Acta Informatica Pragensia
2025, Volume 14, Issue 3, 340–364
https://doi.org/10.18267/j.aip.261

PRAGUE UNIVERSITY OF ECONOMICS AND BUSINESS

Article Open Access

DAC-GCN: A Dual Actor-Critic Graph Convolutional Network with Multi-Hop Aggregation for Enhanced Recommender Systems

Gholamreza Zare [1], **Nima Jafari** [2,3,4], **Mehdi Hosseinzadeh** [5], **Amir Sahafi** [6]

[1] Department of Computer Engineering, Qeshm Branch, Islamic Azad University, Qeshm, Iran
[2] Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran
[3] Future Technology Research Center, National Yunlin University of Science and Technology, Douliou, Taiwan
[4] Research Center of High Technologies and Innovative Engineering, Western Caspian University, Baku, Azerbaijan
[5] Pattern Recognition and Machine Learning Laboratory, School of Computing, Gachon University, Seongnam, Republic of Korea
[6] Department of Computer Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran

Corresponding author: Mehdi Hosseinzadeh (mehdi@gachon.ac.kr)

Editorial Record

First submission received:
October 30, 2024

Revisions received:
January 5, 2025
February 4, 2025

Accepted for publication:
February 6, 2025

Academic Editor:
Zdenek Smutny
Prague University of Economics and Business, Czech Republic

This article was accepted for publication by the Academic Editor upon evaluation of the reviewers' comments.

How to cite this article:
Zare, G., Jafari, N., Hosseinzadeh, M., & Sahafi, A. (2025). DAC-GCN: A Dual Actor-Critic Graph Convolutional Network with Multi-Hop Aggregation for Enhanced Recommender Systems. *Acta Informatica Pragensia*, 14(3), 340–364. https://doi.org/10.18267/j.aip.261

Abstract

Background: Recommender Systems (RSs) frequently face challenges in balancing exploration and exploitation, particularly in dynamic environments where user behaviors evolve over time. Traditional methods struggle to adapt to these complexities, limiting their effectiveness in real-world domains such as e-commerce, streaming services, and social networks.

Objective: The objective of this study is to introduce DAC-GCN, a Dual Actor-Critic Graph Convolutional Network, designed to enhance recommendation accuracy, ranking quality, and adaptability to evolving user preferences. DAC-GCN merges graph-based learning with Deep Reinforcement Learning (DRL) techniques to improve both short-term and long-term user-item interactions.

Methods: DAC-GCN utilizes a dual architecture with separate Graph Convolutional Networks (GCNs) for policy optimization and value estimation. It incorporates Multi-Hop Aggregation (MHA) to capture extended user-item dependencies and an attention mechanism to emphasize pivotal relationships. We evaluate DAC-GCN on benchmark datasets, including MovieLens 100K, MovieLens 1M, Amazon Subscription Boxes, Amazon Magazine Subscriptions, and Mod Cloth, using standard ranking metrics (Precision@K, Recall@K, NDCG@K, MRR@K, and Hit@K).

Results: Experimental results demonstrate that DAC-GCN consistently outperforms state-of-the-art baselines, showing significant improvements in recommendation accuracy, ranking quality, and robustness to shifting user behaviors. The model's ability to capture complex user-item interactions is greatly enhanced by MHA and attention mechanisms, while the dual architecture ensures training stability.

Conclusion: DAC-GCN offers a scalable, high-performance solution for modern recommender systems, effectively addressing challenges such as data sparsity and changing user preferences. By integrating graph-based methods with DRL, this study advances both the theory and practice of recommender systems and provides valuable insights for future research and practical applications.

Index Terms

Recommender system; Graph convolutional network; Actor-critic; Reinforcement learning; Multi-hop aggregation; Personalized recommendations.

1 INTRODUCTION

Recommender Systems (RSs) have become indispensable across modern digital platforms, powering personalized content delivery that boosts user engagement in e-commerce, social networks, healthcare, video, streaming services, and many other domains (Henriques and Pinto 2023). These systems analyze user preferences and behavioral patterns to generate relevant and accurate recommendations, significantly enhancing overall user satisfaction (Hsu 2024). Over the years, RSs have evolved from foundational techniques such as Collaborative Filtering (CF) (Engström et al. 2024), Kalman Filtering (KF) (Darbandi 2017) and Content-Based Filtering (CBF) (Saini and Singh 2024) to advanced frameworks incorporating Deep Learning (DL) and graph-based methods like Graph Convolutional Networks (GCNs) (Guo et al. 2024). GCNs have emerged as a powerful technique for modeling graph-structured data in various applications (Zhu et al. 2022), including RSs, where user–item interactions can be naturally represented as bipartite graphs. In this formulation, users and items serve as nodes, and their interactions define the edges (Xu et al. 2018). Through Multi-Hop Aggregation (MHA), GCN-based approaches effectively capture both local and global interaction patterns, enabling the modeling of high-order connectivity between users and items. However, despite these advantages, many existing GCN-based RS models are constrained by single network architectures (Zhao et al. 2024), limiting their ability to handle diverse tasks—such as recommendation policy generation and value estimation—and to adapt to the continuous evolution of user preferences.

Although recent advances in RS algorithms have led to improvements in several areas, significant challenges remain unsolved—particularly in striking a balance between exploration and exploitation and capturing long-term user preferences in dynamic environments (Alamdari et al. 2022). Much like other traditional approaches, today's RSs continue to grapple with **(I)** data sparsity (Heidari, Moradi, and Koochari 2022), where limited user–item interactions hinder recommendation accuracy; **(II)** cold-start problems (Kannout et al. 2024), where insufficient data for new users or items undermines effectiveness; **(III)** difficulty in modeling long-range dependencies in user–item interaction graphs (Lee et al. 2025), particularly in single-hop methods; **(IV)** limited adaptability to evolving user preferences; and **(V)** training instability in reinforcement learning–based systems, which can adversely affect performance and convergence. Resolving these issues is essential to develop robust and scalable RSs. Indeed, as RSs grow in complexity and scale, they must not only adapt to changing user behaviors but also maintain high accuracy and satisfaction (Zare et al. 2024). Meeting these evolving demands calls for sophisticated and hybrid algorithms capable of robustly tackling these challenges (Salvi et al. 2024).

Against this backdrop, Deep Reinforcement Learning (DRL) has shown promise to confront some of these challenges by framing the recommendation process as a sequential decision problem (Abnoosian, Farnoosh, and Behzadi 2023). One influential DRL architecture is the Actor–Critic (AC) framework, which divides learning into two main components: the actor, responsible for learning a recommendation policy, and the critic, which evaluates the quality of that policy using user feedback (Padhye and Lakshmanan 2023). Unfortunately, existing methods that integrate GCNs with AC frameworks often fail to fully harness the potential of DRL, primarily because they lean on shared network structures. Such designs cannot easily capture complex user–item interactions or achieve an optimal balance between exploration and exploitation—two crucial factors for delivering high-quality, dynamic recommendations.

In this research, we propose the Dual Actor–Critic Graph Convolutional Network (DAC-GCN), a novel framework designed to overcome these challenges through a combination of graph-based learning and RL techniques. The key contributions of this work are outlined below to highlight its novelty and practical relevance:

- **Novel DAC Architecture**: Introduction of a DAC framework leveraging separate GCNs for policy optimization and value estimation, enabling practical task specialization and improved learning outcomes. By decoupling policy generation and value estimation, DAC-GCN ensures task-specific learning that enhances its ability to adapt to evolving user preferences. The framework employs a static reward function to guide the learning process, providing stability in policy optimization.
- **MHA**: Incorporation of MHA to capture long-range dependencies in user-item interaction graphs, tackling challenges such as data sparsity and cold-start problems.
- **Attention Mechanisms**: Integration of attention mechanisms to dynamically prioritize significant interactions, enhancing the model's ability to focus on key user-item relationships (Zhang, Zain, et al. 2024).
- **Comprehensive Evaluation**: Extensive experiments conducted on multiple benchmark datasets, demonstrating significant performance improvements over state-of-the-art methods across key evaluation

metrics such as Hit Ratio (HR@10), Normalized Discounted Cumulative Gain (NDCG@10), and Mean Reciprocal Rank (MRR).

- **Practical Implications**: A design that is adaptable to dynamic user preferences and scalable for real-world applications, providing a robust solution for modern RSs.

To provide readers with a clear understanding of the study, the remainder of this paper is organized as follows: Section 2 reviews the related work, including advancements in RSs and graph-based learning techniques, as well as DRL in RSs. Section 3 details the proposed DAC-GCN architecture, including its components and methodology. Section 4 describes the experimental setup, evaluation metrics, and datasets used, followed by a presentation and analysis of the results. Also, it discusses the contributions of this work to scientific research and its practical implications. Finally, Section 5 concludes the paper and provides suggestions for future research directions.

2 RELATED WORK

Machine learning–driven RSs have progressed substantially over the past decade, chiefly due to their growing ability to capture intricate user–item relationships (Albora, Rossi Mori, and Zaccaria 2023). Early, foundational methods—such as CF and CBF—provided essential insights into personalization but often faltered in scenarios where user feedback was historical data insufficiently supported limited or newly introduced items. More recently, graph-based models leveraging GCNs have demonstrated the capacity to capture both local and global signals by representing users and items as interconnected nodes and propagating information across multiple hops. Nonetheless, many of these GCN-based solutions hinge on single-network architectures, which can restrict their effectiveness in optimizing different learning objectives (e.g., recommendation vs. feedback evaluation).

On another front, DRL has gained traction by framing recommendations as a sequential decision-making process. The AC architectures, in particular, elegantly manage the balance between exploration and exploitation. However, when applied to large-scale or dynamically evolving user–item graphs, these RL-based methods often lack the nuanced representational power necessary to pinpoint relevant interactions and to adapt swiftly to new data patterns.

Recognizing the individual strengths—and inherent limitations—of GCNs and AC frameworks, several recent studies have sought to combine them. Yet, many integrated approaches still rely on shared components that are ill-equipped to simultaneously handle task-specific requirements, such as precise policy optimization and robust value estimation. A more specialized design that can segregate these goals—and also incorporate advanced mechanisms like MHA and attention—stands to offer more stability and higher overall performance.

In this paper, we tackle these shortcomings through our proposed DAC-GCN, which fuses dual Actor–Critic modules with GCN-based embeddings. By separating policy and value estimation into distinct networks, DAC-GCN attains greater flexibility and more effective representation learning, ensuring reliable adaptations to evolving user preferences. Table 1 provides a comparative analysis between DAC-GCN and existing state-of-the-art approaches, summarizing how each method handles key components of recommendation—such as computational scalability, modeling complexity, and the ability to support diverse user–item interactions.

Table 1. *Comparison of DAC-GCN with state-of-the-art methods, highlighting key strengths and addressing main challenges.*

Method	Key Features	Limitations	How DAC-GCN Addresses The issues
Traditional Methods like CF	CF models user-item interactions	Struggles with data sparsity and cold-start problems	MHA captures broader dependencies
Graph-Based Models (e.g., LightGCN, NGCF)	Leverages graph structures for learning	Limited task specialization in single networks	DAC separates tasks effectively
DRL (e.g., Actor-Critic)	Balances exploration and exploitation	Lacks representational power for complex interactions	GCNs enhance interaction modeling and prioritization

To enhance clarity and offer a structured comparison, Table 2 presents the advantages and disadvantages of several existing RS approaches across key dimensions, including computational efficiency, scalability, handling of data sparsity, adaptability to dynamic environments, and the ability to capture long-range dependencies. The table also highlights advanced features such as attention mechanisms, MHA, and specialized learning for AC tasks.

From this comparison, it becomes clear that while methods like NGCF and LightGCN excel in scalability and local connectivity, they lack adaptability to evolving user preferences and do not provide specialized AC structures. Meanwhile, SimGCL and SGL incorporate attention mechanisms but forgo MHA and dedicated AC learning for policy and value estimation. In contrast, the proposed DAC-GCN framework uniquely integrates these critical features—DAC architecture, MHA, and attention mechanisms—thereby addressing the inherent shortcomings of prior methods. As shown in subsequent sections, this holistic approach enables DAC-GCN to achieve superior performance across various evaluation criteria.

