



# OPEN A graph neural network recommendation algorithm based on multi-scale attention and contrastive learning

Dongqi Pu<sup>1,2</sup>, Yaming Zhang<sup>1</sup>✉, Zhenghong Qian<sup>1,2</sup>, Gaoyuan Xie<sup>1</sup> & Die Pu<sup>1</sup>

When facing sparse user–item interaction data, recommendation systems often struggle to learn high-quality representations, which in turn affects the recommendation performance. To address this issue, this paper proposes a graph neural network-based recommendation algorithm with multi-scale attention and contrastive learning (GR-MC). First, a dedicated graph structure augmentation strategy based on user-focused edge dropout is designed to intentionally reduce the dominance of high-degree user nodes in neighbor aggregation, effectively alleviating degree bias and improving the model's generalization ability. Second, a multi-scale attention embedding propagation mechanism is proposed to enhance the modeling of higher-order neighbor relationships. Finally, by treating node self-discrimination as a self-supervised task, contrastive learning is introduced to provide auxiliary signals for representation learning, thereby improving the discriminative ability of embeddings and the robustness of the model. Experimental results show that GR-MC outperforms existing methods on multiple public datasets, especially on the highly sparse Amazon-book dataset, where Recall@20 improves by 24.69%, fully demonstrating its effectiveness and robustness in sparse environments.

**Keywords** Recommendation system, Graph neural network, Multi-scale attention, Data augmentation, Contrastive learning

In the era of information explosion, various technologies have been developed to assist online users in information filtering and decision-making. Among them, recommendation systems (RS)<sup>1</sup>, as one of the most widely applied decision support systems, have become a core technology for alleviating information overload by mining users' latent preferences to achieve precise content distribution. Nowadays, recommendation systems have expanded beyond traditional domains such as movies<sup>2</sup>, music<sup>3</sup>, news<sup>4</sup>, and tourism<sup>5</sup>, penetrating popular fields such as e-commerce<sup>6</sup>, social networks<sup>7</sup>, and location-based services<sup>8</sup>, playing a crucial role in enhancing user experience and increasing platform engagement.

To achieve efficient recommendation, a wide range of recommendation methods have emerged, including collaborative filtering<sup>9,10</sup>, content-based methods<sup>11</sup>, and hybrid methods<sup>12</sup>. In recent years, deep learning-based recommendation methods have attracted widespread attention due to their powerful nonlinear modeling capabilities. These methods utilize neural networks to model the representations of users and items, as well as their complex matching relationships. In particular, recommendation methods based on Graph Neural Network (GNN) provide a new research perspective for recommendation systems. User-item interactions inherently form complex heterogeneous graph structures, containing multi-dimensional information such as user social relationships and item attribute associations. Unlike traditional Euclidean space data, graph data has non-translation invariance and dynamic topological characteristics. GNN aggregate neighborhood information through message-passing mechanisms, effectively modeling these complex relationships, and have become an important representative in deep recommendation methods<sup>13</sup>. Despite the significant progress made in recommendation systems in recent years, the sparsity of user-item interaction data remains a widespread and persistent challenge. Existing studies have shown that most mainstream recommendation approaches follow a supervised learning paradigm, where training signals primarily rely on users' historical interactions. However, compared to the vast space of potential interactions, the observed user-item interactions are extremely sparse, which severely limits the ability to learn high-quality embeddings<sup>14</sup>. This sparsity poses even greater challenges for GNN-based recommendation methods. Current GNN-based approaches suffer from the following major

<sup>1</sup>School of Information Science and Technology, Yunnan Normal University, Kunming 650500, China. <sup>2</sup>Southwest United Graduate School, Kunming 650092, China. ✉email: zhangyaming@ynnu.edu.cn

limitations during modeling: (1) sparse supervision signals; (2) noise introduced by graph augmentation; and (3) rigid neighborhood aggregation. However, the interaction data between users and items is extremely sparse, making it difficult to learn high-quality representations. To address the limitations of existing GNN-based recommendation methods, we propose a GNN recommendation algorithm based on multi-scale attention and contrastive learning (GR-MC). This method treats node self-discrimination as a self-supervised task, introduces contrastive learning to provide auxiliary signals for representation learning, combines it with a multi-scale attention embedding propagation mechanism, and designs a graph structure augmentation strategy based on user-focused edge dropout to form a robust overall framework. This effectively alleviates the problem of data sparsity and enhances the model's generalization and representation capabilities. Specifically, our work mainly includes the following aspects:

- To enhance the generalization ability of the model while preserving user-item interaction structure information, a strategy based on user-focused edge dropout is introduced in the data augmentation layer. By intentionally reducing the influence of high-degree user nodes, this strategy alleviates the degree bias problem, improves the modeling of user preferences, and reduces the interference of noisy information.
- A multi-scale attention embedding propagation mechanism is designed, which combines Large Kernel Attention, Multi-scale Mechanism, and Gated Aggregation strategies to effectively enhance the modeling of multi-level collaborative relationships between users and items, thereby improving the accuracy of the embedding representations.
- Contrastive learning is introduced as an auxiliary training framework for the graph neural network. Through the node self-discrimination task, auxiliary signals are provided. Combined with a dynamic weighting mechanism, this further distinguishes the importance differences among different neighbor nodes while considering their relevance, significantly enhancing the discriminative power of the embeddings and the robustness of the model.
- Extensive experiments were conducted on five benchmark datasets to evaluate the proposed GR-MC algorithm. The experimental results show that, while balancing computational cost and performance, the algorithm achieves an average improvement of 9.98% in Recall@20 and 14.45% in NDCG@20. Additionally, it demonstrates superior performance in cold-start scenarios, significantly alleviating the issue of data sparsity.

The structure of the paper is arranged as follows: Section 2 reviews related research work; Section 3 defines the problem and provides a detailed description of the model architecture; Section 4 presents the experimental results and discussion; Section 5 and 6 conclude the paper and discuss future work.

## Related work

### Data modeling-based recommendation methods

The development of data modeling in recommendation systems has undergone a significant shift from shallow representations to deep learning. Early collaborative filtering algorithms were primarily based on explicit analysis of the user-item interaction matrix. Matrix factorization techniques<sup>15</sup>, such as Singular Value Decomposition (SVD), mapped the high-dimensional sparse interaction matrix into a low-dimensional latent space, showing significant advantages in the Netflix competition<sup>16</sup>. Although these methods are computationally efficient, their linear modeling characteristics struggle to capture the nonlinear features of user preferences. Research by Koren et al.<sup>15</sup> indicates that the performance of traditional matrix factorization significantly deteriorates when user behavior data exhibits complex temporal dynamics.

Nowadays, machine learning plays an important role in many fields. For example, in the study of unstable motion in frictional processes, statistical methods and machine learning tools are employed to address complex nonlinear problems that are difficult to model with traditional analytical approaches<sup>17</sup>. These data-driven methods have achieved significant results, which bears similarity to the challenges faced in predicting user behavior in recommendation systems. In the field of recommendation systems, deep learning techniques have been introduced to enhance the expressive capability of models. He et al.<sup>18</sup> proposed the NeuMF framework, which innovatively combines matrix factorization and multi-layer perceptrons, using neural networks to learn the nonlinear interaction between users and items. Subsequent research, such as that by Xue et al.<sup>19</sup>, further discovered that deep neural networks can automatically learn higher-order feature combinations, performing excellently when handling heterogeneous features. Kuang et al.<sup>20</sup> found that hybrid deep learning methods, by combining the advantages of multiple deep learning models, can more effectively handle the diversity and complexity in recommendation tasks. However, Cheng et al.<sup>21</sup> pointed out that these methods still face the cold-start problem and struggle to effectively utilize the temporal patterns in user behavior sequences.

In recent years, graph representation learning techniques have brought new opportunities to recommendation systems. The DeepWalk algorithm, proposed by Perozzi et al.<sup>22</sup>, was the first to apply the Skip-gram model from natural language processing to graph-structured data, learning network representations by generating node sequences through random walks. Grover et al.<sup>23</sup>'s Node2Vec further improved the walking strategy, enabling more flexible exploration of network structures. Research by Tang et al.<sup>24</sup> showed that these methods can effectively capture higher-order similarities in user-item graphs, but their unsupervised nature leads to a discrepancy between the learning objective and the recommendation task. This limitation prompted researchers to turn towards end-to-end graph neural network methods.

It is worth noting that with the increasing complexity of recommendation scenarios, multimodal data fusion has become a new research direction. Wei et al.<sup>25</sup> explored how to enhance item representations by combining visual features and textual descriptions. Chen et al.<sup>26</sup> further demonstrated that graph neural networks integrating spatiotemporal features can more accurately model dynamic user preferences. These advancements have laid a

solid foundation for building more powerful recommendation systems, but they also bring new challenges such as computational complexity and model interpretability.

### Relationship-enhanced recommendation methods

The development of graph neural networks has driven the rapid evolution of relationship-enhanced techniques in recommendation systems. Early graph neural network-based recommendation models mainly relied on the user-item bipartite graph structure. For instance, PinSage proposed by Ying et al.<sup>27</sup> was the first to demonstrate the effectiveness of GNNs in large-scale recommendation systems. Later, Wang et al.<sup>28</sup> introduced the NGCF framework, which explicitly models higher-order connectivity through multi-layer graph convolutions, but it suffers from high computational complexity. He et al.<sup>29</sup> proposed LightGCN, which significantly improved efficiency by simplifying the message-passing mechanism. Their research showed that removing non-linear transformations and feature transformations can actually lead to better recommendation performance.

Multi-relational modeling is an important approach to improving recommendation performance. Fan et al.<sup>30</sup> pioneered GraphRec, which integrates social relationships and user-item interactions, demonstrating the value of cross-graph information propagation. Wang et al.<sup>31</sup> introduced KGAT, which further incorporates knowledge graphs and learns relation-specific propagation weights through an attention mechanism. However, Yang et al.<sup>32</sup> pointed out that such methods heavily rely on the quality of auxiliary information, and may introduce noise when the data is sparse. To address this, researchers have developed various single-relation enhancement techniques. Wu et al.<sup>14</sup> proposed SGL, which enhances model robustness through self-supervised contrastive learning, while Yu et al.<sup>33</sup> introduced XSimGCL, which achieves comparable performance by applying linear transformations to the user-item interaction matrix, providing new theoretical support for efficient contrastive learning and revealing the feasibility of extreme simplification.

The attention mechanism plays a key role in relationship enhancement. The Transformer architecture proposed by Vaswani et al.<sup>34</sup> has inspired many attention-based recommendation models. In particular, Wang et al.<sup>35</sup> introduced DGCF, which differentiates the influence factors of different relations through disentangled representation learning. Despite significant progress, existing methods still have important limitations. Chen et al.<sup>36</sup> revealed that most GNN-based recommendation models struggle to balance the importance of local and global information. In addition, Adjeisah et al.<sup>37</sup> pointed out that current data augmentation methods generally adopt random strategies, such as random perturbations of nodes or edges, which may damage the structural consistency and semantic integrity of the graph. They further emphasized that future graph data augmentation should be more structure-aware, meaning that the importance of nodes and edges should be reasonably considered during augmentation to avoid damaging critical structures. These findings provide important inspiration for the innovations proposed in this study.

