VIETNAM NATIONAL UNIVERSITY

UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Nguyễn Trần Ngọc Linh

ROBUST AND ADAPTIVE RECOMMENDATION BY DEEP MODELING OF CANONICAL AND AUXILIARY DATA

Major: Information Systems

Code: 9480104

SUMMARY OF THE DOCTORAL DISSERTATION

HA NOI, 2026

The dissertation was completed at: University of Engineering and Technology, Vietnam National University

Scientific Supervisors:

- Supervisors: Assoc.Prof. Phan Xuan Hieu

- Dr. Tran Mai Vu

Reviewer 1: Dr. Do Thanh Ha

Reviewer 2: Assoc.Prof. Bui Thu Lam

Reviewer 3: Assoc.Prof. Do Van Thanh

The dissertation will be defended before the …… -level Doctoral Dissertation Evaluation Committee convened at Hanoi University Of Engineering and Technology, VNU

at o'clock on _______ date _______ month _______ year

The dissertation can be found at:

- National Library of Vietnam
- Information Center - Library, Vietnam National University, Hanoi

Abstract

Modern recommendation systems are increasingly required to operate in large-scale digital ecosystems where user bases grow continuously, item spaces expand rapidly, and interaction patterns evolve in unpredictable ways. These environments intensify fundamental challenges such as data sparsity, cold-start scenarios, scalability constraints, multi-domain adaptation, and the need for conversational interaction. Traditional models often struggle to provide scalable solutions for learning expressive user-item representations or incorporating auxiliary side information at scale, and they fail to maintain stability under changing interaction patterns.

This dissertation develops a unified deep learning framework for robust and adaptive recommendation through deep modeling of both canonical data (user-item interactions) and auxiliary data (side information, domain context, and conversational signals). The research addresses three fundamental challenges across four interconnected directions.

First, the dissertation investigates scalable recommendation through ID-free user representations, neural soft clustering, and contrastive learning. By eliminating dependency on explicit user identifiers and organizing users into probabilistic latent preference groups, the proposed approach dramatically reduces memory requirements while maintaining recommendation quality at the scale of real-world recommendation.

Second, the research explores robust fusion of canonical and auxiliary data through attention-based weight generation mechanisms and masked graph contrastive learning. These techniques dynamically balance the contribution of behavioral embeddings and side information while selectively preserving informative embedding dimensions, enhancing representation robustness under sparse and cold-start conditions.

Third, the dissertation develops continual learning mechanisms for adaptive multi-domain recommendation. Through domain masking and specialization with soft constraints, the proposed framework enables multi-directional knowledge transfer across domains while preserving domain-specific knowledge and ensuring balanced performance optimization.

Fourth, the research proposes hybrid conversational recommendation that bridges canonical and auxiliary data through graph neural network-based preference modeling integrated with large language models and retrieval-augmented generation. This approach combines long-term user behavior with real-time conversational intent to generate context-aware personalized recommendations.

Extensive experiments on benchmark datasets and industrial deployment with real products from Viettel (one of the largest corporations in Vietnam), validating the effectiveness of the proposed framework, demonstrating consistent improvements over state-of-the-art baselines and confirming that deep integration of canonical and auxiliary data provides a robust foundation for scalable, adaptive, and conversational recommendation systems.

Keywords: *Deep learning, recommendation systems, canonical data, auxiliary data, data sparsity, cold-start problem, scalability, soft clustering, contrastive learning, graph neural networks, side information fusion, continual learning, multi-domain recommendation, conversational recommendation, large language models, retrieval-augmented generation.*

Chapter 1

Literature Review of Background and Methods

1.1 Problem Definition and Formulation

Recommender systems are information filtering tools designed to predict user preferences and suggest relevant items from large catalogs [25]. These systems have become essential components of modern digital platforms, powering personalized experiences across e-commerce, streaming services, social networks, and content platforms [16, 39]. The fundamental goal of a recommender system is to estimate the relevance or utility of items that users have not yet interacted with, while guiding users toward items that align with their preferences while helping service providers increase engagement and revenue [29].

The recommendation problem can be understood from multiple perspectives. From an information retrieval viewpoint, it involves ranking items based on their predicted relevance to a user query, where the query is implicitly defined by the user's profile and historical behavior [22]. From a machine learning perspective, recommendation is a prediction task that aims to estimate missing values in a partially observed user item interaction matrix [17]. From an optimization standpoint, the objective is to maximize a utility function that captures both user satisfaction and business objectives [14].

1.1.1 Overview of Recommendation Problem

Problem Statement

Formally, a recommender system operates over three fundamental entities: a set of users $\mathcal{U} = \{u_1, u_2, \ldots, u_M\}$, a set of items $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$, and the interactions between them [25]. The core of any recommendation method is a utility function that quantifies the relevance of an item to a user:

$$f : \mathcal{U} \times \mathcal{V} \rightarrow \mathcal{D} \tag{1.1}$$

where $\mathcal{D}$ represents the domain of utility values.

The primary objective of a recommender system is to identify, for each user $u \in \mathcal{U}$, the items that maximize the utility function [1]:

$$v_u^* = \arg\max_{v \in \mathcal{V}} f(u, v) \tag{1.2}$$

In practice, systems typically generate a ranked list of top-K items rather than a single recommendation, producing an ordered set $\mathcal{D}_u = \{v_1, v_2, \ldots, v_K\}$ where items are sorted in descending order by their predicted utility scores [10]. The recommendation task thus becomes:

$$\mathcal{D}_u = \text{Top-}K_{v \in \mathcal{V}} \big(f(u, v) \big) \tag{1.3}$$

1.1.2 Key Challenges in Recommender Systems

Despite decades of research and significant practical success, recommender systems continue to face several fundamental challenges that limit their effectiveness and scalability [35, 39]. This dissertation specifically addresses three critical challenges: data sparsity, cold-start problems, and scalability constraints.

Data Sparsity

The sparsity problem arises from the inherent imbalance between the vast space of possible user item interactions and the limited number of observed interactions [13, 32]. In large scale systems, users typically interact with only a tiny fraction of available items. For example, Netflix users rate fewer than 1% of available movies [3], and Amazon customers purchase an even smaller fraction of the millions of available products [21].

This sparsity creates several difficulties for recommendation algorithms [2]:

- Limited statistical evidence: Sparse observations make it difficult to distinguish actual preferences from noise, leading to unreliable preference estimates.

- Reduced collaborative signals: Sparsity decreases the overlap between users' interaction histories, weakening the effectiveness of collaborative filtering approaches that rely on finding similar users or items [27].
- Long tail distribution: Popular items receive abundant feedback while the majority of items have very few interactions, leading to biased recommendations that favor popular items over potentially relevant niche content [24].

Traditional approaches address sparsity through regularization [17], dimensionality reduction [26], or incorporation of side information [31]. However, these methods often struggle when sparsity is extreme, motivating the development of more sophisticated techniques such as self supervised learning [36], contrastive learning [34], and graph based representation learning [35].

Cold-Start Problem

The cold-start problem refers to the difficulty of making accurate recommendations for new users or new items that lack sufficient interaction history [18, 30]. This challenge manifests in two distinct forms:

User cold-start occurs when a new user joins the platform without any historical interactions. Since collaborative filtering methods rely on past behavior to infer preferences, they cannot effectively serve users who have not yet provided sufficient feedback [5]. This creates a critical user experience problem: poor initial recommendations may cause new users to disengage before the system can learn their preferences [20].

Item cold-start occurs when new items are added to the catalog without historical interaction data. These items cannot be recommended through collaborative filtering alone, creating a visibility problem where potentially relevant new content remains hidden from users who might appreciate it [28]. This is particularly problematic for platforms with rapidly changing inventories, such as news sites, fashion retailers, or content platforms with frequent new releases [33].

Addressing cold-start typically requires incorporating auxiliary information beyond interaction data, such as user demographics, item attributes, textual descriptions, or social relationships [31, 39]. Content based and hybrid approaches leverage such side information to bootstrap recommendations for cold entities [6], while meta learning [19] and transfer learning techniques [23] enable rapid adaptation to new users or items with minimal data.

Scalability

Modern recommender systems must operate at unprecedented scale, serving millions of users across millions of items while meeting strict latency requirements [9, 40]. Addressing scalability requires innovations across the entire recommendation pipeline, from efficient embedding representations [8] and approximate nearest neighbor search [15] to distributed training frameworks and caching strategies for serving infrastructure [11].

1.1.3 Research Scope and Objectives

This dissertation develops deep learning based solutions that address the interconnected challenges of sparsity, cold-start, and scalability.

Table 1.1 summarizes the mapping between the key challenges and the dissertation chapters that address them.