***Table 2**. Enhanced Comparative Table of Existing Solutions and DAC-GCN.*

Comparative Aspects	Methods							
	BPR	**NGCF**	**LightGCN**	**NeuMF**	**EASE**	**SGL**	**SimGCL**	**DAC-GCN**
Computational Efficiency	Moderate	Moderate	High	Low	Very High	Low	Moderate	Moderate
Scalability	Moderate	Moderate	High	Low	High	Low	Moderate	High
Handling Data Sparsity	Low	Moderate	High	Moderate	Low	High	High	High
Adaptability to Dynamic Environments	Low	Low	Low	Moderate	Very Low	Low	Moderate	High
Capturing Long-Range Dependencies	Low	High	Moderate	Low	Low	High	High	Very High
Integration of Attention Mechanisms	No	No	No	No	No	Yes	Yes	Yes
MHA	No	Yes	No	No	No	No	No	Yes
Specialized Learning for Actor-Critic Tasks	No	No	No	No	No	No	No	Yes

2.1 GCNs for RSs

GCNs have become a foundational tool for learning graph-structured data (Kipf and Welling 2017), yielding significant breakthroughs in scenarios where entities and their interrelationships can be modeled as networks (Halder et al. 2024). In the domain of RSs, GCN-based methods leverage the intrinsic structure of user–item interaction graphs, where users and items are nodes, and their relationships (e.g., ratings, clicks) form edges. Through MHA, GCNs effectively capture higher-order connectivity patterns, substantially boosting recommendation accuracy by uncovering both local and global interaction dynamics (Anjiri, Ding, and Song 2024).

Over the past few years, numerous GCN architectures have been explored to enhance recommendation performance by learning intricate patterns within interaction graphs (Zhou, Ye, and Cao 2024). For example, GC-MC introduced a matrix completion framework utilizing GCN embeddings for users and items (Wei et al. 2023), while NGCF extended this principle by modeling high-order connectivity, thereby improving predictions in data-sparse scenarios. Despite these advancements, traditional GCN-based methods often rely on single-network architectures that struggle to simultaneously address the distinct objectives of policy learning and value estimation in RSs (Huang et al. 2023).

In this work, we build upon these advances by integrating two specialized GCNs within a DAC architecture. The Actor GCN targets recommendation policy learning by capturing intricate interaction patterns, while the Critic GCN focuses on value estimation, assessing the long-term rewards of actions. By decoupling these tasks, the DAC-GCN framework addresses the evolving challenges of dynamic recommendation environments, including data sparsity, shifting user preferences, and cold-start problems.

To further enhance the effectiveness of the DAC-GCN, we incorporate MHA to capture long-range dependencies and an attention mechanism to emphasize pivotal interactions. These design choices enable the model to adapt dynamically, ensuring high-quality, personalized recommendations across diverse datasets and user behaviors. Collectively, the DAC-GCN framework exemplifies the state-of-the-art integration of GCNs and reinforcement learning, achieving superior accuracy, adaptability, and interpretability in RSs.

2.2 DRL in RSs

RSs have evolved significantly with the advent of DRL, a paradigm that reframes recommendation tasks as sequential decision-making problems. Unlike traditional approaches, which often operate in static environments, DRL-based RSs adapt dynamically to user behaviors, enabling the system to provide more personalized and context-aware recommendations.

DRL leverages the principles of RL to optimize long-term rewards, addressing critical limitations in traditional recommendation methods, such as data sparsity, cold-start issues, and the exploration-exploitation trade-off. By treating user interactions as state transitions in a Markov Decision Process (MDP) (Sutton 2018), DRL-based RSs dynamically refine recommendation policies based on user feedback, ensuring continuous improvement in recommendation quality.

The AC framework has emerged as a cornerstone of DRL-based RSs. This architecture divides the task into two complementary components: The **Actor** component is responsible for learning a policy that determines which items to recommend based on the current state (e.g., a user's interaction history), and the **Critic** part evaluates the quality of the Actor's policy by estimating the expected rewards for each recommendation. This separation enhances both exploration and exploitation, allowing the system to balance the discovery of new preferences with the reinforcement of known user interests.

DRL offers transformative contributions to RSs by addressing several core challenges. Unlike traditional CF and CBF methods, DRL explicitly models the sequential nature of user interactions, recognizing that each recommendation action influences future states and rewards. This approach facilitates a more holistic optimization of the user experience. Moreover, DRL excels in capturing dynamic user preferences; as user behaviors evolve, the system continuously updates its policies to ensure relevance. Through its reward-driven mechanism, DRL effectively balances the exploration of new items with the exploitation of known user interests, enhancing both diversity and accuracy in recommendations. Additionally, its reliance on cumulative rewards rather than immediate feedback renders it inherently robust to sparse and noisy interaction data, making it highly effective in scenarios where explicit user feedback is limited.

Advancements in DRL architectures further amplify its utility in RSs. Policy gradient and Q-learning-based models, navigate large knowledge graphs to improve recommendation relevance and interpretability. AC frameworks, like our DAC-GCN, separate policy optimization and value estimation into dedicated networks, leading to improved training stability and higher-quality recommendations. Hierarchical DRL approaches decompose the recommendation process into manageable sub-tasks, enhancing scalability in complex environments. Model-based DRL architectures integrate predictive models of user behavior to streamline policy updates, reducing computational demands while maintaining high performance. Together, these innovations illustrate the versatility and adaptability of DRL in addressing the evolving demands of modern RSs.

While DRL has demonstrated significant promise in RSs, it also presents challenges. The computational complexity of training DRL models, particularly in large-scale environments, remains a concern. Moreover, reward design plays a critical role in guiding policy optimization, yet defining a reward function that aligns with diverse user satisfaction metrics can be challenging. Future research should focus on lightweight architectures, adaptive reward mechanisms, and hybrid models that integrate DRL with other deep learning techniques, such as GNNs and transformers.

In this work, DRL serves as the backbone of the proposed DAC-GCN framework. By embedding DRL within a graph convolutional structure, DAC-GCN benefits from the sequential decision-making capabilities of DRL and the representational power of GCNs. This integration enables the framework to capture both immediate and long-term user preferences, achieving superior recommendation performance in dynamic, real-world environments.

2.3 MHA and Attention Mechanisms

Incorporating MHA in GCNs has become a key technique for enhancing the expressiveness of graph-based models. By aggregating information from multiple hops away in the graph, models can capture long-range dependencies, which are essential for understanding complex relationships between users and items. MHA has been particularly beneficial in RSs, where user preferences are often influenced by indirect interactions that are not immediately apparent in their direct interaction history. For instance, LightGCN demonstrated that simplifying GCNs by focusing on neighborhood aggregation over multiple hops could significantly improve recommendation accuracy.

Building on this idea, DAC-GCN employs MHA to capture both local and global user-item interaction patterns. By aggregating information from distant neighbors in the interaction graph, DAC-GCN can model long-range dependencies that are often missed by single-hop approaches. This enables the model to provide more comprehensive recommendations by considering both immediate preferences and broader behavioral patterns.

Additionally, attention mechanisms have been increasingly integrated into GCN-based models to weigh the importance of different interactions dynamically. Graph Attention Networks (GATs) introduced attention into graph models, allowing the network to assign different importance levels to various neighbors (Cui et al. 2024). This has proven effective in focusing the model on the most relevant interactions for recommendation tasks. In DAC-GCN, we integrate an attention mechanism to ensure that the model prioritizes key interactions within the user-item graph, further improving the relevance of recommendations.

3 PROPOSED METHOD

In this section, we introduce the DAC-GCN framework, which tackles the challenges posed by dynamic user-item interactions, long-term preference modeling, and the need for policy optimization in RSs. The method proceeds through several stages to optimize recommendations:

- **Data Representation:** We begin by structuring the raw input data as a user-item bipartite graph, providing a natural representation of interactions such as clicks, ratings, or purchases.
- **DAC Architecture:** The core of DAC-GCN features two distinct GCNs—one for the Actor (policy optimization) and one for the Critic (value estimation). This division of labor reduces training conflicts and enhances both accuracy and stability.
- **MHA & Attention:** MHA captures higher-order dependencies by collecting signals from distant neighbors in the graph. Meanwhile, the attention mechanism ensures that key interactions receive greater emphasis, further refining user and item representations.
- **Recommendation Generation & Evaluation:** The system generates recommendations based on the Actor's learned policy and then assesses these recommendations through the Critic, guided by a static reward function. This reward function provides consistent feedback for policy updates, improving convergence and performance.

The subsections that follow detail each component of DAC-GCN, illustrating how they collectively tackle the inherent complexities of modern RSs.

3.1 Overview of DAC-GCN Architecture

The core principle of DAC-GCN is to split policy optimization and value estimation into two distinct modules, forming a dual GCN-based Actor-Critic (AC) framework. This decoupling allows the Actor GCN to concentrate on learning recommendation policies while the Critic GCN focuses on assessing the quality of those policies through value estimation. In addition, MHA is used to capture higher-order connections in the user-item graph, and an attention mechanism dynamically adjusts the importance of different interactions. A static reward function further stabilizes the learning process by providing consistent feedback for policy updates.

Graph convolution underpins DAC-GCN by leveraging the inherent graph structure of user-item interactions to enhance recommendation quality. Its key functions include:

- **Capturing Higher-Order Connectivity**: Graph convolution aggregates information from neighboring nodes across multiple hops, thereby identifying both direct and indirect user-item relationships. This enables more nuanced learning of user preferences and item characteristics.
- **Learning Enriched Node Representations**: By combining node features from various neighbors, graph convolution produces richer embeddings for users and items. These embeddings provide a strong basis for generating accurate recommendations.
- **Handling Sparsity and Cold-Start Problems**: In sparse data settings, MHA allows the model to exploit indirect connections. This approach improves performance for users or items with limited interaction histories by inferring preferences through shared neighbors.

The dual AC structure in DAC-GCN ensures that each component targets its specialized objective. The Actor optimizes the recommendation policy, while the Critic evaluates this policy's expected long-term reward. This separation overcomes the issue of conflicting gradients often seen in single-network architectures, improving both training stability and overall performance.

Figure 1 highlights the differences between RSs based on Deep Learning (DL-RSs) and those utilizing Deep Reinforcement Learning (DRL-RSs):

- **DL-RSs**: Typically rely on supervised learning to map user-item interactions to predictions (Dilekh et al. 2024). They excel in relatively static environments but struggle to adapt when user preferences shift, primarily due to limited exploration mechanisms.
- **DRL-RSs:** Formulate recommendation as a sequential decision-making problem, balancing the use of known preferences (exploitation) with identifying new interests (exploration). By employing an AC framework, the recommender learns continuously from user feedback, adapting to evolving behaviors over time.

In the context of DAC-GCN, the Actor decides which items to recommend, while the Critic assesses the likely reward of those recommendations. This design allows DAC-GCN to effectively handle dynamic user preferences and deliver more personalized, context-aware recommendations.