### Existing challenges and research opportunities

Current recommendation systems face three key challenges. First, in relation modeling, fixed-pattern neighborhood aggregation<sup>27,28</sup> struggles to adapt to the information propagation needs of different scenarios. Experiments by Luo et al.<sup>38</sup> show that the diversity of user interests and the heterogeneity of item attributes require a more flexible relation modeling approach. They proposed an adaptive neighborhood-awareness recommendation method (ANGCN), which dynamically adjusts information propagation based on the importance of neighborhoods, effectively enhancing the flexibility and adaptability of recommendation systems. Second, the issue of the universality of data augmentation strategies is becoming increasingly prominent. Ding et al.<sup>39</sup>, in their survey, pointed out that although data augmentation can improve model robustness in graph learning, existing augmentation strategies often rely on simple random operations (such as perturbations of nodes or edges), which may not be suitable for all tasks. Especially when dealing with sparse graphs or complex tasks, these strategies may damage the original structural information and semantic consistency. Finally, the adaptability of optimization strategies is insufficient. The study by Zamfirache et al.<sup>40</sup> shows that by combining reinforcement learning and metaheuristic algorithms, the adaptability of optimization strategies can be improved. Additionally, Li et al.<sup>41</sup> demonstrated that fixed-weight multi-task learning can limit the expressive power of models.

In response to these challenges, this paper proposes three innovative directions.

- First, a multi-scale relation propagation mechanism is introduced, which dynamically models the influence of different-order neighbors on user interests through a hierarchical attention model, aiming to capture finer-grained user behavior patterns.
- Second, a user-aware graph augmentation strategy with targeted optimization is proposed, which focuses more on the key edge dropout of user nodes during the augmentation process, thereby avoiding excessive loss of core information in user behaviors.
- Finally, this paper introduces a dynamically weighted contrastive learning strategy, which adaptively adjusts the optimization strength of the main task and auxiliary tasks based on the evolution pattern of representations during training, thereby improving both the training efficiency and recommendation accuracy of the model.

### Methodology

In this section, we specifically describe the proposed Graph Neural Network-based recommendation algorithm (GR-MC), which incorporates multi-scale attention and contrastive learning, and explain how it addresses several key issues in existing GNN-based recommendation methods. First, to reduce the interference of irrelevant information, we improved the data augmentation method by adopting an edge dropout strategy that only applies to user nodes. Next, to address the issues of fixed neighborhood aggregation scale and

undifferentiated node importance, we designed a Multi-scale Attention Embedding Propagation Layer (MAEP), and enhanced the model's ability to learn multi-level collaborative representations through a dynamic weighting mechanism. Furthermore, by integrating contrastive learning into the embedding prediction and training layer, we further improved the model's representational capacity, thereby effectively improving the performance of the recommendation system in cold start and sparse data scenarios.

### Problem formulation

The main goal of this paper is to enhance the accuracy of personalized recommendation. To address the limitations of existing GNN-based recommendation methods, such as data sparsity and fixed-scale neighborhood aggregation, we propose the GR-MC algorithm by designing a multi-scale attention embedding propagation layer combined with contrastive learning. This approach strengthens the representation capability of user and item features while effectively capturing the complex relationships and latent structural patterns between users and items. Let the set of users be  $U$ , the set of items be  $I$  and the embeddings for users and items are represented as  $e_u$  and  $e_i$ . Firstly, the multi-scale attention mechanism is used to weight and fuse these embeddings, resulting in the initial embeddings of users and items as  $e'_u$  and  $e'_i$ . Subsequently, feature propagation is performed through GCN to obtain the updated embeddings  $e''_u$  and  $e''_i$  for users and items. where,  $e''_u$  and  $e''_i$  are the final user and item features that integrate neighborhood information under the joint effect of the multi-scale attention mechanism and graph convolutional propagation. Assuming that the interaction relationship between users and items is represented by a sparse adjacency matrix  $A$ , and is propagated through multi-layer graph convolution, the following mathematical expression is obtained:

$$e''_u = F_u(A, e'_u), e''_i = F_i(A, e'_i) \quad (1)$$

where  $F_u(\cdot)$  and  $F_i(\cdot)$  respectively represent the embedding propagation process of users and items, and the adjacency matrix  $A$  is used to propagate and update the embeddings.

Final prediction of user interaction probability with items:

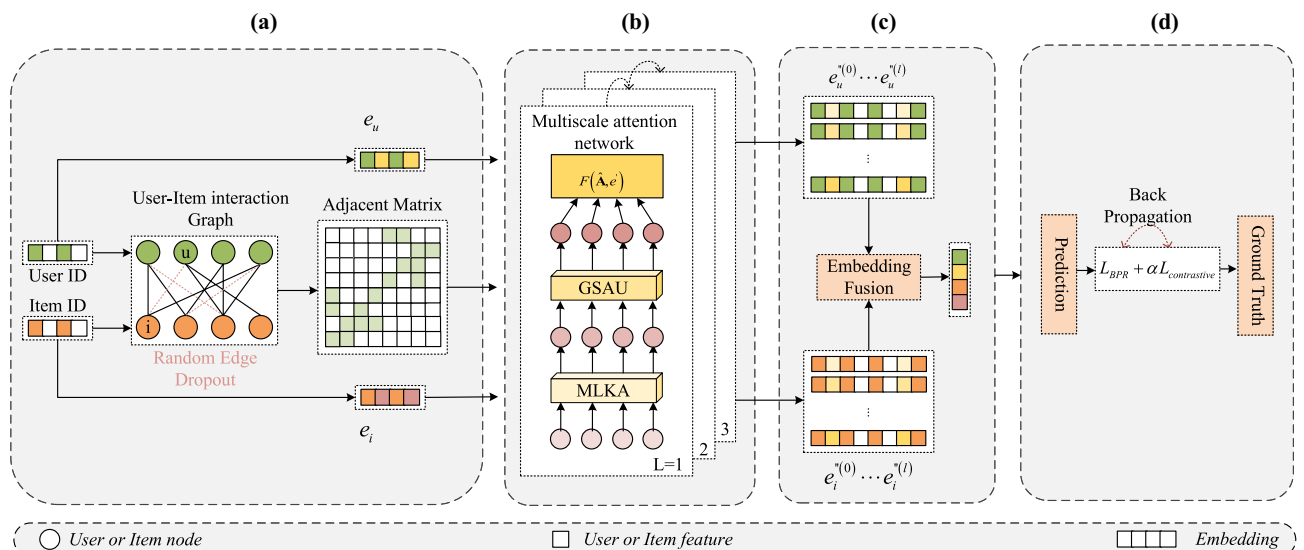
$$Y = g(e''_u, e''_i) \quad (2)$$

$$Y = \sigma(e''_u, e''_i) \quad (3)$$

where  $Y$  is the predicted interaction probability of the user to the item,  $g(\cdot)$  is a Sigmoid function to calculate the interaction probability through the embedding vector, and  $\sigma$  is an activation function to map the output value to the probability interval  $[0, 1]$ .

### Overview of GR-MC

The overview of GR-MC is illustrated in Fig. 1. As shown, GR-MC consists of four key components: (a) Input and Data Augmentation Layer, which enhances the model's generalization ability by randomly dropping some edges, generating two slightly different subgraphs; (b) Multi-Scale Attention Embedding Propagation Layer, which recursively propagates embeddings from neighboring nodes to update their representations, adopts a multi-scale attention mechanism that aggregates multi-scale features to capture multi-level collaborative relationships between users and items, and utilizes a dynamic weighting strategy to adaptively assign weights, enhancing the modeling of neighbor information and improving recommendation accuracy; (c) Embedding Fusion Layer, and (d) Embedding Fusion Layer, which integrates the outputs from the previous layers and performs back propagation to refine the model's performance.



**Fig. 1.** Overview of GR-MC.

which performs weighted fusion of embeddings from different layers in the bipartite graph to generate the final user and item embeddings; (d) Prediction and Training Layer, which predicts the interaction probability between users and items and trains the model parameters, incorporates contrastive learning based on graph structure and embedding augmentation, and minimizes the contrastive loss jointly with the BPR loss to optimize the model. The implementation details of GR-MC are shown in algorithm 1.

---

**Input:** User-item bipartite graph  $G(V, E)$ ; learning rate  $\eta$ ; max epochs  $T$ ; early stopping patience  $P$ ; contrastive loss weight  $\alpha$ ; temperature  $\tau$

- 1: Initialize  $E_u, E_i$  with Xavier uniform distribution
- 2: Perform edge dropout on user nodes to generate augmented views:  $S_1(G) = (V, M_1 \odot \varepsilon)$ ,  $S_2(G) = (V, M_2 \odot \varepsilon)$
- 3: Set best validation metric  $M^* = -\infty$ , counter  $c = 0$
- 4: Initialize epoch counter  $t = 1$
- 5: **while**  $t \leq T$  **do**
- 6:   **/\* Multi-Scale Attention Embedding Propagation (MAEP) \*/**
- 7:   **for** each subgraph  $S_k(G), k \in \{1, 2\}$  **do**
- 8:     Apply LayerNorm:  $N = LN(X)$
- 9:      $H_1 = f_1(N)$    // Linear transform
- 10:     $H_2 = f_2(N)$    // Parallel linear transform
- 11:     $LKA = MLKA(H_1)$    // Large kernel attention
- 12:     $Z = LKA \otimes H_2$    // Element-wise product
- 13:     $X = X + \lambda_1 f_3(Z)$    // Residual update
- 14:     $K = f_4(N)$ ,  $Q = f_5(N)$    // Keys & Queries
- 15:     $s_{u,i} = LeakyReLU(a^\top [K || Q])$    // Raw attention score
- 16:     $\pi_{u,i} = \frac{\exp(s_{u,i})}{\sum_{k \in N_u} \exp(s_{u,k})}$    // Softmax normalization
- 17:     $M = \sum_{i \in N_u} \pi_{u,i} K_i$    // Neighbor aggregation
- 18:     $X = X + \lambda_2 f_6(M)$    // Residual update
- 19:   **end for**
- 20:   **/\* Multi-scale Fusion \*/**
- 21:     $e'_u = \sum_{l=0}^L \tau(l) e_u^{(l)}$ ,  $e'_i = \sum_{l=0}^L \tau(l) e_i^{(l)}$ , where  $\tau(l) = \frac{1}{L+1}$
- 22:   **/\* Contrastive Learning \*/**
- 23:    Construct positive pairs  $\{z_{u_1}, z_{u_2}\}$  and negative pairs  $\{(z_{u_1}, z_{v_2}), u \neq v\}$
- 24:    Compute InfoNCE loss:  $L_{ssl} = \sum_{u_1, u_2 \in U} -\log \frac{\exp(\text{sim}(e''_{u_1}, e''_{u_2})/\tau)}{\sum_{v_2 \in U} \exp(\text{sim}(e''_{u_1}, e''_{v_2})/\tau)}$
- 25:   **/\* BPR Loss \*/**
- 26:     $\hat{y}_{ui} = e''_u \cdot e''_i$
- 27:     $L_{BPR} = -\sum_{(u,i,j) \in O} \log \sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \lambda \|\Theta\|$
- 28:    Total loss:  $L = L_{BPR} + \alpha L_{ssl}$
- 29:    Update  $\Theta$  using Adam optimizer with scheduler
- 30:    Evaluate validation metric  $M^{(t)}$
- 31:    **if**  $M^{(t)} > M^*$  **then**
- 32:       $M^* = M^{(t)}$ ,  $c = 0$    // Update best, reset counter
- 33:    **else**
- 34:       $c = c + 1$    // No improvement
- 35:    **end if**
- 36:    **if**  $c \geq P$  **then**
- 37:      **break**   // Early stopping
- 38:    **end if**
- 39:     $t = t + 1$
- 40: **end while**