Table 1.1: Mapping of Key Challenges to Dissertation Chapters

Challenge	**Chapter 2**	**Chapter 3**	**Chapter 4**	**Chapter 5**
Data Sparsity	✓	✓		✓
Cold-Start	✓	✓		
Scalability	✓			
Multi-Domain Adaptation			✓	
Conversational Context				✓

1.1.4 Summary

This chapter formalizes the proposal problem as learning utility functions that map user-item pairs to preference points, while identifying three core challenges: data sparsity, cold start, and scalability. A model integrating canonical data (user-item interactions) and auxiliary data (demographics, attributes, context, knowledge graphs) is introduced as the theoretical foundation, where deep integration between these two data sources is key to building a robust recommendation system. Traditional methods and modern deep learning architectures (MLP, CNN, RNN, Transformer, GNN, LLM) are examined along with offline (Recall@K, NDCG@K) and online (A/B testing) evaluation metrics.

Chapter 2

Robust Recommendation via Interaction Embedding and Soft Clustering

2.1 EfficientRec: Scalable ID-Free Recommendation via Soft Clustering and Contrastive Learning

To address the identified gaps, this chapter proposes EfficientRec, a novel framework for robust large scale recommendation via interaction embedding and soft clustering. The research objectives guiding the development of EfficientRec are as follows.

2.1.1 Overview

EfficientRec is an end-to-end scalable recommendation framework that abandons explicit user-IDentifiers in favour of representations constructed dynamically from interaction patterns, allowing the model to scale to the larger user populations without growing its parameter footprint. User preferences are encoded through a neural soft-clustering mechanism that maps each user onto a probabilistic mixture of latent preference prototypes, capturing the multifaceted nature of real-world tastes more faithfully than hard-assignment approaches. To prevent representational collapse and improve robustness under sparse interaction data, the framework incorporates contrastive learning objectives that enforce geometric separation between dissimilar users and representation invariance across different observed interaction subsets directly benefiting cold-start scenarios. At inference time, a cluster-based voting pipeline replaces exhaustive item scoring: candidate items are retrieved from the user's most relevant clusters and aggregated through precomputed cluster-level preference scores, achieving sub-linear inference complexity without sacrificing recommendation quality. The framework is validated on public benchmarks as well as an industrial deployment on the TV360 platform, confirming its effectiveness in both controlled and real-world settings.

2.1.2 Model Architecture

This section presents the proposed EfficientRec architecture, a scalable recommendation framework designed to address the fundamental limitations of conventional user-ID based recommendation models. The architecture eliminates the dependency on explicit user-IDentifiers by constructing user representations dynamically from behavioral signals, thereby achieving computational complexity that is independent of the user population size. This design enables the system to scale to large user bases while maintaining consistent recommendation quality and supporting seamless integration of new users without requiring model retraining.

The proposed model consists of three principal components that work together to provide personalized recommendations.

The overall architecture is illustrated in Figure 2.1, which provides a high level view of how the three components interact to produce personalized recommendations.

The first component is the ”Interaction Embedding model”, which is responsible for constructing compact and informative user representations by aggregating information from the user's historical interactions with items in the system.

The second component is the ”Clustering Model”, which organizes users into preference aware groups using contrastive learning [7, 12] and soft clustering techniques [4]. The clustering model learns to map user representations into a latent preference space where each dimension corresponds to a distinct preference cluster. The contrastive learning objective ensures that users with similar preferences are mapped to similar regions in the preference space, while users with different preferences are separated well.

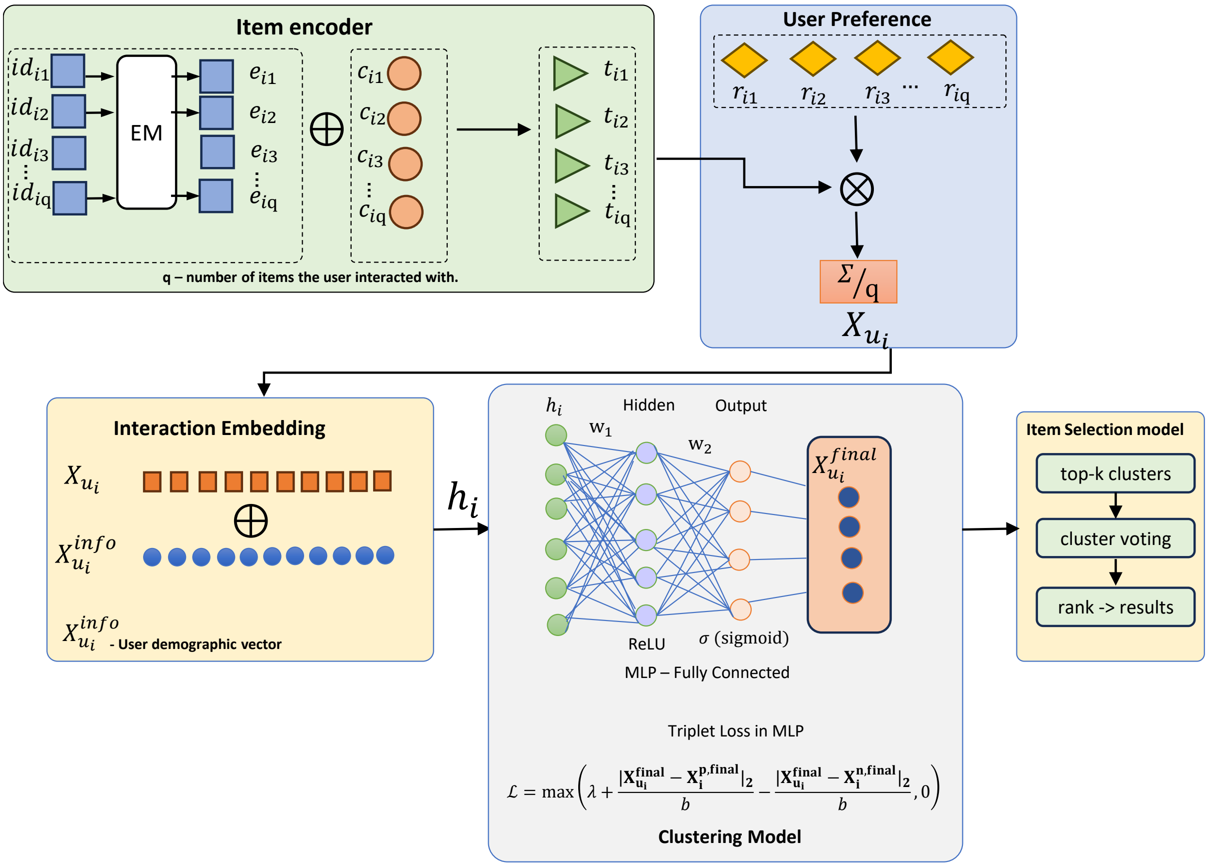

Figure 2.1: EfficientRec Overall Architecture

The third component is the ”Item Selection model”, which efficiently generates personalized recommendations through a two phases cluster based voting mechanism. In the offline phase, the model precomputed preference scores for each cluster and item pair based on the aggregated ratings from all users belonging to that cluster. In the online phase, the model computes the target user’s cluster membership and generates recommendations by aggregating the precomputed scores from the user’s most relevant clusters. This two phases design significantly reduces the computational cost of recommendation generation compared to methods that must score all items for each user.

2.1.3 Offline Experimental Results

All results reported below are evaluated based on ***relative differences****.*

a) Overall Performance

Table 2.1: Overall Performance Comparison (All Users) @30

Model	Recall@30	NDCG@30	Category
Proposed Method			
EfficientRec	**0.1994**	**0.1178**	Proposed
Graph-based Methods			
NGCF	<u>0.1958</u>	<u>0.1174</u>	Graph
LINKX	0.1928	0.1160	Graph
GraphSAGE	0.1658	0.0970	Graph
GAT	0.1639	0.0946	Graph
LightGCN	0.1636	0.0945	Graph
Contrastive Learning Methods			
SSL4Rec	0.1644	0.0950	SSL
MixGCF	0.1591	0.0910	CL
SGL	0.1192	0.0565	CL
Clustering-based and MF Methods			
SCoC	0.1584	0.1107	Clustering
FCCF	0.1090	0.1072	Clustering
BPR-MF	0.1074	0.0912	MF
FCM-Rec	0.1067	0.1068	Clustering

Bold indicates the best result and <u>underline</u> denotes the second best.

EfficientRec achieves the best performance on both metrics: Recall@30 (0.1994) and NDCG@30 (0.1178). Compared to the second best method NGCF, EfficientRec outperforms by +1.8% in Recall@30 and +0.34% in NDCG@30. Against the best clustering baseline SCoC, EfficientRec demonstrates +25.9% improvement in Recall@30, validating the effectiveness of our soft clustering mechanism over traditional clustering approaches.

2.1.4 Online Experimental Results

- Average Content Per User (ACPU) is defined as the average number of distinct recommended content items that each user consumed (i.e., clicked and watched) during the A/B testing period. - Average Duration Per User (ADPU) is defined as the average total viewing time (in minutes) that each user spent on recommended content during the same period.

*All results reported below are evaluated based on **relative differences**.*

Table 2.2: Results of the Online Experiments on TV360

Methods	**TV360 Films**		**TV360 Videos**	
	ACPU	ADPU	ACPU	ADPU
2DNNs	0.0152	13.896	0.0322	10.526
ALS	0.0121	12.190	0.0160	4.277
ER Interaction Split	**0.0186**	**15.018**	**0.0421**	<u>12.290</u>
ER User Group Split	<u>0.0176</u>	<u>14.272</u>	<u>0.0381</u>	**13.155**

Bold = Best, <u>Underline</u> = Second Best.

