Information Fusion 121 (2025) 103167

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

PGA-DRL: Progressive graph attention-based deep reinforcement learning for recommender systems

Jawad Tanveer [a], Sang-Woong Lee [b,*], Amir Masoud Rahmani [c], Khursheed Aurangzeb [d], Mahfooz Alam [e], Gholamreza Zare [f], Pegah Malekpour Alamdari [f], Mehdi Hosseinzadeh [g,h,*]

[a] Department of Computer Science and Engineering, Sejong University, Seoul 05006, Republic of Korea
[b] Pattern Recognition and Machine Learning Lab, School of Computing, Gachon University, Seongnam 13120, Republic of Korea
[c] Future Technology Research Center, National Yunlin University of Science and Technology, Yunlin, Taiwan
[d] Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, P. O. Box 51178, Riyadh 11543, Saudi Arabia
[e] Department of Mathematics and Statistics, Faculty of Science and Technology, Vishwakarma University, Pune 411048, India
[f] DTU AI and Data Science Hub (DAIDASH), Duy Tan University, Da Nang, Vietnam
[g] School of Computer Science, Duy Tan University, Da Nang, Vietnam
[h] Jadara Research Center, Jadara University, Irbid 21110, Jordan

ARTICLE INFO

Keywords:
Recommender systems
Deep reinforcement learning
Graph convolutional networks
Graph attention networks
Progressive fusion
Actor-critic framework

ABSTRACT

Advanced graph models, including Graph Convolutional Networks (GCNs) and Graph Attention Networks (GATs), have demonstrated their effectiveness in capturing intricate user-item interactions. However, their integration into Deep Reinforcement Learning (DRL)-based Recommender Systems (RSs) remains relatively underexplored. To address this gap, we propose PGA-DRL, a Progressive Graph Attention-Based DRL model that incrementally fuses GCN and GAT representations via concatenation, effectively combining their complementary strengths to enhance feature representation within an Actor-Critic (AC) framework. This progressive integration refines both global and localized user-item interaction patterns, Specifically, global patterns capture broader user preferences across the entire graph, and localized patterns focus on specific, detailed interactions between closely connected nodes, enabling a more comprehensive understanding of the recommendation environment. We evaluate our approach using extensive experiments on multiple benchmark datasets, including ML-100K, ML-1M, Amazon Subscription Boxes, Amazon Magazine Subscriptions, and ModCloth, employing standard ranking metrics such as Precision@10, Recall@10, NDCG@10, MRR@10, and Hit@10. The experimental results reveal that PGA-DRL outperforms state-of-the-art baselines, such as BPR, NeuMF, and SimGCL, achieving improvements in NDCG@10 and Recall@10. Our core contributions lie in bridging graph-based learning with reinforcement learning through a novel, efficient, and scalable fusion mechanism that enhances recommendation accuracy and ultimately improves user satisfaction. The source code for PGA-DRL is publicly available at https://github.com/RS-Research/PGA-DRL to enhance transparency and facilitate future research.

1. Introduction

RSs are pivotal in delivering personalized content across various domains, including e-commerce [1–4], entertainment [5–9], social networks [10–12], healthcare [13–15], and digital news platforms [16–18]. By analyzing user behaviors and preferences, these systems aim to provide accurate and relevant suggestions, thereby enhancing user experience and engagement [19–22]. Over the years, RSs have evolved from traditional Collaborative Filtering (CF) [20,23] and Content-Based Filtering (CBF) [24] approaches to more advanced methods that leverage Deep Learning (DL) and graph-based models. Notably, graph-based models like GCNs and GATs have emerged as powerful tools for capturing complex relational structures within user-item interaction graphs, leading to improvements in recommendation accuracy and diversity [25–28]. These graph-based models have also found applications in various other domains [29–33].

* Corresponding authors.
E-mail addresses: Jawadtanveer@sejong.ac.kr (J. Tanveer), slee@gachon.ac.kr (S.-W. Lee), rahmania@yuntech.edu.tw (A.M. Rahmani), kaurangzeb@ksu.edu.sa (K. Aurangzeb), mahfooz.alam@vupune.ac.in (M. Alam), gholamrezazare@duytan.edu.vn (G. Zare), pegahmalekpouralamdari@duytan.edu.vn (P.M. Alamdari), mehdihosseinzadeh@duytan.edu.vn (M. Hosseinzadeh).

https://doi.org/10.1016/j.inffus.2025.103167
Received 13 October 2024; Received in revised form 25 March 2025; Accepted 27 March 2025
Available online 27 March 2025
1566-2535/

Despite these advancements, effectively integrating GCNs and GATs into DRL-based RSs remains a significant challenge [34,35]. DRL approaches, particularly those employing AC frameworks, have shown promise in optimizing long-term user satisfaction by modeling the recommendation task as a sequential decision-making problem [36,37]. However, the incorporation of graph-based models within these frameworks is still underdeveloped. Current methods often treat GCNs and GATs independently, failing to exploit their complementary strengths in capturing both global and localized user-item interactions. This oversight leads to suboptimal representations and limits the model's ability to fully comprehend the complex and dynamic nature of user preferences [38,39].

Also, hybrid and fusion methods have emerged to integrate multiple recommendation techniques, leveraging the strengths of different approaches to build more robust models [40–43]. In this approach, motivated by the need for a more adaptive and progressive integration of GCNs and GATs within DRL frameworks, we propose a novel method that addresses these limitations while maintaining computational efficiency [44,45]. Existing fusion strategies typically rely on complex attention mechanisms or weighted sums, which, although powerful, can introduce unnecessary computational complexity [34,46]. Moreover, these approaches do not leverage the progressive refinement of user-item embeddings across multiple GNN layers—a process that could significantly enhance both accuracy and scalability [47–49].

In this paper, we introduce PGA-DRL for RSs, which progressively integrates the outputs of GCNs and GATs through a straightforward concatenation approach within an AC framework. By leveraging the global structural information captured by GCNs and the localized attention provided by GATs, our model refines user-item interaction representations across multiple layers. This progressive fusion enables the capture of both coarse-grained and fine-grained interaction patterns, leading to a more comprehensive understanding of user preferences. Embedding this progressive learning process within the AC framework allows the model to continuously improve its recommendation policy, resulting in more accurate and diverse recommendations over time.

To validate the effectiveness of PGA-DRL, we conducted several experiments on multiple benchmark datasets. Also, we evaluated the model using several ranking metrics. Our results demonstrate the improvements over state-of-the-art baseline methods. These improvements underscore the advantages of progressive GCN-GAT fusion in capturing diverse user-item interactions.

The remainder of this paper is structured as follows: Section 2 reviews related work on graph-based RSs, DLR, and fusion mechanisms. In Section 3, we present the architecture of our proposed method. Section 4 discusses the experimental results and discussion. Finally, Section 5 concludes the paper.

2. Related work

RSs have transitioned from traditional methods, such as CF and CBF, to more sophisticated approaches that leverage DL and graph-based models [50,51]. These modern systems aim to capture intricate patterns in user-item interactions and deliver highly personalized recommendations [52]. Despite significant advancements, effectively integrating graph-based models, particularly GCNs and GATs, within DRL frameworks remains a considerable challenge [53,54].

2.1. GNN-Based RSs

Graph Neural Networks (GNNs), including GCNs and GATs, have emerged as powerful tools for modeling complex user-item interactions, as they can capture higher-order relationships through graph structures [55]. GCNs propagate and aggregate features uniformly from neighboring nodes, making them well-suited for capturing global structural patterns in user-item bipartite graphs [56–58]. Early extensions, such as GC-MC and PinSage, applied these principles to recommendation tasks, embedding users and items into a shared latent space to improve recommendation quality [59–62]. Recent advancements, including Neural Graph Collaborative Filtering (NGCF) and LightGCN, have refined message-passing techniques to enhance scalability and accuracy, with LightGCN simplifying the architecture for computational efficiency [63–66].

In contrast, GATs introduce attention mechanisms to assign different importance to neighbors, enabling the model to focus on localized, context-specific interactions [67–69]. This selective attention is particularly effective in sparse or noisy environments, allowing GATs to capture fine-grained interaction patterns [70,71]. However, most existing GCN- and GAT-based RSs operate in isolation, limiting their ability to leverage both global and localized patterns simultaneously. The need for efficient fusion mechanisms that integrate these complementary strengths remains a key research challenge. In our framework, 'global' refers to structural patterns learned through uniform neighborhood aggregation (e.g., user communities), while 'local' denotes context-specific interactions inferred via attention (e.g., a user's short-term preferences).

2.2. DRL-Based RSs

DRL has gained traction in RSs due to its ability to optimize long-term user satisfaction by framing recommendation tasks as sequential decision-making problems. These methods commonly employ frameworks such as the Markov Decision Processes (MDPs) to model the dynamic nature of user preferences over time [53,54]. The AC framework is widely used in DRL, where the Actor selects actions (recommendations), and the Critic evaluates their long-term impact, enabling iterative policy refinement [36].

While approaches like Deep Q-Networks (DQNs) and its variants (e.g., A3C) have shown promise, they often struggle to capture the relational structure of user-item interactions. Recent efforts, such as GCN-DQN, have integrated GNNs into DRL frameworks to combine graph-based learning with sequential decision-making. However, these methods do not effectively exploit the complementary strengths of GCNs and GATs, and their reliance on complex attention mechanisms often result in increased computational overhead [53,59]. Therefore, there is still a critical need to develop scalable and efficient fusion strategies that integrate graph-based learning into DRL frameworks.

To address these challenges and effectively harness the combined advantages of GCNs and GATs within DRL, we propose a novel framework, Progressive Graph Attention-based DRL (PGA-DRL). This framework leverages progressive refinement through a specific concatenation-based fusion, enabling the capture of both global and localized interaction patterns while maintaining computational efficiency and scalability. Specifically, PGA-DRL iteratively updates user and item embeddings by concatenating the outputs of GCN and GAT layers at each step. This process allows the model to progressively integrate global structural information with context-specific interactions. Compared to existing methods, this approach provides a more concise and effective fusion strategy, setting a new benchmark for fusion-based RSs and offering an adaptable solution to the identified challenges.

Recent studies increasingly highlight the importance of robustness in RSs, particularly when dealing with heterogeneous multi-behavior interaction data. For instance, Chen, Ma, Zhang, Wang, He, Wang, Liu and Ma [72] proposed Graph Heterogeneous Collaborative Filtering (GHCF), which explicitly leverages heterogeneous user-item interactions through GCNs. Their approach providing robust performance, particularly for cold-start users, by capturing high-hop structural information. Yang, Huang, Xia, Liang, Yu and Li [73] developed the Multi-Behavior Hypergraph-enhanced Transformer (MBHT), which robustly captures both short-term and long-term dynamic dependencies across diverse user behaviors. MBHT significantly improves recommendation quality under varying data distributions and interaction sparsity scenarios. Additionally, Xu, Wang, Wu, Song, Zheng, Wang,

Wang, Zhou and Gai [74] introduced a robust Multi-behavior Self--supervised Learning (MBSSL) framework employing adaptive optimization to address the data sparsity and noisy interactions typical in real-world scenarios. This method demonstrates consistent robustness against sparse and noisy interaction data.

3. Methodology

In this section, we present the architecture of the proposed PGA-DRL method. The design builds on the AC framework, augmented by a progressive fusion of GCNs and GATs. This hybrid approach enables the model to simultaneously capture global and localized interaction patterns in the user-item graph. Additionally, the AC framework allows the system to optimize long-term user satisfaction by learning from dynamic user preferences. Fig. 1 provides an overview of the architecture and workflow of AC-based recommendation methods.

3.1. Overview of the PGA-DRL architecture

The proposed PGA-DRL model addresses key limitations in existing RSs by integrating GCNs and GATs within an AC framework. Specifically, it tackles the challenges of capturing global and local interaction patterns simultaneously. To achieve this, the model progressively refines user-item embeddings through a layer-wise fusion process that combines the strengths of GCNs and GATs.

GCNs aggregate structural information uniformly from neighboring nodes, emphasizing global patterns across the interaction graph. In contrast, GATs dynamically assign importance to different neighbors, focusing on contextually relevant interactions. These outputs are incrementally combined at each layer by concatenating the embeddings from both networks and adaptively weighting them using learnable parameters. During training, these parameters are optimized through backpropagation, ensuring that the fusion process prioritizes the most relevant features based on the recommendation scenario. This progressive refinement generates embeddings that balance global context and local precision, offering a holistic understanding of user preferences and item characteristics.

3.1.1. Necessity of the AC framework

The AC framework plays a critical role in optimizing long-term user satisfaction by separating the decision-making process into two components: the Actor and the Critic. This separation allows for a dynamic approach to recommendation. Specifically, the Actor selects recommendations based on the current state of user-item embeddings, while the Critic evaluates these recommendations by estimating their long-term impact on user satisfaction. Consequently, this structure enables dynamic policy refinement, allowing the model to effectively balance exploration (discovering new items) and exploitation (recommending familiar preferences)—a fundamental requirement for sequential decision-making in RSs.

Achieving this balance between exploration and exploitation presents a significant challenge in RSs, largely due to the evolving nature of user preferences. Users' interests are not static and can shift over time, necessitating a system that can adapt without becoming overly specialized or predictable. To illustrate, consider an e-commerce scenario. Exploration might involve recommending novel products based on emerging trends, perhaps identified through recent surges in searches or purchases. Simultaneously, exploitation ensures the system remains aligned with the user's established preferences, such as frequently purchased categories. By leveraging this dual mechanism, the AC framework can dynamically adapt to both individual user behavior and broader market trends, ultimately ensuring sustained user engagement and satisfaction.

3.1.2. Integration of GCNs and GATs within the AC framework

The integration of GCNs and GATs within the AC framework is a distinguishing feature of PGA-DRL. GCNs aggregate uniform information from neighboring nodes, capturing the broader structure of user-item interaction graphs, while GATs selectively focus on the most relevant neighbors through attention mechanisms. By combining these outputs through a layer-wise fusion process, the model ensures that both global and local patterns are effectively represented.