**Output:** Final user embeddings  $E_u$  and item embeddings  $E_i$

---

**Algorithm 1.** Algorithm Pseudo-code of GR-MC

#### Input and data augmentation layer

The input of the GR-MC model consists of discrete user and item IDs, and a user-item bipartite graph is constructed based on historical interaction data. Specifically, users and items are modeled as two types of nodes in the graph, with edges established based on their interaction records, thereby forming a sparse adjacency



matrix. This matrix serves as the foundational structure for message propagation and representation learning in graph neural networks, carrying key structural signals in collaborative filtering. However, conventional data augmentation methods used in computer vision (CV)<sup>42</sup> and natural language processing (NLP)<sup>43</sup> tasks (such as geometric transformations or blurring) are not applicable to graph-based recommendation systems. The reasons are as follows: (1) user and item features are inherently discrete, such as one-hot encoded IDs and categorical variables, which cannot be directly transformed using operations from image or text domains; (2) Unlike CV and NLP tasks, where data instances are treated in isolation, in recommendation systems, users and items are interconnected and dependent. Therefore, graph recommendation tasks require specifically designed data augmentation strategies. This paper performs data augmentation by leveraging the collaborative filtering signals embedded in the user-item bipartite graph, using an edge dropout strategy. This method randomly drops a portion of the edges in the graph to generate two subgraphs with structural perturbations, aiming to improve the robustness of node embeddings. The formal definition is as follows:

$$S_1(G) = (V, M_1 \odot \varepsilon), S_2(G) = (V, M_2 \odot \varepsilon) \quad (4)$$

where  $M_1, M_2 \in \{0, 1\}^{|\varepsilon|}$  is a mask vector over the edge set, controlling which edges are retained for message propagation. The two subgraphs, perturbed with different structures, guide the model to learn embedding representations that are stable to structural changes, effectively mitigating noise and abnormal interactions. It is important to note that user behaviors are dynamic, while item semantic properties are relatively stable (e.g., book genres or movie styles). Existing studies often apply perturbations to both user and item nodes, which may introduce redundant noise and degrade final performance. To address this, we propose a user-focused edge dropout strategy that randomly removes a portion of edges between users and items, thereby promoting the generalization of user representations and enhancing the modeling of user preferences.

After this augmentation step, we design an embedding initialization strategy to provide a foundation for subsequent propagation and optimization. The embedding layer aims to generate low-dimensional embedding vectors for users and items to capture their latent feature representations. We adopt a widely-used embedding modeling approach<sup>44</sup>, assuming that each user and each item is associated with an embedding vector. To ensure favorable convergence properties of the initial parameters, all embeddings are initialized using Xavier uniform distribution. Let  $E_u$  and  $E_i$  denote the initial embeddings for users and items, respectively. The embedding matrices are defined as:

$$E_{initial} \in \{E_u, E_i\} \quad (5)$$

Where,  $E_{initial}$  encompasses the initial user embeddings  $E_u = \{e_u^1, e_u^2 \dots, e_u^m\}$  and the initial item embeddings  $E_i = \{e_i^1, e_i^2 \dots, e_i^n\}$ . Specifically,  $e_u^m$  denotes the  $m$ -th initial embedding of user  $u$ , and  $e_i^n$  denotes the  $n$ -th initial embedding of item  $i$ . These embedding vectors are stored in a unified embedding dictionary and serve as the input to the graph neural network propagation. In conjunction with the edge dropout strategy, we perform contrastive learning on the two augmented views  $S_1(G)$  and  $S_2(G)$ . Let  $z_{u_1}$  and  $z_{u_2}$  denote the two augmented views of a node  $u$ , and let  $U$  represent the set of all user nodes. For the same node  $u$ , its two augmented views  $z_{u_1}$  and  $z_{u_2}$  are treated as a positive sample pair. For different nodes  $u$  and  $v$ , even if they are augmented to generate different views  $z_{u_1}$  and  $z_{v_2}$ , they are considered as a negative sample pair:

$$\{(z_{u_1}, z_{u_2}) \mid u_1, u_2 \in U\} \quad (6)$$

$$\{(z_{u_1}, z_{v_2}) \mid u_1, v_2 \in U, u_1 \neq v_2\} \quad (7)$$

### Multi-scale attention embedding propagation Layer

To effectively enhance the modeling capability of complex relationships between users and items in recommendation systems, this paper draws inspiration from advanced feature modeling techniques in the field of computer vision. Specifically, we adapt the multi-scale large kernel attention mechanism from the vision domain into graph neural network-based recommendation models, and propose a novel Multi-Scale Attention Embedding Propagation Layer (MAEP). First, we design the Multi-Scale Large Kernel Attention (MLKA) module, which is inspired by the use of large receptive field convolution kernels to model long-range dependencies. By leveraging a multi-scale mechanism, the module extracts features at different levels of granularity, and uses a gating mechanism to perform weighted fusion of local and global features, thereby effectively alleviating the problem of block effects. We adapt this mechanism to the user-item graph, which enhances the expressive power of the embeddings. Then, we construct the Graph Structure Attention Unit (GSAU), which integrates graph neural networks with a self-attention mechanism to further model the structural relationships between nodes. It employs a recursive message-passing approach to reinforce the multi-level collaborative relations between users and items during the embedding propagation process. The formal definition and proof of MAPE are provided in Appendix A (see supplementary material).

Unlike traditional GNN-based recommendation models that rely on single-scale feature extraction methods, MAPE integrates multi-order neighbor information in a recursive manner through multi-scale attention and graph structure awareness, without the need for explicit partitioning into 1-hop, 2-hop, or 3-hop neighbors. This enables adaptive modeling of neighbors at various distances. MAPE consists of two main components: Multi-Scale Large Kernel Attention (MLKA) and Graph Structure Attention Unit (GSAU). The overall process proceeds through the following steps:

$$N = LN(X), X = X + \lambda_1 f_3(MLKA(f_1(N)) \otimes f_2(N)) \quad (8)$$

$$N = LN(X), X = X + \lambda_2 f_6(GSAU(f_4(N), f_5(N))) \quad (9)$$

Where,  $LN(\cdot)$  denotes Layer Normalization,  $\lambda_1$  and  $\lambda_2$  are learnable scaling factors, and  $f_i(\cdot)$  represents a pointwise convolution operation to maintain dimensional consistency. The symbol  $\otimes$  denotes element-wise multiplication. Below, we will introduce the two important modules of MLKA and GSAU.

#### Multi-scale large kernel attention

MLKA is the core component of MAEP. The primary objective of the MLKA module is to capture long-range dependency features through large convolutional kernels, extract hierarchical features via a multi-scale mechanism, and employ a gating mechanism to adaptively aggregate local and global features, ultimately avoiding block artifacts. This module consists of the following three parts:

- (1) **Large Kernel Attention (LKA)** module is designed to extract global features of nodes by leveraging large convolutional kernels, particularly to capture long-range dependencies. In our task, the relationships between users and items are often indirect, especially when users' historical behaviors are sparse or scattered. Capturing long-range dependencies is crucial for understanding these complex interactions. Large convolutional kernels aggregate more contextual information, thereby enhancing the model's global perception capability. Below is the formula for capturing long-range dependency features using large convolutional kernels:

$$LKA(X) = \sigma(X * W_{large} + b_{large}) \quad (10)$$

Where,  $X$  is the input feature,  $W_{large}$  is the weight of the large convolutional kernel,  $b_{large}$  is the bias term,  $*$  denotes the convolution operation, and  $\sigma$  is the activation function.

- (2) **Multi-Scale Mechanism (MSM)** enhances the model's perception capability at multiple levels by combining convolutional operations of different scales. Smaller convolutional kernels capture fine-grained local features, while larger convolutional kernels focus on global information. This combination of multi-scale perception enables the model to more comprehensively understand the complex relationships between users and items.

$$MSM(X) = LKA(X) + Conv_{small}(X) \quad (11)$$

Where,  $Conv_{small}(X)$  is a smaller convolutional operation that captures fine-grained local features. Multi-scale features are fused through addition, ensuring that information from different levels is effectively captured.

- (3) **Gated Aggregation (GA)** module adopts a gating mechanism to dynamically fuse local and global features. Specifically, the model learns a trainable gating parameter to adaptively control the fusion ratio between local and global information, thus avoiding excessive dependence on a single source of features and reducing information redundancy. This helps suppress block effects and enhances the model's representation capacity and convergence stability. In practice, the gating unit is implemented as a lightweight learnable module that efficiently adjusts the output of each layer without introducing significant additional computational overhead.

$$GA(X) = \beta \cdot LKA(X) + (1 - \beta) \cdot Conv_{small}(X) \quad (12)$$

#### Graph structure attention unit

The GSAU module is another core component of the MAEP that we designed. It mainly captures the relationships between nodes through GNN and self-attention mechanisms, and enhances the embedding representations between nodes through a recursive message passing mechanism. Unlike traditional attention mechanisms that rely on learnable attention parameters updated gradually over training epochs, the attention weights in GSAU are dynamically computed at each layer based on the local graph structure and node features. This mechanism mainly consists of the following three steps:

The first step is node embedding propagation. The representation of each node depends not only on its own features but also on the aggregation of information from its neighboring nodes, enriching its embedding representation. This layer-by-layer propagation mechanism enables nodes to integrate useful signals from their local neighbors, thereby obtaining more expressive representations. In our task, node embeddings are used to represent the state of users or items, while the embeddings of neighboring nodes provide additional contextual information, helping the model to more accurately capture user preferences and item characteristics. The computation of node propagation is as follows:

$$e_u^{(t+1)} = \sum_{i \in N_u} \pi_{u,i} \cdot e_i^{(t)} \quad (13)$$

Where,  $e_u^{(t+1)}$  is the embedding of node  $u$  at the  $t+1$ -th layer,  $e_i^{(t)}$  is the embedding of neighbor node  $i$  at the  $t$ -th layer,  $N_u$  is the set of neighbors of node  $u$ , and  $\pi_{u,i}$  is the attention weight between node  $u$  and its neighbor node  $i$ , representing the influence of node  $i$  on node  $u$ . The weight is dynamically computed based on the features and graph structure of nodes  $u$  and  $i$ , and is recalculated at each layer.

