









Interactive Selection Recommendation Based on the Multi-head Attention Graph Neural Network

Shuxi Zhang^(✉), Jianxia Chen, Meihan Yao, Xinyun Wu, Yvfan Ge,
and Shu Li

School of Computer Science, Hubei University of Technology, Wuhan, China
1375018832@qq.com

Abstract. The click-through rate prediction of users is a critical task in the recommendation system. As a powerful machine learning method, graph neural networks have been favored by scholars to solve the task recently. However, most graph neural network-based click-through rate prediction models ignore the effectiveness of feature interaction and generally model all feature combinations, even if some are meaningless. Therefore, this paper proposes a Multi-head attention Graph Neural Network with Interactive Selection, named MGNN-IS in short, to capture the complex feature interactions via graph structures. In particular, there are three sub-graphs to be constructed to capture internal information of users and items respectively, and interactive information between users and items, namely the user internal graph, item internal graph, and user-item interaction graph correspondingly. Moreover, the proposed model designs a multi-head attention propagation module for the aggregation with an interactive selection strategy. This module can select the constructed graph and increase diversity with multiple heads to achieve the high-order interaction from the multiple layers. Finally, the proposed model fuses the features, and predicts. Experiments on three public datasets demonstrate that the proposed model outperformed other advanced models.

Keywords: Click-through Rate · Feature Interaction · Internal Graph · Interaction Graph · Multi-head Attention

1 Introduction

The recommendation system aims to cope with the information [16] overload that users may face. Usually, the click behavior has been regarded as a behavior that expresses the users preferences, thus the click-through rate (CTR) prediction is a crucial task of the recommendation system. Traditional recommendation

This work is supported by National Natural Science Foundation of China (Grant No.61902116).

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2024
B. Luo et al. (Eds.): ICONIP 2023, LNCS 14449, pp. 447–458, 2024.
https://doi.org/10.1007/978-981-99-8067-3_33

methods mainly include the content-based and collaborative filtering (CF)-based approaches [7, 17]. However, there are still some limits of them because of the sparse interaction between users and items [18].

As an effective model to capture the information for the graph data, graph neural networks (GNNs) have achieved a state-of-the-art performance in various tasks such as the semantic segmentation [10], machine translation [1], and recommendation systems [6, 17]. In particular, GNNs show its great potential in modeling the high-order feature interaction to predict the CTR as well. For example, Fi-GNN [9] utilizes a complete graph to interact with each pair of features. However, there are not beneficial for all feature interactions in a complete graph.

Inspired by the Fi-GNN model, the proposed model constructs a feature responding to a graph node and enables different features to interact each other via edges. However, since not all edge interactions are beneficial, Fi-GNN is not a good choice for modeling the interactions. To overcome this limitation, this paper not only enriches the graph construction with the attribute information but also filters out the helpful feature interactions via a special selection step for the interactions.

CF models are good at obtaining more detailed collaborative information to reveal the similarity between attributes via the feature embedding [13]. Usually, if considering the interactions between different features, the feature embedding can utilize more useful information to improve the performance of the prediction [11, 12]. Recently, the features interactions are proposed in an interpretable way with attention mechanisms. For example, HoAFM [15] updates the feature representations by aggregating the representations of the co-occurring features. AutoInt [12] first attempts to utilize a multi-head self-attention mechanism to explicitly model feature interactions. GMCF [14] designs a cross-interaction module before the feature interaction on both users and items sides.

Inspired by the GMCF [14], the proposed model interacts within the user side and the item side at the same time. Different from the GMCF, however, the proposed model revises the propagation aggregation of the GNN structure with a multi-head attention, and added multi-layer structure. Thus, the proposed model can learn the higher-order interaction information.

From the above analysis, this paper proposes a Multi-head attention Graph Neural Network with Interactive Selection, named MGNN-IS in short. In particular the proposed MGNN-IS model explicitly aggregates the internal-interactions and the cross-interactions in various ways in the graph structure. In addition, it also proposes a novel multi-layer network, in which each layer generates the higher-order interaction on the existing basis. The main contributions of the paper are described as follows:

- (1) Designs a feature interaction model with the multi-head attention mechanism that incorporated the idea of the residual connection.
- (2) Calculates an attention score via the feature interaction and the multi-layer perceptron (MLP), in order to select the edges with the highest score in the graph.
- (3) Demonstrates the effectiveness and interpretability of proposed model through the experimental results.

