VIETNAM NATIONAL UNIVERSITY, HANOI
UNIVERSITY OF ENGINEERING AND TECHNOLOGY

NGUYỄN TRẦN NGỌC LINH

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

(Nghiên cứu mô hình hoá học sâu dữ liệu chính tắc và phụ trợ nhằm nâng cao tính vững chắc và thích nghi của hệ thống khuyến nghị)

PhD DISSERTATION INFORMATION SYSTEM

HA NOI - 2026

VIETNAM NATIONAL UNIVERSITY, HANOI
UNIVERSITY OF ENGINEERING AND TECHNOLOGY

NGUYỄN TRẦN NGỌC LINH

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

(Nghiên cứu mô hình hoá học sâu dữ liệu chính tắc và phụ trợ nhằm nâng cao tính vững chắc và thích nghi của hệ thống khuyến nghị)

PhD DISSERTATION INFORMATION SYSTEM

MAJOR: INFORMATION SYSTEM
MAJOR CODE: 9480104

PhD **SUPERVISORS**

CONFIRMATION OF THE TRAINING INSTITUTION

HA NOI - 2026

Declaration - Lời cam đoan

I hereby certify that this Doctoral Dissertation has been completed by me in partial fulfillment of the requirements for the degree of Doctor of Philosophy under the supervision and guidance of my supervisors.

The work presented in this dissertation is entirely my own, except where collaboration has been explicitly acknowledged in the text. This dissertation has not been submitted, in whole or in part, for any other degree, diploma, or qualification at any university or institution, nor is it currently under consideration elsewhere.

Tôi xin cam đoan rằng luận án Tiến sĩ này được thực hiện bởi chính tôi nhằm đáp ứng yêu cầu của chương trình đào tạo Tiến sĩ dưới sự hướng dẫn và giám sát của các giảng viên hướng dẫn.

Các nội dung trình bày trong luận án là kết quả nghiên cứu của riêng tôi, ngoại trừ những phần có sự hợp tác đã được ghi rõ trong nội dung luận án. Luận án này chưa từng được nộp, toàn bộ hoặc một phần, để xét cấp bất kỳ học vị, văn bằng hoặc chứng chỉ nào tại bất kỳ trường đại học hay cơ sở đào tạo nào khác, và hiện cũng không được xem xét ở nơi nào khác.

Hanoi, May 2026
Author

Nguyễn Trần Ngọc Linh

Acknowledgements

I would like to express my sincere gratitude to my university for providing a supportive academic environment and the necessary resources that made this research possible. The institutional support, research facilities, and academic atmosphere have played an essential role in shaping my scholarly development throughout this doctoral journey. I am especially grateful to the faculty members and lecturers whose dedication to teaching and research has continuously inspired me to pursue academic excellence.

My deepest appreciation goes to my supervisors and academic mentors for their invaluable guidance, insightful feedback, and constant encouragement throughout the course of this research. Their expertise, patience, and rigorous academic standards have significantly contributed to the quality of this work and to my growth as a researcher. I am also thankful to the members of my dissertation committee and other faculty members for their constructive comments and suggestions, which have helped refine and strengthen this dissertation.

I would like to acknowledge my fellow doctoral candidates and research colleagues for the stimulating discussions, collaborative spirit, and mutual support we shared during our studies. Exchanging ideas, challenges, and experiences with them has not only enriched my research but also made this academic journey more meaningful and rewarding.

Finally, I am profoundly thankful to my family and friends for their unwavering love, understanding, and moral support. Their belief in me has been a constant source of strength, especially during challenging times, and their encouragement motivated me to persevere and complete this journey. This dissertation would not have been possible without their patience, sacrifices, and unconditional support.

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 deep learning methods 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 gen-

erate 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 models, 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.*

Table of Contents

LIST OF ABBREVIATIONS

ACPU	Average Content Per User
ADPU	Average Duration Per User
AI	Artificial Intelligence
BM25	BM25 Ranking Algorithm
BPR	Bayesian Personalized Ranking
CBF	Content-Based Filtering
CF	Collaborative Filtering
CL	Contrastive Learning
CNN	Convolutional Neural Network
CRS	Conversational Recommender System
DNN	Deep Neural Network
GCL	Graph Contrastive Learning
GCN	Graph Convolutional Network
GNN	Graph Neural Network
GPU	Graphics Processing Unit
InfoNCE	Information Noise-Contrastive Estimation
KB	Knowledge Base
KG	Knowledge Graph
LLM	Large Language Model
LSTM	Long Short-Term Memory
MAML	Model-Agnostic Meta-Learning
MF	Matrix Factorization
MLP	Multilayer Perceptron