The second step is structure-aware weighted propagation based on the adjacency matrix. This process incorporates the graph's structural information into the attention propagation. The adjacency matrix  $A_{u,i}$  indicates whether there is a connection between node  $u$  and node  $i$ , and  $d_i$  denotes the degree of node  $i$ , which is used to normalize the propagation process and maintain model stability. By introducing a structure-aware weighting mechanism, the model relies not only on node features but also leverages the graph's topology to adjust the propagation weights, thereby enhancing its ability to model structural information. The computation of this process is as follows:

$$e_u^{(t+1)} = \sum_{i \in N_u} A_{u,i} \cdot d_i \cdot \pi_{u,i} \cdot e_i^{(t)} \quad (14)$$

The third step is information propagation from high-order neighbors. The information is not limited to first-order neighboring nodes but is also integrated from higher-order neighbors through a multi-hop propagation mechanism. In this way, a node's representation incorporates not only information from its directly connected neighbors but also features from second-order neighbors, enabling the model to capture more complex relationships between nodes. The corresponding formulation is as follows:

$$e_u^{(t+1)} = \sum_{i \in N_u} \pi_{u,i} \cdot e_i^{(t)} + \sum_{j \in N_u^{(2)}} \pi_{u,j} \cdot e_j^{(t)} \quad (15)$$

where  $N_u^{(2)}$  represents the second-order neighbors of node  $u$ .

### Embedding fusion layer

The purpose of the embedding fusion layer is to perform weighted fusion of embeddings from different levels in the bipartite graph, thereby generating the final user and item embeddings. In Embedding Layer, we define the embeddings of user  $u$  as  $z_{u1}$  and  $z_{u2}$  (obtained from the two augmented subgraphs, respectively) and the embedding of item  $i$  as  $e_i$ . After weighted fusion through the multi-scale attention mechanism, the preliminary embeddings  $e_{u1}$ ,  $e_{u2}$ , and  $e_i$  are obtained. The specific fusion process can be expressed as:

$$e'_u = \sum_{l=0}^L \tau(l) e_u^{(l)} \quad (16)$$

$$e'_i = \sum_{l=0}^L \tau(l) e_i^{(l)} \quad (17)$$

Where,  $e_u^{(l)}$  and  $e_i^{(l)}$  are the embeddings of user and item at the  $l$ -th order, respectively, and  $\tau(l)$  is the weighting coefficient for the embeddings. Inspired by LightGCN, we uniformly set  $\tau(l)$  to  $\frac{1}{L+1}$ . After performing  $L$  rounds of propagation on the embeddings using the normalized adjacency matrix  $\hat{A}$  in graph convolution, the final updated embeddings  $e''_{u1}$ ,  $e''_{u2}$ , and  $e'_i$  are obtained as follows:

$$e''_u = F_u(\hat{A}, e'_u) \quad (18)$$

$$e''_i = F_i(\hat{A}, e'_i) \quad (19)$$

### Contrastive learning embedding prediction and training layer

Contrastive learning learns more discriminative embedding representations by reducing the distance between similar samples and increasing the distance between different samples. It uses the self-recognition of nodes as a self-supervised task, providing auxiliary signals for representation learning. In this paper, we combine the multi-scale attention mechanism of GNN and apply an edge drop enhancement strategy. By ignoring the item nodes and enhancing only the user nodes, we generate two different views for each user to construct the contrastive learning task, enabling the model to learn more generalized user representations and improving its ability to model user preferences. In the previous, we have already fused the embeddings to obtain the final user and item embeddings  $e_{u1}$ ,  $e_{u2}$ , and  $e_i$ . To optimize the embeddings, we employ a contrastive loss to maximize the similarity between positive sample pairs while minimizing the similarity between negative sample pairs. InfoNCE has been widely validated as effective in the field of graph contrastive learning for recommendation, offering good training stability and representation discriminability<sup>14,33</sup>, while also integrating naturally with our multi-scale embedding propagation structure. Therefore, we adopt InfoNCE as the loss function in this work. The user-side contrastive loss is defined as follows:

$$L_{ssl}^{user} = \sum_{u_1, u_2 \in U} -\log \frac{\exp(s(e''_{u1}, e''_{u2})/\tau)}{\sum_{v_2 \in U} \exp(s(e''_{u1}, e''_{v2})/\tau)} \quad (20)$$



Where, the similarity between  $s(e''_{u_1}, e''_{u_2})$  and  $s(e''_{u_1}, e''_{v_2})$  is calculated using cosine similarity:

$$s(e''_{u_1}, e''_{u_2}) = \frac{e''_{u_1} \cdot e''_{u_2}}{\|e''_{u_1}\| \|e''_{u_2}\|} \quad (21)$$

$$s(e''_{u_1}, e''_{v_2}) = \frac{e''_{u_1} \cdot e''_{v_2}}{\|e''_{u_1}\| \|e''_{v_2}\|} \quad (22)$$

$\tau \in [0.1, 1]$  controls the smoothness of the distribution, and the negative samples  $e''_{v_2}$  are taken from the augmented views of different users.

Based on the final embeddings of users and items, the interaction score is calculated as follows:

$$\tilde{y}_{ui} = e''_u \cdot e''_i \quad (23)$$

To learn the model parameters, we optimize the Bayesian Personalized Ranking (BPR) loss function<sup>27</sup>, which is widely adopted in recommendation systems. The BPR loss is defined as follows:

$$L_{BPR} = - \sum_{(u,i,j) \in O} \ln \sigma(\tilde{y}_{ui} - \tilde{y}_{uj}) + \lambda \|\Theta\| \quad (24)$$

Where,  $O = \{(u, i, j) \mid (u, i) \in R^+, (u, j) \in R^-\}$  represents the pairwise training dataset,  $\sigma(\cdot)$  is the sigmoid function,  $\Theta$  denotes all learnable parameters, and  $\lambda$  is the L2 regularization coefficient used to prevent model overfitting.

Finally, our objective function is a weighted combination of the BPR loss and the contrastive loss, formulated as follows:

$$L = L_{BPR} + \alpha L_{ssl}^{user} \quad (25)$$

where  $\alpha$  is a hyperparameter used to balance the weights of the BPR loss and the contrastive loss. In this study, we followed the configuration of the representative contrastive learning-based recommendation model SGL<sup>14</sup>, and uniformly set  $\alpha$  to 0.1.

During the training process, we have implemented several measures to optimize model performance, including:

- (1) Early Stopping, to prevent overfitting and improve training efficiency, we employ an early stopping mechanism. If the model fails to significantly improve performance within 50 training epochs, the training process is halted prematurely. This helps save computational resources and avoids overfitting the training data.
- (2) Optimizer and Learning Rate Scheduling, we use the Adam optimizer to adaptively adjust the learning rate for faster convergence. Additionally, a learning rate scheduling strategy is applied to prevent excessively large learning rates in the later stages of training, ensuring stable optimization.

## Experiments

In order to demonstrate the superiority of GR-MC and reveal the reasons for its effectiveness, extensive experiments were conducted and the following research questions were answered:

- RQ1: How does GR-MC perform compared to other recommendation methods?
- RQ2: How do the key components of GR-MC impact its performance?
- RQ3: How do different parameter settings affect the proposed GR-MC?
- RQ4: How does GR-MC perform under cold-start scenarios?
- RQ5: How does the time cost of GR-MC compare to other recommendation methods?

## Experimental settings

### Datasets

This study conducted experiments on five widely used public benchmark datasets: MovieLens-100K, MovieLens-1M, Gowalla, Yelp2018, and Amazon-book. These datasets cover different domains and user scales, with significant differences in the number of users and items as well as interaction density, which can effectively

Dataset	Users	Items	Interactions	Density
MovieLens-100k	943	1682	180,808	0.11399
MovieLens-1M	6040	3706	1,500,542	0.06704
Gowalla	29,858	40,981	1,027,370	0.00084
Yelp2018	31,668	38,048	1,561,406	0.00130
Amazon-book	52,643	91,599	2,984,108	0.00062

**Table 1.** Statistics of the datasets.

reflect the data sparsity and diversity challenges faced by practical recommendation systems. Detailed statistics of each dataset are shown in Table 1.

**MovieLens:** This dataset is sourced from the MovieLens website and includes three stable benchmark datasets with different scales: MovieLens-100K, MovieLens-1M, and MovieLens-20M<sup>45</sup>. Each dataset contains user ratings of movies, movie attributes and tags, as well as demographic features of users. The rating scale ranges from 1 to 5, with a step of 1. In this paper, we select MovieLens-100K and MovieLens-1M for experimentation.

**Gowalla:** This dataset is sourced from the Gowalla platform's check-in data, which contains location information shared by users through check-ins<sup>46</sup>. As a classic dataset in the recommendation domain, in addition to check-in records, this dataset also includes social relationships between users.

**Yelp2018:** This dataset is an academic benchmark dataset built based on data from the Yelp open platform, containing structured data such as user ratings, review texts, social relationships, and business attributes up to 2018<sup>28</sup>. It is widely used in research on recommendation systems and social network analysis.

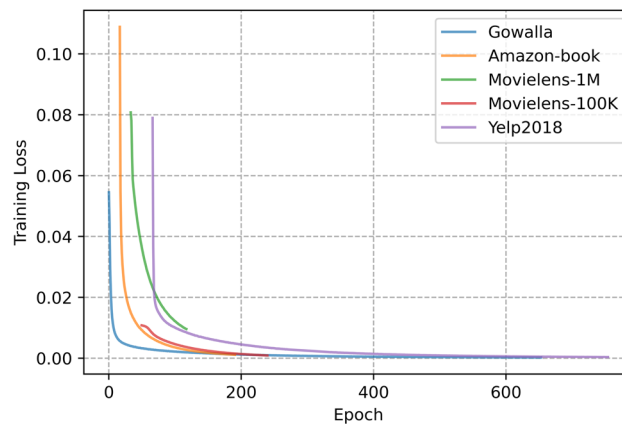
**Amazon-book:** This dataset originates from the Amazon platform, containing user book ratings and review data. It is a widely used public benchmark dataset in recommendation system research. It includes structured data such as user ratings, review texts, and product attributes, making it particularly suitable for studying the performance of recommendation algorithms in sparse data environments<sup>47</sup>.

To ensure the quality of the dataset, we follow the data processing method used in the NGCF model with a 10-core setting<sup>48</sup>, meaning that only users and items with at least 10 interactions are retained. For each dataset, 80% of each user's historical interaction data is randomly selected as the training set, while the remaining 20% is used as the test set. Additionally, 10% of the interaction data from the training set is further randomly sampled to construct the validation set for hyperparameter tuning. For each observed user-item interaction, it is considered a positive sample, and a negative sample is generated by randomly selecting an item that the user has not interacted with using a negative sampling strategy.