This fusion process involves concatenating the embeddings generated by GCNs and GATs at each layer, followed by the application of adaptive weights that are optimized during training. These weights allow the model to dynamically prioritize either global patterns, captured by GCNs, or localized interactions, captured by GATs, depending on the underlying data characteristics. For example, in scenarios with sparse user-item interactions, the model might emphasize GAT embeddings to effectively capture nuanced relationships. Conversely, in dense graphs, GCN embeddings may play a more significant role in the proposed fusion approach. This adaptive approach results in contextually relevant representations, thereby empowering the Actor to make better-informed recommendations while the Critic to accurately evaluate their effectiveness. An overview of the PGA-DRL method is provided on Fig. 2.

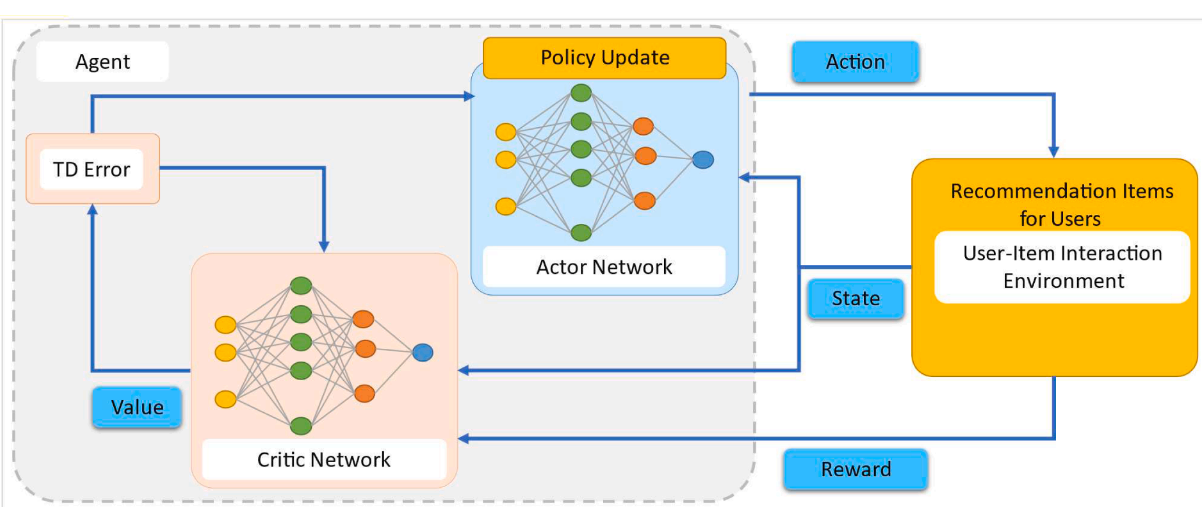

Fig. 1. A general diagram of AC-based RS.

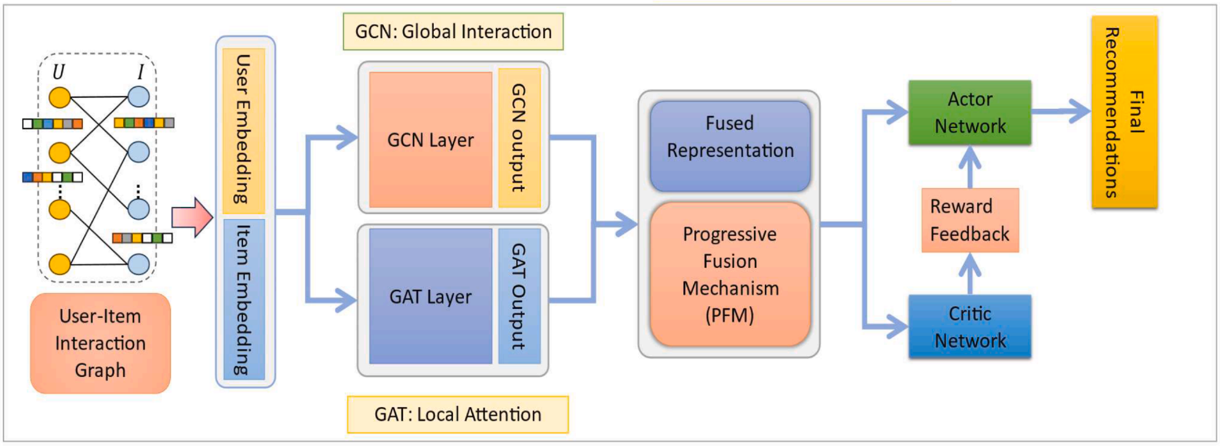

Fig. 2. An overview of the proposed method architecture.

3.1.3. Empirical evidence supporting the AC framework

The effectiveness of the AC framework is supported by extensive empirical evaluations. Specifically, experimental results demonstrate that incorporating the AC framework leads to significant improvements in key recommendation metrics, such as NDCG@10 and Recall@10, across a range of diverse benchmark datasets, including ML-1M and Amazon Subscription Boxes. These datasets were strategically chosen for their contrasting characteristics: ML-1M represents dense user-item interactions with high coverage, providing a benchmark for evaluating global recommendation patterns. Amazon Subscription Boxes, on the other hand, exemplifies sparse data environments, showcasing the model's capability to effectively handle scenarios with limited interactions.

The strong performance of PGA-DRL on these datasets highlights its ability to dynamically adapt to varying interaction densities and prioritize the most relevant patterns. Further supporting the necessity of the AC framework, ablation studies reveal that its removal results in a substantial decline in performance. This finding highlights the critical role of the AC framework in enhancing overall recommendation quality and optimizing long-term user engagement. Consequently, this evidence firmly confirms that the synergistic combination of GCN-GAT fusion with the AC framework is crucial for achieving optimal results in our proposed model.

3.1.4. Real-world applicability and scalability

By effectively integrating the complementary strengths of GCNs and GATs within the AC framework, PGA-DRL consistently improves its ability to capture intricate user-item interactions. This improved capture directly translates into recommendations that are accurate, diverse, and scalable. Specifically, accuracy ensures that the recommendations closely align with individual user preferences. Furthermore, diversity introduces novel and varied suggestions, which is crucial for enhancing user engagement and discovery. Lastly, scalability guarantees efficient performance even when dealing with large datasets and ensures the model's adaptability to various real-world application scenarios.

In this context, scalability encompasses the model's computational efficiency, its inherent capacity to process large-scale datasets seamlessly, and its robustness in maintaining high performance across varying degrees of data sparsity and interaction density. These key attributes render PGA-DRL particularly well-suited for deployment in real-world applications, such as large-scale e-commerce platforms and content recommendation systems, where both computational efficiency and adaptability to diverse data conditions are paramount to achieving success.

3.2. AC framework

At the heart of the PGA-DRL model lies the AC framework, a widely adopted paradigm in RL for tackling sequential decision-making problems. This framework strategically divides the decision-making process into two distinct yet interconnected components: the Actor and the Critic. The Actor is primarily responsible for selecting actions—in our case, recommending items—based on the current state of the user-item environment. Simultaneously, the Critic evaluates the favorability of these actions by estimating their long-term impact, thereby providing crucial feedback that guides the Actor in refining its recommendation policy over time.

The recommender task is modeled as an MDP, where the state s_t at time t represents the current user-item interaction embeddings, and the action a_t refers to the selection of an item to recommend. Upon recommendation, a reward r_t, such as a user click or purchase, provides immediate feedback on the quality of the chosen item. The long-term objective of the Actor is to learn a policy that maximizes the cumulative future reward, which is mathematically expressed as:

$$J(\theta) = \mathbb{E}_{t=0}^{T}[\gamma^t r_t | \pi_\theta] \tag{1}$$

where $\gamma \in [0, 1]$ denotes the discount factor, and θ represents the parameters of the Actor's policy π_θ. By optimizing this objective function, the Actor learns to generate recommendations that not only drive immediate user engagement but also foster sustained long-term user satisfaction.

Complementing the Actor, the Critic network approximates the Q-value $Q(s_t, a_t)$ for each state-action pair. This Q-value represents the expected cumulative future reward of selecting a specific action a_t in a given state s_t. The Critic's training objective is to minimize the temporal difference (Bellman) error, formalized as:

$$L(\omega) = (Q_\omega(s_t, a_t) - (r_t + \gamma Q_\omega(s_{t+1}, \pi_\theta(s_{t+1}))))^2 \tag{2}$$

where ω are the parameters of the Critic network. This separation of responsibilities allows the model to iteratively refine its recommendations by effectively learning from user feedback across both immediate interactions (through the reward signal) and anticipated future outcomes (through the Q-value estimation).

3.3. Graph-based user-item interaction modeling

The effectiveness of any RS depends heavily on its ability to accurately model the complex interplay between users and items. In PGA-DRL, this crucial aspect is addressed through the synergistic integration of GCNs and GATs, which play complementary roles in capturing different aspects of these interactions.

GCNs operate by aggregating feature information from neighboring nodes within the user-item interaction graph, thereby enabling each node to progressively incorporate information from its wider network neighborhood. This mechanism allows GCNs to effectively capture global interaction patterns, which are particularly vital for the CF tasks where users exhibiting similar historical behavior are likely to share preferences. The future update rule for each layer of the GCN can be expressed as:

$$H^{(l+1)} = \sigma\left(D^{-1/2}AD^{-1/2}H^{(l)}W^{(l)}\right) \tag{3}$$

where A is the adjacency matrix representing the user-item interactions, D is the degree matrix, $H^{(l)}$ are the node embeddings at layer l, and $W^{(l)}$ are the learnable weight parameters. By stacking multiple GCN layers, the model gains the capacity to capture higher-order relationships that span several connections (hops) within the graph structure.

In contrast, GATs introduce a layer of attention mechanisms, which empower the model to focus on the most relevant neighboring nodes during the aggregation process. Unlike GCNs, which treat all neighbors with equal importance, GATs assign varying attention coefficients to each neighbor based on their contribution to the representation of the target node. This selective attention mechanism proves particularly beneficial in noisy data environments where certain interactions might be less indicative of genuine user preferences. The attention coefficient α_{ij} between node i and neighbor j is computed as follows:

$$\alpha_{ij} = \frac{\exp\left(LeakyReLU\left(a^T\left[Wh_i \| Wh_j\right]\right)\right)}{\sum_{k\in N(i)}\exp\left(LeakyReLU\left(a^T\left[Wh_i \| Wh_j\right]\right)\right)} \tag{4}$$

where W is the learnable weight matrix, a is the attention vector, and $N(i)$ represents the neighbors of node i. GATs excel at modeling localized, fine-grained interaction patterns, which are critical for capturing context-dependent nuances in user preferences.

3.4. Progressive fusion mechanism (PFM)

A core innovation of the PGA-DRL model lies in its Progressive Fusion Mechanism (PFM), which enables a layer-wise integration of GCN and GAT embeddings. Rather than treating GCNs and GATs as independent modules, the model progressively fuses their outputs at each layer. This strategic approach allows the system to effectively combine the global interaction patterns captured by GCNs with the fine-grained local interaction details discerned by GATs, leading to richer and more informative embeddings.

Specifically, at each layer l, the embeddings generated by the GCN and GAT layers are concatenated to form a unified representation:

$$H_{\mathrm{GCN}}^{(l+1)} = \sigma\left(D^{-1/2}AD^{-1/2}H^{(l)}W_{\mathrm{GCN}}^{(l)}\right) \tag{5}$$

$$H_{\mathrm{GAT}}^{(l+1)} = \sigma\left(\sum_{j\in N(i)} \alpha_{ij} W_{\mathrm{GAT}} h_j\right) \tag{6}$$

These embeddings are concatenated and combined with learnable layer weights w_l that control the importance of each layer's output:

$$H^{(l+1)} = H_{\mathrm{GCN}}^{(l+1)} \| H_{\mathrm{GAT}}^{(l+1)} \tag{7}$$

$$H_{\mathrm{fused}} = \sum_{l=0}^{L} w_l H^{(l)} \tag{8}$$

The fusion weights (w_l) are learnable parameters that adaptively determine the contribution of GCN (global) and GAT (local) at each layer. This allows PGA-DRL to prioritize GAT in sparse/noisy scenarios (where attention is critical) and GCN in dense graphs (where uniform aggregation stabilizes learning).

The synergistic combination of GCN and GAT within this progressive framework ensures that both broad global structural patterns and localized, context-specific interactions are effectively modeled. GCNs provide a consistent and uniform aggregation of neighbor information, which is essential for maintaining the global context of the interaction graph. Complementarily, GATs dynamically assign attention weights to neighbors, enabling the model to focus on the most relevant interactions. Notably, even when GATs assign relatively uniform weights, the GCN component ensures a balanced feature aggregation, which is particularly beneficial in dense graph structures. This inherent synergy significantly enhances the model's ability to generalize effectively across datasets exhibiting varying levels of sparsity.

This layer-wise adaptive weighting allows the model to effectively weigh the contributions of different layers, ensuring that both high-level structural signals and fine-grained localized attention patterns are captured. Consequently, the PFM significantly contributes to the model's ability to generate high-quality recommendations by leveraging a diverse range of interaction signals.

The computational complexity of GCN and GAT layers differs significantly, where GCN operates with $O\left(Nd^2\right)$ complexity, leveraging efficient neighborhood aggregation. In contrast, GAT introduces an attention mechanism that scales with $O\left(Nd^2 + N^2d\right)$, where N is the number of nodes and d is the embedding dimension. The progressive fusion mechanism employed in PGA-DRL maintains efficiency by adaptively weighting these embeddings while avoiding the computational overhead of full pairwise attention computations across all layers. This design choice ensures a well-considered balance between the quality of the learned representations and the computational feasibility of the model.

3.5. Critic network and loss function

The Critic Network in PGA-DRL is specifically designed to evaluate the quality of the Actor's recommendations by estimating the expected future reward of each state-action pair. To achieve this, it employs a Multi-Layer Perceptron (MLP) that processes the concatenated embeddings of a user u and an item i, ultimately outputting a scalar Q-value, as shown below:

$$Q_\omega(s_t, a_t) = \mathrm{MLP}(h_u \| h_i) \tag{9}$$

where h_u and h_i represent the embeddings of user u and item i, respectively, obtained from the PFM.