NDCG	Normalized Discounted Cumulative Gain
NLP	Natural Language Processing
RAG	Retrieval-Augmented Generation
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
SSL	Self Supervised Learning
SVD	Singular Value Decomposition
VSM	Vector-Space Model

List of Symbols

Symbol	Description
Sets and indices	
$\mathcal{U} = \{u_1, \ldots, u_M\}$	Set of users
$\mathcal{V} = \{v_1, \ldots, v_N\}$	Set of items
$u,\ v$	A user and an item, respectively
$M = \lvert\mathcal{U}\rvert,\ N = \lvert\mathcal{V}\rvert$	Number of users and number of items
$\Omega \subseteq \mathcal{U} \times \mathcal{V}$	Set of observed user-item interactions
$\bar{\Omega} = (\mathcal{U} \times \mathcal{V}) \setminus \Omega$	Set of unobserved user-item pairs
(u, i) / i'	An observed (positive) pair / a sampled negative item
K	Number of soft preference clusters
Data and representations	
$R \in \mathbb{R}^{M \times N}$	User–item interaction (rating) matrix
$r_{u,v}$	Interaction / rating of user u on item v
R_Ω	Observed entries of R
$X^{(u)} \in \mathbb{R}^{M \times d_u}$	User-side auxiliary feature matrix
$X^{(i)} \in \mathbb{R}^{N \times d_i}$	Item-side auxiliary feature matrix
$d,\ d_u,\ d_i$	Embedding dimension; user- and item-feature dimensions
$\mathbf{e}_u,\ \mathbf{e}_i$	Learned embedding of user u and item i

List of Figures

List of Tables

Preamble

Research Context and Motivation

With the explosive development of the Internet, e-commerce and social media platforms, recommender systems have become essential tools for various business services to help users cope with information overload and improve user experience, engagement, and decision making quality [48, 133]. These systems are applied in various use cases across industries, such as offering relevant products on online shopping platforms like Amazon and Taobao, suggesting friends on online social networks, or generating personalized playlists for video and music streaming services like YouTube, Netflix, and Spotify. Users depend on recommender systems to deal with the information burden and to discover items of interest from a vast array of options, including products, movies, news articles, or restaurants. Therefore, accurately capturing, analyzing and understanding user's preferences from their past interactions, such as clicks, watches, reads, chats, rates, and purchases, are crucial for an effective recommender system [51, 124]. Under this context, recommendation systems have gradually evolved from complementary ranking tools into core infrastructures that directly shape user experience, service quality, and business performance.

However, real-world online services are becoming increasingly complex than ever before [133]. User behaviors vary significantly and change dynamically over time, the item space grows continuously with strong long tail characteristics, and interaction data remain highly sparse and noisy. In addition, modern recommendation systems are required to operate under strict constraints on scalability, latency, fairness, and robustness, while simultaneously adapting to rapidly changing user preferences and content ecosystems. These challenges reveal fundamental limitations in traditional recommendation methodologies and motivate the integration of different types of data and the exploration of more advanced deep learning approaches [124].

The effectiveness of any recommender system fundamentally depends on the quality and diversity of data it can leverage [89]. In this dissertation, a principled categorization is adopted that distinguishes between two types of data: canonical data and auxiliary data. Canonical data refers to the primary user item interaction signals that directly capture user preferences and behavioral patterns, including explicit feedback such as ratings and reviews, as well as implicit feedback such as clicks, views, and purchases

[51]. Auxiliary data include other sources of information that provide additional context on users, items, or their relationships beyond direct interactions, including demographic attributes, content metadata, contextual signals, and social relations [89]. The auxiliary data can also be the user (purchase) intents revealed in conversational texts in which they are involved. How to integrate both canonical data and auxiliary data in order to enhance the efficacy and efficiency of recommendation systems is still a key question that needs to be explored and investigated thoroughly.