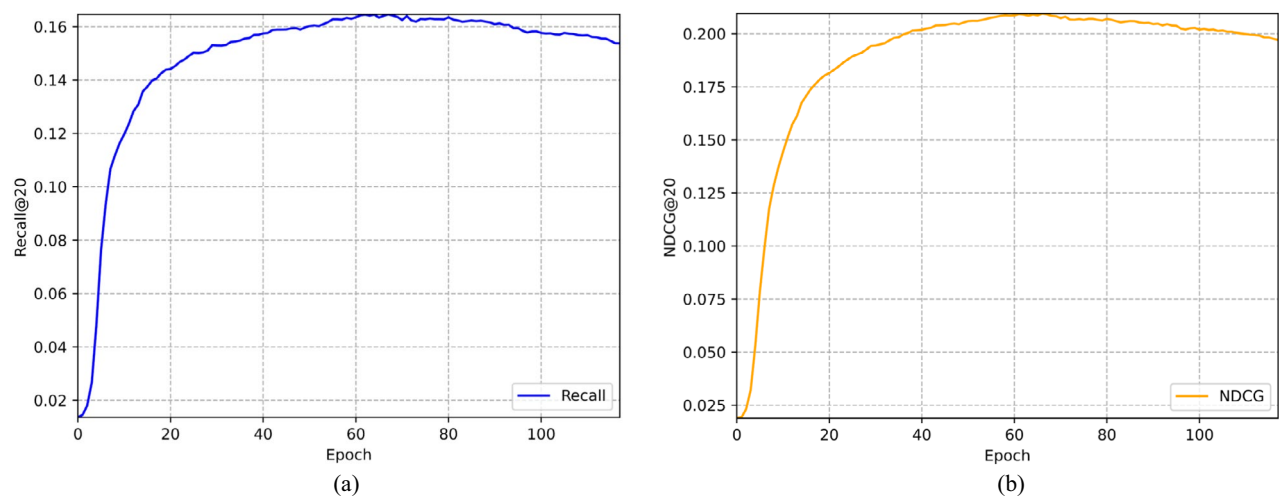
#### Baseline methods

To validate the effectiveness of the proposed GR-MC model, it is compared with seven state-of-the-art baseline models. These baseline models cover different categories of recommendation algorithms, including traditional matrix factorization methods (MF and NeuMF), graph neural network-based recommendation methods with no feature information (NGCF and LightGCN), graph neural network-based recommendation methods with feature information (GC-MC and SGL), as well as self-supervised learning-based SGL and graph attention network-based NGAT4Rec recommendation methods. Specifically, MF and NeuMF are introduced to test the performance of traditional matrix factorization and neural network-based recommendation methods in recommendation tasks, thereby demonstrating the advantages of GNN models in capturing complex relationships between users and items, especially in the absence of explicit feature information. These two baseline models help verify the improvement in recommendation accuracy of GNN over traditional methods. Moreover, NGCF and LightGCN are introduced to compare GNN-based recommendation methods with graph convolution mechanisms and traditional matrix factorization methods, to verify whether GNN models with graph structure information can achieve better performance in capturing complex interactions between users and items. Finally, NGAT4Rec is introduced to evaluate the improvement of the attention mechanism on GNN recommendation methods, and SGL is introduced to verify the role of contrastive learning in enhancing embedding representations and improving model generalization ability. The following outlines these baselines:

- MF<sup>15</sup>: It is a widely used baseline method, simple and efficient, commonly applied in various recommendation systems, and can provide good recommendation results even in scenarios with sparse data.
- NeuMF<sup>18</sup>: It combines the advantages of traditional matrix factorization and neural network models. Through the multi-layer perceptron (MLP), it can learn more complex nonlinear relationships and enhance the expressive power of the recommendation model, thereby improving the modeling accuracy of user-item interactions.
- GC-MC<sup>49</sup>: It combines graph convolution with matrix completion techniques, making full use of graph structural information to enhance the performance of the recommendation system. This method effectively models the implicit relationships between users and items, providing an improvement over traditional matrix factorization.
- NGCF<sup>28</sup>: This is a typical graph neural network-based recommendation method that effectively captures higher-order interaction information between users and items through graph convolution networks. The model propagates information within the graph structure, making the embedding representations more precise, and is suitable for handling complex user-item relationships.
- LightGCN<sup>29</sup>: As a simplified graph neural network model, it removes complex nonlinear activation functions and multiple message-passing operations, using a simple neighbor aggregation mechanism. This approach can significantly improve computational efficiency while maintaining high recommendation performance, making it particularly suitable for large-scale datasets.
- GTN<sup>50</sup>: It automatically generates multi-hop connections (meta-paths) through graph transformation layers, mining potentially useful relationships in heterogeneous graphs without the need for predefined structures, thereby enabling end-to-end node representation learning and enhancing modeling capability in complex graph structures.
- NGAT4Rec<sup>51</sup>: This is a recommendation method based on a neighbor-aware graph attention network. By pairwise calculating attention weights between neighbors, it explicitly models implicit correlations and assigns neighbor-aware attention coefficients to enhance the quality of embedding representations. The model combines multiple layers of neighbor-aware graph attention, aggregating multi-hop neighbor signals while simplifying the network architecture.



**Fig. 2.** Performance of GR-MC training loss on five datasets.



**Fig. 3.** Performance of GR-MC on Movielens-100k Dataset.

- **SGL<sup>14</sup>**: It adopts a self-supervised learning strategy, enhancing the generalization ability of the recommendation system through data augmentation on the graph structure. Additionally, contrastive learning effectively improves the discriminative power of the embeddings, thereby enhancing the model's recommendation performance. SGL introduces three different data augmentation techniques, effectively mitigating the issue of sparse labeled data.

#### Evaluation metrics

This study uses two commonly used metrics, Recall@K and NDCG@K. Recall@K is used to evaluate the proportion of the K recommended items that match the user's actual interactions. NDCG@K further incorporates the ranking of the recommended items, giving higher weights to correct recommendations that appear higher in the ranking. The higher the values of these two metrics, the better the quality of the model's recommendations. In this study, K=20 is selected to analyze the experimental results.

#### Implementation details

All experiments were conducted in the same experimental environment, with the following specifications: NVIDIA A40 GPU, CUDA 11.7, driver version 515.65.01, Python 3.8.19, and PyTorch 2.0.0.

To ensure fair comparisons, the best hyperparameter settings reported in the original baseline papers are referenced. The Xavier is used for initialization, the Adam optimizer is selected, the batch\_size is uniformly set to 1024, reg is set to 1e-5, the initial weight for contrastive learning is set to 0.1, and the learning rate is set to 1e-4. To prevent overfitting, an early stopping strategy is employed to record the best model parameters, which are then used for evaluation on the test dataset.

#### Comparison experiments (Q1)

In this section, we compare GR-MC with seven baseline models across five datasets. For example, Fig. 2 shows the training loss of GR-MC on the five datasets, while Figs. 3 and 4 present the Recall@20 and NDCG@20

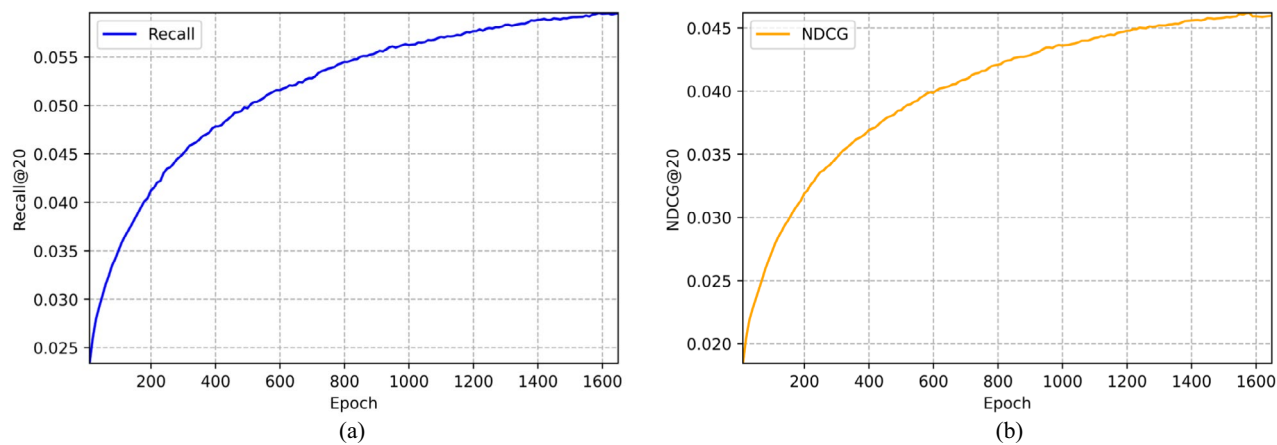


Fig. 4. Performance of GR-MC on Amazon-book Dataset.

Methods	Movielens-100K		Movielens-1M		Gowalla		Yelp2018		Amazon-book	
	R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20
MF	0.0696	0.0467	0.1319	0.1327	0.1291	0.1109	0.0433	0.0354	0.0250	0.0196
NeuMF	0.0719	0.0487	0.1389	0.1395	0.1399	0.1212	0.0451	0.0363	0.0258	0.0200
GC-MC	0.0813	0.0443	0.1554	0.1557	0.1395	0.1204	0.0462	0.0379	0.0288	0.0224
NGCF	0.1433	0.1817	0.2437	0.2330	0.1569	0.1327	0.0579	0.0477	0.0337	0.0261
LightGCN	0.1483	0.1933	0.2501	0.2412	0.1830	<u>0.1554</u>	0.0649	0.0530	0.0411	0.0315
GTN	0.1472	0.1933	0.2654	0.2543	0.1906	0.1173	0.0637	0.0523	0.0465	0.0365
NGAT4Rec	0.1496	0.1988	0.2823	0.2659	0.2009	0.1185	0.0675	0.0554	0.0457	0.0358
SGL	<u>0.1502</u>	<u>0.1998</u>	<u>0.2851</u>	<u>0.2662</u>	<u>0.2018</u>	0.1196	<u>0.0675</u>	<u>0.0555</u>	<u>0.0478</u>	<u>0.0379</u>
GR-MC	<b>0.1646</b>	<b>0.2095</b>	<b>0.2971</b>	<b>0.3380</b>	<b>0.2081</b>	<b>0.1640</b>	<b>0.0731</b>	<b>0.0627</b>	<b>0.0596</b>	<b>0.0462</b>
%Improv.	9.59%	4.85%	4.21%	26.97%	3.12%	5.53%	8.30%	12.97%	24.69%	21.90%

Table 2. Overall performance comparison, where bold indicates the best performance for each row, underlined indicates the best baseline performance, and %Improv. represents the performance improvement of GR-MC compared to the best baseline.

results on MovieLens-100K and Amazon-book, clearly illustrating the training process and advantages of GR-MC. Specifically, the detailed results of GR-MC and the baselines are shown in Table 2, from which it can be observed that our method achieves superior Recall and NDCG on both small-scale datasets (MovieLens-100K and MovieLens-1M) and large-scale sparse datasets (Gowalla, Yelp2018, and Amazon-book).