The overall loss function for training PGA-DRL is a composite one, combining the Bayesian Personalized Ranking (BPR) loss, used to optimize the Actor's policy, and the Mean Squared Error (MSE) loss, used to train the Critic. This combined loss function is formulated as:

$$\mathscr{L}_{\mathrm{Actor}} = \mathrm{BPR}\left(s_{u,i} - s_{u,j}\right) \tag{10}$$

$$\mathscr{L}_{\mathrm{Critic}} = \frac{1}{2}\left(Q_\omega(s_t, a_t) - r_t\right)^2 \tag{11}$$

where $s_{u,i}$ and $s_{u,j}$ are the scores for positive and negative item interactions, respectively. This dual-objective training ensures that the Actor network progressively refines its recommendation policy based on user feedback, while the Critic network simultaneously learns to accurately assess the quality of the generated recommendations.

Overall, the progressive fusion of global and local interaction patterns through GCNs and GATs, combined with the RL optimization process and this composite loss function, equips PGA-DRL with a robust ability to generate more informed, personalized, and high-quality recommendations.

3.6. PGA-DRL algorithm

Algorithm 1 shows the step-by-step procedure of the PGA-DRL

Algorithm 1
PGA-DRL: progressive graph attention-based DRL for RSs.

Input: Dataset D with user-item interactions, configuration parameters, learning rate α, discount factor γ, number of layers L, and number of training iterations.
Output: Trained Actor and Critic networks, user-item embeddings.

1: Initialize:
2: Initialize user and item embeddings
3: Initialize GCN and GAT layers
4: Initialize Actor and Critic networks with random weights
5: Initialize layer weights for progressive fusion
6: Construct normalized adjacency matrix from the interaction graph
7: for each iteration do:
8: Sample a mini-batch of user-item interactions (u, i, j) from the dataset D
9: Forward Propagation through GCN layers:
10: Set initial embeddings $H^{(0)}$
11: for $l = 1$ to L do:
12: Update GCN embeddings using the normalized adjacency matrix and weight matrices
13: End for
14: Forward Propagation through GAT layers:
15: for $l = 1$ to L do:
16: Compute attention coefficients and update GAT embeddings
17: End for
19: Progressive Fusion of GCN and GAT Embeddings:
20: Concatenate GCN and GAT embeddings at each layer
21: Apply *softmax* to layer weights and combine embeddings
22: Split fused embeddings into user and item embeddings
23: Actor Network Update:
24: Compute scores for positive and negative items
25: Calculate BPR loss for Actor network
26: Critic Network Update:
27: Concatenate user and item embeddings, and pass through Critic MLP
28: Compute reward and critic loss
29: Gradient Update:
30: Backpropagate the total loss and update the parameters of the Actor and Critic networks
31: End for
32: Return trained Actor and Critic networks, user and item embeddings

method. This algorithm progressively refines user-item interaction representations by leveraging both GCNs and GATs, integrated into an AC framework. The PFM captures both global and local interaction patterns at each layer, enabling the model to make high-quality, personalized recommendations.

The pseudocode for PGA-DRL begins with the initialization of user and item embeddings, the parameters of the GCN and GAT layers, and the AC networks (lines 1–6). Prior to training, the adjacency matrix, which encodes the user-item interaction graph, is preprocessed to normalize the graph structure. This normalization ensures a balanced aggregation of information from neighboring nodes during the subsequent graph operations.

In each training iteration, the algorithm begins by sampling a mini-batch of user-item interactions (line 8). This mini-batch typically includes a user, an item they have positively interacted with, and a negatively sampled item. The model then processes the embeddings of these entities through the GCN layers (lines 9–13). This step is crucial for capturing the global structural information inherent in the interaction graph. Following this, the embeddings are passed through the GAT layers (lines 14–17), where attention mechanisms are applied to selectively focus on the most relevant neighboring nodes for each user and item.

The PFM, outlined in lines 19–22, then takes the embeddings generated by both the GCN and GAT layers and concatenates them. This step ensures that both the global and local interaction patterns learned by the respective graph networks are preserved in the fused representation. Subsequently, the learned layer weights are applied to adaptively weigh the contribution of each layer's fused output, leading to more informative and context-aware embeddings. Finally, these fused embeddings are split back into separate user and item representations for the downstream tasks.

The Actor network (lines 23–25) uses the BPR loss to optimize the recommendation policy by learning to rank positively interacted items higher than negative ones. Meanwhile, the Critic network (lines 26–28) evaluates the quality of the recommendations using an MLP and estimates the reward for each state-action pair. The total loss, comprising the BPR loss and the Critic's loss, is then backpropagated through the network to update the model parameters (lines 29–30), facilitating learning.

Through this iterative process of refining user-item embeddings and improving the recommendation policy, PGA-DRL effectively learns to capture complex user-item interaction patterns and ultimately generates high-quality recommendations.

3.7. Time complexity analysis

To provide a clear understanding of PGA-DRL's efficiency, we can analyze its computational complexity by examining its main components: the graph convolution and attention layers, and the actor-critic networks. We will also briefly compare this complexity with that of baseline models to provide context. As expected, the graph-based operations, namely GCN and GAT, constitute the primary contributors to the runtime. While the actor-critic stage does introduce some additional overhead, the overall computational cost of PGA-DRL scales roughly linearly with the size of the input graph, considering the number of users, items, and their interactions. This scaling behavior is similar to that observed in standard GCN-based recommender systems, with the attention mechanism in GAT contributing a modest constant-factor increase to the overall complexity.

- **GCN:** Each GCN layer performs neighborhood aggregation across the user-item interaction graph. This involves sparse matrix operations (propagating embeddings along edges) and a linear transformation of node embeddings. The cost per layer scales about linearly with the number of edges (for propagation) plus a smaller cost proportional to the number of nodes (for applying the weight matrix to each embedding). Stacking multiple GCN layers (depth L) multiplies this cost by L, but the overall complexity remains $O(L \cdot |E|)$ in big-O notation, where $|E|$ is the number of interactions (edges). In practice, this means the GCN part can efficiently handle large graphs using optimized sparse operations.

- **GAT:** GAT layers introduce an additional step to compute attention coefficients for each edge. For every node, the model calculates the importance of each neighbor by computing a small attention score (e.g. a dot-product between embeddings) for each connecting edge. Using H attention heads multiplies the operations per edge by H, but these computations are independent per head and can be executed in parallel. As a result, the GAT complexity also scales linearly with the number of edges (each edge's embedding is processed with some extra calculations for attention). The attention mechanism does add a constant-factor overhead – effectively each edge update is a bit more expensive than in a plain GCN – but this overhead is manageable. In summary, the combined graph operations (GCN + GAT) still run in $O(\textit{number of edges})$ per layer, with the GAT's attention making the constant factor larger than models without attention.
- **PFM:** After each layer, PGA-DRL fuses the GCN and GAT outputs by concatenating their embeddings and applying learnable weights. This fusion step has a cost proportional to the number of nodes and the embedding dimension (since we process each node's two embedding vectors). Compared to the graph propagation, this is a minor cost. Even across L layers, the fusion overhead grows as $O(L{\cdot}N)$ (with N nodes), which is typically much smaller than the edge-based costs when the graph is sparse. Thus, PFM adds minimal overhead and does not affect the overall complexity order.
- **Actor-Critic Networks:** In addition to the graph layers, PGA-DRL includes an Actor network and a Critic network (both implemented as MLPs) to learn the recommendation policy. The complexity of forwarding and backpropagating through these networks depends on the network size and the batch size. For a given mini-batch of B user-item interactions, the actor or critic MLP might require on the order of $O\left(B{\cdot}d^2\right)$ operations (if d is the embedding size and the MLP hidden layers are of similar order), which is typical for dense layers. However, this cost is relatively small compared to the graph convolutional part. The actor-critic networks are applied to batches of sampled data rather than the entire graph at once, and B is usually much smaller than N or $|E|$. Therefore, the actor-critic stage contributes only a modest additional cost per training step and does not bottleneck the overall training time.

To provide a clearer perspective on the complexity of our proposed method, we compare it with baseline GCN-based recommenders, specifically LightGCN and NGCF.

- **LightGCN:** This method is a simplified GCN model that omits feature transformation and non-linearities, focusing purely on neighbor aggregation. As a result, it is extremely efficient, roughly scaling linearly with the number of edges and embedding dimension. Each layer of LightGCN essentially performs an embedding averaging over edges, which is similar to the propagation step of PGA-DRL's GCN (but without the extra weight multiplication). This gives LightGCN a lower per-layer constant cost – it requires fewer computations per edge – making it one of the most scalable GCN-based recommenders. In terms of complexity class, LightGCN is $O(L{\cdot}|E|)$ per training iteration (for L layers), with a small constant factor, meaning it handles large graphs with ease.
- **NGCF:** This method is an earlier GCN-based model that includes feature transformation matrices and non-linear activation at each layer. Its complexity per layer is also linear in the number of edges, but with a larger constant factor than LightGCN. NGCF must multiply each node's embedding by a weight matrix and combine it with neighbor information, resulting in an approximate cost on the order of $O(|E|{\cdot}\mathrm{d})$ or $O\left(|E|{\cdot}\mathrm{d}^2\right)$ per layer (depending on implementation details and whether dimensions change between layers). In practice, NGCF is observed to run slower than LightGCN (for example, taking roughly 4 × the training time of a matrix factorization baseline in reported experiments) due to these extra computations. Still, NGCF remains within a linear complexity regime relative to the graph size, and it was designed to be workable on reasonably large datasets.

Our proposed model inherits the graph propagation costs of NGCF (through its GCN layers with transformations) and adds the GAT layer computations on top. Therefore, PGA-DRL has a somewhat higher computational cost per layer compared to both LightGCN and NGCF. The attention mechanism means we do more work for each edge (computing and applying attention weights), and maintaining two parallel graph neural networks (GCN + GAT) doubles up some operations. However, it's important to note that the overall scaling behavior is the same: PGA-DRL's time complexity still grows linearly with $|E|$ and L, just with larger constants. In other words, while LightGCN is considered very fast and NGCF moderately fast, PGA-DRL is slightly slower per epoch than NGCF. However, all three methods exhibit similar computational complexity, as they scale linearly with the number of interactions. The added complexity in PGA-DRL is the trade-off for incorporating attention and the actor-critic framework, which aim to boost recommendation accuracy.

Also, the linear scaling of PGA-DRL with respect to the number of edges implies that it can handle large-scale recommender datasets (with millions of users and items and tens of millions of interactions) so long as we have adequate computational resources. The most expensive operations – the sparse neighbor aggregations and attention computations – can be efficiently implemented using parallel processing on GPUs or distributed systems. In practice, techniques like minibatch training and sampling (processing only a subset of nodes or edges at each step) further ensure that training on large graphs is feasible. The additional overhead from GAT layers and the actor-critic module, while making each iteration a bit slower than simpler models, does not fundamentally hinder scalability. We can still train on large graphs within reasonable time, and the model benefits from the richer computations by potentially achieving better performance. In summary, PGA-DRL remains scalable to large real-world datasets: it operates in roughly linear time complexity like its GCN-based predecessors, and with modern hardware optimizations the extra computations (attention and RL components) are a worthwhile cost for the gains in recommendation quality.

4. Experimental results and discussion

In this section, we present and analyze the experimental results obtained by our proposed PGA-DRL model in comparison to several state-of-the-art baselines. The performance was evaluated across five benchmark datasets: ML-100K, ML-1M, Amazon Subscription Boxes (AmzSB), Amazon Magazine Subscriptions (AmzMS), and Mod Cloth. Table 1 shows their data that selected from different domains. These datasets were chosen to represent a diverse range of domains and data characteristics, including varying levels of sparsity, user-item interactions, and dataset sizes. For instance, ML-100K and ML-1M provide dense user-item interaction data, while Amazon Subscription Boxes and Amazon Magazine Subscriptions represent sparser datasets with fewer interactions. Mod Cloth offers a mid-level sparsity, providing a balance between dense and sparse characteristics. This diversity ensures a

Table 1
Overview of selected datasets for the model's evaluation.

Dataset	#Users	#Items	#Interaction	Sparsity
ML-100K	943	1682	100,000	93.70 %
ML-1M	6040	3,52	1,00,209	95.81 %
Amazon Subscription Boxes	15,237	641	15,953	99.84 %
Amazon Magazine Subscriptions	60,144	3391	70,922	99.97 %
Mod Cloth	47,958	1378	82,790	99.87 %

comprehensive evaluation of the model's effectiveness across different data conditions. For each dataset, we randomly split the data into training, validation, and test sets, maintaining an 80/10/10 split ratio. The datasets used in this study are publicly available and can be accessed at the RecBole repository: https://recbole.io/dataset_list.html, ensuring the reproducibility of our experimental results [75,76].

The performance of the PGA-DRL model is evaluated using the following top-*k* recommendation metrics:

- **Precision@10:** Captures the proportion of recommended items that are relevant. Higher precision values indicate more accurate recommendations.
- **Recall@10:** Measures the fraction of relevant items that are successfully recommended in the top-*k* list. Higher recall values indicate better coverage of relevant items.
- **NDCG@10 (Normalized Discounted Cumulative Gain):** Focuses on both the relevance and the ranking position of recommended items. NDCG rewards methods that rank highly relevant items closer to the top of the recommendation list.
- **MRR@10 (Mean Reciprocal Rank):** Reflects the ranking quality of the first relevant item in the recommendation list. A higher MRR implies that relevant items appear earlier in the ranked list.
- **Hit@10:** Measures the percentage of users for whom at least one relevant item appears in the top-*k* recommendations. A higher hit rate suggests better recommendation coverage.

These metrics complement each other, offering insights into both the ranking quality and the overall recommendation accuracy.

To ensure transparency and reproducibility, we provide the key hyperparameter settings used in the implementation of the PGA-DRL model. Table 2 summarizes these settings for reference.

4.1. Baseline methods

To comprehensively evaluate the effectiveness of PGA-DRL, we compare its performance against several state-of-the-art recommendation models:

- **BPR (Bayesian Personalized Ranking):** A pairwise ranking model optimized for implicit feedback datasets [77].

Table 2
Hyperparameter settings for PGA-DRL.