EfficientRec (Interaction Split) achieves the best overall results, outperforming the second-best method 2DNNs by +22.4% in ACPU on Films (0.0186 vs. 0.0152) and +30.7% on Videos (0.0421 vs. 0.0322). The larger gain on the sparser video catalogue (0.002% density vs. 0.039% for Films) is consistent with the core design of soft clustering, which transfers preference knowledge from similar users and is most effective when per-user interaction data is scarce. For ADPU, EfficientRec (Interaction Split) also leads on Films (+8.1%: 15.018 vs. 13.896), while EfficientRec (User Group Split) leads on Videos (+24.0%: 13.155 vs. 10.526 for 2DNNs), indicating that inter-user discriminability contributes to longer viewing sessions on sparse catalogues. The consistent improvements across both ACPU and ADPU confirm that offline Recall@30 gains translate to real user engagement in production.

2.2 Summary

This chapter presented EfficientRec, a unified deep learning framework for scalable recommendation that addresses three fundamental challenges in industrial recommender systems: data sparsity, cold-start problems, and the dependency on fixed user IDentifiers that limits scalability. The first contribution is the interaction embedding method, which models historical user item behaviors as directional edges within a graph structure. This approach enables the extraction of both first order preference signals and higher order collaborative patterns. The second contribution is the integration of contrastive learning mechanisms to enhance representation robustness under sparse data conditions. The framework constructs positive and negative interaction pairs, incorporating noise injection and learnable feature masking to create diverse contrastive views.

Experimental results on real-world datasets verify the effectiveness of the proposed framework. Online A/B testing on a production streaming platform further confirms the practical effectiveness of EfficientRec in real-world deployment scenarios.

Chapter 3

Boosting Recommendation via Graph-based Fusion of Canonical Interactions and Auxiliary Side Information

3.1 Introduction

This chapter concentrates on enhancing recommendation quality through the modeling and integration of auxiliary information and canonical data within Graph Neural Network (GNN) architectures. While GNN-based methods have demonstrated strong capability in capturing structural patterns from user item graphs and learning meaningful representations, their performance is often constrained by sparse interactions and limited utilization of contextual information. To address these drawbacks, this chapter introduces a framework that enriches GNN representations by incorporating auxiliary side information alongside graph derived interaction embeddings, enabling the recommender system to leverage both descriptive and behavioural signals.

In summary, this chapter presents how side information, interaction patterns, and graph learning can be cohesively combined to improve modern recommendation performance. Through detailed architectural modelling and empirical analysis, this chapter demonstrates that augmenting GNN-based recommendation with structured side information fusion produces more accurate, robust, and semantically aligned recommendation outcomes. These works have been published in peer-reviewed conferences, including:[P2] “GIFT4Rec: An Effective Side Information Fusion Technique Apply to Graph Neural Network for Cold-Start Recommendation” (ACIIDS 2023), and [P3] “The Masked Simple Graph Contrastive Learning for Recommendation” (KSE 2024).

3.2 GIFT4Rec: Auxiliary Information Fusion with Attention-based and Meta-Learning Techniques for Cold-Start Recommendation

3.2.1 Problem Statement

This chapter investigates how graph neural networks (GNNs) combined with side information fusion can be cohesively integrated to improve modern recommendation performance. Through detailed architectural modeling and empirical analysis, this chapter demonstrates that augmenting GNN-based recommendation with structured side information fusion produces more accurate, robust, and semantically aligned recommendation outcomes. These works have been published in peer-reviewed conferences, including: “GIFT4Rec: An Effective Side Information Fusion Technique Apply to Graph Neural Network for Cold-Start Recommendation” (ACIIDS 2023).

3.2.2 Gift4Rec: Model Architecture and Components

This section presents the proposed GIFT4Rec architecture, a unified framework for integrating side information into graph-based recommendation systems. The architecture addresses the fundamental challenge of balancing behavioral signals from user item interactions with semantic information from user attributes, enabling robust recommendation across both warm-start and cold-start scenarios.

The overall architecture is illustrated in Figure 3.1, which provides a high-level view of how the three components interact together for producing personalized recommendations.

The first component is the “GNN Interaction Module”, which learns user and item representations by propagating information through the user item interaction graph. Unlike content-based approaches that rely solely on feature matching, this component captures collaborative signals from the global interaction structure, enabling discovery of preference patterns that emerge from collective user behavior.

The second component is the ”Local Side Information Fusion Module” (LSIF), which adaptively combines behavioral embeddings with side information embeddings for each individual user. The key insight is that the optimal fusion strategy varies across users some users have rich interaction histories that provide strong preference signals, while others have limited interactions where side information becomes more valuable. This component learns personalized fusion weights through an attention-based mechanism called Attention DropoutNet (ADN).

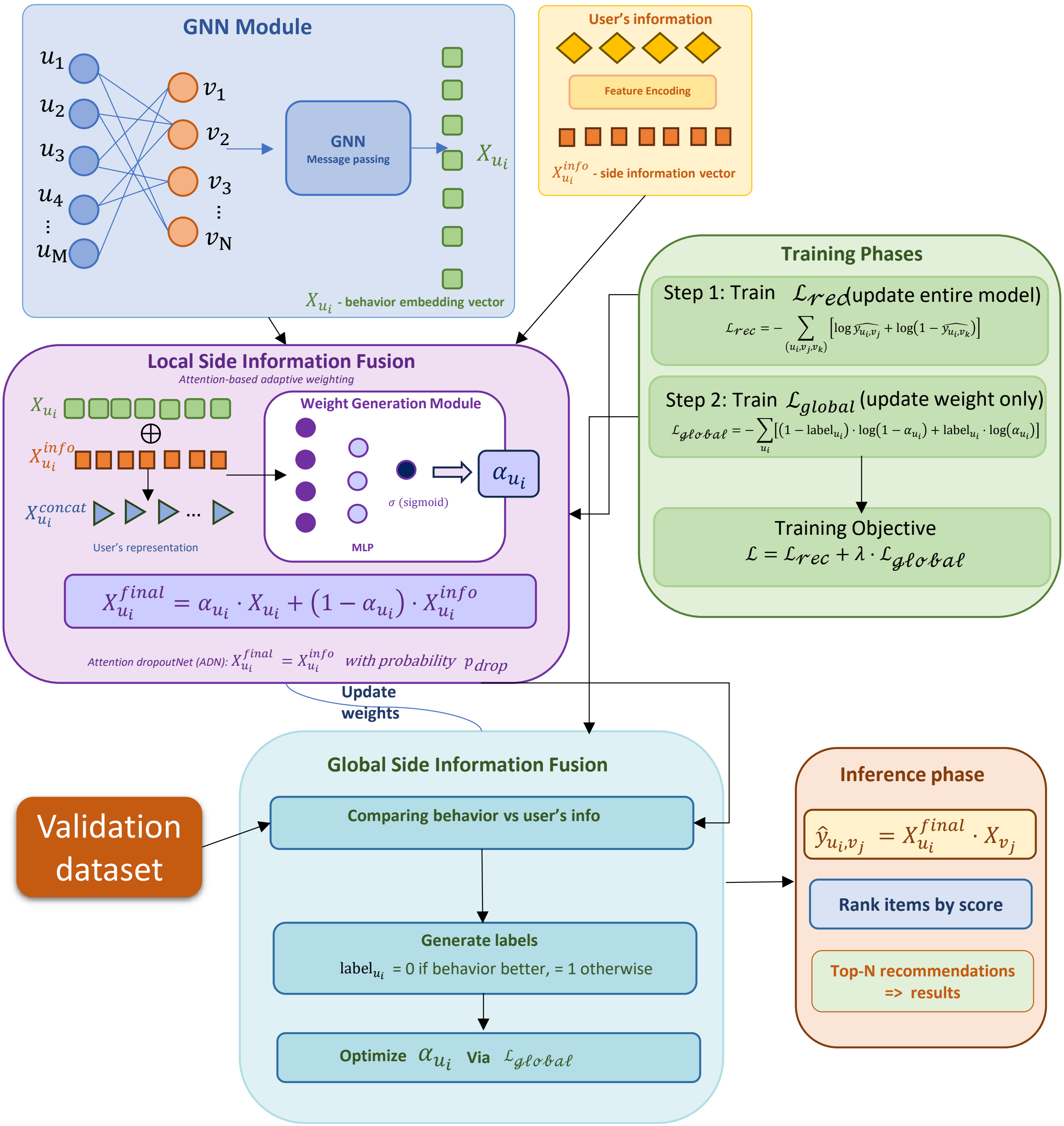

Figure 3.1: GIFT4Rec overall architecture

The third component is the "Global Side Information Fusion" (GSIF) module, which provides meta-level supervision for the weight generation process by evaluating which information source (behavioral or side information) supports better recommendation performance on validation data. This global perspective ensures that the learned fusion weights generalize well beyond the training interactions.