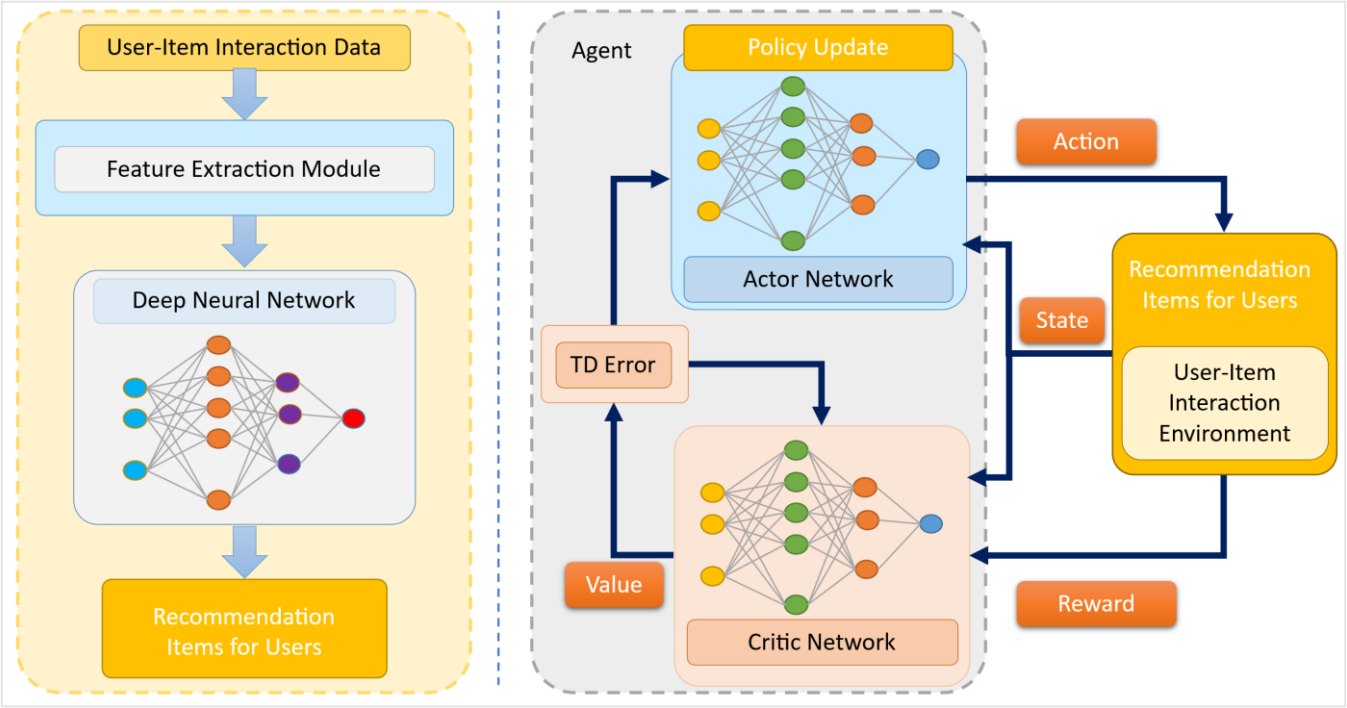

Figure 1. *Deep Learning vs. Deep Reinforcement Learning in RSs. (Left side shows a simple flow diagram of a Deep Learning-based RS and the Right side shows and AC-based RS).*

3.2 DAC Architecture

The DAC framework in DAC-GCN separates policy optimization and value estimation into two specialized networks, known as the Actor and the Critic. This division helps overcome the cold-start problem and improves the model's ability to generalize to new users and items.

- **Actor Network**: This is responsible for policy optimization; the Actor-network explores the user-item interaction graph to generate personalized recommendations.
- **Critic Network**: It focuses on value estimation by predicting the expected reward for the Actor's recommendations, providing continual feedback that refines the policy.

By dedicating specialized resources to each task, DAC-GCN avoids the conflicting objectives often found in single-network approaches. In a traditional single-network architecture, the model simultaneously handles policy learning and value estimation, which can lead to instability and poor exploration-exploitation balance. In contrast, the DAC design allocates distinct capacities for each objective, enhancing adaptability in dynamic recommendation environments. The key advantages to this technique include:

- **Specialized Learning Tasks**: By separating policy optimization (Actor) and value estimation, it allows each network to concentrate on its respective objective and avoids the conflicting gradients in single-network models.
- **Improved Feedback Mechanism**: The Critic's continuous evaluation of the Actor's actions provides a robust signal to refine the policy iteratively. This feedback loop ensures the recommendations are regularly updated to align with user preferences.
- **Adaptability to Dynamic Preferences**: Because the Actor and Critic are distinct, the model can respond more fluidly to shifts in user behavior. As the Actor explores new recommendations, the Critic re-evaluates these actions to stay current with changing preferences.
- **Enhanced Representation Learning**: By maintaining separate GCN layers for policy generation and value estimation, DAC-GCN can learn richer user and item embeddings. The Actor focuses on capturing diverse user-item relationships, while the Critic emphasizes how those relationships contribute to reward maximization.
- **Stabilized Training Process**: It decouples policy and value updates, reduces the risk of oscillations, and accelerates convergence by creating a more stable training procedure.

The Actor learns a policy $\pi(a|s)$, that maps states s (representing user-item interactions) to actions a (recommended items). The Critic, meanwhile, estimates the action-value function $Q_\pi(s,a)$ to guide and refine the Actor's behavior.

- **Actor Objective:** The goal of the Actor is to maximize the expected cumulative reward (Schulman et al. 2015):

$$\max_{\pi} \mathbb{E}_\pi \left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t) \right] \tag{1}$$

  where γ is the discount factor, $r(s_t, a_t)$ is the immediate reward, and T is the horizon length.

- **Critic Objective:** The Critic estimates the action-value function $Q_\pi(s,a)$, given by:

$$Q_\pi(s_t, a_t) = \mathbb{E}_\pi[r(s_t, a_t) + \gamma Q_\pi(s_{t+1}, a_{t+1})] \tag{2}$$

  This value function guides the Actor to select actions that maximize long-term rewards (Schulman et al. 2015).

To model user-item interactions, DAC-GCN constructs a bipartite graph where nodes represent users or items, and edges represent interactions (e.g., clicks, ratings, or purchases). Both the Actor and Critic embed these nodes through specialized GCN layers:

- **Actor GCN**: Learns a recommendation policy by capturing local and global user-item dependencies. This MHA integrates information from distant neighbors in the graph, allowing the Actor to infer latent preferences that enhance recommendation quality. Formally, each layer updates node embeddings $h_v^{(l+1)}$ using (Kipf and Welling 2017):

$$h_v^{(l+1)} = \sigma\left(\sum_{u \in \mathcal{N}(v)} \frac{1}{\sqrt{|N(u)||N(v)|}} W^{(l)} h_u^{(l)}\right) \quad (3)$$

where $\mathcal{N}(v)$ is the set of neighbors of node v, $W^{(l)}$ is the weight matrix at layer l, and $\sigma(\cdot)$ denotes a nonlinear activation (e.g., ReLU).

- **Critic GCN**: Evaluates the Actor's recommendations by estimating expected rewards. In addition to user-item interactions, the Critic GCN can incorporate side information like demographics or item attributes, leading to more precise value estimation.

By jointly optimizing these Actor and Critic GCNs through reinforcement learning, DAC-GCN continuously updates its policy and value estimates. This synergistic design enhances the system's ability to adapt to user behavior and effectively personalize recommendations over time.

3.3 MHA

To capture both short-term and long-term user preferences, we use MHA in the GCN layers. This approach extends beyond immediate neighbors by gathering information from nodes multiple hops away, allowing the model to uncover complex user-item dependencies that are often overlooked in single-hop interactions. For example, a user's interest in an unfamiliar item might be inferred through shared preferences with other users several hops away in the interaction graph. By integrating these indirect relationships, MHA enriches the learned embeddings and enhances recommendation quality, particularly under data sparsity. Its specific contributions include:

- **Capturing Long-Range Dependencies**: MHA propagates information across extended neighborhoods, enabling the model to identify indirect relationships between users and items. This higher-order connectivity allows the system to infer preferences even when direct interactions are lacking.
- **Mitigating Data Sparsity**: By incorporating signals from distant nodes, MHA strengthens the representations for users and items with limited direct interactions. This approach helps ensure robust embeddings for sparse nodes, improving overall recommendation performance.
- **Enhancing Representation Learning**: Combining local and global interaction patterns into comprehensive embeddings enables the model to capture complex user-item relationships that simple, single-hop methods might miss. As a result, DAC-GCN can deliver diverse and more accurate recommendations.
- **Balancing Information Flow**: The aggregation process balances the contributions of immediate neighbors against those of distant neighbors, preventing overfitting to local patterns and maintaining global context. This balance is crucial for reflecting both short-term and long-term user preferences.

Formally, after k-hop aggregation, the embedding for a user node u is defined as:

$$h_u^{(k)} = \sum_{v \in \mathcal{N}_k(u)} \alpha_{uv} h_v \quad (4)$$

where $\mathcal{N}_k(u)$ is the set of neighbors at hop distance k, and α_{uv} represents the attention weight assigned to each neighbor based on its importance to the user's preferences (Wang et al. 2019).

3.4 Attention Mechanism

The attention mechanism in DAC-GCN builds on MHA by selectively highlighting the most influential interactions, ensuring that the propagated information remains both relevant and meaningful. In particular, it assigns higher weights to interactions that exert a stronger impact on recommendation outcomes while mitigating the influence of noisy or marginal data. By dynamically adapting these interaction weights based on context, the attention mechanism allows DAC-GCN to capture evolving user preferences with greater flexibility and accuracy. Its specific purposes include:

- **Prioritizing Relevant Interactions**: The attention mechanism focuses on user-item interactions that most affect recommendation results. For example, if a user predominantly engages with items in a specific category, the mechanism will emphasize these interactions, ensuring that subsequent recommendations remain aligned with the user's core interests.

- **Mitigating Noise**: Real-world datasets often contain interactions that are transient or of limited importance. The attention mechanism downweighs such interactions, preventing the model from overfitting to noisy signals and improving overall recommendation quality.
- **Dynamic Adaptation**: Because the attention mechanism adjusts interaction weights on the fly, it facilitates rapid adaptation to diverse user behaviors and changing preferences. This enables DAC-GCN to remain robust and responsive in dynamic environments.

By assigning higher weights to influential interactions, the attention mechanism not only enriches user and item embeddings but also improves DAC-GCN's ability to capture subtle or niche preferences. This leads to more personalized recommendations and a better understanding of individual user behavior. Moreover, attention scores make the model more interpretable by revealing which interactions are most crucial for its decisions. Such transparency can guide practitioners in refining model components and improving system performance.

Formally, the attention weight α_{uv} is computed as:

$$\alpha_{uv} = \frac{exp\left(\text{LeakyReLU}(a^T[h_u||h_v])\right)}{\sum_{w\in\mathcal{N}(u)} exp\left(\text{LeakyReLU}(a^T[h_u||h_w])\right)} \quad (5)$$

where a is the learnable attention vector, and $||$ denotes concatenation (Veličković et al. 2017). This formulation ensures that each neighbor's contribution is weighted according to its importance to the user's preferences, thereby strengthening DAC-GCN's overall recommendation performance.

3.5 Static Reward Function

In the DAC-GCN framework, the static reward function serves as a cornerstone for both training stability and policy optimization. By providing a fixed, predefined feedback signal (e.g., +1 for a user interaction and 0 otherwise), it mitigates the volatility introduced by fluctuating or noisy user feedback. This consistency eliminates frequent reward updates based on changing contexts or temporal factors, enabling smoother convergence in the Actor-Critic process.

A key benefit of the static reward is its capacity to guide policy optimization without inducing abrupt policy shifts. Because the reward signal remains stable, the Actor network can focus on incremental improvements rather than reacting to transient patterns, thus reducing the likelihood of overfitting. This regularized exploration helps the learned policy remain robust across diverse scenarios, even in dynamic environments.

Formally, we define the static reward in DAC-GCN as:

$$r(s_t, a_t) = \begin{cases} +1 & \text{if the user interacts with} \\ & \text{the recommended item} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Such a straightforward, predefined feedback mechanism prevents large policy oscillations, allowing the model to refine recommendations iteratively.

To assess how well this static reward aligns with overall recommendation objectives, we employ standard evaluation metrics, including Precision@K, Recall@K, and NDCG@K. These metrics quantify the relevance of the recommended items and reflect the degree to which the reward function successfully drives the Actor network toward accurate predictions. In addition, metrics such as Mean Reciprocal Rank (MRR@K) and Hit Ratio (Hit@K) help evaluate ranking performance, further confirming the reward's effectiveness in encouraging high-quality recommendations. By consistently reinforcing reward signals aligned with these metrics, the static reward function enables DAC-GCN to achieve stable and robust policy learning in real-world recommendation scenarios.

