed as mean of all warm-start users and active users embedding just to recommend them the popular items that many users have interests in to our model knowledge, then combine with side information embedding for final representations:

$$X_{u_i} = \frac{1}{N_u - N_{U_{cold}}} \sum_{u_j \notin U_{cold}} X_{u_i}$$

**Global Side Information Module.** We proposed a new metric-based meta learning method observing the model performance computed by our metrics at two case:

- We define  $y_{behavior_{u_i,i}}$  as the list of the probability user  $i$  having interest of each item if we just using behavior of user  $i$  to model:

$$y_{behavior_{u_i,i}} = [y_{behavior_{u_i,i_1}}, y_{behavior_{u_i,i_2}}, \dots, y_{behavior_{u_i,i_{n_I}}}]$$

$$y_{behavior_{u_i,i_j}} = X_{u_i} \cdot X_{i_j}$$

- We use  $y_{behavior_{u_i,i}}$  to calculate model performance each metrics and then average them.
- Similar to the case if we just using side information of user  $i$  to model

We choose to use the validation set to test our model performance at two case above that help model Weight Generated module indirectly learning more objective knowledge from each user unseen interaction

We define  $label_{u_i} = 0$  if the model performance at case one is better. If not, then  $label_{u_i} = 1$

We encourage our Weight Generated module to learn more objectively and globally by optimizing a loss function called  $L_{global}$  defined as:

$$L_{global} = - \sum_i [(1 - label_{u_i}) * \log(1 - a_{u_i}) + label_{u_i} * \log(a_{u_i})]$$

$L_{global}$  would be training separately with  $L_{CF}$  in each epoch.

## 4 Experiments

### 4.1 Experiment Setting

**Dataset.** We use Movielen 1M (ML1M) [16], a relatively large and popular data set with the demographic of each user, item ratings and user’s interaction in the research field to test our proposed architecture performance. In addition, we use the Douban Dataset [17] to examine the effectiveness of side information fusion techniques (Table 1).

**Table 1.** Dataset Information

Dataset	Users	Items	Ratings
ML1M [16]	6040	3955	715477
Douban [17]	2442	8349	1053069

We split users into three set:

- Top 80% users having highest number of interactions will be choose as active users set
- Top 10% users having lowest number of interactions will be choose as cold-start users set that their interactions we don’t use for training and the first item each user interacts are used during testing.
- The remain users will be choose as warm-start users set

To evaluate model performance efficiently, for each user in the active users set, we hold out the last item for the testing set for active users, treat one random item before the last item as the validation set, and use the remaining items for the training set. For each user in the warm-start users set, we hold out the last item as a testing set for warm-start users, we combine the remaining interactions with the training set to create our graph structure. For each user in the cold-start users set, we just hold out the first item as a testing set for cold-start users.

**Baseline Methods.** To verify the effectiveness of our method, we compare it with the following representative baselines:

- GAT: a model only using graph neural network module from KGAT [7] paper along with using mean of all non cold-start users embedding for each cold-start users during testing
- GAT + DropoutNet: a model only using combination of graph neural network module from KGAT paper and DropoutNet technique
- GIFT4Rec (w/o Local): our proposed model without updating Weight Generated Module parameters via optimizing  $L_{CF}$
- GIFT4Rec (w/o Global): our proposed model without updating Weight Generated Module parameters via optimizing  $L_{Global}$
- GIFT4Rec: our proposed model



**Metrics.** We define  $A_i$  as top  $k$  highest ranking items generated by model for user  $i$ ,  $B_i$  as the real items set that user  $i$  interacted,  $N$  as the number of users.

Recall@k:

$$\frac{\sum \frac{|A_i \cap B_i|}{|B_i|}}{N}$$

We define overall score as mean of three sets scores. This metrics could evaluate model performance more fairly than just calculating score of combination of three sets which each of them always has different number of users that the more numbers of users one set has, the more its affection to overall score. In our experiment, we choose  $k$  as 50.