- GR-MC demonstrates significantly better overall performance compared to all baseline models. On average, it achieves improvements of 9.98% in Recall@20 and 14.45% in NDCG@20 across the five datasets. Notably, on the MovieLens-1M dataset, NDCG@20 improves substantially by 26.97%, which can be attributed to GR-MC’s enhanced capability to capture user interests, as well as its improved ranking accuracy through dynamic modulation and contrastive signal enhancement. On Amazon-book, a typical long-tail sparse dataset, GR-MC improves Recall@20 and NDCG@20 by 24.69% and 21.90%, respectively, verifying its effectiveness and robustness in complex recommendation scenarios.
- GNN-based methods show clear advantages in recommendation systems. Unlike MF models, which compute static associations between users and items via inner product, GNNs recursively aggregate information from neighboring nodes, enabling the capture of richer high-order collaborative signals. For example, NGCF introduces a hierarchical neighbor aggregation mechanism that significantly enhances representational capacity, while LightGCN reduces model complexity by removing redundant modules without sacrificing performance, performing particularly well on datasets like Gowalla and Amazon-book. These approaches demonstrate that GNNs can effectively improve node representations, thereby significantly boosting recommendation performance.
- Compared to the self-supervised baseline SGL, GR-MC further enhances the consistency and discriminability of embedding representations. SGL improves the robustness of user and item embeddings by generating contrastive views of graphs for self-supervised learning. GR-MC, on the other hand, adopts a contrastive learning framework as an auxiliary strategy, integrates user-side enhancement, and designs the MLKA and GSAU modules to further improve the modeling of neighborhood information. Through joint optimization

- guided by contrastive loss, the model not only achieves global contrastive learning, but also extracts more discriminative local features, ultimately improving recommendation accuracy.
- From a model design perspective, GR-MC enhances the modeling of crucial neighbor information through the MLKA and GSAU modules while maintaining moderate computational complexity. This targeted enhancement is particularly effective for long-tail sparse datasets such as Amazon-book, where GR-MC demonstrates stronger robustness and accuracy. Overall, these improvements enable GR-MC to maintain excellent performance across various complex recommendation scenarios.

**Ablation study (Q2)**

In this section, in order to further validate the necessity of the key components in the proposed method, a series of ablation experiments were conducted, as shown in Table 3. These experiments separately evaluate the effects of MLKA and GSAU. Recall@20 and NDCG@20 were used as evaluation metrics for result analysis, leading to the following main conclusions:

- (1) The advantages of Multi-Scale Large Kernel Attention (MLKA). The ablation experiment was conducted on the MLKA module, and when switched off, the model's performance declined. The average performance of Recall@20 and NDCG@20 across the five datasets decreased by 6.09%, 8.12%, 4.72%, 12.07%, and 18.44%, respectively. This indicates that MLKA can extract information from multiple scales, enhancing the effectiveness of GNN propagation. Removing this module leads to a reduced ability to extract information, making it difficult for the model to effectively aggregate distant neighbor information, which affects GNN propagation and results in an overall performance drop.
- (2) The effectiveness of the Graph Structure Attention Unit (GSAU). The ablation experiment was conducted on the GSAU module. After removing GSAU from each dataset, the average performance of the model decreased by 2.92%, 5.61%, 1.41%, 7.98%, and 9.06%, respectively. The drop in experimental results further indicates that while edge dropout, as a data augmentation technique, has significant advantages in improving model robustness, alleviating overfitting, and enhancing long-tail user modeling, it can also lead to a reduction in the neighbors of certain users, affecting the completeness of information propagation. This effect is more pronounced in sparsely-interacted datasets. However, our designed GSAU, through adaptive attention mechanisms and high-order information aggregation strategies, effectively compensates for the information loss caused by edge dropout, allowing the model to still learn crucial interaction information even with reduced neighbors, thus improving recommendation performance.
- (3) This paper compares three different edge dropout strategies: applying augmentation only to the edges adjacent to user-only, only to the edges adjacent to item-only, and to both user and item nodes simultaneously. Specifically, under the item-only strategy, the average performance on Recall@20 and NDCG@20 across the five datasets drops by 1.43%, 2.05%, 3.04%, 5.41%, and 6.69%, respectively; while under the symmetric strategy, the average drops are 1.14%, 1.06%, 1.65%, 2.68%, and 3.46%. These results indicate that structural perturbation on the user side plays a more critical role in enhancing model generalization and training stability in user-item interaction graphs. This is mainly because user behavior patterns and interest distributions are more diverse, and moderately augmenting their adjacent edges helps improve the robustness of representation learning. In contrast, item nodes are typically denser, and excessive perturbation tends to introduce noise, negatively affecting the accuracy of information propagation. Appendix B (see supplementary material) provides simplified preliminary theoretical analysis of (a) the gradient variance under the edge dropout mechanism and (b) the expected performance gain per training epoch, as an attempt to support these observations.

**Model architecture study (Q3)**

*Effect of layer numbers*

The results in Table 4 show that the 3-layer model achieves the highest Recall@20 and NDCG@20, effectively balancing information aggregation and over-smoothing. This indicates that moderately increasing the number of layers can enhance the model's expressive power. However, performance degrades in the 4- to 6-layer models due to excessive smoothing caused by overly deep networks, which weakens the model's discriminative ability. Additionally, on sparser datasets such as Amazon-Book and Yelp2018, the 1-layer model performs significantly worse, suggesting that shallow networks struggle to sufficiently capture complex user-item interactions. In contrast, the 3-layer model better leverages neighbor information, resulting in improved recommendation performance. Overall, the 3-layer architecture consistently delivers optimal performance across different

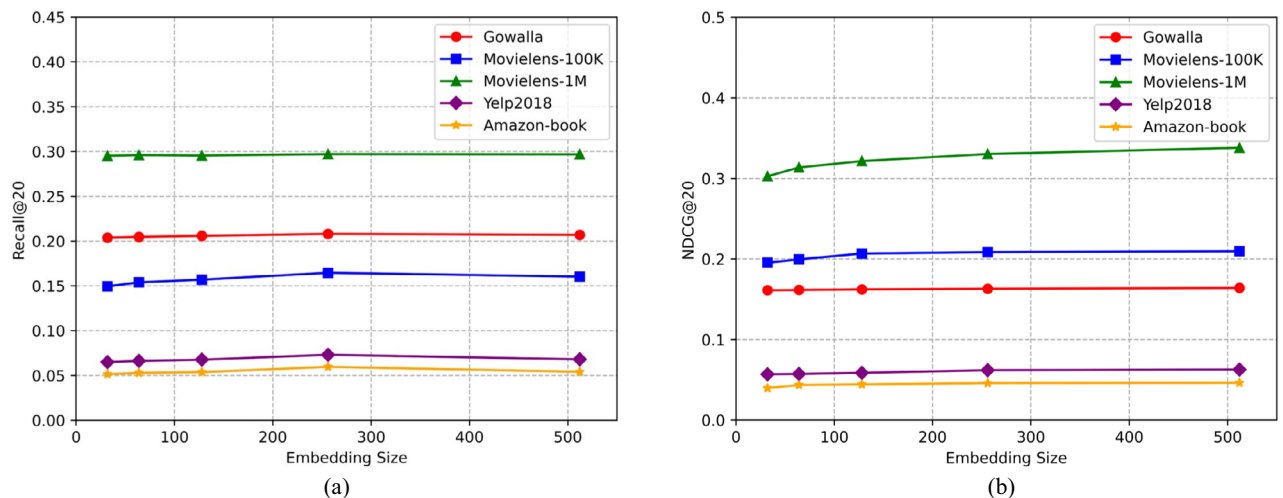
Methods	Movielens-100K		Movielens-1M		Gowalla		Yelp2018		Amazon-book	
	R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20	Re@20	N@20
GR-MC	0.1646	0.2095	0.2971	0.3380	0.2081	0.1640	0.0731	0.0627	0.0596	0.0462
item-only	0.1627	0.2059	0.2918	0.3302	0.2027	0.1583	0.0695	0.0590	0.0564	0.0425
symmetric	0.1639	0.2056	0.2942	0.3341	0.2049	0.1611	0.0714	0.0608	0.0578	0.0444
w/o MLKA	0.1531	0.1986	0.2884	0.2930	0.1976	0.1568	0.0649	0.0546	0.0473	0.0387
w/o GSAU	0.1593	0.2040	0.2961	0.3012	0.2044	0.1623	0.0682	0.0569	0.0528	0.0431

**Table 3.** The impact of user-only Edge Dropout, MLKA and GSAU on the performance of the GR-MC model.



Layer	Movielens-100K		Movielens-1M		Gowalla		Yelp2018		Amazon-book	
	R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20
1 Layer	0.1620	0.2079	0.2960	0.3324	0.1988	0.1567	0.0602	0.0487	0.0531	0.0413
2 Layer	0.1593	0.2032	0.2955	0.3368	0.2043	0.1603	0.0609	0.0496	0.0538	0.0418
3 Layer	<b>0.1646</b>	<b>0.2095</b>	<b>0.2971</b>	<b>0.3380</b>	<b>0.2081</b>	<b>0.1640</b>	<b>0.0619</b>	<b>0.0506</b>	<b>0.0596</b>	<b>0.0462</b>
4 Layer	0.1635	0.2065	0.2969	0.3376	0.2026	0.1600	0.0615	0.0504	0.0591	0.0455
5 Layer	0.1624	0.2056	0.2951	0.3352	0.1973	0.1578	0.0608	0.0497	0.0586	0.0452
6 Layer	0.1621	0.2028	0.2940	0.3341	0.1938	0.1543	0.0595	0.0489	0.0581	0.0450

**Table 4.** The impact of the number of embedding propagation Layers (L).



**Fig. 5.** The impact of the number of embedding size.

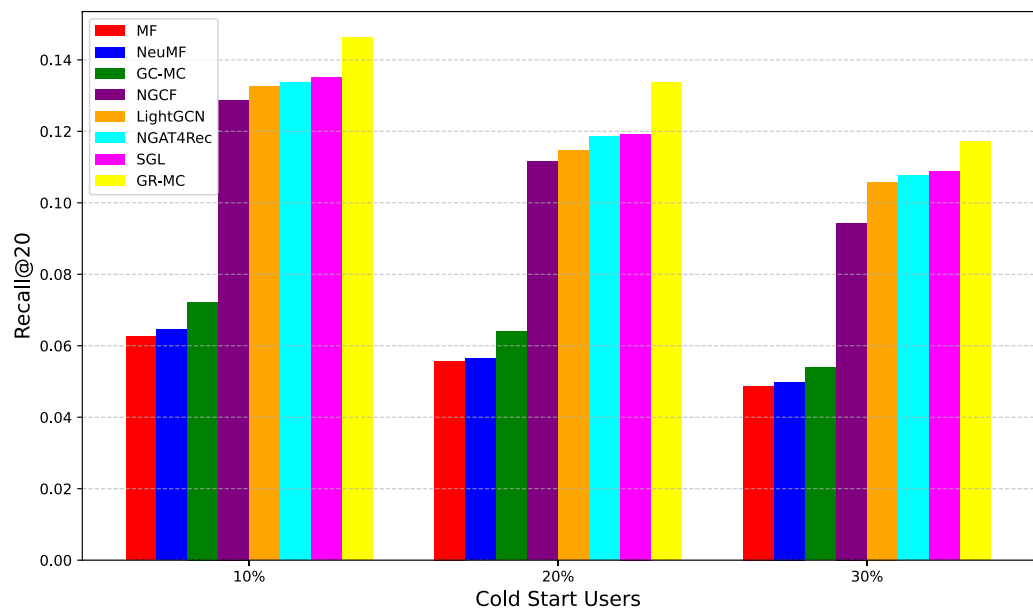
datasets; therefore, we adopt a 3-layer propagation depth in this work. At the same time, to further mitigate the overfitting risk potentially caused by deeper networks, we incorporated strategies such as neighbor sampling, user-focused edge dropout, L2 regularization, and early stopping in the model.