Hyperparameter	Value	Description
Embedding Size	128	Dimensionality of user and item embeddings.
Number of Layers	8	Number of message-passing layers for GCN and GAT.
Learning Rate	0.001	Learning rate for training Actor, Critic, and overall model.
Dropout Rate	0.3	Dropout probability for layers to prevent overfitting.
Edge Dropout Rate	0.3	Probability of dropping edges in the graph during training.
Batch Size	8192	Number of samples per batch for training and evaluation.
Regularization Weight	1×10^{-5}	Weight for L2 regularization to control overfitting.
Discount Factor (γ)	0.99	Discount factor used in the Actor-Critic framework for long-term reward.
Number of Heads	8	Number of attention heads in GAT layers.
Loss Function	BPR	Bayesian Personalized Ranking for optimizing recommendations.
Optimizer	Adam	Optimizer used for training the model.
Evaluation Metrics	Recall, NDCG, MRR	Metrics used for evaluating the top-k recommendations.
Early Stopping	10 epochs	Stop training if validation performance does not improve for 10 epochs.

- **NeuMF (Neural Collaborative Filtering):** A DL-based model that combines matrix factorization and neural networks to learn user-item interactions [78].
- **NGCF (Neural Graph Collaborative Filtering):** A graph-based model that propagates user-item interactions through graph convolutions [66].
- **LightGCN:** A simplified graph convolutional model that focuses on the propagation of embeddings without non-linear transformations [65].
- **SGL (Self-supervised Graph Learning):** A contrastive learning-based graph model that improves embedding robustness through self-supervision [79,80].
- **SimGCL:** A similarity-based graph contrastive learning model designed to improve the robustness of collaborative filtering methods [67].
- **AutoCF:** By utilizing a graph autoencoder, this model reconstructs obscured user-item interactions to expand training data. This augmentation process strengthens the model's ability to learn more informative representations, improving recommendation accuracy [81].
- **LightGCL:** This model enhances collaborative filtering by incorporating singular value decomposition, refining structural relationships within the data. This method facilitates better representation learning by capturing essential patterns in user-item interactions [82].
- **GFormer:** This model enhances recommendation performance by leveraging a graph autoencoder to reconstruct masked user-item interactions, thereby generating augmented training data. This approach enables the model to refine user and item embeddings more effectively [83].
- **AMLDM:** The Attention-based Multilayer Linear Diffusion Model (AMLDM) addresses challenges in recommendation systems, such as incomplete interaction data and preference biases. It introduces Gaussian noise into user interaction histories, then uses attention-based layers to restore the data, ensuring it aligns with the original distribution. This method improves robustness and better captures users' true preferences, outperforming benchmark models in performance [84].
- **MGDCF:** This approach integrates GNN-based collaborative filtering with Network Representation Learning (NRL) through a Markov process. Instead of training GNNs in a traditional manner, it uses them as a static process to generate contextualized features. These features are then combined with an NRL model via a fully connected layer, emphasizing the role of ranking loss in optimizing collaborative filtering performance [85].
- **XSimGCL:** Unlike conventional graph augmentation techniques, this model employs a noise-based embedding enhancement strategy within a contrastive learning framework. This unique approach promotes a more uniform distribution of user and item representations, which in turn effectively reduces popularity bias and increases the exposure of users to less frequently interacted items [86].

4.2. Statistical significance testing

To evaluate our model, we conducted multiple independent experimental runs (specifically, 10 independent trials with different random seeds) using randomized splits of 80 % training, 10 % validation, and 10 % testing sets. We report average performance metrics (Precision@10, Recall@10, NDCG@10, MRR@10, and Hit@10) across these runs. Furthermore, we performed paired two-tailed *t*-tests to assess whether observed improvements of our proposed PGA-DRL model over baseline methods are statistically significant. Significant differences at $p < 0.05$ and $p < 0.01$ are clearly marked with * and **, respectively, in our results tables. This statistical analysis enhances confidence in our findings, clearly indicating that the observed improvements are robust and not due to random variations in data partitioning or model initialization.

4.3. Results and analysis

This section offers a detailed evaluation of PGA-DRL's performance across the benchmark datasets, comparing its results against several well-established recommendation models.

The Precision@10 metric, which reflects the proportion of recommended items that are relevant, provides insight into the effectiveness of each model in identifying pertinent recommendations. As seen in Table 3, PGA-DRL consistently outperforms the other models on the ML-100K and ML-1M datasets, achieving Precision@10 scores of 0.2069 and 0.2155, respectively. These results highlight PGA-DRL's ability to effectively rank and recommend relevant items, particularly in datasets with dense user-item interactions.

In comparison, models such as NeuMF and NGCF show slightly lower precision values on these dense datasets, underscoring PGA-DRL's strength in ranking relevant items at the top of the recommendation list. However, when considering Amazon Subscription Boxes, a more sparse dataset, the performance of PGA-DRL (0.0145) drops notably compared to NeuMF (0.0248), which performs better in handling sparse interactions. This difference in performance highlights the impact of dataset sparsity on the effectiveness of different recommendation approaches, suggesting that PGA-DRL excels in settings with richer interaction data but can be affected by the limitations inherent in sparse datasets.

The Recall@10 metric, which measures the fraction of relevant items that appear in the top 10 recommendations, provides further insights into the ability of each model to capture a broad range of user preferences. As shown in Table 4, PGA-DRL achieves the highest Recall@10 score on ML-100K, with 0.2570, outperforming all other models. This demonstrates the model's ability to recommend a diverse set of relevant items, enhancing the breadth of coverage.

On the Amazon Subscription Boxes dataset, PGA-DRL yields a lower recall (0.1451) compared to NeuMF (0.2483). This difference can be attributed to NeuMF's strengths in utilizing latent factor models, which are well-suited to sparse data. In contrast, PGA-DRL's performance reflects the challenge of capturing relevant items in settings with fewer interactions. Nonetheless, even in these conditions, PGA-DRL continues to perform reasonably well, suggesting that its hybrid approach provides solid recall capabilities, even when data is sparse.

The NDCG@10 metric, which emphasizes both relevance and ranking position, provides a more nuanced view of how effectively the models prioritize highly relevant items. As shown in Table 5, PGA-DRL achieves the highest NDCG scores on both the ML-100K and ML-1M datasets, with values of 0.3056 and 0.2774, respectively. These results suggest that PGA-DRL excels not only in recommending relevant items but also in positioning them toward the top of the ranked list, enhancing the user experience by ensuring that the most relevant recommendations are presented first.

Conversely, models like AutoCF and GFormer show lower NDCG scores, particularly on ML-100K, indicating that while they perform adequately in retrieving relevant items, their ranking quality is not as refined. This observation further underscores the advantage of PGA-DRL's progressive fusion mechanism, which uniquely integrates the strengths of both GCNs and GATs. By doing so, PGA-DRL effectively balances the capture of global structural patterns with the nuanced, localized insights provided by its attention mechanisms, ultimately leading to a more refined ranking quality.

On the Amazon Subscription Boxes, PGA-DRL achieves an NDCG@10 score of 0.1384, which is relatively high compared to some other models. However, it still falls short of the performance observed on denser datasets, with NeuMF outperforming in this case. This highlights that while PGA-DRL is effective across a range of datasets, its ranking quality can be influenced by the degree of data sparsity.

The MRR@10 metric, which measures the ranking quality of the first relevant item in the recommendation list, also confirms PGA-DRL's strong ranking performance. As seen in Table 6, PGA-DRL achieves the highest MRR scores on ML-100K (0.4988) and ML-1M (0.4749). These results underscore the model's effectiveness in ensuring that the most relevant items are positioned higher in the ranked list, thus enhancing user satisfaction.

Other models, including NeuMF and AutoCF, achieve slightly lower MRR values, which further supports PGA-DRL's advantage in ranking the most relevant items early in the recommendation list. The ability to position these items more prominently is critical in improving the user experience, as users are more likely to engage with recommendations that appear early in the list.

The Hit@10 metric, which measures the proportion of users for whom at least one relevant item appears in the top-10 recommendations, provides further evidence of the model's ability to recommend relevant items across different datasets. As shown in Table 7, PGA-DRL

Table 3
Precision@10 Prediction metrics results of the experiments. (Statistically significant results are indicated by: * $p < 0.05$, ** $p < 0.01$.).

Method	ML-100K	ML-1M	AmzSB	AmzMS	Mod Cloth
BPR	0.1998	0.2081	0.0063	0.0050	0.0110
NeuMF	0.2006	0.2058	**0.0248**	0.0155	0.0148
NGCF	0.2019	0.2019	0.0101	0.0109	0.0146
LightGCN	0.1435	0.1495	0.0105	0.0124	0.0163
SGL	0.0279	0.0291	0.0068	**0.0163**	0.0187
SimGCL	<u>0.1052</u>	0.1096	0.0045	0.0038	0.0076
AutoCF	0.2035	<u>0.2119</u>	0.0142	0.0155	<u>0.0191</u>
LightGCL	0.1743	0.1815	0.0122	0.0133	0.0164
GFormer	0.2021	0.2104	0.0141	0.0154	0.0191
AMLDM	0.2004	0.2087	0.0140	0.0152	0.0188
MGDCF	0.2017	0.2097	0.0138	0.0153	0.0189
XSimGCL	0.2013	0.1987	0.0134	0.0145	0.0179
PGA-DRL	**0.2069***	**0.2155***	<u>0.0145</u>	<u>0.0158</u>	**0.0195***

Table 4
Recall - Prediction metrics results of the experiments. (Statistically significant results are indicated by: * $p < 0.05$, ** $p < 0.01$.).

Method	ML-100K	ML-1M	AmzSB	AmzMS	Mod Cloth
BPR	0.2557	0.1710	0.0629	0.0498	0.1096
NeuMF	0.2552	<u>0.1740</u>	**0.2483**	0.1546	0.1478
NGCF	<u>0.2566</u>	0.1673	0.1014	0.1093	0.1459
LightGCN	0.1885	0.1261	0.1049	0.1239	0.1627
SGL	0.0682	0.0456	0.0682	**0.1630**	**0.1868**
SimGCL	0.1563	0.1045	0.0455	0.0385	0.0754
AutoCF	0.2528	0.1691	0.1421	0.1547	0.1621
LightGCL	0.2497	0.1714	0.1403	0.1528	0.1601
GFormer	0.2512	0.1716	0.1418	0.1536	0.1611
AMLDM	0.2507	0.1698	0.1415	0.1534	0.1608
MGDCF	0.2519	0.1705	0.1423	0.1542	0.1615
XSimGCL	0.2531	**0.2116**	0.1431	0.1553	0.1623
PGA-DRL	**0.2570***	0.1719	<u>0.1451</u>	<u>0.1573</u>	<u>0.1648</u>

Table 5
NDCG Prediction metrics results of the experiments. (Statistically significant results are indicated by: * $p < 0.05$, ** $p < 0.01$.).

Method	ML-100K	ML-1M	AmzSB	AmzMS	Mod Cloth
BPR	0.2944	0.2672	0.0322	0.0265	0.0604
NeuMF	0.2980	0.2655	**0.1550**	0.0760	0.0810
NGCF	0.2981	0.2584	0.0496	0.0509	0.0799
LightGCN	0.2143	0.1945	0.0573	0.0629	<u>0.0962</u>
SGL	0.0521	0.0473	0.0374	**0.0888**	**0.1098**
SimGCL	0.1623	0.1473	0.0289	0.0193	0.0433
AutoCF	0.2462	0.2712	0.1353	0.0852	0.0882
LightGCL	0.2250	0.2479	0.1236	0.0779	0.0806
GFormer	0.2401	0.2643	0.1319	0.0831	0.0861
AMLDM	0.2465	0.2715	0.1358	0.0857	0.0881
MGDCF	0.2455	0.2705	0.1369	0.0849	0.0873
XSimGCL	0.2496	0.2750	0.1372	0.0864	0.0892
PGA-DRL	**0.3056***	**0.2774***	<u>0.1384</u>	<u>0.0872</u>	0.0902

Table 6
MRR - Prediction metrics results of the experiments. (Statistically significant results are indicated by: * $p < 0.05$, ** $p < 0.01$.).

Method	ML-100K	ML-1M	AmzSB	AmzMS	Mod Cloth
BPR	0.4856	0.4623	0.0230	0.0195	0.0457
NeuMF	0.4917	0.4585	**0.1261**	0.0526	0.0610
NGCF	0.4948	0.4488	0.0344	0.0336	0.0602
LightGCN	0.3733	0.3554	0.0428	0.0446	0.0760
SGL	0.0867	0.0825	0.0283	**0.0662**	**0.0864**
SimGCL	0.3026	0.2881	0.0240	0.0135	0.0337
AutoCF	0.4713	0.4703	0.0891	0.0533	0.0756
LightGCL	0.4398	0.4283	0.0853	0.0485	0.0701
GFormer	0.4761	0.4619	0.0917	0.0521	0.0763
AMLDM	0.4824	0.4593	0.0918	0.0517	0.0759
MGDCF	0.4912	0.4681	0.0939	0.0524	0.0776
XSimGCL	0.4871	0.4638	0.0942	0.0513	0.0758
PGA-DRL	**0.4988***	**0.4749***	0.0953	0.0538	0.0787

Table 7
Hit - Prediction metrics results of the experiments. (Statistically significant results are indicated by: * $p < 0.05$, ** $p < 0.01$.).

Method	ML-100K	ML-1M	AmzSB	AmzMS	Mod Cloth
BPR	0.7996	0.7591	0.0629	0.0499	0.1098
NeuMF	0.7943	0.7611	**0.2483**	0.1547	0.1480
NGCF	**0.8070**	0.7493	0.1014	0.1094	0.1461
LightGCN	0.6829	0.6483	0.1049	0.1240	0.1629
SGL	0.2524	0.2396	0.0682	**0.1630**	**0.1870**
SimGCL	0.6225	0.5910	0.0455	0.0385	0.0756
AutoCF	0.7983	0.7609	0.1437	0.1517	0.1619
LightGCL	0.7752	0.7370	0.1397	0.1463	0.1591
GFormer	0.7981	0.7597	0.1442	0.1518	0.1607
AMLDM	0.7612	0.7146	0.1356	0.1421	0.1534
MGDCF	0.7992	0.7620	0.1449	0.1498	0.1639
XSimGCL	0.8002	0.7603	0.1441	0.1513	0.1628
PGA-DRL	0.8038	**0.7631***	0.1451	0.1524	0.1647

achieves the highest Hit@10 scores on both ML-100K (0.8038) and ML-1M (0.7631), underscoring its consistent ability to offer relevant recommendations to a large proportion of users. This metric is particularly important in real-world applications, where ensuring that at least one relevant item is presented to users can significantly improve engagement and satisfaction.