3.2.3 Experimental Results

All results reported below are evaluated-based on ***relative differences****.*

Table 3.1: GIFT4Rec - Overall Performance Comparison (All Users) @30

Model	**Recall@30**	**NDCG@30**	**Category**
GIFT4Rec	**0.2162**	**0.1263**	Proposed
KGAT	0.1793	0.1067	Aux+KG
LINKX	0.1928	0.1160	Aux+Graph
GAT	0.1639	0.0946	Aux+Attn
KGAT DropoutNet	0.0908	0.0359	Aux+KG
NGCF	0.1958	<u>0.1178</u>	GNN
LightGCN	0.1636	0.0945	GNN
SSL4Rec	0.1644	0.0950	SSL
EfficientRec (Ch.2)	<u>0.1994</u>	0.1174	Proposed

Bold = Best, <u>Underline = Second Best</u>

Table 3.1 reports the overall recommendation performance of GIFT4Rec and all baseline methods. GIFT4Rec achieves the best performance on both metrics, with Recall@30 of 0.2162 and NDCG@30 of 0.1263. Among GNN-based methods, NGCF is the strongest competitor (Recall@30 = 0.1958); GIFT4Rec outperforms it by $+10.4\%$ in Recall@30 and $+7.2\%$ in NDCG@30, demonstrating that graph-based message passing alone is insufficient when interactions are sparse and that the incorporation of auxiliary side information provides a substantial complementary signal. Among auxiliary fusion methods, LINKX achieves Recall@30 of 0.1928, yet GIFT4Rec surpasses it by $+12.1\%$, confirming that static, uniform fusion of side information is inferior to the per-user adaptive weighting learned by the Weight Generated module. The improvement over EfficientRec (Chapter 2, Recall@30 = 0.1994) of $+8.4\%$ further shows that attention-based side information fusion provides gains beyond what interaction-only soft clustering can achieve. Overall, the consistent improvements across all baseline categories validate that the two core designs of GIFT4Rec adaptive per-user fusion and meta-learning-based generalisation are both necessary for robust recommendation performance.

3.3 The Masked Simple Graph Contrastive Learning for Recommendation

3.3.1 Problem Statement

Contrastive learning (CL), which is capable of extracting generalizable representations from unlabeled raw data, has recently emerged as an effective solution to the problem of data sparsity and has attracted significant attention in recommendation research. By constructing positive and negative pairs through data augmentation, CL enables models to learn robust and discriminative representations without relying heavily on dense supervision signals. This property makes contrastive learning particularly suitable for large scale recommendation scenarios where explicit feedback is limited and highly sparse.

Motivated by these advantages, this work extends the SimGCL [37] framework by introducing a learnable masking mechanism that adaptively controls the importance of different dimensions in node representations during contrastive learning. Instead of treating all embedding dimensions equally, the proposed masking strategy allows the model to explicitly identify and preserve task relevant parameters, while suppressing less informative or noisy dimensions. As a result, the model is encouraged to focus on semantically meaningful features, which effectively reduces the risk of overfitting.

3.3.2 MaskSimGCL: Model Architecture and Components

This section presents the proposed MaskSimGCL (Masked Simple Graph Contrastive Learning) architecture, a novel framework designed to address the limitations of existing graph-based contrastive learning methods for recommendation. The architecture extends the SimGCL framework by integrating learnable masking mechanisms that adaptively identify and weight the importance of different embedding dimensions, thereby achieving more robust representation learning under sparse data conditions.

The proposed model consists of four principal components that operate together to deliver personalized recommendations. The first component is the graph neural network backbone, which is responsible for learning user and item representations through message passing operations on the user item bipartite graph. Following the LightGCN design, this component employs simplified graph convolutions that propagate collaborative signals without feature transformation, capturing neighborhood patterns through layer-wise aggregation.

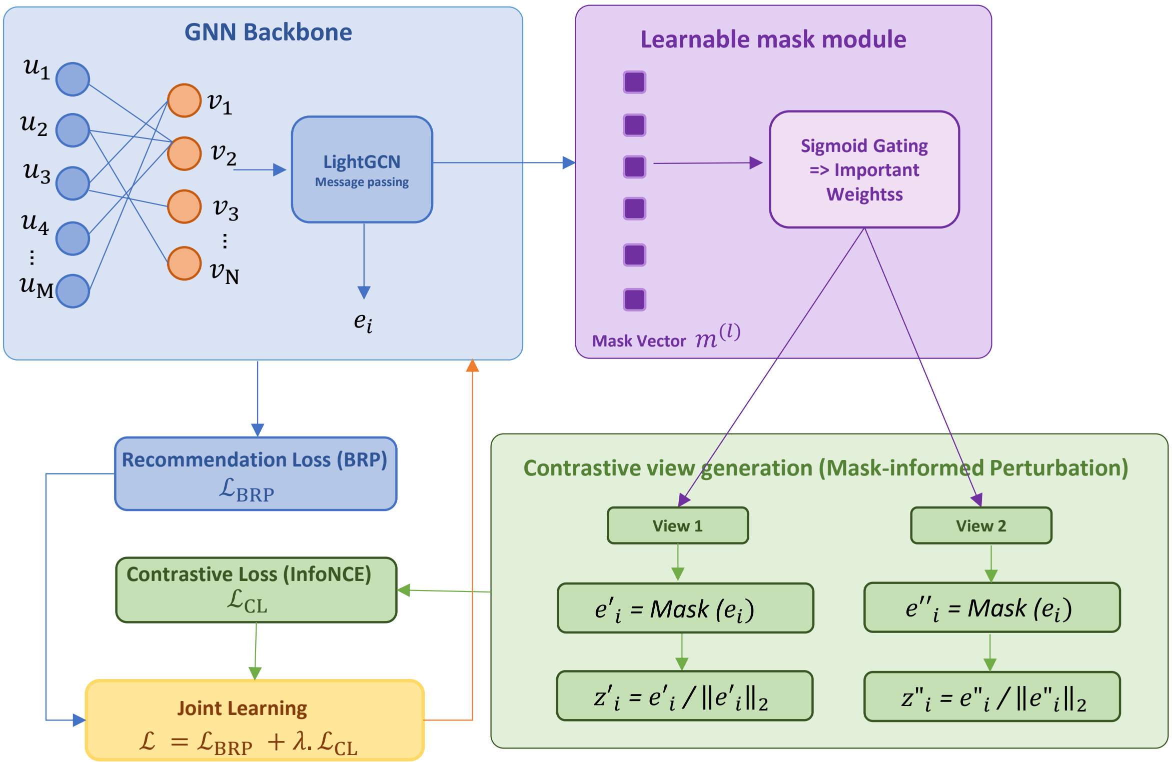

Figure 3.2: MaskSimGCL overall architecture

The second component is the learnable mask module, which introduces trainable mask vectors at each graph neural network layer. These masks serve as importance filters that adaptively weight each dimension of the node embeddings-based on their relevance to the recommendation task. By focusing model capacity on informative parameters while suppressing noisy or redundant dimensions, this component provides implicit regularization that effectively mitigates overfitting in sparse data environments.

The third component is the contrastive view Generation module, which constructs augmented representations for contrastive learning. Unlike SimGCL's uniform noise injection, MaskSimGCL employs mask informed perturbations that apply differential noise magnitudes-based on the learned importance scores. Dimensions identified as less important receive larger perturbations, while critical dimensions are preserved with smaller noise, resulting in consistent contrastive views that enhance representation learning.

The fourth component is the joint optimization framework, which combines the supervised recommendation objective with the self-supervised contrastive learning objective. This multitask learning formulation enables the model to simultaneously optimize for accurating preference prediction and robust representation learning, with the contrastive loss providing uniformity regularization that promotes more evenly distributed embeddings in the representation space.

3.3.3 Experimental Results

All results reported below are evaluated-based on ***relative differences****.*

a. Overall Performance

Table 3.2: Overall Performance Comparison (All Users) @30

Model	Recall@30	NDCG@30	Category
MaskSimGCL	**0.2404**	**0.1322**	Proposed
XSimGCL	<u>0.2301</u>	0.1250	GCL
SimGCL	0.2292	0.1249	GCL
LightGCL	0.2128	0.1103	GCL
DirectAU	0.2110	0.1226	GCL
SGL	0.1192	0.0565	GCL
SSL4Rec	0.1644	0.0950	GNN
LightGCN	0.1636	0.0945	GNN
GIFT4Rec (Section 3.2)	0.2162	<u>0.1263</u>	Proposed
EfficientRec (Chapter 2)	0.1994	0.1174	Proposed

Bold = Best, Underline = Second Best

MaskSimGCL achieves the best performance on both metrics: Recall@30 of 0.2404 and NDCG@30 of 0.1322. The second-best method is XSimGCL in (Recall@30 = 0.2301, NDCG@30 = 0.1250); MaskSimGCL outperforms it by $+4.5\%$ in Recall@30 and $+5.8\%$ in NDCG@30. Since both models share the same SimGCL backbone, this gap is directly attributable to the learnable mask mechanism: by identifying and suppressing uninformative embedding dimensions, MaskSimGCL generates higher-quality contrastive views that preserve semantically meaningful structure, whereas XSimGCL applies uniform noise indiscriminately across all dimensions.