2 MGNN_IS Model

The MGNN_IS model mainly consists of four sub-modules: graph construction & feature embedding, interaction selection & propagation aggregation, feature fusion, and prediction. The model architecture is shown in Fig. 1. Symbol definition and each sub-module are described as follows.

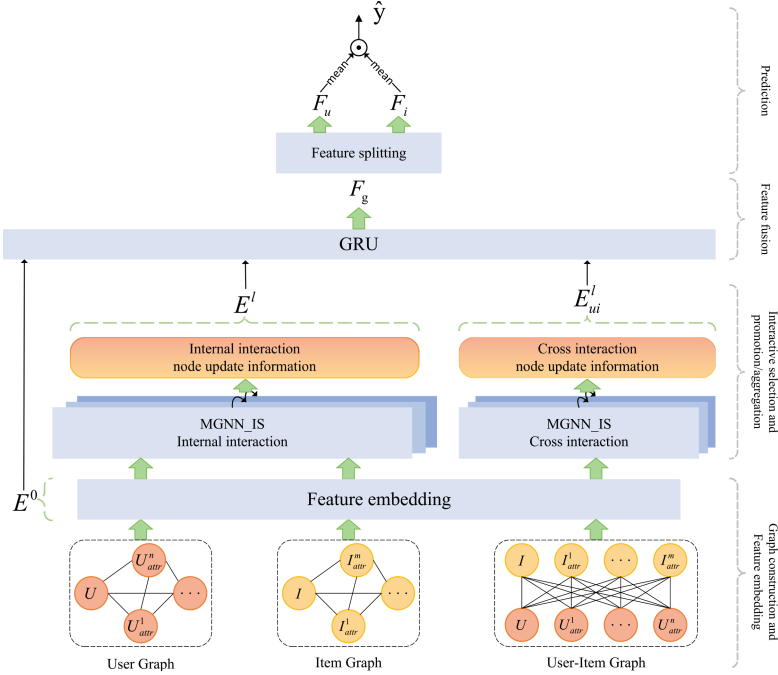


Fig. 1. Overall architecture of MGNN_IS model

2.1 Symbol Definition

The set of users and their attributes is defined as $\mathcal{U} = \{u_1, u_2, \dots, u_a, u_{attr_1}, u_{attr_2}, \dots, u_{attr_b}\}$, the set of items and their attributes is presented as $\mathcal{I} = \{i_1, i_2, \dots, i_c, i_{attr_1}, i_{attr_2}, \dots, i_{attr_d}\}$, and the set of all nodes is $\mathcal{V} = \mathcal{U} \cup \mathcal{I}$, and \mathcal{E} is presented as the set of relations generated by users, items and their attributes.

Each different user u has multiple attributes u_{attr} , and each different item i has multiple attributes i_{attr} . Since the training data of a recommendation system usually consists of historical interactions between users and items, each pair of (u, i) is utilized to represent them, where $u \in \mathcal{U}$ and $i \in \mathcal{I}$.

The input of the task that this paper deals with is a graph \mathcal{G} , which includes the users and its attributes, items and its attributes, and structural-semantic

information. The final output result includes a class label \hat{y} , which is the label of (u, i) , indicating whether u and i interact.

2.2 Graph Construction and Feature Embedding Sub-module

This paper constructs three sub-graphs, including users and their attributes $\mathcal{G}_{uu} = \{(u, u_{attr}, e_{uu}) \mid u \in \mathcal{U}, u_{attr} \in \mathcal{U}, e_{uu} \in \mathcal{E}\}$, items and their attributes $\mathcal{G}_{ii} = \{(i, i_{attr}, e_{ii}) \mid i \in \mathcal{I}, i_{attr} \in \mathcal{I}, e_{ii} \in \mathcal{E}\}$, and the interactions between users and items $\mathcal{G}_{ui} = \{(u, i, u_{attr}, i_{attr}, e_{ui}) \mid u \in \mathcal{U}, i \in \mathcal{I}, u_{attr} \in \mathcal{U}, i_{attr} \in \mathcal{I}, e_{ui} \in \mathcal{E}\}$. Defined respectively, where $e_{uu} \in \mathcal{E}$ represents the relationship between a user and its attribute, $e_{ii} \in \mathcal{E}$ represents the relationship between an item and its attribute, $e_{ui} \in \mathcal{E}$ represents the interaction between a user and an item. It should be noted that \mathcal{G}_{uu} and \mathcal{G}_{ii} are complete graphs, while \mathcal{G}_{ui} is the interconnection between nodes of users and items.