## 4.2 Experiment Result

**Table 2.** Benchmark

Dataset name	Method	Active users	Warm-start user	Cold-start user	Overall
ML1M	GAT [7]	19.64	0.066	24.52	14.94
	GAT [7] + DropoutNet	<b>20.24</b>	<b>0.5</b>	24.78	15.17
	GIFT4Rec (w/o Global)	<b>14.76</b>	<b>17.22</b>	<b>24.83</b>	<b>18.94</b>
	GIFT4Rec (w/o Local)	14.22	17.05	24.81	18.7
	GIFT4Rec	14.66	<b>18.71</b>	24.55	<b>19.3</b>
Douban	GAT	2.3	<b>2.95</b>	5.33	3.53
	GAT + DropoutNet	5.44	0.07	4.07	6.17
	GIFT4Rec (w/o Global)	<b>10.5</b>	1.48	7.52	<b>6.5</b>
	GIFT4Rec (w/o Local)	8.94	<b>2.58</b>	<b>7.94</b>	6.49
	GIFT4Rec	<b>10.18</b>	2.21	<b>8.36</b>	<b>6.92</b>

Our experiments results on ML1M dataset show GAT + DropoutNet model having the good performance at cold-start users set and active users simultaneously, which proves the efficient of DropoutNet. But this model has a very bad scores at warm-start users, which is nearly zero. Besides, this model also perform worse at all sets of Douban dataset than ours. That proves the accurate of our insight into the bad affection to model performance caused by uncontrollable side information learning of DropoutNet (Table 2).

GAT performs very good at active users set but worst at almost all of the remains methods on ML1M dataset, that is considered biased. In additions, this model performance at almost all sets of Douban dataset are the worst, compared to other models in experiment that prove the existence of large information about each user interest hidden inside their side information.

Our model without Global module gets a very good result at active users set of each dataset. The lower results of our model proposed and itself without Local module can be explained with the difference of distribution between two tasks we learning which one task is directly observing via each user that is also in active users set. That 's also an open challenge for meta-learning method.

Our model achieves the best result at cold-start users set of Douban dataset as soon as warm-start users set of ML1M dataset. Moreover, it also gets the second best result at active-users set of Douban dataset. But most of all, based on the most important metrics, our model outperforms the remains method on both datasets that is clearly the most fairness and unbiased recommendation system.

## 5 Conclusion

In this paper, we applied the attention-based side information fusion technique to cold-start users problem resolving and an unbiased and fairness recommendation system. Experiment results on two popular datasets show that our model outperforms the remains method which are variants of our model ore based on many popular algorithms for recommendation systems in recent years.

In future, we will upgrade our model to apply to cold-start items problem resolving. Another directions for future work would be research about how to combine  $L_{CF}$  and  $L_{Global}$  for less time consuming and efficient knowledge transfer between local and global modules to resolve open challenge that we describe in experiment result section.

## Appendix B

We defines  $\Theta$  and  $\Theta_{WG}$  as our model parameters and the parameters of Weight Generated module. In addition,  $scores_{u_B}$  and  $scores_{u_Binfo}$  denote users  $u_B$  (batch of U) model performance in validation set if just using their behavior embedding or their side information. Here is a pseudo code for our proposed model learning algorithm.

---

### Algorithm 1. GIFT4Rec

---

Random initialize  $\Theta$  and  $\Theta_{WG}$

while not converged:

  for  $u_B$  in  $U$ :

    update  $\Theta$  via optimizing  $L_{CF}$  with each user in  $u_B$

$X_{final_{u_B}} \leftarrow X_{u_B}$

    calculate  $scores_{u_B}$

$X_{final_{u_B}} \leftarrow X_{info_{u_B}}$

    calculate  $scores_{u_Binfo}$

    compare  $scores_{u_B}$  and  $scores_{u_Binfo}$  to get output labels for optimizing  $L_{Global}$