3.4 Summary

This chapter presented a unified graph-based framework for enhancing recommendation quality through the joint modeling of canonical interaction data and auxiliary side information. The proposed approach addresses the limitations of conventional GNN-based recommender systems, which often underutilize rich semantic attributes and suffer from performance degradation under sparse and cold-start conditions. Overall, this chapter establishes graph-based modeling, when combined with adaptive side information fusion and masked contrastive learning, forms a paradigm for improving recommendation accuracy, stability, and cold-start generalization.

Chapter 4

Enhancing Multi-Domain Recommendation with Continual Learning

4.1 Introduction

Modern recommendation systems increasingly operate across multiple service domains within unified platforms, where users interact with heterogeneous content categories such as e-commerce products, video streaming, music services, and news feeds. These multi-domain environments present unique challenges that extend beyond traditional single domain recommendation paradigms. While cross domain recommendation (CDR) has emerged as a promising direction to leverage rich information across domains [38], existing approaches predominantly focus on improving target domain performance while often neglecting the preservation of source domain knowledge and the fairness of performance across all participating domains.

4.1.1 Model Architecture and Components

This section presents the complete architecture of the proposed CNL4Rec (Continual learning for multi-domain recommendation) framework. The CNL4Rec framework introduces a task masking based continual learning mechanism that operates at the embedding level. The central design principle is to treat each domain as a sequential learning task and apply domain specific masks that identify and protect important latent dimensions for each domain.

The architecture introduces learnable mask vectors for each domain that have the same dimensionality as the user and item embeddings. These masks identify which latent dimensions are essential for representing domain specific behavioral patterns, enabling the system to selectively update only relevant parameters during training. Parameters deemed unimportant for the current domain remain protected to maintain performance on previously learned domains.

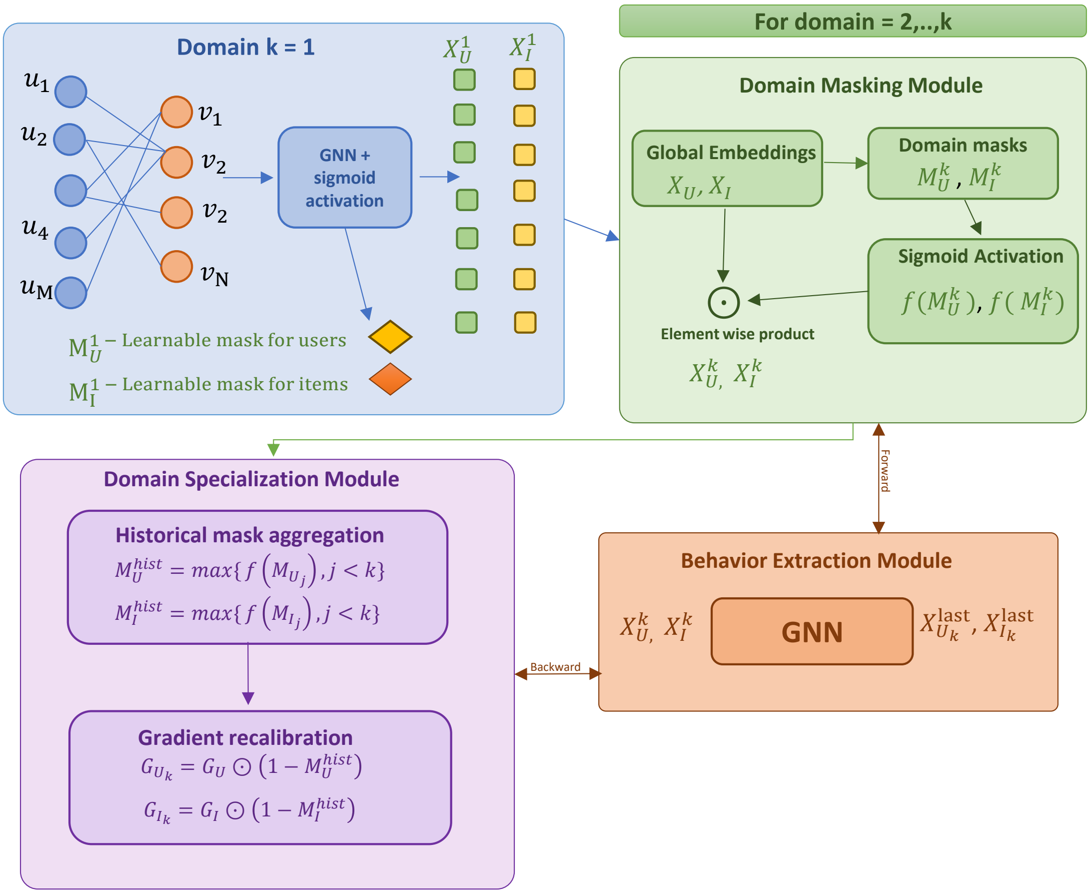

Figure 4.1: CNL4Rec Overall Architecture

As illustrated in Figure 4.1, CNL4Rec addresses the continual multi-domain recommendation problem through three tightly integrated modules. The ”Domain Masking Module” learns domain-specific binary masks over the shared user and item embedding matrices, identifying which latent dimensions are relevant for each domain and producing filtered domain-specific representations $\mathbf{X}_U^k$ and $\mathbf{X}_I^k$. Its role is to partition the shared embedding space into domain-specific subspaces without requiring separate parameters per domain. The ”Domain Specialization Module” protects previously acquired knowledge by aggregating the masks of all past domains into a historical mask $\mathbf{M}^{\text{hist}}$, which is then used to zero out gradients flowing into dimensions already claimed by prior domains during backpropagation. This selective gradient suppression is the core anti-forgetting mechanism, ensuring that learning a new domain does not overwrite representations essential to earlier ones. The ”Behavior Extraction Module” receives the masked, domain-specific embeddings and passes them through a GNN backbone that aggregates high-order collaborative signals within the current domain, producing final embeddings used for preference scoring and recommendation ranking. Together, the three modules implement domain knowledge isolation at the embedding level, knowledge protection at the gradient level, and preference learning at the inference level collectively resolving the stability plasticity trade off inherent in continual multi-domain recommendation.

4.1.2 Experimental Results

All results reported below are evaluated based on ***relative differences****.*

A. Overall results

a. Results on MovieLens-1M

On MovieLens-1M (density 4.21%), CNL4Rec achieves the best mean Recall@30 of 0.2288, outperforming the second-best method CTNet (0.1616) by $+41.6\%$, and leads across all five genre domains. The largest gains are in Comedy (0.2583 vs. CTNet 0.0838, $+208.2\%$) and Action (0.2481 vs. 0.1600, $+55.1\%$), where concentrated and differentiable user preferences allow the domain mask to cleanly identify domain-relevant embedding dimensions and protect them from cross-genre interference.

Table 4.1: Performance Comparison on MovieLens-1M (Recall@30)

Method	Action	Comedy	Sci-Fi	Thriller	Drama	Mean
CNL4Rec	**0.2481**	**0.2583**	**0.1865**	**0.2349**	<u>0.2163</u>	**0.2288**
CTNet	<u>0.1600</u>	<u>0.0838</u>	<u>0.1342</u>	<u>0.2034</u>	**0.2268**	<u>0.1616</u>
DIIT	0.1246	0.0632	0.1106	0.1181	0.1061	0.1045
KEEP	0.0598	0.0060	0.0841	0.1005	0.1796	0.0860
MF	0.0744	0.0060	0.0286	0.0485	0.2061	0.0727
ECAT	0.0063	0.0273	0.0173	0.0744	0.1665	0.0583

Bold = Best, <u>Underline</u> = Second Best

Sci-Fi also benefits substantially ($+39.0\%$) due to its strong user overlap with Action, which enables effective multi-directional knowledge transfer. The only exception is Drama, where CTNet marginally outperforms CNL4Rec (0.2268 vs. 0.2163,

-4.6%): as the largest domain by interaction volume, Drama provides sufficient within-domain data that cross-domain transfer offers diminishing returns, and CTNet's domain-specific columns exploit this dense signal effectively. Despite this single-domain loss, CNL4Rec's consistent strength across the remaining four genres confirms that domain masking and gradient recalibration provide the greatest benefit in settings where domain boundaries are sharp and interaction patterns are domain-specifics.

b. Results on Yelp

On the Yelp dataset (interaction density 1.15%), CNL4Rec achieves the best mean Recall@30 of 0.0174, outperforming the second-best method ECAT (0.0165) by $+5.5\%$.

At the domain level, CNL4Rec leads in Restaurant (0.0194 vs. ECAT 0.0164, $+18.3\%$) and Health (0.0182 vs. CNET 0.0179, $+2.2\%$).