3.6 Training Process and Algorithm

The training of DAC-GCN follows an iterative procedure, where the Actor and Critic networks are updated based on the observed rewards. The optimization process is based on gradient ascent for the Actor and gradient descent for the Critic. The loss for the Critic network is the temporal difference (TD) error, defined as (Chen et al. 2023):

$$L_{\text{Critic}} = (r(s_t, a_t) + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))^2 \quad (7)$$

The policy gradient for the Actor is computed using the policy gradient theorem (Sutton et al. 1999):

$$\nabla_\theta J(\theta) = \mathbb{E}_\pi[\nabla_\theta log\ \pi_\theta(a_t|s_t)Q_\pi(s_t, a_t)] \quad (8)$$

While the incorporation of MHA and attention mechanisms introduces additional computational overhead, these components significantly enhance the model's ability to capture nuanced user-item interactions. MHA improves the representation of both local and global patterns by capturing long-range dependencies, and the attention mechanism dynamically prioritizes the most relevant relationships in the user-item graph. These enhancements ensure that the model effectively balances complexity with improved recommendation quality.

Figure 2 shows the flow diagram of our proposed method in a high-level structure.

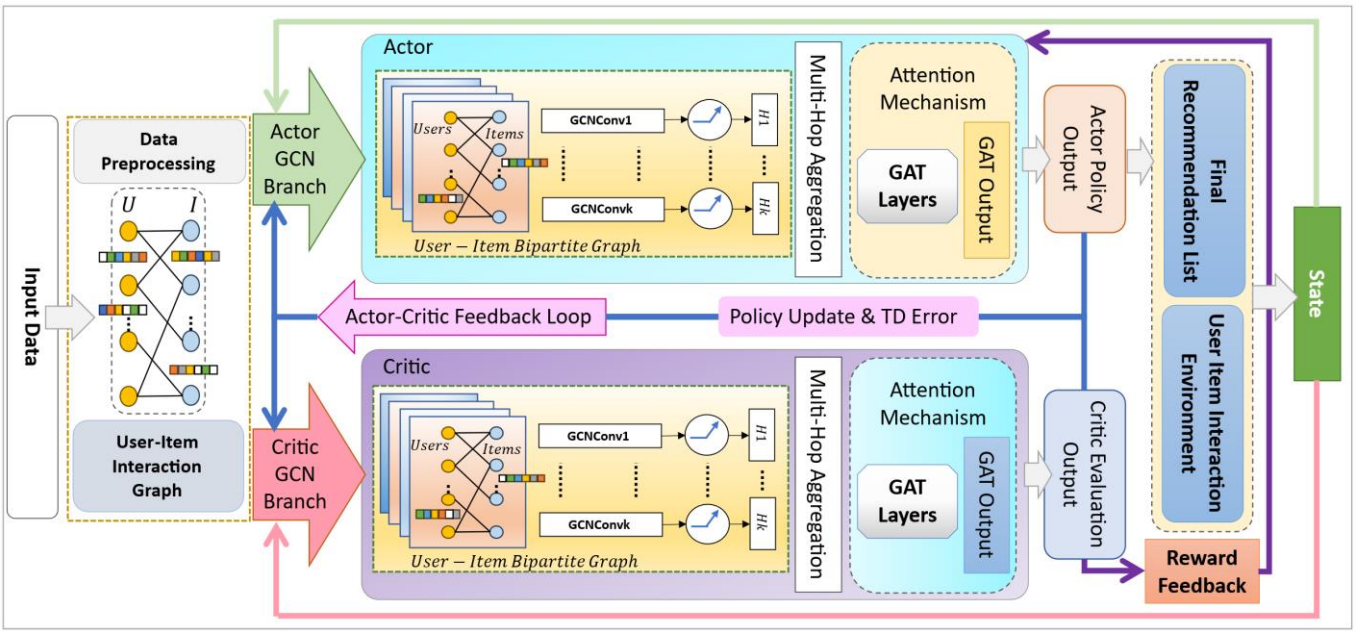

Figure 2. *Flow Diagram of the Proposed DAC-GCN Method.*

Algorithm 1 shows the training process of the proposed DAC-GCN for RSs. The algorithm begins by initializing the Actor and Critic GCN networks with random weights. The training proceeds iteratively over multiple episodes, where for each user-item interaction, the Actor selects a recommended item based on the user's current state (interaction history), and the environment provides a reward based on user feedback. The Critic evaluates both the current and next state-action pairs, and the TD target is computed to guide the Critic's update. The Critic is updated to minimize the TD error, while the Actor is updated using a policy gradient to maximize the expected reward. The algorithm continues this process, gradually improving the Actor's ability to recommend better items and the Critic's capacity to evaluate them, ultimately returning the trained models.

Table 3. *Hyperparameters and settings of the DAC-GCN model, including GCN configurations, RL components, and evaluation metrics.*

Parameter	Value	Description
`embedding_size`	64	Size of the embedding vectors for nodes.
`n_hops`	3	Number of hops in the graph convolution network.
`node_dropout`	0.3	Dropout rate for nodes during training.
`message_dropout`	0.3	Dropout rate for messages passed in the graph.
`dropout`	0.3	General dropout rate applied to layers.
`edge_dropout_rate`	0.1	Dropout rate for graph edges during training.
`reg_weight`	1e-5	Regularization weight for preventing overfitting.
`n_layers`	3	Number of layers in the model.
`actor_lr`	0.001	Learning rate for the actor model.
`critic_lr`	0.01	Learning rate for the critic model.

Parameter	Value	Description
train_batch_size	8192	Batch size used for training.
eval_batch_size	4096	Batch size used for evaluation.
gamma	0.99	Discount factor for reward in reinforcement learning.
num_heads	8	Number of attention heads in the multi-head attention mechanism.
temperature	0.2	Temperature parameter for scaling logits during sampling.
learning_rate	0.001	The learning rate for the optimizer.
epochs	200	Number of training epochs.
batch_size	8192	Batch size used for processing data.
optimizer	Adam	Optimizer used for training.
loss_type	BPR	Type of loss function used, Bayesian Personalized Ranking (BPR).
metrics	Recall, MRR, NDCG, Hit, Precision	Evaluation metrics considered for validation.
topk	10	Number of top recommendations to evaluate metrics.

Algorithm 1. *DAC-GCN for RSs.*

Input: G_{UI}: User-item bipartite graph, γ: Discount factor, α_{actor}: Learning rate for Actor, α_{critic}: Learning rate for Critic, T: Number of episodes, d: Embedding dimension

Output: Trained Actor GCN, Trained Critic GCN

Initialize Actor GCN π_θ and Critic GCN Q_ϕ with random weights
for each episode $t = 1,2, \dots, T$ **do**
for each user u and item i interaction in G_{UI} **do**
    Set current state $s_t \leftarrow \text{get\_state}(u)$ (user interaction history)
    Actor selects action: $a_t \leftarrow \pi_\theta(s_t)$ (recommend item)
    Simulate environment step:
        Receive reward r_t and next state $s_{t+1} \leftarrow \text{environment\_step}(u, a_t)$
    Critic evaluates current state-action pair:
        $Q_\phi(s_t, a_t) \leftarrow$ Critic GCN
    Actor selects next action: $a_{t+1} \leftarrow \pi_\theta(s_{t+1})$
    Critic evaluates next state-action pair:
        $Q_\phi(s_{t+1}, a_{t+1}) \leftarrow$ Critic GCN
    Compute TD target:
        TD Target $= r_t + \gamma Q_\phi(s_{t+1}, a_{t+1})$
    Compute Critic loss:
        $L_{\text{critic}} = (\text{TD Target} - Q_\phi(s_t, a_t))^2$
    Update Critic network:
        $\phi \leftarrow \phi - \alpha_{\text{critic}} \nabla_\phi L_{\text{critic}}$
    Compute Actor loss:
        $L_{\text{actor}} = -\log(\pi_\theta(a_t|s_t)) \cdot Q_\phi(s_t, a_t)$
    Update Actor network:
        $\theta \leftarrow \theta - \alpha_{\text{actor}} \nabla_\theta L_{\text{actor}}$
end for
end for
return Trained Actor GCN π_θ, Trained Critic GCN Q_ϕ

To enhance the clarity and reproducibility of our proposed DAC-GCN framework, we summarize the key hyperparameters and settings used in the model in Table 3. It outlines the essential parameters, including those governing the GCNs, RL components, and evaluation metrics. These settings were chosen based on prior work and fine-tuning experiments to optimize model performance across the benchmark datasets.

DAC-GCN's training poses several computational challenges due to its complex architecture. MHA, for instance, involves propagating information through multiple layers of the user-item graph, significantly increasing memory usage and training time—especially for large-scale datasets. Similarly, attention mechanisms, which dynamically weight user-item interactions, add extra matrix operations during training and inference. Balancing these components alongside DAC networks introduces substantial model complexity and can compromise training stability if not carefully managed. Moreover, scaling DAC-GCN for larger datasets or real-time recommendations intensifies these challenges further, since it requires handling substantial computational loads from both graph-based operations and RL optimization.

To overcome these challenges, we propose several practical strategies that can be adopted when developing and training DAC-GCN. First, employing graph sampling techniques can limit the size of the computation graph, thereby reducing memory requirements and training times without substantially compromising performance. Next, exploring sparse attention mechanisms helps focus computations on the most impactful interactions, improving overall efficiency. Regularization and gradient clipping are also recommended to counter overfitting and prevent exploding gradients, thus maintaining training stability. Finally, a batch-processing approach distributes the computational load more effectively, facilitating scalability and reducing the time needed for each training epoch.

4 RESULTS AND DISCUSSION

In this section, we present the experimental results of our proposed DAC-GCN, comparing its performance with state-of-the-art models across several benchmark datasets. We evaluate DAC-GCN on key recommendation metrics. The experiments are designed to evaluate the ability of DAC-GCN to overcome challenges such as data sparsity, relevance ranking, and capturing complex user-item interactions. Additionally, we conduct an ablation study to assess the impact of individual components on the model's overall performance.

4.1 Datasets, Baseline Methods, and Evaluation Metrics

To evaluate the performance of the proposed DAC-GCN model, we conducted experiments on several widely-used benchmark datasets sourced from RecBole (Zhao et al. 2022), each of which represents different domains and interaction types. The selected datasets include MovieLens 100K, MovieLens 1M, Amazon Subscription Boxes, Amazon Magazine Subscriptions, and Mod Cloth. These datasets are well-known in the RS research community. These datasets were chosen due to their diversity in domain, interaction types, and sparsity levels, providing a comprehensive benchmark for evaluating the robustness and adaptability of DAC-GCN. A detailed overview of the selected datasets is provided in Table 4.

Table 4. *Characteristics of the datasets used for evaluating DAC-GCN, highlighting varying interaction densities and challenges for assessing model robustness.*

Dataset	#Users	#Items	#Interaction	Reason
ML-100K	943	1,682	100,000	Well-balanced datasets with sufficient interactions, suitable for assessing DAC-GCN's general recommendation quality and scalability under standard conditions.
ML-1M	6,040	3,952	1,000,209	
Amazon Subscription Boxes	5,223	3,874	82,456	Sparse datasets with limited interactions, used to evaluate DAC-GCN's ability to address data sparsity and enrich representations via MHA.
Amazon Magazine Subscriptions	3,900	2,748	57,823	
Mod Cloth	47,958	1,378	82,790	A dataset with a dense user-to-item ratio, ideal for testing the model's scalability and adaptability to large-scale recommendation scenarios.