#### Effect of embedding size

Both excessively large and small embedding dimensions can affect the expressive power and computational efficiency of the GR-MC model. Larger embedding dimensions enhance the model's ability to capture complex user-item relationships, especially improving recommendation accuracy in the multi-scale attention mechanism, but at the cost of increased computational and storage overhead. Conversely, smaller embedding dimensions can improve training efficiency and help prevent overfitting in sparse data scenarios but may be insufficient to represent complex relationships. Therefore, this paper analyzes the impact of different embedding dimensions on the proposed model's performance by varying the dimension size, e.g.,  $d=32, 64, 128, 256, 512$ , and evaluates performance using Recall@20 and NDCG@20 metrics. Through experiments, a balance between expressive power and computational efficiency is achieved. As shown in Fig. 5, the Recall@20 and NDCG@20 generally improve across five datasets as the embedding dimension increases, but after  $d=256$ , further increases yield diminishing performance gains and even marginal declines. Thus, the embedding dimension of 256 is finally adopted, combined with L2 regularization and early stopping mechanisms, to further prevent redundancy and overfitting risks associated with larger embeddings.

#### Cold start study (Q4)

The cold-start problem is one of the major challenges in recommendation systems, referring to the difficulty of generating personalized recommendations for users with no historical interaction data. To assess the robustness of GR-MC in cold-start scenarios, we conducted user-level cold-start experiments on the Movielens-100K dataset. Specifically, we randomly selected a subset of users and excluded all their interaction data from the training set to simulate a real-world cold-start situation. The model then generated recommendations for these users without relying on any prior behavior data, and we evaluated the results using the Recall@20 metric. As shown in Fig. 6, GR-MC significantly outperforms various baseline methods under the cold-start condition, demonstrating its superior recommendation performance. These results indicate that, through its neighbor-enhanced mechanism and contrastive learning strategy, GR-MC is able to infer user preferences effectively even in the absence of interaction history, achieving robust and accurate recommendations.



**Fig. 6.** Cold start user Recall@20 result.

Model	Movielens-100K		Movielens-1M		Gowalla		Yelp2018		Amazon-book	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing	Training	Testing
MF	1.8	1.85	28.2	4.9	16.4	128.0	28.5	127.8	59.0	257.3
NeuMF	5.3	2.55	84.5	7.8	49.2	214.0	85.4	221.7	176.9	431.5
GC-MC	9.4	3.25	140.7	10.5	82.0	299.5	142.3	307.8	294.9	602.5
NGCF	18.4	5.0	281.8	18.8	163.9	213.8	284.6	335.5	589.8	862.5
LightGCN	5.1	2.52	83.5	7.6	49.2	108.5	85.1	141.7	176.9	431.5
NGAT4Rec	15.2	3.8	198.4	11.2	83.5	120.1	253.2	168.4	569.4	534.2
SGL	12.2	3.2	187.1	10.1	72.4	108.5	232.7	147.9	517.6	508.3
GR-MC	<b>13.4</b>	<b>3.3</b>	<b>196.3</b>	<b>10.3</b>	<b>81.8</b>	<b>114.7</b>	<b>240.5</b>	<b>152.4</b>	<b>550.9</b>	<b>529.4</b>

**Table 5.** The training and testing time (unit: seconds/epoch).

### Complexity and time efficiency analysis (Q5)

To comprehensively evaluate the potential computational overhead introduced by incorporating the Multi-Scale Large Kernel Attention (MLKA) and the Graph Structure Attention Unit (GSAU) in GR-MC, this paper records the training time and the total time cost during the testing phase for each model on five datasets, as shown in Table 5, and provides both theoretical and empirical analyses. It should be noted that the testing time includes not only the forward inference on the test data but also the computational cost of calculating evaluation metrics.

In each GNN propagation layer, the core operations of GR-MC are neighbor aggregation and embedding update. Let the total number of user and item nodes be  $N$ , the average number of neighbors per node be  $C$ , the embedding dimension be  $d$ , and the number of attention scales be  $S$ . The complexity of each module is as follows:

- **Single GNN layer propagation:**

$$\mathcal{O}(N \cdot C \cdot d) \quad (26)$$

- **Multi-scale Large Kernel Attention (MLKA):**

$$\mathcal{O}(S \cdot N \cdot C \cdot d) \quad (27)$$

- **User-focused Edge Dropout for contrastive views:**

$$\mathcal{O}(2 \cdot S \cdot N \cdot C \cdot d) \quad (28)$$

- **Contrastive loss computation:** Let  $M$  be the number of nodes sampled for contrastive learning and  $K$  the number of negative samples per node.

$$\mathcal{O}(M \cdot K \cdot d) \quad (29)$$

Thus, the total time complexity of GR-MC can be approximated as:

$$\mathcal{O}(2 \cdot S \cdot N \cdot C \cdot d + M \cdot K \cdot d) \quad (30)$$

Compared to baseline models, the multi-scale attention strategy in GR-MC introduces additional but controllable overhead, while significantly enhancing the representational capacity of the model. To further demonstrate the above phenomenon, the following section analyzes the time cost of GR-MC from an experimental perspective.

From the experimental results, it can be observed that GR-MC exhibits favorable control over time costs during both training and inference stages. For example, on the MovieLens-1M dataset, the per-epoch training time of GR-MC is 196.3 seconds, which is shorter than that of NGCF. On large-scale sparse graphs such as Amazon-book, GR-MC also outperforms NGCF and NGAT4Rec, validating the practical effectiveness of its structural optimization strategy. In the inference stage, GR-MC maintains high efficiency as well, requiring only 114.7 seconds on the Gowalla dataset, which is about 46% less than NGCF. However, on small-scale datasets like MovieLens-100K, the advantage in inference time is less obvious due to the relatively high proportion of fixed computational overhead, where minimalist models such as LightGCN perform better.

In summary, GR-MC achieves performance improvement while keeping the time cost within an acceptable range, thus meeting the computational efficiency requirements of practical recommendation tasks.

## Discussion

Our model needs to consider potential social impacts. First, due to the inherent message-passing mechanism in graph neural networks favoring high-degree nodes, the model may unintentionally amplify popularity bias, tending to recommend popular items while neglecting long-tail content. To address this issue, we have incorporated neighbor sampling and regularization strategies in the current design for preliminary mitigation. In the future, we plan to integrate diversity-aware sampling or weighting mechanisms to further balance the popularity distribution in recommendation results. Secondly, modeling user behavior may pose privacy risks, especially when it comes to social relationships or long-term behavioral sequences, where user preference patterns may be indirectly inferred or abused. So in the future, privacy protection technology should be introduced to enhance the usability and security of the model in real recommendation scenarios. In terms of fairness, recommendation models may have performance differences between different user groups, resulting in some groups receiving suboptimal recommendations. To alleviate this issue, this algorithm avoids explicit preference amplification in its design and verifies its stability on multiple datasets at different sparsity levels. In the future, we will introduce explicit fairness constraints or multi-objective optimization strategies to enhance the fairness of recommendations. Although offline experiments have been conducted on five publicly available datasets covering multiple domains and sparsity, simulating diverse real-world scenarios, offline evaluation still cannot completely replace dynamic user feedback in real environments. Therefore, the proposed method will be deployed in the real-time recommendation system for future work.

## Conclusion and future work

In this paper, we propose GR-MC to address the challenges of insufficient accuracy and robustness in recommendation tasks under data sparsity and missing neighbor interactions. First, we design a graph structure augmentation operator based on an edge dropout strategy applied solely to user nodes, which helps capture the interaction between users and items more effectively, while also aiding in combating noise and anomalous interactions. Secondly, we design a multi-scale attention embedding propagation mechanism to enhance the model's ability to capture high-order neighbor information, improving the accuracy of user and item representations. Third, we incorporate contrastive learning as an auxiliary framework into the GNN model, along with a dynamic weighting mechanism, to improve the discriminative power of the learned embeddings. Finally, benefiting from the efficient design of the multi-scale attention mechanism and contrastive learning framework, GR-MC maintains model performance while keeping computational cost within a manageable range, achieving a good balance between effectiveness and resource consumption. Extensive experiments were conducted on five datasets with different sparsity characteristics to further validate the effectiveness and robustness of the proposed algorithm in addressing data sparsity and cold-start problems compared to other baseline methods.

Similar to the effectiveness of machine learning in handling complex nonlinear problems such as unstable motion in frictional processes, user behavior prediction in recommendation systems also faces challenges involving complex relationships and dynamic features. Although this study has achieved promising experimental results, we acknowledge that there is still room for further improvement and extension. First, the current algorithm mainly focuses on modeling multi-scale interactions between users and items within the graph structure, but does not explicitly incorporate temporal or sequential information. This may limit its ability to fully capture the dynamic evolution of user interests over time. In future work, we plan to integrate time-aware modules and sequential modeling methods, combining the multi-scale attention mechanism with temporal information to improve the accuracy and personalization of recommendations. Second, as artificial intelligence continues to advance in the field of recommendation systems, intelligent and dynamic user interest modeling has become an important development trend<sup>52</sup>. Therefore, the GR-MC framework has great potential to be extended to application scenarios with higher demands for real-time and dynamic recommendations, such as social media and news recommendations. In future research, we will continue to explore the potential of

graph neural networks for complex relationship modeling, and integrate multi-dimensional information such as temporal signals to provide more practical ideas and methods for recommendation system research.

## Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Received: 1 April 2025; Accepted: 28 August 2025

Published online: 01 September 2025

## References

- Wu, L., He, X., Wang, X., Zhang, K. & Wang, M. A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation. *IEEE Trans. Knowl. Data Eng.* **35**, 4425–4445. <https://doi.org/10.1109/TKDE.2022.3145690> (2023).
- Walek, B. & Fojtik, V. A hybrid recommender system for recommending relevant movies using an expert system. *Expert Syst. Appl.* **158**, 113452. <https://doi.org/10.1016/j.eswa.2020.113452> (2020).
- Moscato, V., Picariello, A. & Sperli, G. An emotional recommender system for music. *IEEE Intell. Syst.* **36**, 57–68. <https://doi.org/10.1109/MIS.2020.3026000> (2020).
- Hu, L., Li, C., Shi, C., Yang, C. & Shao, C. Graph neural news recommendation with long-term and short-term interest modeling. *Inf. Process. Manag.* **57**, 102142. <https://doi.org/10.1016/j.ipm.2019.102142> (2020).
- Nittu, P., Coelho, J. & Madiraju, P. Improvising personalized travel recommendation system with recency effects. *Big Data Min. Anal.* **4**, 139–154. <https://doi.org/10.26599/BDMA.2020.9020026> (2021).
- Ma, D., Wang, Y., Ma, J. & Jin, Q. Sgnr: A social graph neural network based interactive recommendation scheme for e-commerce. *Tsinghua Sci. Technol.* **28**, 786–798. <https://doi.org/10.26599/TST.2022.9010050> (2023).
- Li, X., Sun, L., Ling, M. & Peng, Y. A survey of graph neural network based recommendation in social networks. *Neurocomputing* **549**, 126441. <https://doi.org/10.1016/j.neucom.2023.126441> (2023).
- Liu, J., Chen, Y., Huang, X., Li, J. & Min, G. Gnn-based long and short term preference modeling for next-location prediction. *Inf. Sci.* **629**, 1–14. <https://doi.org/10.1016/j.ins.2023.01.131> (2023).
- Nguyen, L. V., Hong, M.-S., Jung, J. J. & Sohn, B.-S. Cognitive similarity-based collaborative filtering recommendation system. *Appl. Sci.* **10**, 4183. <https://doi.org/10.3390/app10124183> (2020).
- Natarajan, S., Vairavasundaram, S., Natarajan, S. & Gandomi, A. H. Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data. *Expert Syst. Appl.* **149**, 113248. <https://doi.org/10.1016/j.eswa.2020.113248> (2020).
- Patra, B. G. et al. A content-based literature recommendation system for datasets to improve data reusability-a case study on gene expression omnibus (geo) datasets. *J. Biomed. Inform.* **104**, 103399. <https://doi.org/10.1016/j.jbi.2020.103399> (2020).
- Cai, X., Hu, Z., Zhao, P., Zhang, W. & Chen, J. A hybrid recommendation system with many-objective evolutionary algorithm. *Expert Syst. Appl.* **159**, 113648. <https://doi.org/10.1016/j.eswa.2020.113648> (2020).
- Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* <https://doi.org/10.48550/arXiv.1609.02907> (2016).
- Wu, J. et al. Self-supervised graph learning for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* 726–735. <https://doi.org/10.1145/3404835.3462862> (2021).
- Koren, Y., Bell, R. & Volinsky, C. Matrix factorization techniques for recommender systems. *Computer* **42**, 30–37. <https://doi.org/10.1109/MC.2009.263> (2009).
- Bell, R. M. & Koren, Y. Lessons from the netflix prize challenge. *ACM SIGKDD Explor. Newsl.* **9**, 75–79. <https://doi.org/10.1145/1345448.1345465> (2007).
- Nosonovsky, M. & Aglikov, A. S. Triboinformatics: Machine learning methods for frictional instabilities. *Facta Univers. Ser.: Mech. Eng.* **22**, 423–433. <https://doi.org/10.22190/FUME231208013N> (2024).
- He, X. et al. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web* 173–182. <https://doi.org/10.1145/3038912.3052569> (2017).
- Xue, H.-J., Dai, X., Zhang, J., Huang, S. & Chen, J. Deep matrix factorization models for recommender systems. *IJCAI* **17**, 3203–3209. <https://doi.org/10.24963/ijcai.2017/447> (2017).
- Kuang, M., Safa, R., Edalatpanah, S. A. & Keyser, R. S. A hybrid deep learning approach for sentiment analysis in product reviews. *Facta Univers. Ser.: Mech. Eng.* **21**, 479–500. <https://doi.org/10.22190/FUME230901038K> (2023).
- Cheng, H.-T. et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems* 7–10. <https://doi.org/10.1145/2988450.2988454> (2016).
- Perozzi, B., Al-Rfou, R. & Skiena, S. Deepwalk: online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 701–710. <https://doi.org/10.1145/2623330.2623732> (2014).
- Grover, A. & Leskovec, J. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 855–864. <https://doi.org/10.1145/2939672.2939754> (2016).
- Tang, J. et al. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web* 1067–1077. <https://doi.org/10.1145/2736277.2741093> (2015).
- Wei, Y. et al. Mmgcn: Multi-modal graph convolution network for personalized recommendation of micro-video. In *Proceedings of the 27th ACM International Conference on Multimedia* 1437–1445. <https://doi.org/10.1145/3343031.3351034> (2019).
- Chen, Y., Huang, G., Wang, Y., Huang, X. & Min, G. A graph neural network incorporating spatio-temporal information for location recommendation. *World Wide Web* **26**, 3633–3654. <https://doi.org/10.1007/s11280-023-01193-9> (2023).
- Ying, R. et al. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* 974–983. <https://doi.org/10.1145/3219819.3219890> (2018).
- 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* 165–174. <https://doi.org/10.1145/3331184.3331267> (2019).
- He, X. et al. 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* 639–648. <https://doi.org/10.1145/3397271.3401063> (2020).
- Fan, W. et al. Graph neural networks for social recommendation. In *The World Wide Web Conference* 417–426. <https://doi.org/10.1145/3308558.3313488> (2019).
- Wang, X., He, X., Cao, Y., Liu, M. & Chua, T.-S. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* 950–958. <https://doi.org/10.1145/3292500.3330989> (2019).

32. Yang, Y. et al. Debiased contrastive learning for sequential recommendation. *Proc. ACM Web Conf.* **1063–1073**, 2023. <https://doi.org/10.1145/3543507.3583361> (2023).
33. Yu, J. et al. Xsimgcl: Towards extremely simple graph contrastive learning for recommendation. *IEEE Trans. Knowl. Data Eng.* **36**, 913–926. <https://doi.org/10.1109/TKDE.2023.3288135> (2023).
34. Vaswani, A. et al. Attention is all you need. *Adv. Neural. Inf. Process. Syst.* **30**, 5998–6008. <https://doi.org/10.48550/arXiv.1706.03762> (2017).
35. Wang, X. et al. Disentangled graph collaborative filtering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* 1001–1010. <https://doi.org/10.1145/3397271.3401137> (2020).
36. Chen, H., Huang, B., Wang, X., Zhou, Y. & Zhu, W. Global-local graphformer: Towards better understanding of user intentions in sequential recommendation. In *Proceedings of the 5th ACM International Conference on Multimedia in Asia* 1–7. <https://doi.org/10.1145/3595916.3626377> (2024).
37. Adjeisah, M., Zhu, X., Xu, H. & Ayall, T. A. Towards data augmentation in graph neural network: An overview and evaluation. *Comput. Sci. Rev.* **47**, 100527. <https://doi.org/10.1016/j.cosrev.2022.100527> (2023).
38. Luo, M., Su, Z., Tang, Y. & Ding, X. Angcn: Adaptive neighborhood-awareness for recommendation. In *International Conference on Knowledge Science, Engineering and Management* 272–287. [https://doi.org/10.1007/978-981-97-5501-1\\_21](https://doi.org/10.1007/978-981-97-5501-1_21) (2024).
39. Ding, K., Xu, Z., Tong, H. & Liu, H. Data augmentation for deep graph learning: A survey. *ACM SIGKDD Expl. Newsl.* **24**, 61–77. <https://doi.org/10.1145/3575637.3575646> (2022).
40. Zamfirache, I. A., Precup, R.-E. & Petriu, E. M. Q-learning, policy iteration and actor-critic reinforcement learning combined with metaheuristic algorithms in servo system control. *Facta Univers. Ser. Mech. Eng.* **21**, 615–630. <https://doi.org/10.22190/FUME231011044Z> (2023).
41. Li, C. et al. Multi-task learning with dynamic re-weighting to achieve fairness in healthcare predictive modeling. *J. Biomed. Inform.* **143**, 104399. <https://doi.org/10.1016/j.jbi.2023.104399> (2023).
42. He, K., Fan, H., Wu, Y., Xie, S. & Girshick, R. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 9726–9735. <https://doi.org/10.1109/CVPR42600.2020.00975> (2020).
43. Wada, S. & Morimoto, N. Investigating relationship between data augmentation intensity and model performance in natural language processing. In *2024 International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan)* 445–446. <https://doi.org/10.1109/ICCE-Taiwan62264.2024.10674562> (2024).
44. Zhang, X. & Gan, M. C-gdn: Core features activated graph dual-attention network for personalized recommendation. *J. Intell. Inf. Syst.* **62**, 317–338. <https://doi.org/10.1007/s10844-023-00816-x> (2024).
45. Harper, F. M. & Konstan, J. A. The movielens datasets: History and context. *Acm Trans. Interact. Intell. Syst.* **5**, 1–19. <https://doi.org/10.1145/2827872> (2015).
46. Liang, D., Charlin, L., McInerney, J. & Blei, D. M. Modeling user exposure in recommendation. In *Proceedings of the 25th International Conference on World Wide Web* 951–961. <https://doi.org/10.1145/2872427.2883090> (2016).
47. He, R. & McAuley, J. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web* 507–517. <https://doi.org/10.1145/2872427.2883037> (2016).
48. He, R. & McAuley, J. Vbpr: Visual bayesian personalized ranking from implicit feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence* vol. 30. <https://doi.org/10.1609/aaai.v30i1.9973> (2016).
49. van den Berg, R., Kipf, T. N. & Welling, M. Graph convolutional matrix completion. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* 1–9. <https://doi.org/10.48550/arXiv.1706.02263> (2017).
50. Yun, S., Jeong, M., Kim, R., Kang, J. & Kim, H. J. Graph transformer networks. *Adv. Neural Inf. Process. Syst.* <https://doi.org/10.5555/3454287.3455360> (2019).
51. Song, J., Chang, C., Sun, F., Song, X. & Jiang, P. Ngat4rec: Neighbor-aware graph attention network for recommendation. *arXiv preprint arXiv:2010.12256* <https://doi.org/10.48550/arXiv.2010.12256> (2020).
52. He, J.-H. Transforming frontiers: The next decade of differential equations and control processes. *Adv. Differ. Equ. Control Process* **32**, 2589–2589. <https://doi.org/10.5940/adecp2589> (2025).

## Acknowledgements

This work was supported by the National Natural Science Foundation of China (Program No. 62341124), the Natural Science Basic Research Plan in Shanxi Province of China (Program No. 2017JM6068), and the Yunnan Fundamental Research Projects (Program No. 202201AT070030).

## Author contributions

Dongqi Pu and Yaming Zhang led the analysis and writing of this paper. All authors read the paper and Yaming Zhang contributed to polishing it. Die Pu and Gaoyuan Xie did the interpretation of the data. Dongqi Pu and Zhenghong Qian conceived and designed the experiment. All authors reviewed the manuscript and approved the publication of our manuscript.

## Declarations

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary Information** The online version contains supplementary material available at <https://doi.org/10.1038/s41598-025-17925-y>.

**Correspondence** and requests for materials should be addressed to Y.Z.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025