This module characterizes all users and their attributes, items and their attributes. First, each node as the input is represented as a one-hot vector $Node = [node_1, node_2, \dots, node_z]$, where z represents the number of nodes, Node includes the total number of all user IDs and its attributes, item IDs and its attributes in the datasets. The $node_i$ represents the one-hot vector of the i -th node. Since the one-hot vectors are very sparse and high-dimensional, a trainable matrix $V \in \mathbb{R}^{z \times d}$ is needed to map these one-hot vectors to a low-dimensional latent space.

Specifically, the vector $node_i$ is mapped to a dense embedding $e_i \in \mathbb{R}^d$, as shown in the Eq. (1):

$$e_i = V node_i \quad (1)$$

Therefore, the feature embedding matrix can be composed by feature embedding as shown in the Eq. (2):

$$E^0 = [e_1, e_2, \dots, e_z] \quad (2)$$

2.3 Interaction Selection and Propagation Aggregation Sub-module

This sub-module adopts a multi-head attention mechanism to perform the message propagation and aggregation. As shown in Fig. 2, this module consists of multiple layers, in which each layer includes a GNNs and an Add & Norm part. Meanwhile, the left side shows the multi-layer structure of the model, and the right side shows the specific calculation of each layer in the multi-layer. In particular, the output $H^{(l)}$ of the GNNs results from the updating of the node features in each layer. The output $H^{(l)'}$ of the Add & Norm is the input of the next layer. The result of feature embedding E^0 is the input in the first layer represented by $\{H^{(0)}, H_{attr1}^{(0)}, \dots, H_{attrn}^{(0)}\}$. Finally, the results of node feature updates at each layer is concatenated to be the final output $E^l \in \mathbb{R}^{n \times l \times d}$.

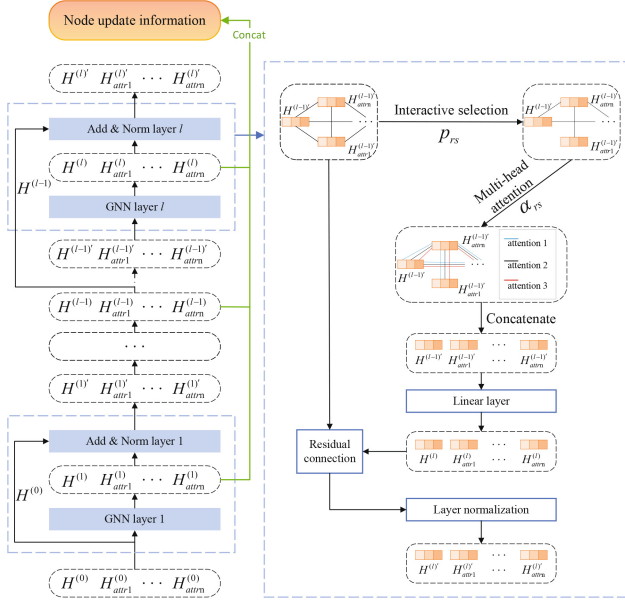


Fig. 2. Structure diagram of the interaction selection and propagation aggregation sub-module.