To provide a comprehensive comparison, we selected several state-of-the-art baseline methods. The selected baselines represent a variety of approaches—ranging from traditional CF methods to advanced graph-based techniques—allowing us to evaluate DAC-GCN's performance comprehensively against diverse methodologies. The selected baselines are as follows:

- **BPR (Bayesian Personalized Ranking)** (Hu et al. 2021): A popular CF method that optimizes pairwise ranking for implicit feedback. This method serves as a strong baseline for comparing ranking performance.
- **LightGCN (Light Graph Convolutional Network)** (Zhang, Zhu, et al. 2024): A highly efficient GCN-based method that focuses solely on neighborhood aggregation without feature transformations, making it ideal for evaluating graph-based interactions.
- **NeuMF (Neural Matrix Factorization)** (Chen et al. 2020): A widely-used deep learning-based method that combines matrix factorization with multi-layer perceptrons, enabling a comparison of DAC-GCN's graph-based learning against traditional deep learning models.
- **NGCF (Neural Graph Collaborative Filtering)** (Wang et al. 2019): Another GCN-based approach that captures high-order connectivity in user-item graphs, providing a valuable baseline to assess the graph convolutional aspects of DAC-GCN.
- **EASE (Embarrassingly Shallow Autoencoders)** (Steck 2019): A shallow CF model that demonstrates strong performance in sparse data environments, offering a contrast to deep and graph-based methods.
- **SGL (Self-supervised Graph Learning)** (Wu et al. 2021): A state-of-the-art model that incorporates self-supervised learning into graph-based recommendation tasks, providing an advanced benchmark for graph-based learning approaches.
- **SimGCL** (Liu et al. 2024): A similarity-based graph contrastive learning model designed to improve the robustness of CF methods.

To assess the performance of DAC-GCN and the baseline methods, we employed **ranking metrics**. These metrics provide a comprehensive evaluation of each method's ability to predict user preferences and rank relevant items effectively. The metrics used are defined as follows:

- Precision@K: The proportion of relevant items among the top-K recommendations, measuring the accuracy of predictions.
- Recall@K: The fraction of all relevant items successfully retrieved within the top-K recommendations, reflecting the model's ability to retrieve all relevant items.
- NDCG@K (Normalized Discounted Cumulative Gain): Evaluates the ranking quality by considering both the relevance and the position of recommended items, assigning higher importance to relevant items ranked higher in the list.
- MRR@K (Mean Reciprocal Rank): Calculates the reciprocal rank of the first relevant item in the recommendation list, averaged across all users, providing insight into the model's ability to rank relevant items at the top.
- Hit@K: Measures the fraction of users for whom at least one relevant item is present in the top-K recommendations, indicating the success rate of recommendations.

These metrics are widely used in RS research as they comprehensively evaluate the accuracy, relevance, and ranking quality of recommendations. By combining these metrics, we can capture both the precision of individual recommendations and the overall system effectiveness. For example, Precision@K and Recall@K assess the relevance and retrieval quality of the recommendations, ensuring that the top-K items align with user preferences. NDCG@K and MRR@K focus on ranking performance, emphasizing the importance of correctly ordering relevant items, while Hit@K provides an overall success rate of including relevant items in the top-K recommendations. All metrics are evaluated with $K = 10$, a commonly used cutoff in RS algorithms. These metrics provide a robust evaluation of each method's ability to generate accurate and relevant recommendations across different datasets. By defining and applying these metrics, we ensure a comprehensive assessment of the proposed DAC-GCN model and its effectiveness compared to baseline methods.

4.2 Results Analysis

Table 5 provides a comprehensive comparison of DAC-GCN against several baseline models across five datasets, revealing notable advantages in terms of Precision@10, Recall@10, NDCG@10, MRR@10, and Hit@10. DAC-GCN

consistently outperforms methods such as BPR, LightGCN, NeuMF, and NGCF, demonstrating its capacity to capture user-item interactions more effectively and deliver high-quality recommendations. Notably, it achieves top Precision@10 scores on the ML-100K (0.2073) and ML-1M (0.2159) datasets, surpassing both NeuMF and NGCF. This illustrates DAC-GCN's ability to retrieve highly relevant items in the top 10 positions, which is particularly impressive given the complexity of these datasets. Moreover, DAC-GCN shows a pronounced improvement on the Mod Cloth dataset, achieving a Precision@10 of 0.0195, thereby outperforming LightGCN and NeuMF by a considerable margin.

Beyond precision, DAC-GCN excels at retrieving relevant items, evidenced by its strong Recall@10 scores. On the ML-100K dataset, it attains a Recall@10 of 0.2575—slightly higher than NGCF (0.2566) and NeuMF (0.2552)—and a Recall@10 of 0.1722 on ML-1M, edging out NGCF (0.1673) and closely matching NeuMF (0.1740). This indicates DAC-GCN's proficiency in not only identifying pertinent items but also ranking them effectively within the top recommendations.

NDCG@10 further highlights the ranking quality of DAC-GCN. On ML-100K (0.3062) and ML-1M (0.2780), it achieves the highest NDCG@10 values among all baseline models, reflecting its ability to prioritize the most relevant items at the forefront of the recommendation list. DAC-GCN also exhibits robust performance on smaller datasets, such as Amazon Magazine Subscriptions and Mod Cloth, where it achieves NDCG@10 scores of 0.0874 and 0.0904, respectively, underscoring its versatility across different dataset sizes and characteristics.

Mean Reciprocal Rank (MRR@10) provides further evidence of DAC-GCN's aptitude for ranking relevant items in top positions. The model surpasses all baselines on ML-100K (0.4998) and ML-1M (0.4758), reinforcing its consistent ability to produce accurate recommendations. Its strong MRR@10 score on Amazon Subscription Boxes (0.0955) underscores its effectiveness in managing data sparsity.

DAC-GCN also excels in Hit@10, reporting a hit ratio of 0.8054 on ML-100K and 0.7646 on ML-1M—both higher than those of NGCF and NeuMF—indicating more frequent successful recommendations within the top 10 results. On Amazon Subscription Boxes, DAC-GCN achieves a Hit@10 of 0.1454, confirming its capacity to retrieve a wide range of relevant items even in datasets with sparse user-item interactions.

Overall, these results validate the effectiveness of DAC-GCN's DAC architecture, which separately optimizes policy and value estimation within GCN layers to capture both immediate and long-term preferences. MHA further bolsters its ability to model complex user-item relationships, resulting in notable improvements across all key metrics. Its robust and scalable performance makes DAC-GCN well-suited for various recommendation scenarios, including those with challenging data sparsity.

Figure 3 visually summarizes these findings, displaying the average metric values across all datasets and emphasizing DAC-GCN's superior performance compared to baselines like BPR, LightGCN, and NeuMF. In particular, the model's higher NDCG@10 scores underscore its adeptness at ranking relevant items at the top of recommendations. The figure also highlights DAC-GCN's consistent effectiveness on both dense datasets (e.g., ML-100K) and sparse datasets (e.g., Amazon Subscription Boxes), attesting to the versatility of its dual architecture and MHA approach.

Table 5. *Performance comparison of DAC-GCN and state-of-the-art methods across multiple datasets using Precision@10, Recall@10, NDCG@10, MRR@10, and Hit@10 metrics.*

Metric	Method	ML-100K	ML-1M	Amazon Subscription Boxes	Amazon Magazine Subscriptions	Mod Cloth
Precision@10	BPR	0.1998	0.2081	0.0063	0.0050	0.0110
	LightGCN	0.1435	0.1495	0.0105	0.0124	0.0163
	NeuMF	0.2006	0.2058	0.0248	0.0155	0.0148
	NGCF	0.2019	0.2019	0.0101	0.0109	0.0146
	EASE	0.0595	0.0679	0.0044	0.0046	0.0062
	SGL	0.0279	0.0291	0.0068	0.0163	0.0187

Metric	Method	ML-100K	ML-1M	Amazon Subscription Boxes	Amazon Magazine Subscriptions	Mod Cloth
	SimGCL	0.1052	0.1096	0.0045	0.0038	0.0076
	DAC-GCN	0.2073	0.2159	0.0145	0.0158	0.0195
Recall@10	BPR	0.2557	0.1710	0.0629	0.0498	0.1096
	LightGCN	0.1885	0.1261	0.1049	0.1239	0.1627
	NeuMF	0.2552	0.1740	0.2483	0.1546	0.1478
	NGCF	0.2566	0.1673	0.1014	0.1093	0.1459
	EASE	0.0992	0.0763	0.0437	0.0472	0.0631
	SGL	0.0682	0.0456	0.0682	0.1630	0.1868
	SimGCL	0.1563	0.1045	0.0455	0.0385	0.0754
	DAC-GCN	0.2575	0.1722	0.1454	0.1576	0.1651
NDCG@10	BPR	0.2944	0.2672	0.0322	0.0265	0.0604
	LightGCN	0.2143	0.1945	0.0573	0.0629	0.0962
	NeuMF	0.2980	0.2655	0.1550	0.0760	0.0810
	NGCF	0.2981	0.2584	0.0496	0.0509	0.0799
	EASE	0.0894	0.0774	0.0274	0.0281	0.0441
	SGL	0.0521	0.0473	0.0374	0.0888	0.1098
	SimGCL	0.1623	0.1473	0.0289	0.0193	0.0433
	DAC-GCN	0.3062	0.2780	0.1387	0.0874	0.0904
MRR@10	BPR	0.4856	0.4623	0.0230	0.0195	0.0457
	LightGCN	0.3733	0.3554	0.0428	0.0446	0.0760
	NeuMF	0.4917	0.4585	0.1261	0.0526	0.0610
	NGCF	0.4948	0.4488	0.0344	0.0336	0.0602
	EASE	0.1549	0.1405	0.0222	0.0217	0.0389
	SGL	0.0867	0.0825	0.0283	0.0662	0.0864
	SimGCL	0.3026	0.2881	0.0240	0.0135	0.0337
	DAC-GCN	0.4998	0.4758	0.0955	0.0539	0.0789
Hit@10	BPR	0.7996	0.7591	0.0629	0.0499	0.1098
	LightGCN	0.6829	0.6483	0.1049	0.1240	0.1629
	NeuMF	0.7943	0.7611	0.2483	0.1547	0.1480
	NGCF	0.8070	0.7493	0.1014	0.1094	0.1461
	EASE	0.4380	0.4078	0.0437	0.0472	0.0631
	SGL	0.2524	0.2396	0.0682	0.1630	0.1870
	SimGCL	0.6225	0.5910	0.0455	0.0385	0.0756
	DAC-GCN	0.8054	0.7646	0.1454	0.1527	0.1650

Additionally, Table 6 compares the performance of various recommendation models against our proposed DAC-GCN on the ML-100K dataset. We evaluated each model using ranking metrics at multiple top-k thresholds (k=5,10,15,20,25,50,100), providing insights into how effectively these approaches retrieve and rank relevant items within the top-k recommendations.

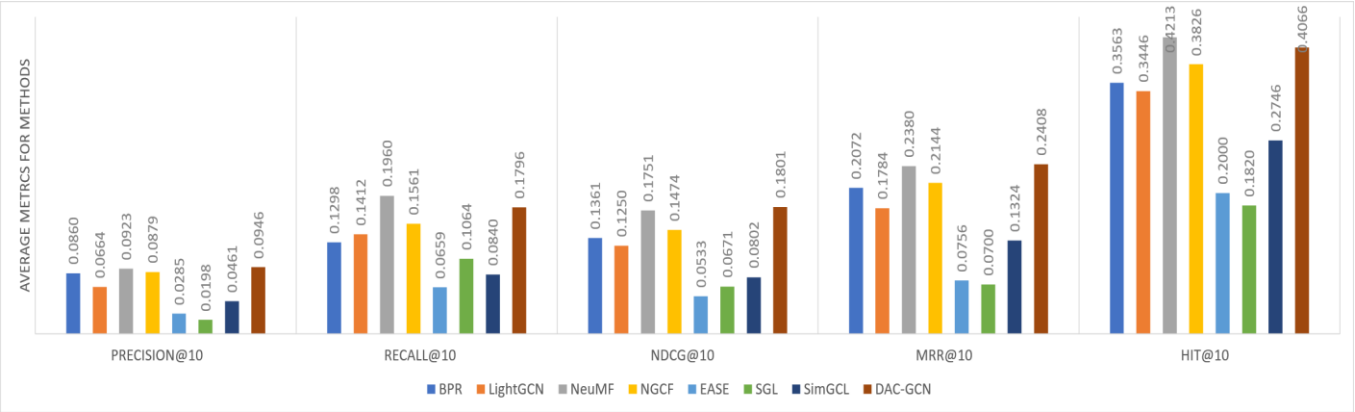

Figure 3. *Average prediction metrics across all datasets, showing DAC-GCN's superior performance.*

Across all metrics, DAC-GCN demonstrates consistently superior performance, particularly at higher k values (e.g., k=50 and k=100). This robustness highlights its ability to maintain accuracy even when users view extended recommendation lists. Figures 4 through 8 illustrate the comparative performance of each model on Precision, Recall, NDCG, MRR, and Hit, revealing that DAC-GCN consistently surpasses the baseline methods at different k values. Notably, the model's elevated NDCG and MRR scores underscore its effectiveness at ranking relevant items near the top of recommendation lists, while its enhanced Recall and Hit metrics attest to broader coverage of relevant items. Together, these findings solidify DAC-GCN as a powerful solution for both ranking and retrieval tasks, delivering high-quality recommendations to users.