While PGA-DRL leads in Hit@10 on denser datasets, it performs less strongly on sparser datasets, such as Amazon Subscription Boxes (0.1451), where NeuMF leads with 0.2483. This further highlights the variability of recommendation models across different data characteristics, as PGA-DRL's performance is more robust when the user-item interaction graph is dense, but it faces challenges in the cold-start and sparse scenarios.

The results from the various datasets reveal that the performance of PGA-DRL is strongly influenced by the characteristics of the data, particularly its sparsity. On denser datasets such as ML-100K and ML-1M, PGA-DRL demonstrates consistently high performance across all metrics. This outcome aligns with our expectations, as PGA-DRL is intentionally designed to effectively leverage both global and local user-item interaction patterns, rendering it particularly well-suited for recommendation environments where abundant interaction data is available.

However, on sparser datasets like Amazon Subscription Boxes and Amazon Magazine Subscriptions, PGA-DRL's performance is somewhat diminished, particularly in metrics such as Precision@10 and Recall@10. This indicates that while PGA-DRL is a robust model capable of handling diverse interaction patterns, its effectiveness can be limited by the sparsity of the data, as it may struggle to learn meaningful representations from a limited number of interactions.

Despite this, PGA-DRL remains competitive in ranking tasks, as demonstrated by its performance in NDCG@10 and MRR@10. These results suggest that the model's ability to refine recommendations through the integration of GCNs and GATs, along with its reinforcement learning-based optimization, provides advantages even in sparse datasets by helping to better prioritize relevant items.

The experimental results confirm that PGA-DRL excels in ranking quality and recommendation accuracy, particularly on denser datasets. The PFM, which combines the strengths of GCNs and GATs, allows PGA-DRL to effectively capture both global and localized user-item interactions, resulting in superior performance in terms of NDCG@10, MRR@10, and Hit@10. While PGA-DRL is particularly strong in dense recommendation settings, its performance in sparse environments can be less consistent, with some models like NeuMF performing better in such contexts.

Nonetheless, PGA-DRL offers significant improvements in ranking tasks, ensuring that relevant items are not only retrieved but also positioned at the top of recommendation lists. This capability is crucial in enhancing user engagement and satisfaction, making PGA-DRL a strong contender for real-world RSs, where both accuracy and ranking quality are paramount.

Tables 8–12 provide a detailed comparison of the performance of PGA-DRL and baseline methods on the ML-100K dataset, evaluated across various top-K values ($k = 5, 10, 15, 20, 25, 50, 100$). These results offer a comprehensive view of how each model performs as the number of recommended items increases, providing key insights into the models' ability to maintain recommendation quality across different K-values.

In modern RS, increasing the number of recommended items is crucial for improving user engagement, content discovery, and personalization. This is particularly pertinent given the rapid expansion of digital platforms such as e-commerce, streaming services, and social media, where users are routinely presented with vast catalogs of options. By increasing K, we ensure that users receive a more diverse set of relevant recommendations, thereby significantly enhancing their chances of encountering new items that align with their interests. This becomes especially critical in scenarios where user preferences are dynamic and evolve over time, as a limited recommendation list might otherwise fail to capture the full spectrum of their potential interests.

Moreover, in applications like online retail and video streaming, increasing K allows recommendation systems to cater to different user intents, such as immediate purchase decisions, exploratory browsing, or long-term interest development. In personalized experiences, where different users exhibit varying levels of engagement, an adaptive approach that considers a broader recommendation list can significantly improve user satisfaction. Furthermore, in competitive platforms, where user retention is a priority, providing a larger pool of relevant recommendations helps reduce churn by continuously offering fresh and engaging content.

By analyzing the models' performance across different K-values, we assess not only their ability to retrieve relevant items but also their effectiveness in maintaining quality and ranking accuracy as the recommendation list grows. A well-performing model should be able to scale effectively, preserving recommendation relevance even when K increases. As the results demonstrate, PGA-DRL consistently outperforms baseline models across all K-values, making it a robust choice for applications that require both high precision in top-ranked items and strong coverage for diverse user preferences.

The results in Table 8 show that PGA-DRL consistently achieves the highest Precision@K across all K-values, outperforming other models at larger values of K, particularly at $k = 50$ and $k = 100$. This highlights PGA-DRL's strength in delivering accurate recommendations even when more items are considered for evaluation. Other models, such as NeuMF and NGCF, also perform well but fall short of PGA-DRL's precision at larger K-values, indicating that PGA-DRL is better at maintaining high relevance in extended recommendation lists. This is particularly important for recommendation systems that need to scale to longer lists while ensuring that users are presented with highly relevant content.

The results in Table 9 demonstrate that PGA-DRL maintains high

Table 8
Performance comparison of various methods on the ML-100K dataset, showing Precision metric at different values of k (5, 10, 15, 20, 25, 50, 100). (Statistically significant results are indicated by: * $p < 0.05$, ** $p < 0.01$.).

Method	@5	@10	@15	@20	@25	@50	@100
BPR	0.2484	0.1998	0.1733	0.1538	0.1389	0.0986	0.0654
NeuMF	0.2558	0.2006	0.1717	0.1528	0.1379	0.0989	0.0664
NGCF	0.2477	0.2019	0.1735	0.1540	0.1391	0.0986	0.0657
LightGCN	0.1803	0.1435	0.1224	0.1113	0.1021	0.0747	0.0517
SGL	0.0312	0.0279	0.0255	0.0240	0.0229	0.0194	0.0160
SimGCL	0.1283	0.1052	0.0922	0.0837	0.0766	0.0553	0.0386
AutoCF	0.2537	0.2035	0.1757	0.1562	0.1421	0.1005	0.0671
LightGCL	0.2172	0.1743	0.1505	0.1338	0.1217	0.0861	0.0575
GFormer	0.2519	0.2021	0.1745	0.1552	0.1411	0.0998	0.0667
AMLDM	0.2498	0.2004	0.1730	0.1539	0.1399	0.0989	0.0661
MGDCF	0.2514	0.2017	0.1741	0.1548	0.1409	0.0996	0.0665
XSimGCL	0.2509	0.2013	0.1738	0.1545	0.1406	0.0994	0.0664
PGA-DRL	**0.2579***	**0.2069***	**0.1786***	**0.1588***	**0.1445***	**0.1021***	**0.0682***

Table 9
Performance comparison of various methods on the ML-100K dataset, showing Recall metric at different values of k (5, 10, 15, 20, 25, 50, 100). (Statistically significant results are indicated by: * $p < 0.05$, ** $p < 0.01$.).

Method	@5	@10	@15	@20	@25	@50	@100
BPR	0.1614	0.2557	**0.3295**	**0.3823**	**0.4254**	0.5742	0.7297
NeuMF	**0.1650**	0.2552	0.3197	0.3728	0.4171	0.5705	0.7329
NGCF	0.1583	0.2566	0.3266	0.3794	0.4210	0.5718	0.7254
LightGCN	0.1187	0.1885	0.2404	0.2828	0.3169	0.4427	0.5791
SGL	0.0395	0.0682	0.0901	0.1128	0.1312	0.1977	0.2899
SimGCL	0.0961	0.1563	0.2066	0.2464	0.2791	0.3862	0.5085
AutoCF	0.1593	0.2528	0.3235	0.3733	0.4173	0.5669	0.7311
LightGCL	0.1574	0.2497	0.3196	0.3687	0.4122	0.5599	0.7221
GFormer	0.1583	0.2512	0.3215	0.3709	0.4147	0.5633	0.7265
AMLDM	0.1580	0.2507	0.3208	0.3702	0.4139	0.5622	0.7250
MGDCF	0.1587	0.2519	0.3224	0.3720	0.4158	0.5649	0.7285
XSimGCL	0.1595	0.2531	0.3239	0.3737	0.4178	0.5676	0.7320
PGA-DRL	0.1620	**0.2570***	0.3290	0.3795	0.4243	**0.5764***	**0.7434***

Table 10
Performance comparison of various methods on the ML-100K dataset, showing NDCG metric at different values of k (5, 10, 15, 20, 25, 50, 100). (Statistically significant results are indicated by: * $p < 0.05$, ** $p < 0.01$.).

Method	@5	@10	@15	@20	@25	@50	@100
BPR	0.2930	0.2944	0.3060	0.3178	0.3294	0.3786	0.4293
NeuMF	0.3021	0.2980	0.3056	0.3172	0.3290	0.3802	0.4333
NGCF	0.2955	0.2981	0.3077	0.3192	0.3303	0.3799	0.4307
LightGCN	0.2135	0.2143	0.2204	0.2305	0.2399	0.2811	0.3260
SGL	0.0413	0.0521	0.0598	0.0673	0.0733	0.0941	0.1200
SimGCL	0.1549	0.1623	0.1744	0.1866	0.1973	0.2332	0.2715
AutoCF	0.2455	0.2462	0.2547	0.2638	0.2740	0.3151	0.3593
LightGCL	0.2244	0.2250	0.2327	0.2411	0.2504	0.2880	0.3284
GFormer	0.2394	0.2401	0.2484	0.2573	0.2672	0.3073	0.3504
AMLDM	0.2458	0.2465	0.2550	0.2641	0.2744	0.3155	0.3598
MGDCF	0.2448	0.2455	0.2539	0.2630	0.2732	0.3142	0.3583
XSimGCL	0.2489	0.2496	0.2582	0.2674	0.2778	0.3194	0.3643
PGA-DRL	**0.3047***	**0.3056***	**0.3161***	**0.3274***	**0.3401***	**0.3910***	**0.4460***

Recall@K values, ensuring that relevant items are recommended across a broader range of K-values. In particular, PGA-DRL achieves a Recall@10 of 0.2570, outperforming other models in the middle-range values of k, where models like AutoCF and NeuMF perform similarly but do not reach the same level of performance across all K-values. This indicates that PGA-DRL is particularly effective at recommending a wider range of relevant items, ensuring better coverage of user preferences. This capability is of paramount importance in real-world applications, where maximizing the exposure of users to as many relevant items as possible can significantly enhance their overall experience and engagement with the platform.

Table 10 demonstrates that PGA-DRL excels in ranking the most relevant items higher in the recommendation list, achieving superior NDCG@K scores compared to other models, particularly for $k = 50$ and $k = 100$. This supports earlier findings that PGA-DRL not only retrieves relevant items but also ensures they are ranked appropriately, maximizing user satisfaction in top-K recommendations. NDCG is a critical metric for real-world systems as it reflects both the relevance of items and their order in the ranking, ensuring that the best items are placed at the top of the recommendation list.

The MRR@K results in Table 11 reinforce PGA-DRL's advantage in placing relevant items earlier in the recommendation list. At all K-values, PGA-DRL achieves the highest MRR@K, particularly excelling at $k = 5$ and $k = 10$, ensuring that users see relevant items as early as possible in the recommended list. This is crucial for improving user engagement, as items placed higher in the list are more likely to be interacted with. The ability of PGA-DRL to prioritize the most relevant items early in the ranking enhances the likelihood that users will engage

Table 11
Performance comparison of various methods on the ML-100K dataset, showing MRR metric at different values of k (5, 10, 15, 20, 25, 50, 100). (Statistically significant results are indicated by: * $p < 0.05$, ** $p < 0.01$.).

Method	@5	@10	@15	@20	@25	@50	@100
BPR	0.4685	0.4856	0.4909	0.4926	0.4936	0.4950	0.4954
NeuMF	0.4741	0.4917	0.4966	0.4989	0.4996	0.5013	0.5016
NGCF	0.4770	0.4948	0.5003	0.5015	0.5024	0.5039	0.5042
LightGCN	0.3560	0.3733	0.3791	0.3819	0.3836	0.3859	0.3864
SGL	0.0726	0.0867	0.0921	0.0952	0.0976	0.1025	0.1049
SimGCL	0.2794	0.3026	0.3094	0.3128	0.3143	0.3169	0.3179
AutoCF	0.4540	0.4713	0.4761	0.4775	0.4786	0.4800	0.4803
LightGCL	0.4237	0.4398	0.4443	0.4456	0.4466	0.4479	0.4482
GFormer	0.4587	0.4761	0.4810	0.4823	0.4835	0.4849	0.4852
AMLDM	0.4647	0.4824	0.4873	0.4887	0.4899	0.4913	0.4916
MGDCF	0.4732	0.4912	0.4962	0.4976	0.4988	0.5002	0.5005
XSimGCL	0.4693	0.4871	0.4921	0.4935	0.4947	0.4961	0.4964
PGA-DRL	**0.4805***	**0.4988***	**0.5039***	**0.5053***	**0.5065***	**0.5080***	**0.5083***

Table 12
Performance comparison of various methods on the ML-100K dataset, showing Hit metric at different values of k (5, 10, 15, 20, 25, 50, 100). (Statistically significant results are indicated by: * $p < 0.05$, ** $p < 0.01$.).

Method	@5	@10	@15	@20	@25	@50	@100
BPR	0.6713	0.7996	0.8653	0.8950	0.9183	0.9661	0.9894
NeuMF	0.6649	0.7943	0.8558	0.8950	0.9120	**0.9671**	0.9862
NGCF	0.6734	**0.8070**	**0.8749**	0.8950	0.9152	**0.9671**	0.9873
LightGCN	0.5525	0.6829	0.7572	0.8070	0.8452	0.9290	0.9576
SGL	0.1463	0.2524	0.3213	0.3765	0.4327	0.6055	0.7688
SimGCL	0.4496	0.6225	0.7094	0.7688	0.8049	0.8908	0.9544
AutoCF	0.6755	0.7983	0.8634	0.8935	0.9199	0.9584	0.9844
LightGCL	0.6560	0.7752	0.8384	0.8677	0.8932	0.9307	0.9559
GFormer	0.6754	0.7981	0.8631	0.8933	0.9196	0.9582	0.9842
AMLDM	0.6442	0.7612	0.8232	0.8520	0.8771	0.9139	0.9387
MGDCF	0.6763	0.7992	0.8643	0.8946	0.9209	0.9595	0.9855
XSimGCL	0.6772	0.8002	0.8654	0.8957	0.9221	0.9607	0.9868
PGA-DRL	**0.6802***	0.8038	0.8693	**0.8997***	**0.9262***	0.9650	**0.9912***

with the content presented.