Interaction Selection Mechanism. Since not all node interactions are beneficial, the MGNN-IS model designs an interaction selection mechanism, in which a MLP with a hidden layer is designed to calculate the weight of the edge between two nodes by the dot product of the node pair, as shown in the Eq. (3):

$$p_{rs} = \sigma(W_2\delta(W_1(H_r \odot H_s) + b_1) + b_2) \quad (3)$$

Where, (H_r, H_s) are the feature vectors of a pair of neighboring nodes; \odot represents the dot product; $W_1 \in \mathbb{R}^{e \times d \times \text{hidden}}$ represents the weight of the first linear layer of MLP; $b_1 \in \mathbb{R}^{e \times 1}$ represents the bias of the first linear layer of MLP; δ is the activation function ReLU of the first layer of MLP; $W_2 \in \mathbb{R}^{e \times \text{hidden} \times 1}$ represents the weight of the second linear layer of MLP; $b_2 \in \mathbb{R}^{e \times 1}$ represents the bias of the second linear layer of MLP; σ is the activation function Sigmoid; $p_{rs} \in \mathbb{R}^{e \times 1}$ is the result obtained by calculating equation.

After obtaining the attention score p_{rs} , the top k edges are selected and the weights of the other edges are set to 0. The number of k is set as a fixed proportion multiplied by the number of edges in the graph. The calculation process is shown in the Eq. (4):

$$\begin{aligned} id_k &= \text{argtop}_k p_{rs} \\ p_{rs}[-id_k] &= 0 \end{aligned} \quad (4)$$

Where, argtop_k represents the operation of selecting the top k scores of p_{rs} ; id_k is the index of the top k scores, $-id_k$ is the rest of the index of p_{rs} excluding id_k .

After the interactive selection, the remaining neighbor node set of nodes H_r is defined as $\mathcal{N}_r = \{H_s \mid p_{rs} > 0, s = 1, 2, \dots, n_r\}$.

Message Propagation Aggregation. To capture the polysemous of feature interactions in different semantic sub-spaces, the MGNN_IS model adopts a multi-head attention (MHA) mechanism. Specifically, there are H-independent attentions, and the node features H_r that are evenly split into the H parts. To make the feature vector $H_r \in \mathbb{R}^d$ be split by the any number of heads, the proposed model maps it to $H_r \in \mathbb{R}^{H*d}$ with a linear transformation. The split features independently perform the update of Eq. (5) as follows:

$$\begin{aligned} H_r^o &= \text{MultiHead}(H_r \odot H_s) = \text{Concat}[head_1; \dots; head_h; \dots; head_H] \\ &\quad \text{where } head_h = \sigma(\sum_{s \in \mathcal{N}_r} \alpha_{rs}^h p_{rs} W_b^h (H_r^h \odot H_s^h)) \\ &\quad \alpha_{rs}^h = \text{Softmax}(\text{LeakyReLU}(W_a^h (H_r^h \odot H_s^h))) \end{aligned} \quad (5)$$

Where, Concat represents concatenation; both W_a^h and W_b^h are trainable linear transformation matrices at the h-th head; p_{rs} and α_{rs}^h are attention scores calculated by different functions; σ and LeakyReLU are activation functions; H_r^o is the updated node feature.

Moreover, the proposed model links the above features together to obtain the updated feature $H_r^o \in \mathbb{R}^{H*d}$. Afterward, it utilizes another linear transformation to make $H_r^o \in \mathbb{R}^d$ to facilitate subsequent calculations. In the case of multiple layers, the paper performs the addition operation for the output of the current GNN layer and the output of the previous GNN layer, followed by layer normalization, to obtain the result of the Add & Norm sub-module as $H_r^{O'} \in \mathbb{R}^d$. The purpose of Add & Norm is to improve the performance and stability of the network.

2.4 Feature Fusion Sub-module

As shown in Fig. 1, the MGNN_IS module utilizes the RNN to integrate three kinds of the node information. In particular, through the feature embedding module, the MGNN_IS model can obtain the set of all node features E^0 in the graph \mathcal{G} . Meanwhile, it can obtain the updated set of all node features E^l of the internal-interaction graphs \mathcal{G}_{uu} and \mathcal{G}_{ii} , and the updated set of all node features E_{ui}^l of the cross-interaction graph \mathcal{G}_{ui} . Afterward, the node features in E^l and E_{ui}^l are the concatenation of the outputs of each layer of the GNN module.