Regarding recommendation methods, various techniques have been applied successfully to building recommender systems so far, such as collaborative filtering (like neighborhood user based or item based, clustering based, SVD, matrix factorization), content based, and hybrid approach. However, the recent success of deep learning in computer vision, natural language processing, and speech recognition has inspired the application and evolution of deep models in recommendation systems [17, 124]. Unlike traditional models that are largely based on manually engineered features and complex mathematical transformations (e.g., SVD, matrix factorization), deep learning enables automatic extraction of high level semantic representations from massive volumes of unstructured data. Architectures such as graph neural networks, autoencoders, contrastive self supervised learning frameworks, and more recently large language models (LLMs) have greatly expanded the representational power of recommender systems. These deep architectures offer capabilities for modeling complex patterns from a wide range of structured and unstructured data. Nevertheless, despite their remarkable modeling capacity, deep learning based recommendation models still face critical challenges in practical deployment, particularly in terms of robustness, scalability, adaptability, and dynamicity. Robustness means that a recommender system can work well in extreme data sparsity and cold-start scenarios. Scalability means that a system can still serve efficiently under massive user bases where traditional per ID personalization becomes unmanageable. Adaptability means that a system can make the most of multiple services and domains where user preferences and data distributions continuously evolve. Dynamic adaptability, in the scope of this thesis, items of interest based on both long term preferences and real time user intents revealed in a conversation.

Challenge 1: Scalable and Robust Deep Modeling of Canonical and Auxiliary Data The first fundamental challenge lies in achieving scalable and robust recommendation when canonical interaction data is sparse or unavailable, necessitating effective integration with auxiliary information sources while maintaining computational efficiency at web scale.

Data Sparsity and cold-start: Even in large scale platforms, the vast majority of users interact with only a very small fraction of available items, resulting in highly sparse

canonical user item interaction matrices. For example, Netflix users rate fewer than 1% of available movies [7], and Amazon customers purchase an even smaller fraction of the millions of available products [63]. This sparsity in canonical data creates several difficulties: limited statistical evidence makes it difficult to distinguish actual preferences from noise, reduced collaborative signals weaken the effectiveness of collaborative filtering approaches [82], and long tail distributions lead to biased recommendations that favor popular items.

The cold-start problem emphasizes the importance of auxiliary data integration. When new users or items lack sufficient canonical interaction history, auxiliary information including demographic attributes, content metadata, contextual signals, and social relationships offers a promising direction to compensate for the absence of behavioral signals [124]. Graph based learning has emerged as a powerful paradigm for the unification of canonical and auxiliary data [112], as it naturally models high order relationships between users, items, and auxiliary entities within a unified structure. LightGCN [33] established the foundation by demonstrating that simplified graph convolutions achieve superior performance for collaborative filtering. Self supervised and contrastive learning techniques [111, 119] further enhance the robustness of representation by constructing additional training signals from unlabeled data.

Scalability Constraints: Beyond data quality challenges, modern recommendation systems face critical scalability constraints that limit practical deployment. The standard paradigm of learning unique embedding vectors for each user and item creates impractical memory requirements at web scale. Chen et al. [14] calculated that a system serving 1 billion users with 64 dimensional embeddings requires approximately 238 GB of memory solely for user embeddings. When combined with item embeddings and categorical features commonly used in industrial systems, the total memory requirements can easily exceed the capacity of commodity hardware. This constraint forces practitioners to make difficult trade offs between model expressiveness and computational resources.

Furthermore, the exhaustive item scoring paradigm where all candidate items must be evaluated for each user request creates a fundamental misalignment between offline training and online serving requirements. Production systems must respond within strict latency constraints, often measured in tens of milliseconds, yet most academic research continues to optimize for full matrix reconstruction objectives that are computationally infeasible in real time scenarios.

Research Gap 1: Despite progress in graph based methods, several critical limitations remain in modeling of canonical and auxiliary data at scale. First, auxiliary information may introduce systematic bias if the data are incomplete, noisy, or reflect societal

biases [12]. Most existing fusion methods employ static combination rules that cannot adapt to the varying informativeness of canonical versus auxiliary data sources across diverse user populations. Second, existing contrastive learning methods treat all dimensions of node embeddings uniformly, failing to account for the diverse informativeness of different embedding dimensions that encode canonical versus auxiliary signals. The integration of learnable masks with graph based contrastive learning for adaptive data fusion remains unexplored. Third, current approaches rely heavily on explicit user identifiers with dedicated embedding vectors, creating memory bottlenecks and preventing efficient scaling to web scale user populations. The development of ID independent representation learning that maintains recommendation quality while dramatically reducing model size remains an open challenge.