Table 12 shows that PGA-DRL achieves the highest Hit@K values across all K-values, especially for $k = 100$, indicating that it ensures at least one relevant item is recommended to the majority of users. This metric is important for measuring the general effectiveness of a recommendation system in diverse user scenarios. PGA-DRL outperforms other models in providing consistent coverage of relevant items, ensuring a higher likelihood of user engagement. Indeed, Hit@K is especially valuable in assessing a model's capability to deliver at least one relevant recommendation to users, a key factor in sustaining user interest and satisfaction.

To summarize the findings on the ML-100K dataset, PGA-DRL consistently demonstrates superior performance across all evaluated metrics, particularly excelling at larger K-values where maintaining the quality of recommendations becomes increasingly critical. These results highlight PGA-DRL's robustness in both ranking accuracy and coverage, ensuring that relevant items are not only retrieved but also presented in an optimal order. While NeuMF and NGCF show competitive results, PGA-DRL consistently outperforms these models in both ranking quality and diversity, positioning it as a leading approach for top-K recommendation tasks. This comprehensive evaluation solidifies the effectiveness of PGA-DRL in delivering high-quality, accurate, and diverse recommendations across various settings and recommendation list lengths.

Also, Figs. 3–7 provide a visual representation of the comparative performance across all the evaluation metrics, presented through bar

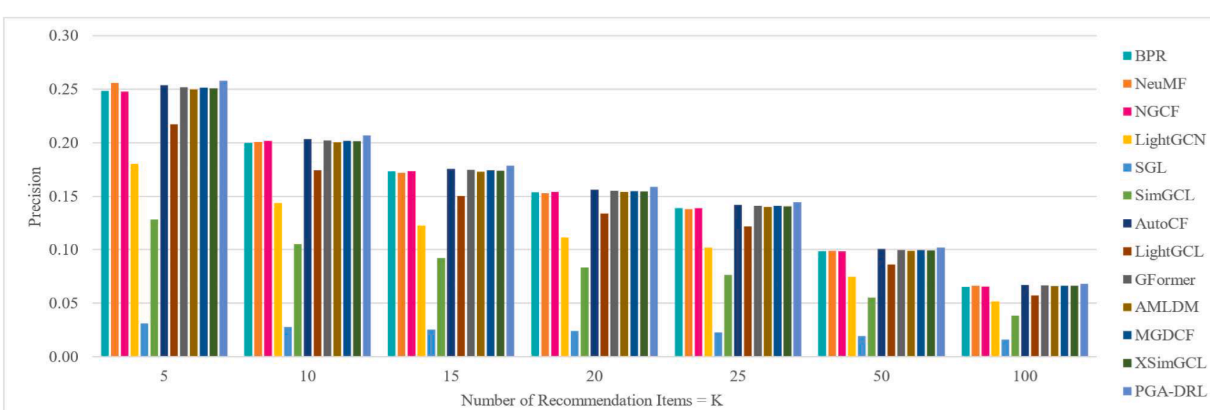

Fig. 3. Precision at different values of k for various recommendation models on the ML-100K dataset, showing PGA-DRL's superior performance.

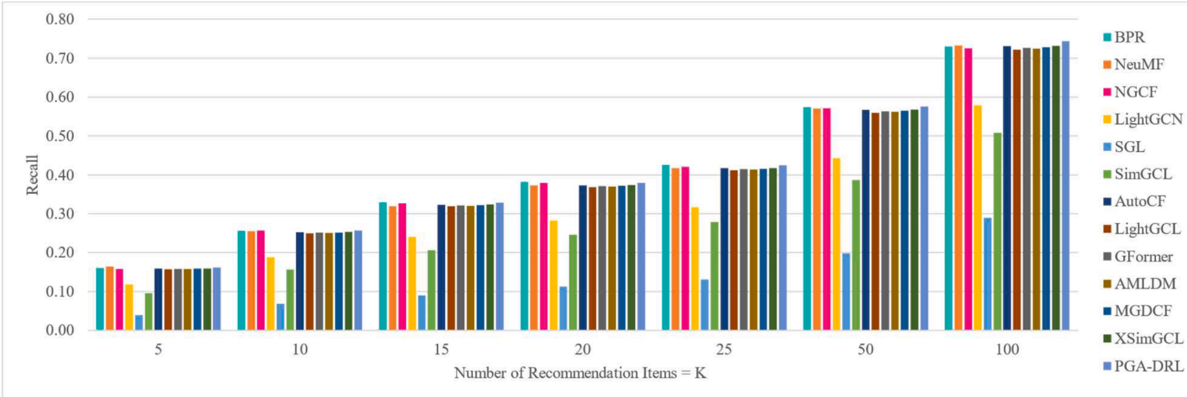

Fig. 4. Recall at different values of k for various recommendation models on the ML-100K dataset, highlighting PGA-DRL's ability to capture relevant items.

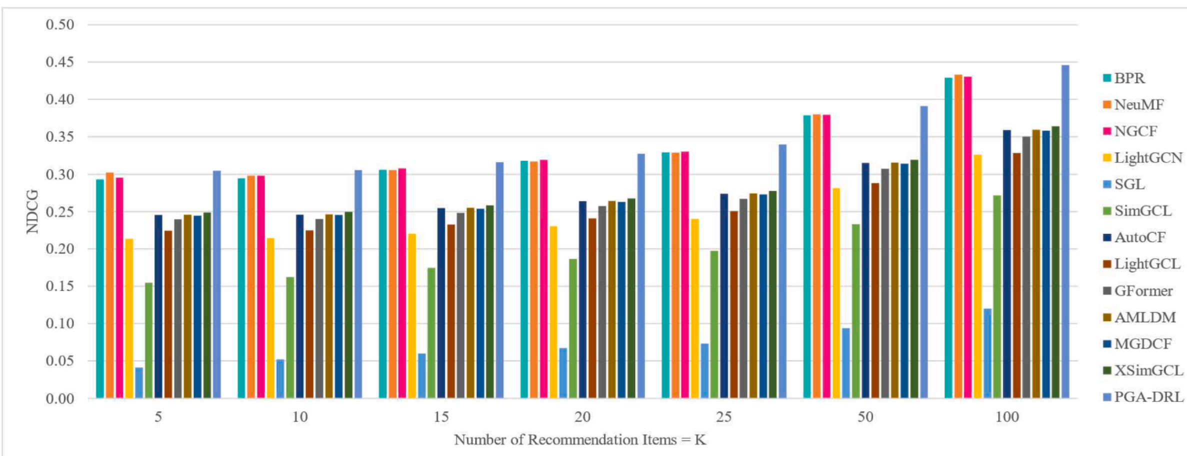

Fig. 5. NDCG at different values of k for various recommendation models on the ML-100K dataset, illustrating PGA-DRL's ability to prioritize highly relevant items.

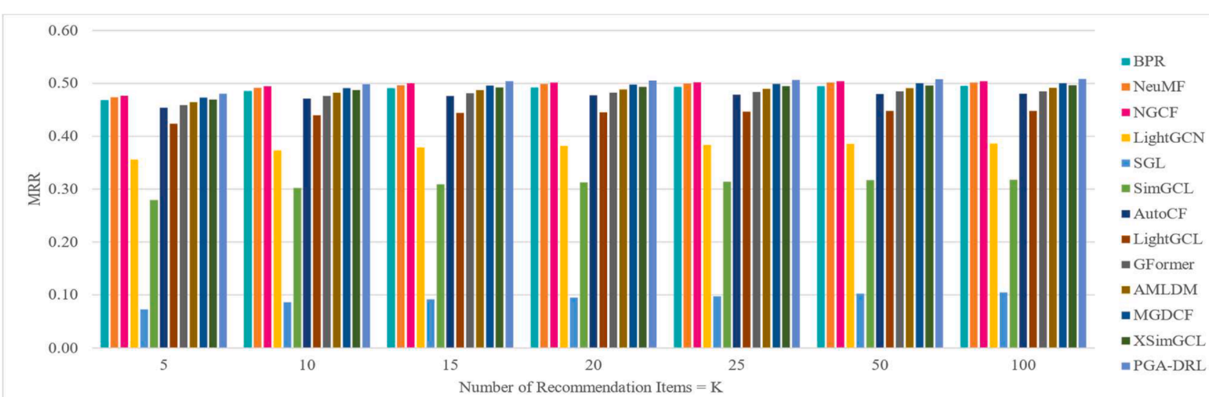

Fig. 6. MRR at different values of k for various recommendation models on the ML-100K dataset, showcasing PGA-DRL's superior ranking of relevant items.

charts. These charts clearly illustrate that PGA-DRL consistently outperforms all baseline models at various K-values, confirming its ability to deliver more accurate, relevant, and diverse recommendations. The charts emphasize the ability of PGA-DRL to achieve superior performance in both ranking and retrieval tasks, which are pivotal to the success of recommender systems, particularly in dynamic environments where ranking relevance and diversity are critical.

5. Discussion

The experimental results demonstrate the effectiveness of the proposed PGA-DRL model in capturing user-item interactions for RSs. This discussion focuses on analyzing the model's performance across various datasets, the contributions of different components, and the implications of these findings.

The PFM of the PGA-DRL model, which integrates GCNs and GATs, has demonstrated significant improvements in recommendation

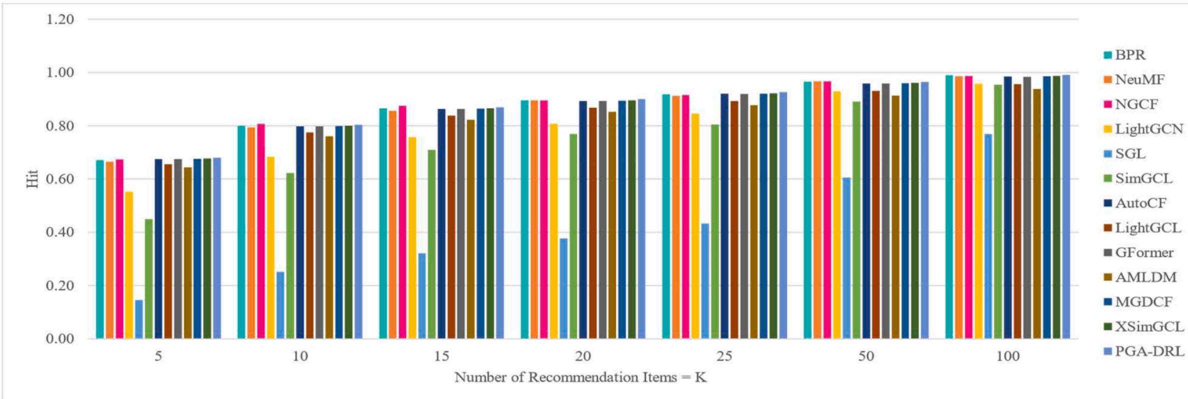

Fig. 7. Hit rate at different values of k for various recommendation models on the ML-100K dataset, with PGA-DRL demonstrating higher success in recommending relevant items to users.

accuracy compared to state-of-the-art baselines. By progressively combining GCN and GAT embeddings, the model leverages both global structural relationships and localized interaction patterns. This fusion effectively enhances the representation quality of user and item embeddings, leading to superior ranking performance, as evidenced by the higher NDCG@10 and MRR@10 scores across multiple datasets. The progressive approach outperformed models that employed either GCNs or GATs independently, such as LightGCN and NGCF, demonstrating the benefits of leveraging complementary strengths of both global aggregation and localized attention mechanisms. Also, even when GAT assigns uniform weights, GCN acts as a structural regularizer, ensuring stable propagation of collaborative signals. This redundancy mitigates overfitting and enhances robustness, particularly in noisy or cold-start scenarios.

The results show that PGA-DRL outperforms existing methods in ranking quality, which is crucial for enhancing user satisfaction. Metrics like NDCG@10, Precision@10, and MRR@10 indicate that PGA-DRL is particularly effective in positioning highly relevant items towards the top of the recommendation list. This improved ranking quality suggests that the fusion mechanism not only integrates diverse types of interactions but also learns to prioritize them appropriately, resulting in more personalized and relevant recommendations.

Moreover, the model demonstrated competitive performance in terms of recall and hit rate metrics, indicating its capacity to cover a wide range of relevant items. Specifically, the Recall@10 and Hit@10 results reveal that PGA-DRL can effectively capture broader user preferences. This robustness is evident across diverse datasets, including both dense (ML-100k and ML-1M) and sparse datasets (Amazon Subscription Boxes and ModCloth).

The performance of PGA-DRL across five benchmark datasets highlights its robustness and generalizability. The datasets chosen for the experiments represent diverse domains and vary significantly in terms of sparsity and user-item interactions. PGA-DRL's ability to consistently outperform or match the performance of baselines across these datasets suggests that it effectively captures complex user-item relationships regardless of dataset characteristics. This adaptability is particularly important for real-world recommendation scenarios where user behavior and data characteristics are dynamic and heterogeneous.

The AC framework was shown to be an effective approach for optimizing long-term user satisfaction. The Critic's role in evaluating the actions of the Actor significantly contributes to refining the recommendation policy, leading to improved outcomes in terms of sustained user engagement. Without the Critic, the model's ability to optimize long-term rewards was diminished, highlighting the importance of the AC framework in achieving high-quality recommendations.

Despite incorporating both GCNs and GATs, PGA-DRL maintains competitive computational efficiency compared to other models. The progressive fusion approach effectively reduces redundancy by combining representations at each layer, rather than treating them independently. This design not only enhances the expressive power of the model but also keeps computational complexity manageable, allowing the model to scale well even for larger datasets. This balance between performance and efficiency makes PGA-DRL suitable for real-world applications where computational resources may be limited. While the primary focus of this study is on demonstrating PGA-DRL's effectiveness in improving recommendation accuracy, future work will involve a comprehensive analysis of its runtime and memory usage to more precisely quantify the computational trade-offs inherent in the progressive fusion mechanism.