Table 6. *Performance comparison of DAC-GCN and state-of-the-art models on the ML-100K dataset, evaluated across multiple metrics (Precision, Recall, NDCG, MRR, and Hit) at varying cutoff values (k = 5, 10, 15, 20, 25, 50, 100).*

Metric	Method	@5	@10	@15	@20	@25	@50	@100
Precision	BPR	0.2484	0.1998	0.1733	0.1538	0.1389	0.0986	0.0654
	LightGCN	0.1803	0.1435	0.1224	0.1113	0.1021	0.0747	0.0517
	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
	EASE	0.0628	0.0595	0.0568	0.0529	0.0503	0.0411	0.0321
	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
	DAC-GCN	0.2584	0.2073	0.1790	0.1591	0.1448	0.1023	0.0683
Recall	BPR	0.1614	0.2557	0.3295	0.3823	0.4254	0.5742	0.7297
	LightGCN	0.1187	0.1885	0.2404	0.2828	0.3169	0.4427	0.5791
	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
	EASE	0.0566	0.0992	0.1336	0.1614	0.1850	0.2760	0.3888
	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
	DAC-GCN	0.1623	0.2575	0.3297	0.3803	0.4251	0.5776	0.7449
NDCG	BPR	0.2930	0.2944	0.3060	0.3178	0.3294	0.3786	0.4293
	LightGCN	0.2135	0.2143	0.2204	0.2305	0.2399	0.2811	0.3260
	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

Metric	Method	@5	@10	@15	@20	@25	@50	@100
	EASE	0.0749	0.0894	0.1011	0.1101	0.1181	0.1493	0.1860
	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
	DAC-GCN	0.3053	0.3062	0.3167	0.3281	0.3408	0.3918	0.4469
MRR	BPR	0.4685	0.4856	0.4909	0.4926	0.4936	0.4950	0.4954
	LightGCN	0.3560	0.3733	0.3791	0.3819	0.3836	0.3859	0.3864
	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
	EASE	0.1327	0.1549	0.1638	0.1677	0.1698	0.1741	0.1754
	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
	DAC-GCN	0.4815	0.4998	0.5049	0.5063	0.5075	0.5090	0.5093
Hit	BPR	0.6713	0.7996	0.8653	0.8950	0.9183	0.9661	0.9894
	LightGCN	0.5525	0.6829	0.7572	0.8070	0.8452	0.9290	0.9576
	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
	EASE	0.2683	0.4380	0.5514	0.6193	0.6681	0.8144	0.9024
	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
	DAC-GCN	0.6816	0.8054	0.8710	0.9015	0.9281	0.9669	0.9932

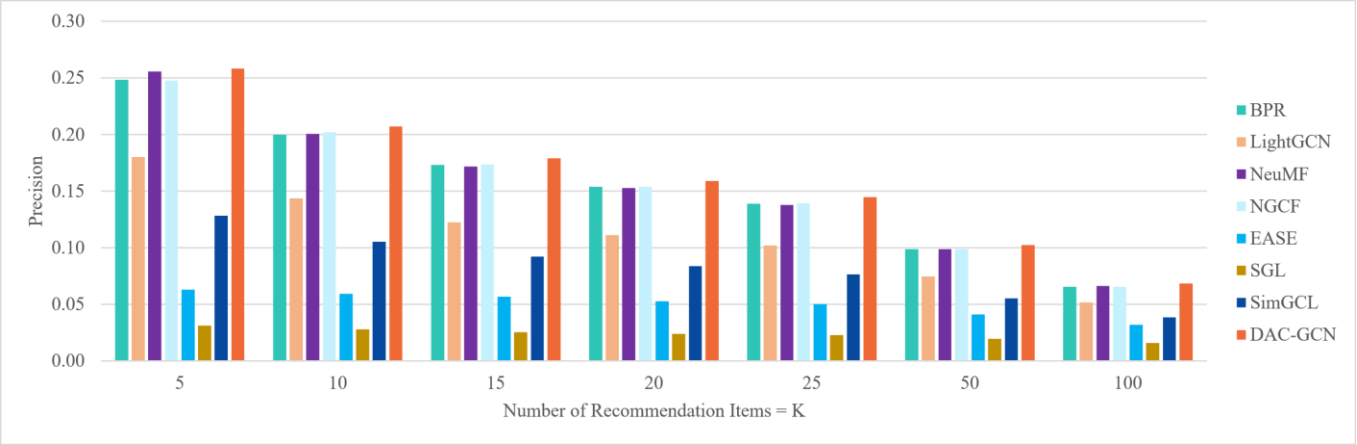

Figure 4. *Precision at different values of k for various recommendation models on the ML-100k dataset, showing DAC-GCN's superior performance.*

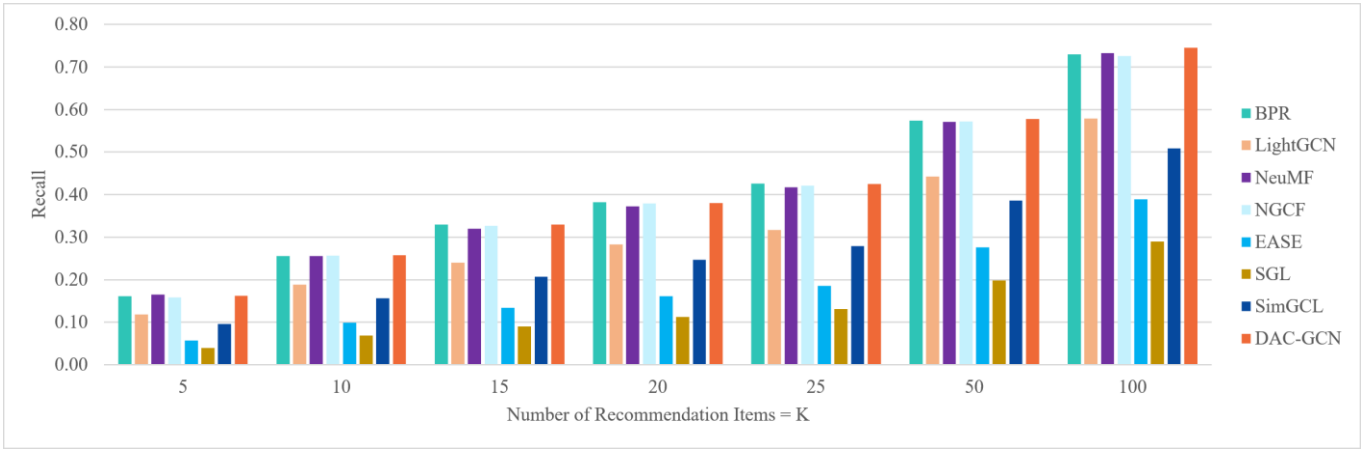

Figure 5. *Recall at different values of k for various recommendation models on the ML-100k dataset, highlighting DAC-GCN's ability to capture relevant items.*

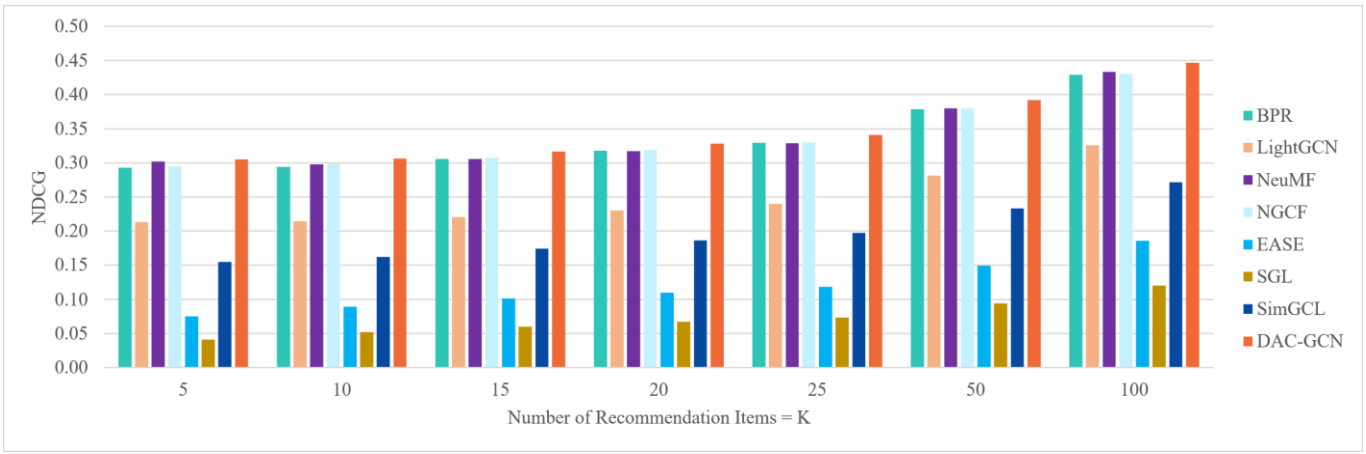

Figure 6. *NDCG at different values of k for various recommendation models on the ML-100k dataset, illustrating DAC-GCN's ability to prioritize highly relevant items.*

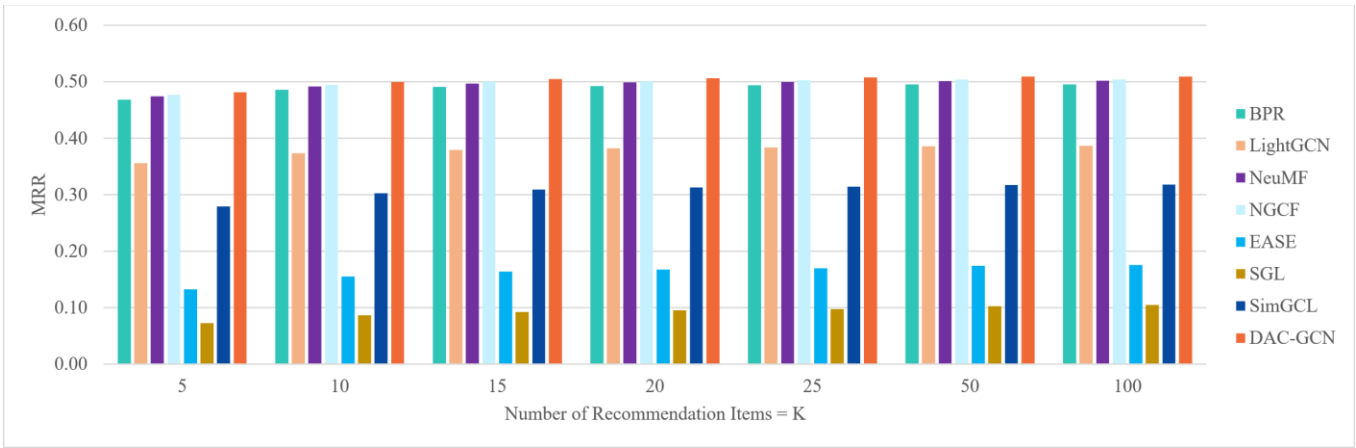

Figure 7. *MRR at different values of k for various recommendation models on the ML-100k dataset, showcasing DAC-GCN's superior ranking of relevant items.*