Table 4.2: Performance Comparison on Yelp (Recall@30)

Method	Restaurant	Shopping	Food	Beauty	Health	Mean
CNL4Rec	**0.0194**	0.0150	<u>0.0184</u>	<u>0.0166</u>	**0.0182**	**0.0174**
ECAT	<u>0.0164</u>	<u>0.0172</u>	0.0162	**0.0172**	0.0154	<u>0.0165</u>
CTNet	0.0146	0.0152	**0.0192**	0.0148	<u>0.0179</u>	0.0164
KEEP	0.0158	**0.0182**	0.0156	0.0156	0.0164	0.0165
MF	0.0150	0.0158	0.0179	0.0162	0.0156	0.0163
DIIT	0.0138	0.0154	0.0150	0.0156	0.0178	0.0155

Bold = Best, <u>Underline</u> = Second Best

The large margin in Restaurant reflects the fact that this domain has the densest user item interactions among the five Yelp categories, providing the clearest gradient signal for the domain mask to identify informative embedding dimensions.

The domain Shopping is notably competitive: KEEP outperforms CNL4Rec in Shopping (0.0182 vs. 0.0150, -21.3%). In conclude, CNL4Rec's consistent strength across Restaurant, Food, Beauty and Health is sufficient to achieve the best mean performance, confirming that the continual masking strategy provides a net benefit across the full multi-domain setting even when individual domains present challenges.

c. Results on Amazon

Table 4.3: Performance Comparison on Amazon (Recall@30)

Method	**Electronics**	**Books**	**Movies**	**Home**	**Sports**	**Mean**
CNL4Rec	0.0139	0.0125	0.0135	**0.0133**	0.0126	**0.0132**
ECAT	**0.0147**	0.0130	0.0128	0.0132	0.0116	0.0131
MF	0.0129	0.0109	**0.0170**	0.0125	0.0124	0.0131
KEEP	0.0114	**0.0145**	0.0142	0.0118	0.0127	0.0129
CTNet	0.0115	0.0120	0.0137	0.0113	**0.0136**	0.0124
DIIT	0.0134	0.0135	0.0105	0.0103	0.0116	0.0119

Bold = Best, Underline = Second Best

On the Amazon dataset (interaction density 0.83%, the sparsest of the three benchmarks), CNL4Rec achieves a mean Recall@30 of 0.0132, marginally outperforming the second-best method ECAT (0.0131) by only +0.8%. The narrow margin across all methods reflects the difficulty of the Amazon setting: with fewer than 1% of user item pairs observed, all models struggle to learn reliable representations, compressing the performance range.

CNL4Rec leads in only the Home domain (0.0133 vs. ECAT 0.0132, +0.8%). The fact that MF outperforms all multi-domain methods on Movies and that ECAT leads on Electronics highlights the key challenge of Amazon: in highly sparse settings, cross-domain knowledge transfer can introduce noise rather than benefit, and simpler per-domain models can outperform on individual categories. CNL4Rec's gradient recalibration mechanism mitigates this risk by preventing domain interference at the embedding level, which explains why it achieves the best *mean* performance despite not winning every individual domain.

Comparing the three datasets, the improvement of CNL4Rec over the second-best method decreases from +41.6% on MovieLens-1M (density 4.21%) to +5.5% on Yelp (1.15%) and +0.8% on Amazon (0.83%). This monotonic relationship between dataset density and performance margin confirms that the domain masking and gradient recalibration mechanisms of CNL4Rec are most effective when interactions are sufficiently dense to train reliable domain-specific masks, and that the advantage diminishes in extremely sparse regimes where mask learning itself is data-limited.

4.2 Summary

This chapter presented CNL4Rec, a continual learning based framework for multi-domain recommendation designed to preserve domain specific knowledge while maintaining model adaptability across heterogeneous user groups and content categories. The chapter highlights that conventional recommendation approaches suffer from performance degradation in multi-domain environments due to distributional differences, parameter interference, and catastrophic forgetting during sequential training.

Overall, this chapter establishes that integrating continual learning with behavioral modeling provides a powerful pathway toward scalable, resilient, and fair recommendation systems capable of long term evolution in real-world multi-domain environments.

Chapter 5

Conversational Recommendation with a GNN and RAG-Based Hybrid System

5.1 Introduction

5.1.1 Motivation

A recommendation chatbot is an interactive conversational agent designed to assist users in discovering items tailored to user preferences through natural language dialogue. In regard to the streaming platform context, by integrating with messaging or streaming interfaces, such chatbots provide personalized movie suggestions, enhancing user engagement and satisfaction by making the discovery process more intuitive and accessible.

This chapter proposes a film recommendation chatbot that combines the understanding of past behaviors from historical interaction data studied through a graph based deep learning model and real time preferences acquired from large language models (LLMs) via ensemble learning. This framework accentuates the complementary strengths of both advanced techniques to provide meaningful recommendations to users. In particular, graph based deep learning excels in deploying interactions into graph structured data, thus uncovering latent patterns and complex relationships between node entities. The graph based model also uses scalable algorithms and distributed computing techniques to handle vast volumes of interaction data, ensuring the system remains responsive and accurate as user and content data grow. On the other hand, LLMs collect individual preferences through conversational sessions, process natural language input, and flexibly adapt to immediate requirements. By combining these components, the system achieves a comprehensive, robust, and adaptive recommendation approach that balances past behavior insights with current user context, optimized for real time performance and large scale data processing.

5.2 CG-RAG: Conversational Recommendation via Graph-Enhanced Retrieval-Augmented Generation

In this section, we briefly define our problem formulation and the proposed methods for solving it.

The recommendation generator integrates dialogue data with both contextual understanding and user behavior analysis to provide relevant movie suggestions. Our method comprises three main components: the conversational engine, the recommendation engine, and the feature matching and retrieval layer.

Figure 5.1 illustrates the overall architecture of the proposed hybrid conversational recommendation system, which is organized into two parallel branches converging at a shared fusion layer. The *conversational engine* processes the user's natural language query through three sequential stages: a pre-processing stage that normalizes the raw query and simultaneously extracts a Film Vector and a Metadata Vector from the film database, encoding item content and structural attributes respectively; an Intent Detection stage that applies Named Entity Recognition and vector embedding to identify domain-specific entities (e.g., genres, actors) and classify the user's intent into a compact Intent Vector; and a Retrieval stage that computes semantic similarity between the Intent Vector and item vectors to produce a ranked list of semantically relevant candidates. The *recommendation engine* operates independently on historical user-item interaction dataThe recommendation engine operates on historical user-item interaction data and is built upon a systematic empirical evaluation of graph-based neural network methods. Specifically, we benchmark twelve graph-based approaches. The behavioral ranking score derived from historical user-item interaction patterns is independently fused with the retrieval score produced by the conversational retrieval engine, which grounds candidate items in the current dialogue context via RAG. Finally, the *feature matching and retrieval layer* serves as the fusion point of the entire system, aligning the candidate lists from both branches by item identity and jointly re-scoring each item based on its semantic relevance to the expressed user intent and its predicted preference strength from behavioral history, thereby producing a final recommendation list that is simultaneously context-aware and personalized.