    update  $\Theta_{WG}$  via optimizing  $L_{global}$  with each user both in  $u_B$  and validation set

users

return  $\Theta$ ,  $\Theta_{WG}$

---

## References

1. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. *Computer* **42**(8), 30–37 (2009). <https://doi.org/10.1109/MC.2009.263>
2. Truong, Q.-T., Salah, A., Lauw, H.: Bilateral variational autoencoder for collaborative filtering, pp. 292–300 (2021). <https://doi.org/10.1145/3437963.3441759>
3. Vančura, V., Kordík, P.: Deep variational autoencoder with shallow parallel path for top-N recommendation (VASP). In: Farkaš, I., Masulli, P., Otte, S., Wermter, S. (eds.) *ICANN 2021*. LNCS, vol. 12895, pp. 138–149. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-86383-8\\_11](https://doi.org/10.1007/978-3-030-86383-8_11)
4. Krichene, W., et al.: Efficient training on very large corpora via Gramian estimation. In: *ICLR 2019* (2019)
5. Mehrotra, R., Lalmas, M., Kenney, D., Lim-Meng, T., Hashemian, G.: Jointly leveraging intent and interaction signals to predict user satisfaction with slate recommendations. In: *WWW 2019* (2019)
6. Wu, S., Sun, F., Zhang, W., Xie, X., Cui, B.: Graph neural networks in recommender systems: a survey, 2 April 2022
7. Wang, X., He, X., Cao, Y., Liu, M., Chua, T.-S.: KGAT: knowledge graph attention network for recommendation, 8 June 2019
8. Zhao, C., Li, C., Xiao, R., Deng, H., Sun, A.: CATN: cross-domain recommendation for cold-start users via aspect transfer network, 23 May 2020
9. Bi, Y., Song, L., Yao, M., Wu, Z., Wang, J., Xiao, J.: A heterogeneous information network based cross domain insurance recommendation system for cold start users, 30 July 2020
10. Bi, Y., Song, L., Yao, M., Wu, Z., Wang, J., Xiao, J.: MeLU: meta-learned user preference estimator for cold-start recommendation, 31 July 2019
11. Dong, M., Yuan, F., Yao, L., Xu, X., Zhu, L.: MAMO: memory-augmented meta-optimization for cold-start recommendation, 7 July 2020
12. Volkovs, M., Yu, G., Poutanen, T.: DropoutNet: addressing cold start in recommender systems. In: *NIPS* (2017)
13. Li, Y., et al.: Fairness in recommendation: a survey, 1 June 2022
14. Xie, Y., et al.: KoMen: domain knowledge guided interaction recommendation for emerging scenarios. In: *ACM SIGWEB* (2022)
15. Zhu, Z., Sefati, S., Saadatpanah, P., Caverlee, J.: Recommendation for new users and new items via randomized training and mixture-of-experts transformation. In: *SIGIR* (2020)
16. Harper, F.M., Konstan, J.A.: The MovieLens datasets: history and context. *ACM Trans. Interact. Intell. Syst. (TiiS)* **5** (2015). Article 19, 19 pages. <https://doi.org/10.1145/2827872>
17. Zhu, F., Wang, Y., Chen, C., Liu, G., Zheng, X.: A graphical and attentional framework for dual-target cross-domain recommendation. In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pp. 3001–3008 (2020)
18. Ying, R., He, R., Chen, K., Eksombatchai, P.: Graph convolutional neural networks for web-scale recommender systems (2018)
19. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S.: A comprehensive survey on graph neural networks. *TNNLS* **32**(1), 4–24 (2020)
20. Zhou, J., et al.: Graph neural networks: a review of methods and applications. *AI Open* **1**(2020), 57–81 (2020)

21. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs. In: NeurIPS, pp. 1025–1035 (2017)
22. Li, Y., Tarlow, D., Brockschmidt, M., Zemel, R.: Gated graph sequence neural networks. In: ICLR (2016)
23. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. In: ICLR (2017)