These gaps motivate Chapters 2 and 3, where supportive solutions are proposed: EfficientRec (Chapter 2) achieves scalable deep modeling of canonical interaction data through behavior driven embeddings that eliminate user ID dependency, soft clustering that enables probabilistic user representation across latent preference groups, and contrastive learning that enhances robustness under sparse conditions. GIFT4Rec and MaskSimGCL (Chapter 3) address robust fusion of canonical and auxiliary data through attention based mechanisms that adaptively weight different information sources and masked contrastive learning that selectively preserves informative embedding dimensions while filtering noise.

Challenge 2: Adaptive Multi-Domain Recommendation

Another fundamental feature of modern recommendation environments is their multi-domain nature, which presents unique challenges for both canonical and auxiliary data modeling. Users frequently interact across multiple services, platforms, and content categories such as movies, music, short videos, e-commerce products, and social content. The canonical interaction patterns learned in one domain may be partially transferable to another, yet each domain also shows specific characteristics in both canonical behaviors and auxiliary contexts [122, 132]. In industrial recommendation systems, platforms such as Taobao and streaming services simultaneously serve users across multiple business domains where each domain exhibits distinct canonical user behavior patterns and auxiliary semantic characteristics [132].

However, most existing deep recommendation models suffer from catastrophic forgetting when deployed in sequential multi-domain settings [47]. When models are updated to accommodate new domains with different canonical and auxiliary data characteristics, previously learned knowledge is often overwritten, resulting in severe performance degradation. Recent industrial deployments have addressed this challenge through approaches such as CTNet [65], KEEP [127], and DIIT [39], but these methods predom-

inantly follow a unidirectional transfer paradigm.

Research Gap 2: A systematic analysis reveals limitations in existing approaches for adaptive multi-domain modeling. Firstly, the predominant paradigm follows a unidirectional transfer approach where knowledge flows exclusively from source domains to target domains creating an inherent imbalance where source domain representations may experience performance degradation over time. Secondly, existing continual learning approaches employ hard constraints that completely freeze parameters considered crucial for previous domains, limiting bidirectional knowledge sharing. Thirdly, no existing framework explicitly optimizes for balanced performance across all domains. These gaps motivate Chapter 4, where CNL4Rec is proposed a continual learning framework for adaptive multi-domain recommendation that employs domain masking and domain specialization mechanisms with soft constraints, enabling multi directional knowledge transfer and fairness oriented optimization of both canonical and auxiliary data.

Challenge 3: Conversational Recommendation Bridging Canonical and Auxiliary Data

As the challenges mentioned above indicate, recommendation systems are undergoing a paradigm shift toward conversational and language driven interaction [21, 41]. This shift basically changes how canonical and auxiliary data are utilized: long term canonical interaction histories must be integrated with real time auxiliary signals from natural language conversations. With the rapid development of large language models, users increasingly expect recommendation systems to support natural language queries, interactive feedback, and explainable suggestions.

User intent in conversational settings can be classified into two primary forms: implicit intent inferred from canonical behavioral data (e.g., browsing history, viewing patterns), and explicit intent communicated through auxiliary conversational input (e.g., natural language queries, chatbot). Traditional recommendation models show strengths in capturing implicit intent from historical canonical data, while chatbot systems are effective at handling explicit intent through conversational understanding. However, large language models alone are not optimized for structured preference learning from canonical data, while traditional recommenders lack flexibility for natural language interaction [62]. Graph based deep learning models have been applied for their ability to produce expressive representations from canonical interaction graphs [33], while LLMs display remarkable capabilities for processing auxiliary textual information [6, 15].

Research Gap 3: Bridging structured recommendation models for canonical data with language based generative intelligence for auxiliary conversational signals requires hybrid integration [55]. Existing conversational recommendation systems struggle with data sparsity in conversational contexts, available canonical interaction data within a

single dialogue session is essentially limited, necessitating effective utilization of auxiliary conversational context. The challenge lies in combining long term canonical user interaction histories with real time user intent to generate accurate, personalized recommendations. This gap motivates Chapter 5, where a hybrid conversational recommendation framework is proposed that combines GNN based preference modeling of canonical interaction data with LLM and RAG powered semantic reasoning over auxiliary conversational context.