To make the concatenated features be able to perform subsequent calculations, the MGNN_IS model utilizes a linear layer to map the concatenated dimension to the original dimension size. Moreover, it utilizes a gated recurrent unit (GRU) [3] model to combine the three sets of node features E^0 , E^l , and E_{ui}^l , to result in the final set of node features \mathcal{F}_g , in which $\mathcal{F}_g = \text{GRU}(E^0, E^l, E_{ui}^l) = \{e_g^* \mid g \in V\}$.

2.5 Prediction Sub-module

The prediction module divides the nodes into two parts such as the users and items nodes, and corresponding average values of nodes are the feature representation for the users and items respectively. Afterward, the dot product is utilized to calculate whether the user and item interact, that is, to predict \hat{y} .

In particular, the MGNN-IS model divides \mathcal{F}_g into the user feature set \mathcal{F}_u and the item feature set \mathcal{F}_i . Moreover, to calculate the whole-graph attributes of both the user-graph and the item-graph, the MGNN-IS model utilizes the average values of their respective node sets \mathcal{F}_u and \mathcal{F}_i to capture the user-graph attributes E_u^F and the item-graph attribute E_i^F .

Finally, MGNN-IS model predicts the final value \hat{y} with the sum of the dot products based on the two graph of the user and the item attributes, as shown in the Eq. (6):

$$\hat{y} = \sigma(\text{sum}(E_u^F \odot E_i^F)) \quad (6)$$

Where $E_u^F, E_i^F \in \mathbb{R}^{b \times l * d}$, b is the batch size, σ represents the Sigmoid function, and the values in the result \hat{y} ranging from 0 to 1.

Since the task of this paper is the binary classification whether the user is interested in the item or not, the proposed model utilizes the binary cross-entropy loss function (BCELoss) shown in the Eq. (7):

$$\mathcal{L} = -(y \cdot \log \hat{y} + (1 - y) \cdot \log (1 - \hat{y})) \quad (7)$$

Where y is the true label, \hat{y} is the predicted value, and the optimizer utilizes the Adam [8] algorithm.

3 Experiment

3.1 Datasets

The MGNN-IS model was tested on the following three benchmark datasets. Table 1 summarizes the statistical details of these datasets.

- MovieLens 1M [5]: Contains user-movie ratings, and the user attributes and movie attributes.
- Bookcrossing [20]: Contains user-book ratings, and both users and books have attributes.
- AliEC [19]: Displays advertising click-through rate prediction datasets from Taobao.com.

3.2 Parameter Settings

This paper randomly splits each dataset into the training, validation, and test sets at a ratio of 6:2:2. It utilizes three evaluation metrics, namely Area Under the ROC Curve (AUC), Normalized Discounted Cumulative Gain top 5 (NDCG@5), and Normalized Discounted Cumulative Gain top 10 (NDCG@10). The specific hyper-parameter settings are shown in Table 2.

Table 1. Statistical information of the datasets.

| Datasets | Interaction | User_ID | Item_ID | User_features | Item_features |
|--------------|-------------|---------|---------|---------------|---------------|
| MovieLens 1M | 1 144 739 | 6060 | 3952 | 30 | 6049 |
| Bookcrossing | 1 050 834 | 4873 | 53 168 | 87 | 43 157 |
| AliEC | 2 599 463 | 4532 | 371 760 | 36 | 4 344 254 |

Table 2. Hyper-parameter description.

| Symbol | Size | Meaning |
|------------|--------------------|--|
| batch_size | 128 | Batch size |
| d | 64 | Feature embedding dimension size |
| epochs | 50 | Number of training iterations |
| hidden | 64 | Number of hidden units in the interaction selection function |
| lr | 1×10^{-3} | Learning rate |

3.3 Baseline Model

This paper compares the following baseline models with the MGNN-IS model.

- FM [11]: Computes relevance in a low-dimensional dense space, rather than directly computing the relevance of the input vectors themselves.
- NFM [7]: Combines FM with neural networks to capture multi-order interactions between features.
- W&D [2]: A hybrid model composed of a single-layer Wide part and a multi-layer Deep part with a strong “memory ability” and “generalization ability”.
- DeepFM [4]: Utilizes FM to replace the Wide side of W&D to simultaneously learn low-order explicit feature combinations and high-order implicit feature combinations.
- AutoInt [12]: Proposes a multi-head attention mechanism to implement the high-order explicit interactions between features.
- Fi-GNNs [9]: Models features as a complete graph and utilizes gated graph neural networks to model feature interactions.
- GMCF [14]: A graph-based CF method that utilizes both internal and cross interactions.