5.1. Sensitivity analysis

To assess the robustness of PGA-DRL, we conducted sensitivity experiments on key hyperparameters, including embedding size, number of GNN layers, and dropout rate, as these significantly impact model performance. The goal of this analysis is to determine how variations in these parameters affect recommendation accuracy and to identify optimal configurations.

- **Effect of Embedding Size:** The embedding size d controls the dimensionality of user and item representations. We experimented with four different sizes: 32, 64, 128, and 256, using ML-100K as the benchmark dataset. The results, shown in Table 13, indicate that performance improves as the embedding size increases up to 128, but begins to slightly decline at 256, suggesting potential overfitting. This aligns with findings in prior literature, where excessively large embeddings can introduce redundancy and increase computational cost without substantial performance gains.

- **Effect of Number of GNN Layers:** We varied the number of GCN/GAT layers from 2 to 10, keeping all other hyperparameters constant. As seen in Table 14, performance improves when increasing layers up to 8, but adding more layers leads to diminishing returns or even slight performance degradation. This is likely due to over-smoothing, where deeper layers aggregate too much neighborhood information, leading to less discriminative representations. Therefore, a depth of 8 layers is optimal for this dataset.

- **Effect of Dropout Rate:** Dropout is used to prevent overfitting by randomly deactivating units during training. We tested dropout rates of 0.0 (no dropout), 0.1, 0.3, and 0.5. As shown in Table 15, 0.3 dropout provides the best performance, suggesting it effectively balances regularization without overly disrupting training. Using no dropout (0.0) results in slightly better training performance but

Table 13
Effect of Embedding Size on Recommendation Performance of the PGA-DRL Model on ML-100K Dataset.

Embedding Size	Precision@10	Recall@10	NDCG@10	MRR@10	Hit@10
32	0.1985	0.2459	0.2927	0.4781	0.7701
64	0.2023	0.2517	0.2995	0.4884	0.7873
128 (default)	0.2069	0.2570	0.3056	0.4988	0.8038
256	0.2031	0.2523	0.3007	0.4893	0.7885

Table 14
Effect of GNN layers on recommendation performance of the PGA-DRL model on ML-100K dataset.

Embedding Size	Precision@10	Recall@10	NDCG@10	MRR@10	Hit@10
2	0.1935	0.2403	0.2859	0.4673	0.7524
4	0.1981	0.2472	0.2939	0.4794	0.7731
6	0.2047	0.2536	0.3011	0.4923	0.7931
8 (default)	0.2069	0.2570	0.3056	0.4988	0.8038
10	0.2051	0.2548	0.3029	0.4941	0.7971

Table 15
Effect of dropout rate on recommendation performance of the PGA-DRL model on ML-100K dataset.

Dropout Rate	Precision@10	Recall@10	NDCG@10	MRR@10	Hit@10
0.0	0.2019	0.2506	0.2981	0.4863	0.7847
0.1	0.2031	0.2521	0.2997	0.4895	0.7891
0.3 (default)	0.2069	0.2570	0.3056	0.4988	0.8038
0.5	0.2008	0.2491	0.2963	0.4837	0.7795

worsens generalization on test data. Conversely, excessive dropout (0.5) harms performance by removing too much information.

From these experiments, we conclude that embedding size = 128, number of layers = 8, and dropout = 0.3 provide the best performance for PGA-DRL. While minor variations from these specific values do not significantly impair the model's effectiveness, we observed that extreme settings—such as excessively large embedding dimensions, an overly deep network architecture, or an overly aggressive dropout rate—tend to negatively impact recommendation accuracy. This analysis demonstrates that PGA-DRL is relatively robust to hyperparameter choices but benefits from careful tuning to optimize performance.

5.2. Ablation study

To understand the contribution of each component in the PGA-DRL model, we conducted an ablation study using the ML-100K dataset, which is widely used in recommendation system research due to its balanced size and well-established benchmarks. Specifically, we evaluated the impact of removing or modifying critical parts of the model, including the GAT layers, the PFM, and the Critic network in the AC framework. The performance of each variant was assessed using Precision@10, Recall@10, NDCG@10, MRR@10, and Hit@10, as these metrics provide a comprehensive evaluation of both the ranking quality and the overall recommendation accuracy, capturing different aspects of user satisfaction.

The following ablation settings were considered:

- **Full PGA-DRL:** The complete model with GCN, GAT, PFM, and AC framework, representing the fully integrated approach to capturing both global and localized interaction patterns.
- **Without GAT Layers:** This variant uses only the GCN component without GAT to capture local user-item interactions.
- **Without PFM:** In this variant, GCN and GAT embeddings are not progressively fused. Instead, their outputs are independently processed.
- **Without Critic:** The Critic network, responsible for evaluating actions and providing feedback, was removed to assess its role in optimizing long-term rewards.

Table 16 presents the results of the ablation study on the ML-100K dataset using the selected metrics.

The Full PGA-DRL model outperformed all other variants across all metrics, demonstrating the effectiveness of combining GCN and GAT for capturing both global and local interaction patterns, and the importance of the AC framework for optimizing long-term recommendations. Specifically:

- **Without GAT Layers:** Removing the GAT layers resulted in a noticeable decline in Precision@10, Recall@10, and NDCG@10, indicating the importance of GAT in capturing localized attention patterns. This suggests that GAT layers are crucial for modeling fine-grained user-item interactions, leading to more personalized recommendations.
- **Without PFM:** The absence of the PFM significantly impacted all metrics, especially Precision@10, NDCG@10, and MRR@10. This highlights the effectiveness of progressively fusing GCN and GAT embeddings, which ensures that both global and local interaction signals are integrated throughout the model, leading to richer user-item representations.
- **Without Critic:** The removal of the Critic network led to moderate decreases in performance. The Critic's role in providing feedback for action evaluation is essential for optimizing long-term rewards, which improves the model's ability to recommend items that lead to sustained user satisfaction.

The results of the ablation study highlight the contribution of each component to the overall performance of PGA-DRL. These findings can inform future work by guiding improvements in model design, such as

Table 16
Performance comparison of different variants of the PGA-DRL model on ML-100K dataset.

Model Variant	Precision@10	Recall@10	NDCG@10	MRR@10	Hit@10
Full PGA-DRL	0.2069	0.2570	0.3056	0.4988	0.8038
Without GAT Layers	0.1895	0.2431	0.2903	0.4797	0.7754
Without PFM	0.1824	0.2367	0.2842	0.4711	0.7685
Without Critic	0.1957	0.2485	0.2970	0.4862	0.7851

optimizing the fusion mechanism or enhancing the integration of attention-based components. The PFM plays a crucial role in balancing global and localized interaction patterns, while the GAT layers add value by emphasizing important neighbors during aggregation. The Critic network ensures that the model not only optimizes immediate rewards but also considers long-term user satisfaction, leading to more effective recommendations.

Overall, the Full PGA-DRL model achieves the highest scores in all ranking metrics, demonstrating the effectiveness of combining GCN, GAT, PFM, and the Actor-Critic framework in a progressive manner. The ablation study clearly shows that each of these components contributes significantly to capturing complex user-item interactions and improving recommendation quality.

6. Conclusion

This paper introduced PGA-DRL, a Progressive Graph Attention-Based DRL model designed to effectively capture both global and local user-item interaction patterns within RSs. By integrating GCNs and GATs in a progressive manner within an AC framework, PGA-DRL enables the model to leverage both coarse-grained and fine-grained interaction signals, resulting in a more nuanced and adaptive recommendation strategy. Our experiments across multiple benchmark datasets demonstrate that PGA-DRL consistently outperforms existing state-of-the-art models, particularly in ranking quality, as reflected by improvements in metrics such as NDCG@10 and MRR@10. The PFM adopted by PGA-DRL offers a pathway for balancing global graph structure insights with localized attention-based information, leading to superior recommendation performance. Moreover, the model's robust performance across datasets of varying sparsity indicates its adaptability and scalability for diverse recommendation scenarios.

In addition to its performance gains, PGA-DRL demonstrates the feasibility of integrating advanced graph-based models within a reinforcement learning framework for RSs. The specific integration of GCNs and GATs, achieved through a progressive concatenation approach, not only enhances the model's capacity to capture intricate user-item relationships but also provides a structured and systematic method for refining user and item embeddings across multiple processing layers. This progressive refinement contributes significantly to the model's robustness and ability to adapt to dynamic user behaviors, making it well-suited for practical applications where user preferences evolve over time.

The AC framework employed in PGA-DRL plays a crucial role in optimizing the recommendation policy. The Actor component selects actions (recommendations) based on the current state, while the Critic evaluates these actions to ensure they lead to long-term user satisfaction, thereby improving the overall quality of the recommendations. By separating action selection (Actor) from action evaluation (Critic), the model effectively learns to generate recommendations that maximize long-term user satisfaction. This approach allows the system to strike a balance between immediate engagement and sustained user interest, which is critical for enhancing user retention in RS. The use of reinforcement learning, combined with graph-based techniques, ensures that the recommendations are not only relevant but also contextually aware, adapting to changes in user preferences and item popularity.

Moreover, the experimental results underscore the model's robust ability to generalize effectively across datasets exhibiting varying degrees of sparsity. This adaptability highlights PGA-DRL's versatility in handling both dense and sparse recommendation environments, a crucial attribute for real-world applications where data characteristics can differ significantly across various domains. Building upon this, the inherent scalability of PGA-DRL suggests its strong potential for deployment in large-scale RSs. This scalability is facilitated by several key features, including its efficient graph representation, the reduced computational overhead achieved through the progressive fusion mechanism, and its capacity to effectively manage large user-item interaction graphs. Consequently, these combined characteristics render PGA-DRL particularly well-suited for practical applications where both high performance and scalability are paramount.

CRediT authorship contribution statement

Jawad Tanveer: Writing – review & editing, Resources. **Sang-Woong Lee:** Writing – review & editing, Resources. **Amir Masoud Rahmani:** Writing – review & editing, Resources. **Khursheed Aurangzeb:** Writing – review & editing, Data curation. **Mahfooz Alam:** Writing – review & editing, Data curation. **Gholamreza Zare:** Writing – original draft, Visualization, Methodology, Conceptualization. **Pegah Malekpour Alamdari:** Writing – original draft, Validation, Methodology, Conceptualization. **Mehdi Hosseinzadeh:** Writing – review & editing, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This Research is funded by Researchers Supporting Project Number (RSPD2025R947), King Saud University,Riyadh, Saudi Arabia.

Data availability

No data was used for the research described in the article.

References

[1] B. Walek, P. Fajmon, A hybrid recommender system for an online store using a fuzzy expert system, Expert. Syst. Appl. 212 (2023) 118565.
[2] A. Valencia-Arias, H. Uribe-Bedoya, J.D. González-Ruiz, G.S. Santos, E.C. Ramírez, E.M. Rojas, Artificial intelligence and recommender systems in e-commerce. Trends and research agenda, Intell. Syst. Appl. 24 (2024) 200435.
[3] R. Esmeli, H. Abdullahi, M. Bader-El-Den, A.S. Can, Session context data integration to address the cold start problem in e-commerce recommender systems, Decis. Support. Syst. 187 (2024) 114339.
[4] Z. Khan, M.I. Hussain, N. Iltaf, J. Kim, M. Jeon, Contextual recommender system for E-commerce applications, Appl. Soft. Comput. 109 (2021) 107552.
[5] D. Wang, X. Zhao, Affective video recommender systems: a survey, Front. Neurosci. 16 (2022).
[6] P. Gu, H. Hu, A holistic view on positive and negative implicit feedback for micro-video recommendation, Knowl. Based. Syst. 284 (2024) 111299.
[7] Y. Gao, MOOCs video recommendation using low-rank and sparse matrix factorization with inter-entity relations and intra-entity affinity information, Inf. Process. Manage 61 (2024) 103861.
[8] M. Perano, G.L. Casali, Y. Liu, T. Abbate, Professional reviews as service: a mix method approach to assess the value of recommender systems in the entertainment industry, Technol. Forecast. Soc. Change 169 (2021) 120800.
[9] W. Liu, Design and user behavior analysis of an English learning social platform based on digital entertainment content recommendation algorithm, Entertain. Comput. 51 (2024) 100734.
[10] J. Liao, W. Zhou, F. Luo, J. Wen, M. Gao, X. Li, J. Zeng, SocialLGN: light graph convolution network for social recommendation, Inf. Sci. (Ny) 589 (2022) 595–607.
[11] X. Zan, A. Semenov, C. Wang, X. Xian, W. Geremew, Causality-aware social recommender system with network homophily informed multi-treatment confounders, Inf. Sci. (Ny) 676 (2024) 120729.
[12] H. Hamidi, R. Moradi, Design of a dynamic and robust recommender system based on item context, trust, rating matrix and rating time using social networks analysis, J. King Saud Univ. - Comput. Inf. Sci. 36 (2024) 101964.
[13] M. Etemadi, S. Bazzaz Abkenar, A. Ahmadzadeh, M. Haghi Kashani, P. Asghari, M. Akbari, E. Mahdipour, A systematic review of healthcare recommender systems: open issues, challenges, and techniques, Expert. Syst. Appl. 213 (2023) 118823.
[14] Y. Sun, J. Zhou, M. Ji, L. Pei, Z. Wang, Development and evaluation of health recommender systems: systematic scoping review and evidence mapping, J. Med. Internet Res. 25 (2023) e38184.
[15] Y. Sun, M. Leng, W. Lu, B. Li, F. Lv, W. Zhang, Z. Wang, A knowledge graph-based recommender system for dementia care: design and evaluation study, Int. J. Med. Inform. 191 (2024) 105554.