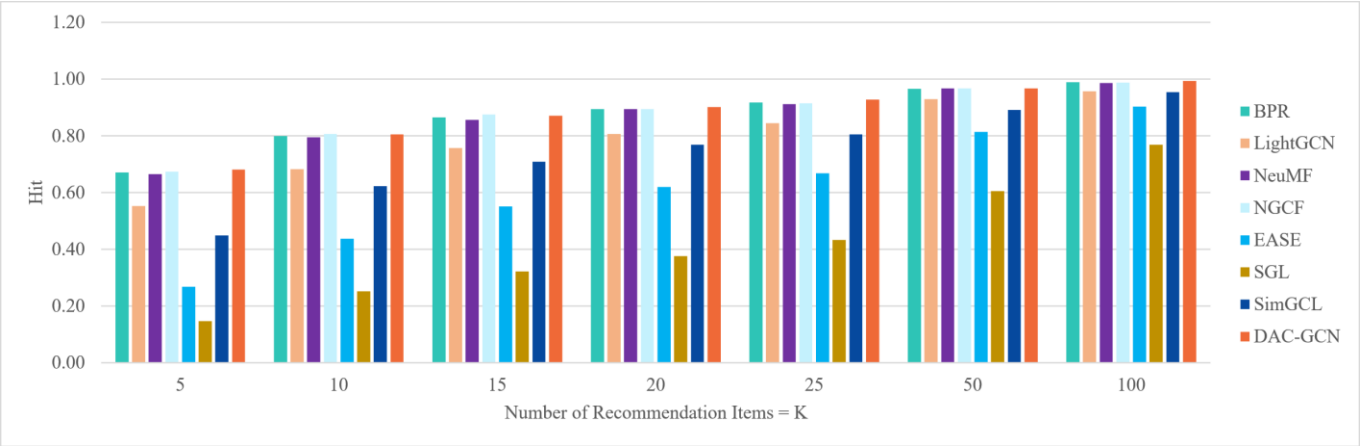

Figure 8. *Hit rate at different values of k for various recommendation models on the ML-100k dataset, with DAC-GCN demonstrating higher success in recommending relevant items to users.*

4.3 Discussion

In this section, we consolidate the experimental findings and examine the practical as well as theoretical implications of our proposed DAC-GCN framework. We begin by highlighting its performance gains and analyzing the factors that contribute to its success. We then discuss real-world applicability, focusing on scalability and resource considerations. Finally, we outline the model's current limitations and propose directions for future research.

4.3.1 Performance Gains and Key Observations

The results demonstrate DAC-GCN's remarkable effectiveness in critical RS tasks, including ranking accuracy, item retrieval, and mitigating data sparsity. Notably, it consistently outperforms baseline models such as BPR, LightGCN, NeuMF, and NGCF across diverse evaluation metrics. This superior performance largely stems from the DAC architecture, which specializes in both policy generation and value estimation through separate networks. MHA further augments the model's ability to capture long-range user–item dependencies and thereby enrich its representation of user preferences.

A significant highlight of DAC-GCN is its success on top-ranking metrics like NDCG@10 and MRR@10, which directly gauge the ability to surface highly relevant items. In every dataset tested, DAC-GCN achieves higher scores than competing approaches, underlining the model's prowess in prioritizing pertinent recommendations. This robust ranking accuracy is attributable to the dual architecture's capacity to integrate both local and global interaction patterns, aided by an attention mechanism that dynamically emphasizes the most influential user–item connections.

Data sparsity remains a formidable challenge in many real-world scenarios, particularly for datasets such as Amazon Subscription Boxes and Amazon Magazine Subscriptions. Here, DAC-GCN demonstrates clear advantages by substantially boosting Recall@10 and Hit@10, surpassing conventional models by a wide margin. These gains can be ascribed to its dual-GCN structure, which merges immediate user preferences with extended behavioral patterns. Additionally, DAC-GCN exhibits strong performance across a range of recommendation contexts, from smaller, denser datasets (e.g., ML-100K) to larger, more complex ones (e.g., ML-1M). Its capacity to deliver consistent gains in both dense and sparse environments underscores its adaptability to variable user interactions.

From a research standpoint, this work advances RSs by introducing a DAC paradigm for DRL with graph-based embeddings. By decoupling policy learning and value estimation, it transcends the limitations of single-network frameworks and demonstrates the value of task-specific GCNs for handling intricate user–item relationships.

4.3.2 Real-World Applicability and Resource Considerations

DAC-GCN's heightened accuracy and ability to adapt to dynamic user preferences make it well suited for practical deployments in areas like e-commerce and media streaming. Here, personalized recommendations are critical and user behaviors often shift rapidly. The model's strong performance on both dense and sparse datasets indicates it can manage varying data volumes and structures, a vital attribute for real-world systems.

Despite its strengths, DAC-GCN introduces non-trivial computational demands. MHA, which propagates information across multiple graph layers, increases both memory usage and training time, while the attention mechanism adds overhead through the dynamic weighting of interactions. Running separate actor and Critical networks effectively doubles the computational workload compared to single-network solutions. Nevertheless, techniques such as graph sampling, sparse attention, parameter sharing, and parallelization can help alleviate these pressures. With these optimizations, DAC-GCN can be deployed at scale or even adapted for near real-time scenarios without a significant trade-off in recommendation quality.

4.3.3 Limitations and Future Directions

Notwithstanding its promising results, DAC-GCN faces several constraints that merit further investigation:

- **Computational Complexity:** MHA and attention mechanisms substantially increase training costs, especially for large datasets or time-sensitive applications. While graph sampling and sparse attention alleviate some of these burdens, more advanced optimizations are necessary for truly scalable solutions.
- **Training Time:** The DAC structure, in conjunction with GCN-based learning, can prolong training relative to simpler models. Future work should focus on improved optimization strategies that expedite convergence without undermining accuracy.
- **Applicability to Real-Time Systems:** Although DAC-GCN excels in offline batch settings, rapidly evolving user behaviors in real-time contexts may pose bottlenecks. Incremental learning methods or more lightweight aggregation strategies could help bridge this gap.
- **Dependence on Graph Structure:** The quality of the user–item graph critically influences model performance. Poorly connected or extremely sparse graphs may limit the effectiveness of MHA, diminishing recommendation quality.
- **Generalizability Across Domains:** While DAC-GCN has shown strong results on the datasets evaluated, its adaptability to diverse domains with varied interaction patterns or item types remains uncertain, calling for broader testing and potential domain-specific modifications.

Addressing these limitations offers several promising pathways for future research. Real-time optimizations and more advanced graph processing methods could expand DAC-GCN's applicability to rapidly changing environments, while domain-specific adaptations and hybrid data representations may strengthen its generalizability. By pursuing these directions, the model has the potential to deliver even more robust, high-quality recommendations across a wide range of use cases. Building on the insights gained from our comprehensive evaluation, we next present an ablation study that examines the individual contributions of DAC-GCN's key components to its overall performance.

4.4 Ablation Study

To evaluate the contribution of each key component of the DAC-GCN model, we conducted an ablation study on the ML-100K dataset. The objective of this study is to assess the impact of individual components, including the DAC architecture, MHA, and the attention mechanism, on the overall performance of the model. By isolating and removing specific components, we aim to understand their respective roles and quantify their contributions to the performance improvements seen in the full DAC-GCN model.

We created three variants of DAC-GCN for the ablation study:

- **V1 → DAC-GCN without Multi-Hop:** In this variant, the MHA is removed, and only single-hop aggregation is used.
- **V2 → DAC-GCN without Attention:** This version excludes the attention mechanism, relying solely on the default aggregation without dynamic importance weighting.
- **V3 → Single Actor-Critic GCN (SAC-GCN)**: In this baseline, the DAC structure is replaced with a single GCN network that performs both recommendation policy generation and value estimation without task separation.

The results of these experiments measured using the same evaluation metrics (Precision@10, Recall@10, NDCG@10, MRR@10, and Hit@10), are presented in Table 7.

Table 7. *Ablation Study Results on ML-100K Dataset.*

Variant	Precision@10	Recall@10	NDCG@10	MRR@10	Hit@10
Full Model	0.2073	0.2575	0.3062	0.4998	0.8054
V1	0.1894	0.2312	0.2789	0.4672	0.7806
V2	0.1953	0.2405	0.2856	0.4790	0.7902
V3	0.1815	0.2217	0.2634	0.4511	0.7642

The results of the ablation study clearly illustrate the importance of each component in the proposed DAC-GCN model. The full DAC-GCN model, which integrates MHA, attention mechanisms, and a DAC structure, achieves the highest performance across all metrics, confirming the effectiveness of combining these elements.

Removing the MHA (V1) leads to a notable decrease in performance across all metrics. Precision@10 drops from 0.2073 to 0.1894, and NDCG@10 falls from 0.3062 to 0.2789. This decline highlights the critical role that MHA plays in capturing long-range dependencies and complex interaction patterns within the user-item graph. Without it, the model becomes less capable of incorporating global user preferences, resulting in lower recommendation accuracy and ranking quality.

Similarly, excluding the attention mechanism (V2) results in a decrease in performance, though the drop is less severe compared to removing MHA. This suggests that while the attention mechanism significantly enhances the model's ability to prioritize key user-item interactions, its absence does not cripple the model entirely. The decline in NDCG@10 (from 0.3062 to 0.2856) and MRR@10 (from 0.4998 to 0.4790) indicates that the attention mechanism helps the model focus on more relevant items, improving ranking quality and user satisfaction.

Finally, the single actor-critic GCN variant (V3) performs the worst among the variants, with a Precision@10 of 0.1815 and an NDCG@10 of 0.2634. This demonstrates the clear advantage of separating the recommendation tasks between the Actor and Critic networks. The dual architecture allows each network to specialize in distinct tasks, resulting in more effective learning and better recommendations. The drop in performance for SAC-GCN suggests that a single GCN structure is insufficient for effectively capturing both immediate and long-term user preferences.

Thus, the ablation study demonstrates that the combination of DAC GCNs, MHA, and attention mechanisms significantly contributes to the superior performance of DAC-GCN. Each component plays a critical role in enhancing the model's ability to capture complex user-item interactions, optimize recommendation policies, and deliver high-quality recommendations. These findings validate the design choices of the proposed method and highlight the necessity of integrating advanced graph-based techniques in modern RSs.

5 CONCLUSION

This paper presented DAC-GCN, a novel framework that advances DAC architectures within graph-based RSs. By decoupling policy optimization and value estimation into task-specific learning components, DAC-GCN illuminates how specialized architectural choices address challenges such as long-range dependency modeling and training stability. From a practical standpoint, DAC-GCN demonstrates heightened recommendation accuracy and robustness in dynamic environments, offering compelling solutions for industrial applications in e-commerce, media streaming, and personalized content delivery. Its capacity to prioritize critical interactions and capture intricate user-item relationships establishes a sound basis for next-generation RSs.

From the point of view of theoretical contributions, first, the introduction of a DAC architecture that separates policy learning from value estimation facilitates more targeted and efficient learning. Second, the incorporation of MHA effectively captures long-range user-item dependencies, mitigating data sparsity and cold-start issues. Third, attention mechanisms provide a means to dynamically emphasize important interactions, thereby enhancing recommendation relevance and improving user satisfaction. Collectively, these innovations address pivotal gaps left

by conventional and contemporary methods, demonstrating the potential of integrating hybrid DRL with graph-based techniques.