3.4 Experimental Results and Analysis

Comparison with Baselines. As shown in Table 3, the best-performing model is shown in bold, the second-best model is shown with an underline, and the last row is the relative improvement of the proposed MGNN-IS model compared to the best baseline.

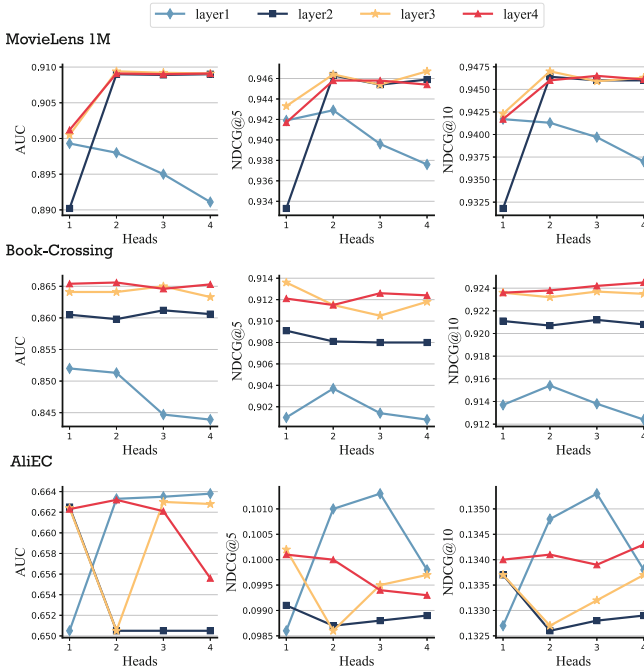
Compared with the best performance of the baseline models, the proposed MGNN-IS model improved the AUC score by 4.86%, the NDCG@5 score by 3.08%, and the NDCG@10 score by 2.77% on the Book-Crossing datasets; the

Table 3. Model performance comparison.

| | MovieLens 1M | | | Book-Crossing | | | AliEC | | |
|------------|---------------|----------------|-----------------|---------------|----------------|-----------------|---------------|----------------|-----------------|
| | <i>AUC</i> | <i>NDCG @5</i> | <i>NDCG @10</i> | <i>AUC</i> | <i>NDCG @5</i> | <i>NDCG @10</i> | <i>AUC</i> | <i>NDCG @5</i> | <i>NDCG @10</i> |
| FM | 0.8761 | 0.8761 | 0.8761 | 0.7417 | 0.7616 | 0.8029 | 0.6171 | 0.0812 | 0.1120 |
| NFM | 0.8985 | 0.8486 | 0.8832 | 0.7988 | 0.7989 | 0.8326 | 0.6550 | 0.0997 | 0.1251 |
| W&D | 0.9043 | 0.8538 | 0.8538 | 0.8105 | 0.8048 | 0.8381 | 0.6531 | 0.0959 | 0.1242 |
| DeepFM | 0.9049 | 0.8510 | 0.8848 | 0.8127 | 0.8088 | 0.8400 | 0.6550 | 0.0974 | 0.1243 |
| AutoInt | 0.9034 | 0.8619 | 0.8931 | 0.8130 | 0.8127 | 0.8472 | 0.6434 | 0.0924 | 0.1206 |
| Fi-GNN | <u>0.9063</u> | 0.8705 | 0.9029 | 0.8136 | 0.8094 | 0.8522 | 0.6462 | 0.0986 | 0.1241 |
| GMCF | 0.8998 | <u>0.9412</u> | <u>0.9413</u> | <u>0.8255</u> | <u>0.8843</u> | <u>0.8989</u> | <u>0.6566</u> | <u>0.0995</u> | <u>0.1347</u> |
| MGNN-IS | 0.9091 | 0.9458 | 0.9460 | 0.8656 | 0.9115 | 0.9238 | 0.6635 | 0.1013 | 0.1353 |
| Improve(%) | 0.31 | 0.49 | 0.50 | 4.86 | 3.076 | 2.77 | 1.05 | 1.81 | 0.45 |

AUC score by 0.31%, the NDCG@5 score by 0.49%, and the NDCG@10 score by 0.50% on the MovieLens 1M datasets; and the AUC score by 1.05%, the NDCG@5 score by 1.81%, and the NDCG@10 score by 0.45% on the AliEC datasets. Therefore, it is obvious that the proposed MGNN-IS model improved the performance on all three datasets and achieved the best improvement.