[16] M.M. Talha, H.U. Khan, S. Iqbal, M. Alghobiri, T. Iqbal, M. Fayyaz, Deep learning in news recommender systems: a comprehensive survey, challenges and future trends, Neurocomputing. 562 (2023) 126881.
[17] M. Jesse, D. Jannach, Digital nudging with recommender systems: survey and future directions, Comput. Hum. Behav. Rep. 3 (2021) 100052.
[18] E. Fernandes, S. Moro, P. Cortez, Towards a news recommendation system to increase reader engagement through newsletter content personalization, Procedia Comput. Sci. 239 (2024) 217–225.
[19] C.-L. Hsu, Does a gamified website matter? An extended UTAUT model with regulatory fit, user experience, engagement, and attitudes, Entertain. Comput. 50 (2024) 100658.
[20] E. Engström, I. Vartanova, J. Viberg Johansson, M. Persson, P. Strimling, Comparing and modeling the use of online recommender systems, Comput. Hum. Behav. Rep. 15 (2024) 100449.
[21] P.M. Alamdari, N.J. Navimipour, M. Hosseinzadeh, A.A. Safaei, A. Darwesh, A systematic study on the recommender systems in the E-commerce, IEEe Access. 8 (2020) 115694–115716.
[22] G. Zare, N. Jafari Navimipour, M. Hosseinzadeh, A. Sahafi, Network link prediction via deep learning method: a comparative analysis with traditional methods, Eng. Sci. Technol., Int. J. 56 (2024) 101782.
[23] S. Natarajan, S. Vairavasundaram, S. Natarajan, A.H. Gandomi, Resolving data sparsity and cold start problem in collaborative filtering recommender system using Linked Open Data, Expert. Syst. Appl. 149 (2020) 113248.
[24] K. Saini, A. Singh, A content-based recommender system using stacked LSTM and an attention-based autoencoder, Measurement: Sensors 31 (2024) 100975.
[25] F. Wang, Z. Zheng, Y. Zhang, Y. Li, K. Yang, C. Zhu, To see further: knowledge graph-aware deep graph convolutional network for recommender systems, Inf. Sci. (Ny) 647 (2023) 119465.
[26] T. Wei, T.W.S. Chow, FGCR: fused graph context-aware recommender system, Knowl. Based. Syst. 277 (2023) 110806.
[27] F. Guo, Z. Wang, X. Wang, Q. Lu, S. Ji, Dual-view multi-modal contrastive learning for graph-based recommender systems, Comput. Electric. Eng. 116 (2024) 109213.
[28] S.-W. Lee, J. Tanveer, A.M. Rahmani, H. Alinejad-Rokny, P. Khoshvaght, G. Zare, P. Malekpour Alamdari, M. Hosseinzadeh, SFGCN: synergetic fusion-based graph convolutional networks approach for link prediction in social networks, Inf. Fusion 114 (2025) 102684.
[29] Y. Zhu, J. Ma, C. Yuan, X. Zhu, Interpretable learning based dynamic graph convolutional networks for alzheimer's disease analysis, Inf. Fusion 77 (2022) 53–61.
[30] Y. Wang, J. Hu, F. Teng, L. Peng, S. Du, T. Li, KaTaGCN: knowledge-augmented and time-aware graph convolutional network for efficient traffic forecasting, Inf. Fusion 111 (2024) 102542.
[31] Y. Wang, D. Zhu, A hypergraph-based hybrid graph convolutional network for intracity human activity intensity prediction and geographic relationship interpretation, Inf. Fusion 104 (2024) 102149.
[32] X. Jin, D. Kong, M. Xie, Y. Huang, M. Liu, W. Yang, H. Shi, Y. Liu, Self-supervised reconstructed graph learning for link prediction in bipartite graphs, Neurocomputing. 602 (2024) 128250.
[33] D. Kumar Jain, X. Zhao, S. Garcia, S. Neelakandan, Robust multi-modal pedestrian detection using deep convolutional neural network with ensemble learning model, Expert. Syst. Appl. 249 (2024) 123527.
[34] X. Chen, L. Yao, J. McAuley, G. Zhou, X. Wang, Deep reinforcement learning in recommender systems: a survey and new perspectives, Knowl. Based. Syst. 264 (2023) 110335.
[35] R.K. Jha, S. Bag, D. Koley, G.R. Bojja, S. Barman, An appropriate and cost-effective hospital recommender system for a patient of rural area using deep reinforcement learning, Intell. Syst. Appl. 18 (2023) 200218.
[36] V. Padhye, K. Lakshmanan, A deep actor critic reinforcement learning framework for learning to rank, Neurocomputing. 547 (2023) 126314.
[37] G. Zare, N. Jafari, M. Hosseinzadeh, A. Sahafi, DAC-GCN: a dual actor-critic graph convolutional network with multi-hop aggregation for enhanced recommender systems, Acta Inf. Pragensia (2025).
[38] P. Agrawal, S. Raj, S. Saha, N. Onoe, A meta-learning based generative model with graph attention network for multi-modal recommender systems, Procedia Comput. Sci. 222 (2023) 581–590.
[39] Y. Kim, S. Kim, Y. Lee, J. Hong, Y. Lee, Multi-attention recommender system for non-fungible tokens, Eng. Appl. Artif. Intell. 137 (2024) 109179.
[40] M. Gridach, Hybrid deep neural networks for recommender systems, Neurocomputing. 413 (2020) 23–30.
[41] M.S. Mahmud, J.Z. Huang, S. García, Clustering approximation via a fusion of multiple random samples, Inf. Fusion 101 (2024) 101986.
[42] M. Salvi, H.W. Loh, S. Seoni, P.D. Barua, S. García, F. Molinari, U.R. Acharya, Multi-modality approaches for medical support systems: a systematic review of the last decade, Inf. Fusion 103 (2024) 102134.
[43] Y. Niu, Y. Su, S. Li, S. Wan, X. Cao, Deep adversarial autoencoder recommendation algorithm based on group influence, Inf. Fusion 100 (2023) 101903.
[44] P. Yang, Y. Xiao, W. Zheng, Y. Liu, C.-H. Hsu, A graph attention network with contrastive learning for temporal review-based recommendations, Appl. Soft. Comput. 159 (2024) 111652.
[45] V. Bhatia, DLSF: deep learning and semantic fusion based recommendation system, Expert. Syst. Appl. 250 (2024) 123900.
[46] S. Gheewala, S. Xu, S. Yeom, S. Maqsood, Exploiting deep transformer models in textual review based recommender systems, Expert. Syst. Appl. 235 (2024) 121120.
[47] N. Pourmoazemi, S. Maleki, A music recommender system based on compact convolutional transformers, Expert. Syst. Appl. 255 (2024) 124473.
[48] R. Patel, P. Thakkar, V. Ukani, CNNRec: convolutional neural network based recommender systems - a survey, Eng. Appl. Artif. Intell. 133 (2024) 108062.
[49] C. Zhang, X. Huang, J. An, S. Zou, Improving conversational recommender systems via multi-preference modelling and knowledge-enhanced, Knowl. Based. Syst. 286 (2024) 111361.
[50] B. Alhijawi, G. Al-Naymat, N. Obeid, A. Awajan, Novel predictive model to improve the accuracy of collaborative filtering recommender systems, Inf. Syst. 96 (2021) 101670.
[51] G. Albora, L.Rossi Mori, A. Zaccaria, Sapling similarity: a performing and interpretable memory-based tool for recommendation, Knowl. Based. Syst. 275 (2023) 110659.
[52] S.N. Anjiri, D. Ding, Y. Song, HyGate-GCN: hybrid-gate-based graph convolutional networks with dynamical ratings estimation for personalized POI recommendation, Expert. Syst. Appl. 258 (2024) 125217.
[53] H. Lu, Personalized music teaching service recommendation based on sensor and information retrieval technology, Measur.: Sensors 33 (2024) 101207.
[54] Y. Cui, H. Yu, X. Guo, H. Cao, L. Wang, RAKCR: reviews sentiment-aware based knowledge graph convolutional networks for personalized recommendation, Expert. Syst. Appl. 248 (2024) 123403.
[55] T. Zhou, H. Ye, F. Cao, Node-personalized multi-graph convolutional networks for recommendation, Neural Networks 173 (2024) 106169.
[56] S. Halder, K.H. Lim, J. Chan, X. Zhang, A survey on personalized itinerary recommendation: from optimisation to deep learning, Appl. Soft. Comput. 152 (2024) 111200.
[57] P. Zhou, Y. Cui, X. Guo, J. Wei, H. Cao, Phase-wise attention GCN for recommendation denoising, Appl. Soft. Comput. 163 (2024) 111910.
[58] R.v.d. Berg, T.N. Kipf, M. Welling, Graph convolutional matrix completion, arXiv preprint arXiv:1706.02263, (2017).
[59] X. Liu, H. Feng, X. Zhang, X. Zhou, A. Bouyer, Graph contrast learning for recommendation based on relational graph convolutional neural network, J. King Saud Univ. - Comput. Inf. Sci. 36 (2024) 102168.
[60] Y. Dong, Y. Liu, Y. Dong, Y. Wang, M. Chen, Multi-knowledge enhanced graph convolution for learning resource recommendation, Knowl. Based. Syst. 291 (2024) 111521.
[61] T. Wei, T.W.S. Chow, J. Ma, M. Zhao, ExpGCN: review-aware Graph Convolution Network for explainable recommendation, Neural Networks 157 (2023) 202–215.
[62] X. Wang, Q. Li, D. Yu, W. Huang, Q. Li, G. Xu, Neural Causal Graph collaborative filtering, Inf. Sci. (Ny) 677 (2024) 120872.
[63] X. Wang, X. He, M. Wang, F. Feng, T.S. Chua, Neural graph collaborative filtering, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), 2019, pp. 165–174.
[64] W. Huang, F. Hao, J. Shang, W. Yu, S. Zeng, C. Bisogni, V. Loia, Dual-LightGCN: dual light graph convolutional network for discriminative recommendation, Comput. Commun. 204 (2023) 89–100.
[65] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, M. Wang, 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, Association for Computing Machinery, Virtual Event, China, 2020, pp. 639–648.
[66] X. Wang, X. He, M. Wang, F. Feng, T.-S. Chua, Neural graph collaborative filtering, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Association for Computing Machinery, Paris, France, 2019, pp. 165–174.
[67] C. Liu, C. Yu, N. Gui, Z. Yu, S. Deng, SimGCL: graph contrastive learning by finding homophily in heterophily, Knowl. Inf. Syst. 66 (2024) 2089–2114.
[68] J.-C. Zhang, A.M. Zain, K.-Q. Zhou, X. Chen, R.-M. Zhang, A review of recommender systems based on knowledge graph embedding, Expert. Syst. Appl. 250 (2024) 123876.
[69] S. Lotfi, M. Mirzarezaee, M. Hosseinzadeh, V. Seydi, Detection of rumor conversations in Twitter using graph convolutional networks, Appl. Intell. 51 (2021) 4774–4787.
[70] J. Fang, X. Meng, X. Qi, A top-k POI recommendation approach based on LBSN and multi-graph fusion, Neurocomputing. 518 (2023) 219–230.
[71] L. Li, P. Wang, X. Zheng, Q. Xie, X. Tao, J.D. Velásquez, Dual-interactive fusion for code-mixed deep representation learning in tag recommendation, Inf. Fusion 99 (2023) 101862.
[72] C. Chen, W. Ma, M. Zhang, Z. Wang, X. He, C. Wang, Y. Liu, S. Ma, Graph heterogeneous multi-relational recommendation, Proc. AAAI Conf. Artific. Intell. 35 (2021) 3958–3966.
[73] Y. Yang, C. Huang, L. Xia, Y. Liang, Y. Yu, C. Li, Multi-behavior hypergraph-enhanced transformer for sequential recommendation, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, Washington DC, USA, 2022, pp. 2263–2274.
[74] J. Xu, C. Wang, C. Wu, Y. Song, K. Zheng, X. Wang, C. Wang, G. Zhou, K. Gai, Multi-behavior self-supervised learning for recommendation, in: Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval, 2023, pp. 496–505.
[75] W.X. Zhao, S. Mu, Y. Hou, Z. Lin, Y. Chen, X. Pan, K. Li, Y. Lu, H. Wang, C. Tian, Recbole: towards a unified, comprehensive and efficient framework for recommendation algorithms, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 4653–4664.
[76] W.X. Zhao, Y. Hou, X. Pan, C. Yang, Z. Zhang, Z. Lin, J. Zhang, S. Bian, J. Tang, W. Sun, RecBole 2.0: towards a more up-to-date recommendation library, in: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, 2022, pp. 4722–4726.

[77] Y. Hu, F. Xiong, S. Pan, X. Xiong, L. Wang, H. Chen, Bayesian personalized ranking based on multiple-layer neighborhoods, Inf. Sci. (Ny) 542 (2021) 156–176.
[78] C. Chen, M. Zhang, Y. Zhang, Y. Liu, S. Ma, Efficient neural matrix factorization without sampling for recommendation, ACM Trans. Inf. Syst. (TOIS) 38 (2020) 14.
[79] J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, X. Xie, Self-supervised graph learning for recommendation, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Association for Computing Machinery, Virtual Event, Canada, 2021, pp. 726–735.
[80] Y. Li, K. Liu, R. Satapathy, S. Wang, E. Cambria, Recent developments in recommender systems: a survey, IEEE Comput. Intell. Mag. 19 (2024) 78–95.
[81] L. Xia, C. Huang, C. Huang, K. Lin, T. Yu, B. Kao, Automated self-supervised learning for recommendation, in: Proceedings of the ACM Web Conference 2023, Association for Computing Machinery, Austin, TX, USA, 2023, pp. 992–1002.
[82] X. Cai, C. Huang, L. Xia, X. Ren, LightGCL: simple yet effective graph contrastive learning for recommendation, arXiv preprint arXiv:2302.08191, (2023).
[83] C. Li, L. Xia, X. Ren, Y. Ye, Y. Xu, C. Huang, Graph transformer for recommendation, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, Association for Computing Machinery, Taipei, Taiwan, 2023, pp. 1680–1689.
[84] Y. Zhang, Attention-based multilayer linear diffusion model, Acad. J. Comput. Inf. Sci. 7 (2024) 56–61.
[85] J. Hu, B. Hooi, S. Qian, Q. Fang, C. Xu, MGDCF: distance learning via markov graph diffusion for neural collaborative filtering, IEEe Trans. Knowl. Data Eng. 36 (2024) 3281–3296.
[86] J. Yu, X. Xia, T. Chen, L. Cui, N.Q.V. Hung, H. Yin, XSimGCL: towards extremely simple graph contrastive learning for recommendation, IEEE Trans. Knowl. Data Eng. 36 (2024) 913–926.