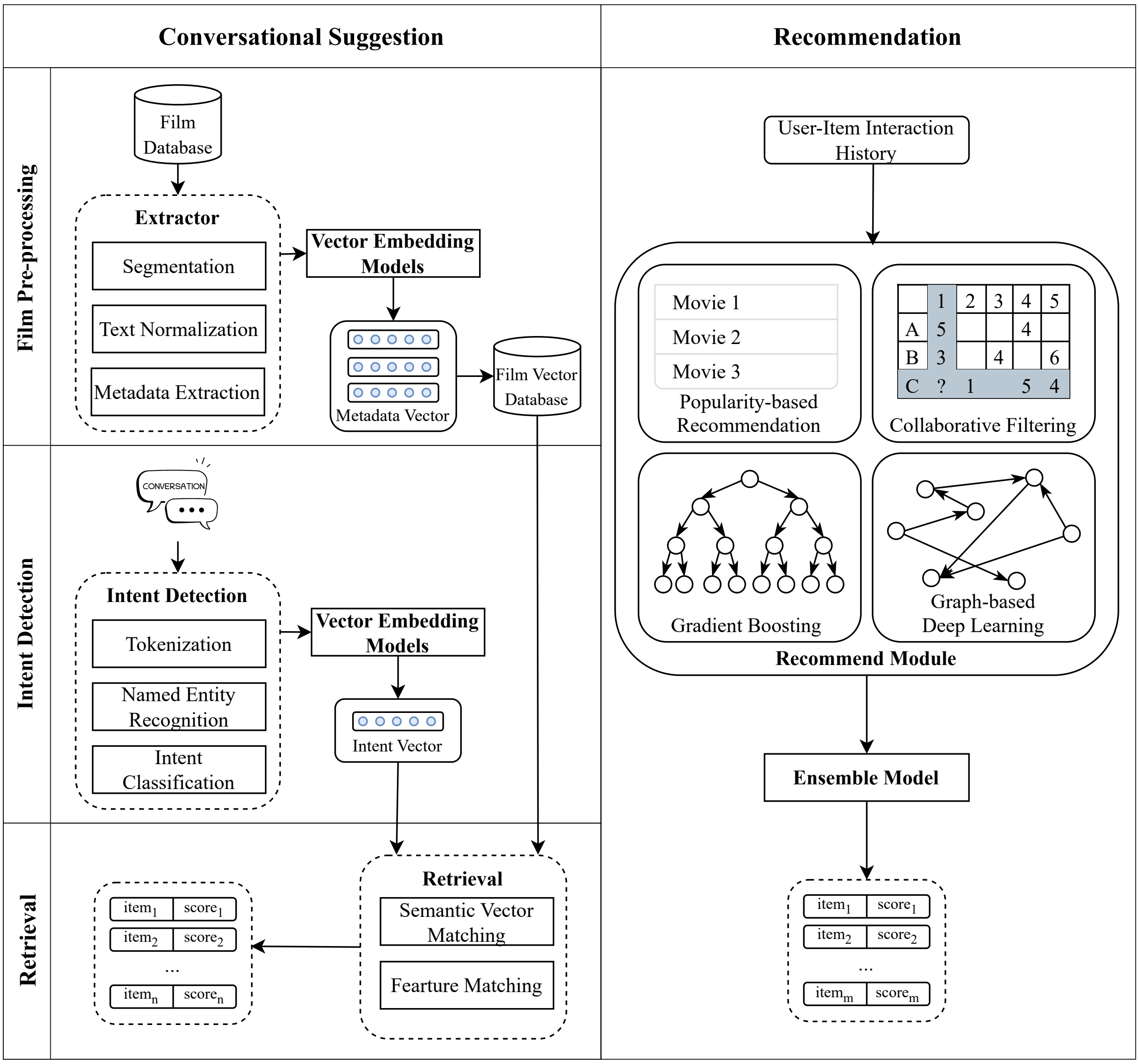

Figure 5.1: Architecture of the conversational suggestion and recommendation generator models

5.2.1 Experimental Results

Recommendation Engine Result

The running results of all models are demonstrate in Table 5.1.

Table 5.3 evaluates twelve recommendation models on two datasets MovieLens-1M and TV360 using Recall@30, Precision@30, Recall@50, and Precision@50. The results consistently show that models incorporating self-supervised contrastive learning or representation alignment objectives outperform standard graph propagation ap-

Table 5.1: Performance comparison between baseline methods on Movielens-1M and TV360 datasets. R@K refers to Recall top-K, and P@K represents Precision top-K. The best result for each dataset is highlighted in bold, while the second best is determined by underline.

Dataset	Movielens-1M					TV360				
Metric	R@30	P@30	R@50	P@50	Time	R@30	P@30	R@50	P@50	Time
LightGCN	0.1628	0.1040	0.2373	0.0964	1e-6	0.0423	0.1018	0.0701	0.1184	1e-6
GAT	0.1650	0.0564	0.2395	0.0515	1e-5	0.0323	0.1067	0.0529	**0.1406**	1e-5
PMLP	0.1292	0.0467	0.1920	0.0430	1e-6	0.0127	0.0433	0.0223	0.0449	1e-5
GraphSAGE	0.1482	0.0526	0.2152	0.0482	1e-6	0.0400	0.1246	0.0666	0.1251	1e-6
LINKX	0.1184	0.0406	0.1672	0.0371	1e-6	0.0376	0.1212	0.0609	0.1176	1e-6
MixGCF	0.0192	0.0208	0.0432	0.0256	1e-5	0.0369	0.0269	0.0550	0.0248	1e-5
SGL	0.0192	0.0208	0.0295	0.0196	1e-6	0.0681	0.0449	0.0933	0.0385	1e-6
SimGCL	0.2125	0.1467	0.2872	0.1224	1e-6	0.1038	0.0672	0.1467	0.0593	1e-5
XSimGCL	0.2209	0.1494	0.2905	0.1267	1e-6	0.1725	0.1026	0.2362	0.0885	1e-6
NCL	**0.2497**	**0.1990**	**0.3351**	**0.1685**	1e-6	0.0915	0.0584	0.1296	0.0514	1e-6
SSL4Rec	0.1933	0.1208	0.2713	0.1068	1e-5	0.1082	0.0730	0.1644	0.0678	1e-5
DirectAU	0.1315	0.0835	0.1820	0.0723	1e-5	**0.1932**	**0.1250**	**0.2714**	0.1095	1e-5

Bold = Best, Underline = Second Best

proaches by a significant margin. Non-graph or augmentation-only methods such as PMLP, LINKX, MixGCF, and SGL fall into the bottom tier across both datasets.

Conversational Retrieval Engine Result

Table 5.2: Recall@K scores in Movielens-1M dataset across 3 LLMs. The results are measured in various K values from 10 to 200, accompanied by answer generation time in seconds. The generated response outputs the top film provided by the chatbot modules.

	Recall@10	**Recall@20**	**Recall@30**	**Recall@50**	**Recall@100**	**Recall@200**	**Time (s)**
Gemma-2B	0.186	0.201	0.214	0.259	0.401	0.829	2.05
LLaMA-3B	0.204	0.227	0.239	0.291	0.428	0.869	2.28
Qwen-3B	**0.207**	**0.232**	**0.246**	**0.299**	**0.435**	**0.888**	**2.42**

Bold = Best, Underline = Second Best

Qwen-3B achieves the highest recall scores across most K values in the TV360 dataset, with Recall@10 at 0.073 and Recall@200 at 0.740, indicating its superior retrieval capability in this context. LLaMA-3B follows closely, showing slightly lower performance but maintaining strong consistency, such as Recall@10 at 0.072 and Recall@200 at 0.731. Gemma-2B exhibits the lowest recall metrics across all thresholds, particularly underperforming at broader recall levels like Recall@100 (0.221) and Recall@200 (0.703). In terms of response time, Gemma-2B is the fastest at 1.88 seconds, while Qwen-3B is the slowest at 2.18 seconds, showing a minor latency trade off for improved accuracy.

Table 5.3: Recall@K scores on TV360 dataset across 3 LLMs. The results are measured in various K values from 10 to 200, accompanied by answer generation time in seconds. The generated response outputs the top film provided by both modules.

	Recall@10	**Recall@20**	**Recall@30**	**Recall@50**	**Recall@100**	**Recall@200**	**Time (s)**
Gemma-2B	0.060	0.071	0.084	0.106	0.221	0.703	1.88
LLaMA-3B	0.072	**0.086**	0.093	0.112	0.226	0.731	2.13
Qwen-3B	**0.073**	0.083	**0.095**	**0.118**	**0.233**	**0.740**	**2.18**

Bold = Best, Underline = Second Best

Ensemble Engine Result

Table 5.4: Performance comparison between baseline recommendation methods combined with conversational engine on Movielens-1M and TV360 datasets. R@K refers to Recall top-K and P@K represents Precision top-K

Dataset	Movielens-1M				TV360			
Metric	R@10	R@20	R@30	R@50	R@10	R@20	R@30	R@50
XSimGCL	0.2046	0.2353	0.2503	0.3012	**0.1258**	**0.1532**	**0.1812**	**0.2681**
NCL	**0.2104**	**0.2387**	**0.2524**	**0.3402**	0.0657	0.0884	0.1021	0.1458
DirectAU	0.1365	0.1581	0.1872	0.2447	0.1106	0.1582	0.1808	0.2606

Bold = Best, Underline = Second Best

Compared to using only a chatbot engine, the ensemble models exceed the chatbot baselines by a considerable margin. This demonstrates the effectiveness of combining two types of recommendation strategies through a traditional matrix factorization framework and a conversational engine.

Regarding the Movielens-1M dataset, NCL remains the most effective recommendation technique even when combined with the conversational engine, as the ensemble between the NCL and Qwen-3B model yields the best result. In particular, it witnesses a slight increase of 1.64%, 2.89%, 2.60%, and 13.78% on the Recall metric at k equals 10, 20, 30, and 50, respectively, compared to the standalone chatbot engine. Comparing to the recommendation module, XSimGCL achieves recall gains of 13.3% at *k=30* while also improving precision by 11.6% at *k=30* and 7.7% at *k=50*. DirectAU experiences a similar trend of higher magnitude. Specifically, recall improvements are substantial: 42.4% at *k=30* and 34.5% at *k=50*, which is the largest relative recall gains of any model. Precision likewise improves by +20.0% at *k=30* and +23.7% at *k=50*. However, ensemble gains with NCL are modest.

Regarding the TV360 dataset, the ensemble between the XSimGCL and Qwen-3B model yields the best result. Specifically, compared to the chatbot module, the ensemble result demonstrates a drastic increase of more than 70% at Recall@10 to roughly under 130% at Recall@50. This can be attributed to the performance gap between the two separate modules. It is noteworthy that DirectAU performs best among the baseline models in the TV360 dataset when working individually, but performance drops when combined with the conversational module, leading to XSimGCL surpassing this model when combined with the conversational engine. As a result, while XSimGCL and NCL witness an increase from approximately 5% to 13% depending on the value of k on the recall metric, DirectAU shows a slight drop of roughly 5%. This can be explained by the fact that DirectAU is specifically designed to learn well-aligned and uniformly distributed latent representations from historical interaction data. However, when combined with the conversational module, the additional dialogue-derived signals may introduce noise or weakly calibrated preferences that partially dilute the stronger collaborative signal learned by DirectAU. In summary, the ensemble module shows more significant improvements at higher item retrieval thresholds. The current evaluation emphasizes relative performance improvements under controlled model capacity. While larger backbone models may further increase absolute performance, the observed gains in these experiments indicate that the proposed fusion mechanism consistently improves over strong baselines within the same parameter scale.