Comparison of Different Numbers of Heads and Layers. As shown in Fig. 3, the comparison with different numbers of heads and layers on datasets, it should be noted that the best performance of the model is not obtained with the same number of heads and layers for different datasets.

**Fig. 3.** Comparison of different numbers of heads and layers.

In Fig. 3, the paper conducted experiments on three datasets and utilized line charts to visualize the model performance when the number of heads was 1, 2, 3, and 4 and the number of layers was 1, 2, 3, and 4 respectively. The horizontal axis identifies the number of heads, the vertical axis identifies the scores of three different evaluation indicators, and different line colors indicate different numbers of layers. The legend shows the line colors and their corresponding numbers of layers. It is easy to find that for the MovieLens 1M and Book-Crossing datasets, the model has the best performance when the number of heads is 2 and the number of layers is 4, and after three layers, the number of layers has very little impact on the model performance. However, this situation is not the same for the AliEC datasets. This is because AliEC has a large amount of data that has more interactions between users and items and their attributes, in which the number of layers increasing will make the final features smoother.

In addition, it can be found that the proposed model has best performance while the number of heads is defined as two or three. With the number of heads increasing, however, it does not necessarily improve the performance of the model. The features need to be split evenly before sending them into the multiple heads. This means that each head gets less information as the number of heads increases. The purpose of the multi head attention mechanism is to learn information in multiple semantic sub-spaces. This can increase diversity and make the model more generalizable. By adjusting the number of heads, the model can balance the amount of information obtained by each head and the variation of the generalization performance.

Ablation Experiments. As shown in Fig. 4, this paper conducted an ablation experiment to verify the effectiveness of the interactive selection of the model. The experiments remove the interactive selection step and utilize the optimal number of heads and layers for each dataset: 2 heads and 4 layers for MovieLens 1M, 2 heads and 4 layers for Book-Crossing, and 3 heads and 1 layer for AliEC.

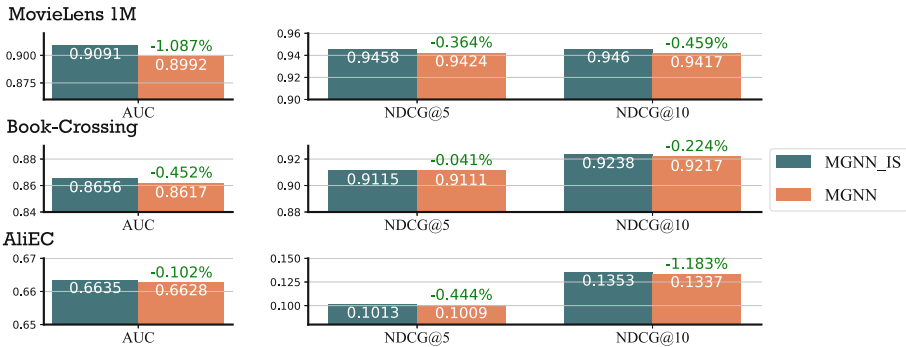


Fig. 4. Ablation experiment on the interactive selection step.

In the Fig. 4, the horizontal axis identifies the evaluation indicators, the green bars are the results of MGNN_IS, and the orange bars are the results of MGNN_IS without the interactive selection step. The white font is the specific value, and the green font on the orange bar is the decrease in the evaluation indicator score after ablating the interactive selection step.

The ablation experimental results demonstrate that interactive selection improves the performance of MGNN_IS model. In addition, it is best for the interactive selection sub-module to be combined with the multi-layer and multi-head attention sub-module together. If only interactive selection modules or multi-layer multi-head attention sub-modules are utilized, the performance is not as good as the separate multi-layer attention sub-module.