From the perspective of practical contributions, our DAC-GCN method delivers more accurate and diverse recommendations, thereby boosting user engagement and retention. Its adaptability to different datasets underscores its suitability for a broad range of domains, from dense interaction environments to those where sparse feedback complicates recommendation generation. However, the framework's reliance on MHA and attention mechanisms increases its computational footprint, posing challenges for large-scale, real-time deployments. Additionally, the model's performance depends on the completeness of the user-item graph, which highlights potential vulnerabilities in highly sparse or disconnected scenarios. Furthermore, the training process requires substantial time and computational resources, creating opportunities for optimization to enhance both efficiency and scalability.

Looking ahead, future work can explore lightweight and scalable variants of DAC-GCN tailored for real-time usage and huge datasets. Investigating adaptive reward functions may also prove beneficial in improving training efficiency and responding more effectively to evolving user behavior. Additionally, extending DAC-GCN to handle heterogeneous graphs or multi-modal data, including textual and visual information, would broaden its applicability in complex recommendation tasks.

In conclusion, DAC-GCN represents a significant leap forward in RS research, particularly for dynamic and data-sparse environments. By continuing to refine its architecture, optimize its computational demands, and adapt to emerging modalities, DAC-GCN sets the stage for future advancements in personalized recommendation technologies.

ADDITIONAL INFORMATION AND DECLARATIONS

Acknowledgments: This article was derived from PhD degree thesis from the Islamic Azad University, Qeshm Branch (Zare, 2025).

Conflict of Interests: The authors declare no conflict of interest.

Author Contributions: G.Z.: Writing – original draft, Visualization, Methodology, Conceptualization, Data curation. N.J.: Writing – review & editing, Data curation. M.H: Writing – review & editing, Project administration, Resources. A.S.: Writing – review & editing.

Statement on the Use of Artificial Intelligence Tools: The authors declare that they didn't use artificial intelligence tools for text or other media generation in this article.

Data Availability: The data that support the findings of this study are openly available on the RecBole website at https://recbole.io/dataset_list.html

REFERENCES

Abnoosian, K., Farnoosh, R., & Behzadi, M. H. (2023). A pipeline -based framework for early prediction of diabetes. *Journal of Health and Biomedical Informatics*, 10(2), 125–140. https://doi.org/10.34172/jhbmi.2023.19

Alamdari, P. M., Navimipour, N. J., Hosseinzadeh, M., Safaei, A. A., & Darwesh, A. (2022). Image-based product recommendation method for e-commerce applications using convolutional neural networks. *Acta Informatica Pragensia*, 11(1), 15–35. https://doi.org/10.18267/j.aip.167

Albora, G., Mori, L. R., & Zaccaria, A. (2023). Sapling Similarity: A performing and interpretable memory-based tool for recommendation. *Knowledge-Based Systems*, 275, 110659. https://doi.org/10.1016/j.knosys.2023.110659

Anjiri, S. N., Ding, D., & Song, Y. (2024). HyGate-GCN: Hybrid-Gate-Based Graph Convolutional Networks with dynamical ratings estimation for personalised POI recommendation. *Expert Systems With Applications*, 258, 125217. https://doi.org/10.1016/j.eswa.2024.125217

Chen, C., Zhang, M., Zhang, Y., Liu, Y., & Ma, S. (2020). Efficient Neural Matrix Factorization without Sampling for Recommendation. *ACM Transactions on Information Systems*, 38(2), Article 14. https://doi.org/10.1145/3373807

Chen, X., Yang, G., Yang, S., Wang, H., Dong, S., & Gao, Y. (2023). Online attentive kernel-based temporal difference learning. *Knowledge-Based Systems*, 278, 110902. https://doi.org/10.1016/j.knosys.2023.110902

Cui, Y., Yu, H., Guo, X., Cao, H., & Wang, L. (2024). RAKCR: Reviews sentiment-aware based knowledge graph convolutional networks for Personalized Recommendation. *Expert Systems With Applications*, 248, 123403. https://doi.org/10.1016/j.eswa.2024.123403

Darbandi, M. (2017). Proposing new intelligent system for suggesting better service providers in cloud computing based on Kalman filtering. *HCTL Open International Journal of Technology Innovations and Research*, 24, 1–9.

Dilekh, T., Benharzallah, S., Mokeddem, A., & Kerdoudi, S. (2024). Dynamic Context-Aware Recommender System for home automation through synergistic unsupervised and supervised learning algorithms. *Acta Informatica Pragensia*, 13(1), 38–61. https://doi.org/10.18267/j.aip.228

Engström, E., Vartanova, I., Johansson, J. V., Persson, M., & Strimling, P. (2024). Comparing and modeling the use of online recommender systems. *Computers in Human Behavior Reports*, 15, 100449. https://doi.org/10.1016/j.chbr.2024.100449

Guo, F., Wang, Z., Wang, X., Lu, Q., & Ji, S. (2024). Dual-view multi-modal contrastive learning for graph-based recommender systems. *Computers & Electrical Engineering*, 116, 109213. https://doi.org/10.1016/j.compeleceng.2024.109213

Halder, S., Lim, K. H., Chan, J., & Zhang, X. (2024). A survey on personalized itinerary recommendation: From optimisation to deep learning. *Applied Soft Computing*, 152, 111200. https://doi.org/10.1016/j.asoc.2023.111200

Heidari, N., Moradi, P., & Koochari, A. (2022). An attention-based deep learning method for solving the cold-start and sparsity issues of recommender systems. *Knowledge-Based Systems*, 256, 109835. https://doi.org/10.1016/j.knosys.2022.109835

Henriques, R., & Pinto, L. (2023). A novel evaluation framework for recommender systems in big data environments. *Expert Systems With Applications*, 231, 120659. https://doi.org/10.1016/j.eswa.2023.120659

Hsu, C. (2024). Does a gamified website matter? An extended UTAUT model with regulatory fit, user experience, engagement, and attitudes. *Entertainment Computing*, 50, 100658. https://doi.org/10.1016/j.entcom.2024.100658

Hu, Y., Xiong, F., Pan, S., Xiong, X., Wang, L., & Chen, H. (2021). Bayesian personalized ranking based on multiple-layer neighborhoods. *Information Sciences*, 542, 156–176. https://doi.org/10.1016/j.ins.2020.06.067

Huang, W., Hao, F., Shang, J., Yu, W., Zeng, S., Bisogni, C., & Loia, V. (2023). Dual-LightGCN: Dual light graph convolutional network for discriminative recommendation. *Computer Communications*, 204, 89–100. https://doi.org/10.1016/j.comcom.2023.03.018

Kannout, E., Grzegorowski, M., Grodzki, M., & Nguyen, H. S. (2024). Clustering-Based frequent pattern mining framework for solving Cold-Start problem in recommender systems. *IEEE Access*, 12, 13678–13698. https://doi.org/10.1109/access.2024.3355057

Kipf, T. N, & Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations 2017*. https://openreview.net/pdf?id=SJU4ayYgl

Lee, S., Tanveer, J., Rahmani, A. M., Alinejad-Rokny, H., Khoshvaght, P., Zare, G., Alamdari, P. M., & Hosseinzadeh, M. (2025). SFGCN: Synergetic Fusion-based Graph Convolutional Networks Approach for link prediction in social networks. *Information Fusion*, 114, 102684. https://doi.org/10.1016/j.inffus.2024.102684

Liu, C., Yu, C., Gui, N., Yu, Z., & Deng, S. (2024). SimGCL: graph contrastive learning by finding homophily in heterophily. *Knowledge and Information Systems*, 66(3), 2089–2114. https://doi.org/10.1007/s10115-023-02022-1

Padhye, V., & Lakshmanan, K. (2023). A deep actor critic reinforcement learning framework for learning to rank. *Neurocomputing*, 547, 126314. https://doi.org/10.1016/j.neucom.2023.126314

Saini, K., & Singh, A. (2024). A content-based recommender system using stacked LSTM and an attention-based autoencoder. *Measurement Sensors*, 31, 100975. https://doi.org/10.1016/j.measen.2023.100975

Salvi, M., Loh, H. W., Seoni, S., Barua, P. D., García, S., Molinari, F., & Acharya, U. R. (2024). Multi-modality approaches for medical support systems: A systematic review of the last decade. *Information Fusion*, 103, 102134. https://doi.org/10.1016/j.inffus.2023.102134

Schulman, J., Levine, S., Moritz, P., Jordan, M., I., & Abbeel, P. (2015). Trust Region Policy optimization. *arXiv*. https://doi.org/10.48550/arxiv.1502.05477

Steck, H. (2019). Embarrassingly Shallow Autoencoders for Sparse Data. In *WWW '19: The World Wide Web Conference,* (pp. 3251–3257). ACM. https://doi.org/10.1145/3308558.3313710

Sutton, R. S. (2018). *Reinforcement learning: An introduction*. Bradford Book.

Sutton, R. S, McAllester, D., Singh, S., & Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. In *NIPS'99: Proceedings of the 13th International Conference on Neural Information Processing Systems*, (pp. 1057–1063). NIPS.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2017). Graph attention networks. *arXiv*. https://doi.org/10.48550/arxiv.1710.10903

Wang, X., He, X., Wang, M., Feng, F., & Chua, T. (2019). Neural Graph Collaborative Filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 165–174). ACM. https://doi.org/10.1145/3331184.3331267

Wei, T., Chow, T. W., Ma, J., & Zhao, M. (2023). ExpGCN: Review-aware Graph Convolution Network for explainable recommendation. *Neural Networks,* 157, 202–215. https://doi.org/10.1016/j.neunet.2022.10.014

Wu, J., Wang, X., Feng, F., He, X., Chen, L., Lian, J., & Xie, X. (2021). Self-supervised graph learning for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 726–735). ACM. https://doi.org/10.1145/3404835.3462862

Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2018). How Powerful are Graph Neural Networks? *arXiv*. https://doi.org/10.48550/arxiv.1810.00826

Zare, G. (2025). *A Deep Reinforcement Learning-Based Approach for Improving the Accuracy of the Recommendations in Social Networks*. Doctoral dissertation. Islamic Azad University, Qeshm Branch.

Zare, G., Navimipour, N. J., Hosseinzadeh, M., & Sahafi, A. (2024). Network link prediction via deep learning method: A comparative analysis with traditional methods. *Engineering Science and Technology an International Journal*, 56, 101782. https://doi.org/10.1016/j.jestch.2024.101782

Zhang, J., Zain, A. M., Zhou, K., Chen, X., & Zhang, R. (2024). A review of recommender systems based on knowledge graph embedding. *Expert Systems With Applications*, 250, 123876. https://doi.org/10.1016/j.eswa.2024.123876

Zhang, Y., Zhu, J., Zhang, Y., Zhu, Y., Zhou, J., & Xie, Y. (2024). Social-aware graph contrastive learning for recommender systems. *Applied Soft Computing*, 158, 111558. https://doi.org/10.1016/j.asoc.2024.111558

Zhao, W. X., Hou, Y., Pan, X., Yang, C., Zhang, Z., Lin, Z., Zhang, J., Bian, S., Tang, J., Sun, W., Chen, Y., Xu, L., Zhang, G., Tian, Z., Tian, C., Mu, S., Fan, X., Chen, X., & Wen, J. (2022). RecBole 2.0: Towards a More Up-to-Date Recommendation Library. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management* (pp. 4722–4726). ACM. https://doi.org/10.1145/3511808.3557680

Zhao, Z., Tong, X., Wang, Y., & Zhang, Q. (2024). Multi-behavior contrastive learning with graph neural networks for recommendation. *Knowledge-Based Systems*, 300, 112221. https://doi.org/10.1016/j.knosys.2024.112221

Zhou, T., Ye, H., & Cao, F. (2024). Node-personalized multi-graph convolutional networks for recommendation. *Neural Networks*, *173*, 106169. https://doi.org/10.1016/j.neunet.2024.106169

Zhu, Y., Ma, J., Yuan, C., & Zhu, X. (2021). Interpretable learning based Dynamic Graph Convolutional Networks for Alzheimer's Disease analysis. *Information Fusion*, 77, 53–61. https://doi.org/10.1016/j.inffus.2021.07.013

Acta Informatica Pragensia is published by the Prague University of Economics and Business, Czech Republic | eISSN: 1805-4951