5.3 Summary

This chapter presents CG-RAG, a novel hybrid conversational recommendation framework that unifies graph neural network based structured preference modeling with large language model driven conversational understanding through retrieval-augmented generation. The proposed architecture is organized into two complementary branches: a recommendation engine that leverages graph-based deep learning methods to capture long-term user item interaction patterns, and a conversational engine that processes real-time natural language queries through metadata extraction, intent detection, and semantic retrieval. These two branches converge at a shared fusion layer, where behavioral ranking scores derived from historical interaction data are jointly combined with retrieval scores grounded in the current dialogue context via Polyak averaging, producing a final recommendation list that is simultaneously context-aware and personalized.

Conclusions

Summary of Contributions

This dissertation presents a comprehensive deep learning framework for modern recommendation systems, addressing fundamental challenges in real world large scale environments including scalability, data sparsity, cold start, multi domain learning, and conversational recommendation.

Chapter 2 proposes EfficientRec, a scalable user ID independent recommendation framework based on deep interaction modeling, contrastive learning, and soft clustering.

Chapter 3 develops a unified auxiliary interaction fusion framework based on Graph Neural Networks to address data sparsity and cold start challenges.

Chapter 4 introduces CNL4Rec, a continual learning framework for multi domain recommendation based on domain masking and domain specialization mechanisms.

Chapter 5 advances conversational recommendation by proposing a hybrid framework that integrates GNN based structured preference modeling with Large Language Models and Retrieval Augmented Generation.

Overall, the proposed models achieve superior performance across multiple benchmark datasets and deployment scenarios while providing strong practical value for real world recommendation systems operating under dynamic, sparse, and multi domain conditions.

Limitations

Despite the contributions of this dissertation, several limitations warrant acknowledgment. First, while the proposed frameworks improve scalability by eliminating explicit user ID dependency, training deep neural architecturesparticularly graph neural networks and continual learning mechanismsstill demands substantial computational resources, posing challenges for ultra-large industrial deployments. Second, the effectiveness of side information fusion relies on the quality of auxiliary data, which is often noisy or incomplete in practice; issues of fairness and bias induced by such data remain unaddressed. Third, the continual learning framework assumes clear domain boundaries, which may not hold in highly overlapping or rapidly evolving scenarios. Fourth, the hybrid conversational recommendation system, while enhancing reasoning capabilities, introduces challenges including inference latency, hallucination risks, and unresolved privacy concerns. Finally, although extensive experiments were conducted across multiple datasets, generalization to other domains such as finance and healthcare requires further empirical validation. These limitations indicate promising directions for future resea

Future Work

Future research can extend this work in several promising directions: (1) production-oriented optimization through incremental updates, online learning, and real-time feedback mechanisms; (2) multimodal personalization incorporating images, audio, and video via cross-modal fusion techniques; (3) efficient deployment using lightweight architectures and privacy-preserving learning for on-device inference; and (4) fairness and sustainability considerations including bias mitigation and energy efficiency. These directions point toward recommendation systems that are continuously optimized, multimodally enriched, and deeply integrated into real-world operational environments.

References

[1] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” *IEEE transactions on knowledge and data engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[2] C. C. Aggarwal *et al.*, *Recommender systems*. Springer, 2016, vol. 1.

[3] J. Bennett, S. Lanning *et al.*, “The Netflix prize,” *Proceedings of KDD Cup and Workshop*, vol. 2007, p. 35, 2007.

[4] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum Press, 1981.

[5] J. Bobadilla, F. Ortega, and A. Hernando, “Collaborative filtering based on significances,” *Information Sciences*, vol. 185, no. 1, pp. 1–17, 2012.

[6] R. Burke, “Hybrid recommender systems: Survey and experiments,” *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002.

[7] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *International Conference on Machine Learning*. PMLR, 2020, pp. 1597–1607.

[8] T. Chen, H. Yin, Q. V. H. Nguyen, W.-C. Peng, X. Li, and X. Ji, “Learning elastic embeddings for customizing on-device recommenders,” in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 2021, pp. 138–147.

[9] P. Covington, J. Adams, and E. Sargin, “Deep neural networks for youtube recommendations,” in *Proceedings of the 10th ACM conference on recommender systems*, 2016, pp. 191–198.

[10] P. Cremonesi, Y. Koren, and R. Turrin, “Performance of recommender algorithms on top-n recommendation tasks,” in *Proceedings of the Fourth ACM Conference on Recommender Systems*. ACM, 2010, pp. 39–46.

[11] U. Gupta *et al.*, "Architectural implications of embedding dimension during recommendation inference," *arXiv preprint arXiv:2006.02479*, 2020.

[12] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9729–9738.

[13] Z. Huang, H. Chen, and D. Zeng, "Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 116–142, 2004.

[14] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender systems: an introduction.* Cambridge university press, 2010.

[15] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2019.

[16] H. Ko, S. Lee, Y. Park, and A. Choi, "A survey of recommendation systems: Recommendation models, techniques, and application fields," *Electronics*, vol. 11, no. 1, p. 141, 2022.

[17] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[18] X. N. Lam, T. Vu, T. D. Le, and A. D. Duong, "Addressing cold-start problem in recommendation systems," in *Proceedings of the 2nd international conference on Ubiquitous information management and communication*, 2008, pp. 208–211.

[19] H. Lee, J. Im, S. Jang, H. Cho, and S. Chung, "MeLU: Meta-learned user preference estimator for cold-start recommendation," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2019, pp. 1073–1082.

[20] B. Lika, K. Kolomvatsos, and S. Hadjiefthymiades, "Facing the cold start problem in recommender systems," *Expert Systems with Applications*, vol. 41, no. 4, pp. 2065–2073, 2014.

[21] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003.

[22] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval.* Cambridge University Press, 2008.

[23] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[24] Y.-J. Park and A. Tuzhilin, “The long tail of recommender systems and how to leverage it,” *ACM RecSys*, pp. 11–18, 2008.

[25] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*, 2nd ed. Springer, 2015.

[26] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Application of dimensionality reduction in recommender system – a case study,” in *ACM WebKDD Workshop*, 2000.

[27] ——, “Item-based collaborative filtering recommendation algorithms,” in *Proceedings of the 10th international conference on World Wide Web*, 2001, pp. 285–295.

[28] M. Saveski and A. Mantrach, “Item cold-start recommendations: Learning local collective embeddings,” in *Proceedings of the 8th ACM Conference on Recommender Systems*. ACM, 2014, pp. 89–96.

[29] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, “Collaborative filtering recommender systems,” in *The Adaptive Web*. Springer, 2007, pp. 291–324.

[30] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, “Methods and metrics for cold-start recommendations,” in *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2002, pp. 253–260.

[31] Y. Shi, M. Larson, and A. Hanjalic, “Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges,” *ACM Computing Surveys*, vol. 47, no. 1, pp. 1–45, 2014.

[32] X. Su and T. M. Khoshgoftaar, “A survey of collaborative filtering techniques,” *Advances in Artificial Intelligence*, vol. 2009, p. 421425, 2009.

[33] J. Wei, J. He, K. Chen, Y. Zhou, and Z. Tang, “Collaborative filtering and deep learning based recommendation system for cold start items,” in *Expert Systems with Applications*, vol. 69. Elsevier, 2017, pp. 29–39.

[34] J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, and X. Xie, “Self-supervised graph learning for recommendation,” in *Proceedings of the 44th International ACM*

SIGIR Conference on Research and Development in Information Retrieval. ACM, 2021, pp. 726–735.

[35] S. Wu, F. Sun, W. Zhang, X. Xie, and L. Cui, Bin, "Graph neural networks in recommender systems: a survey," *ACM Computing Surveys*, vol. 55, no. 5, pp. 1–37, 2022.

[36] J. Yu, H. Yin, X. Xia, T. Chen, J. Li, and Z. Huang, "Self-supervised learning for recommender systems: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 1, pp. 335–355, 2023.

[37] X. Yu, X. Wang, X. He, F. Feng, M. Wang, and T.-S. Chua, "Are graph augmentations necessary? simple graph contrastive learning for recommendation," in *Proceedings of the 30th ACM International Conference on Multimedia.* ACM, 2022, pp. 35–44.

[38] T. Zang, Y. Zhu, H. Liu, R. Zhang, and J. Yu, "A survey on cross-domain recommendation: taxonomies, methods, and future directions," *ACM Transactions on Information Systems*, vol. 41, no. 2, pp. 1–39, 2022.

[39] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM computing surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.

[40] Z. Zhao, L. Hong, L. Wei, J. Chen, A. Nber, S. Shaked, D. Verna, K. Singh, C. Rosenberg, and E. H. Chi, "Recommending what video to watch next: A multitask ranking system," in *Proceedings of the 13th ACM Conference on Recommender Systems.* ACM, 2019, pp. 43–51.