4 Conclusion

This paper proposes a novel interactive selection recommendation model named MGNN_IS, which solves the click-through rate prediction problem and improves performance of the generalization and interpretability. In particular, the MGNN_IS model constructs three sub-graphs including the user internal-interaction, item internal-interaction, and user-item cross-interaction. After feature encoding, it utilizes the MHA-based GNN with the proposed interactive selection to propagate and aggregate messages for the internal-interaction and the cross-interaction separately. Moreover, it utilizes the GRU to fuse all features of above interactions. Afterward, the MGNN_IS model divides the nodes into user's and item's nodes and combines their respective information to calculate the features of the user and item graph separately. Finally, the MGNN_IS model utilizes the dot product to predict the final click-through rate.

Compared with the baselines, the experimental results demonstrate that the MGNN_IS model improves the recommendation performance greatly. In addition, the paper also explores the function of multi-head and multi-layer, and verifies the effectiveness of the interactive selection step by the ablation study. In the future work, the paper would like to propose the cross features while reduce the noise information and achieve the personalized cross features at the sample granularity.

References

1. Beck, D., Haffari, G., Cohn, T.: Graph-to-sequence learning using gated graph neural networks. arXiv preprint [arXiv:1806.09835](https://arxiv.org/abs/1806.09835) (2018)
2. Cheng, H.T., et al.: Wide & deep learning for recommender systems. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, pp. 7–10 (2016)
3. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint [arXiv:1406.1078](https://arxiv.org/abs/1406.1078) (2014)
4. Guo, H., Tang, R., Ye, Y., Li, Z., He, X.: DeepFM: a factorization-machine based neural network for CTR prediction. arXiv preprint [arXiv:1703.04247](https://arxiv.org/abs/1703.04247) (2017)

5. Harper, F.M., Konstan, J.A.: The movielens datasets: history and context. *ACM Trans. Interact. Intell. Syst. (TIIS)* **5**(4), 1–19 (2015)
6. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: LightGCN: simplifying and powering graph convolution network for recommendation. In: *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pp. 639–648 (2020)
7. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: *Proceedings of the 26th International Conference on World Wide Web*, pp. 173–182 (2017)
8. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. *arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)* (2014)
9. Li, Z., Cui, Z., Wu, S., Zhang, X., Wang, L.: FI-GNN: modeling feature interactions via graph neural networks for CTR prediction. In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 539–548 (2019)
10. Qi, X., Liao, R., Jia, J., Fidler, S., Urtasun, R.: 3D graph neural networks for RGBD semantic segmentation. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5199–5208 (2017)
11. Rendle, S.: Factorization machines. In: *2010 IEEE International Conference on Data Mining*, pp. 995–1000. IEEE (2010)
12. Song, W., et al.: AutoInt: automatic feature interaction learning via self-attentive neural networks. In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1161–1170 (2019)
13. Su, Y., Erfani, S.M., Zhang, R.: MMF: attribute interpretable collaborative filtering. In: *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE (2019)
14. Su, Y., Zhang, R.M., Erfani, S., Gan, J.: Neural graph matching based collaborative filtering. In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 849–858 (2021)
15. Tao, Z., Wang, X., He, X., Huang, X., Chua, T.S.: HoAFM: a high-order attentive factorization machine for CTR prediction. *Inf. Process. Manag.* **57**(6), 102076 (2020)
16. Wang, H., Zhao, M., Xie, X., Li, W., Guo, M.: Knowledge graph convolutional networks for recommender systems. In: *The World Wide Web Conference*, pp. 3307–3313 (2019)
17. Wang, X., He, X., Wang, M., Feng, F., Chua, T.S.: Neural graph collaborative filtering. In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 165–174 (2019)
18. Wang, X., Wang, C.: Recommendation system of e-commerce based on improved collaborative filtering algorithm. In: *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pp. 332–335. IEEE (2017)
19. Zhou, G., et al.: Deep interest network for click-through rate prediction. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1059–1068 (2018)
20. Ziegler, C.N., McNee, S.M., Konstan, J.A., Lausen, G.: Improving recommendation lists through topic diversification. In: *Proceedings of the 14th International Conference on World Wide Web*, pp. 22–32 (2005)