Deep Modeling of Canonical and Auxiliary Data for Robust and Adaptive Recommendation

Motivated by these fundamental challenges and research gaps, this dissertation focuses on developing deep learning based recommendation methods that achieve robust and adaptive recommendation through deep modeling of both canonical and auxiliary data. At the core of this framework lies the principle of learning informative, robust, and transferable user representations by integrating canonical interaction signals with auxiliary contextual information through advanced deep learning architectures. Table 3 summarizes the mapping between the key challenges and the dissertation chapters.

Table 3: Mapping of key challenges to dissertation chapters

Challenge	Ch. 2	Ch. 3	Ch. 4	Ch. 5
Canonical data sparsity	✓	✓		✓
cold-start (auxiliary fusion)	✓	✓		
Scalability	✓			
multi-domain adaptation			✓	
Conversational context				✓

Robust and scalable recommendation is advanced by constructing interaction based user representations from canonical data that eliminate the dependency on explicit user identifiers (Chapter 2). Through soft clustering and contrastive learning, large user populations are effectively organized into latent preference structures derived from canonical behavioral patterns. To address cold-start through canonical versus auxiliary fusion, interaction signals and auxiliary side information are integrated through graph neural representation learning and masked contrastive self supervision (Chapter 3).

For adaptive multi-domain recommendation, a continual learning framework based on domain masking and domain specialization is introduced to regulate parameter updates across domains with varying canonical and auxiliary data characteristics (Chapter 4). Domain critical parameters are selectively protected while adaptive learning for new

domains is enabled.

In the conversational recommendation setting, a hybrid framework integrating graph neural networks for canonical data modeling with retrieval augmented generation and large language models for auxiliary conversational processing is proposed (Chapter 5). This design effectively bridges canonical behavioral signals with auxiliary linguistic inputs.

Research Questions

The challenges and research gaps identified above motivate the central research question of this dissertation: How can robust and adaptive recommendations be achieved through deep modeling of both canonical interaction data and auxiliary contextual information? This question is decomposed into three sub-questions (SQs):

SQ1: How can deep learning architectures achieve robust modeling of canonical interaction data and effective fusion with auxiliary side information to address sparsity, cold-start, and scalability challenges? (Chapter 2 and 3)

SQ2: How can deep learning models adaptively preserve prior knowledge of both canonical and auxiliary representations while achieving stable adaptation across multiple recommendation domains? (Chapter 4)

SQ3: How can a hybrid conversational recommendation framework unify long term canonical user behavior and real time auxiliary conversational intent to generate context aware personalized recommendations? (Chapter 5)

Research Objectives

To answer the research questions posed above, this dissertation pursues the following concrete objectives, each addressing one sub-question:

O1: Proposing deep learning based recommendation models that achieve robustness against cold-start, data sparsity, and scalability challenges by learning informative representations from canonical user item interactions and effectively integrating auxiliary side information.

O2: Developing an adaptive recommendation model capable of delivering balanced and reliable performance across multiple domains with varying canonical and auxiliary data characteristics, while preserving prior knowledge during domain adaptation.

O3: Building a hybrid conversational recommendation framework that combines

the long term user interaction history (canonical data) with real time conversational user intent (auxiliary data) in order to produce accurate, personalized, and contextually aligned recommendations.

Research Methodology

To address the research objectives and questions, this dissertation obeys the following research methods:

- **Quantitative research methods** involve the statistical analysis of large-scale user-item interaction and auxiliary data, testing the hypotheses addressed within the dissertation.
- **Qualitative research methods** focus on understanding the structure and context of canonical interaction data and heterogeneous auxiliary information, identifying the strengths and weaknesses of current research approaches in order to propose new solutions and models to deal with the data sparsity, cold-start, scalability, multi-domain adaptation, and conversational recommendation problems.
- **Experimental research methods** were intensively used to conduct experiments in order to confirm the hypotheses and validate the accuracy and effectiveness of the proposed models on both public benchmark datasets and real-world industrial data.

Each method plays a critical role in problem understanding, designing and developing models for the recommendation tasks over canonical and auxiliary data. This dissertation integrates these research methods to leverage the strengths of each method and ensure a comprehensive, appropriate, and reliable evaluation of both the technical and practical aspects of developing deep learning based recommender systems.

Main Contributions of the Dissertation

This dissertation presents three key contributions that directly address the identified research gaps in deep modeling of canonical and auxiliary data for recommendation:

Contribution 1: (Chapters 2 and 3): Robust deep modeling of canonical and auxiliary data is developed through complementary innovations addressing scalability, sparsity, and cold-start challenges. First, EfficientRec (Chapter 2) achieves scalable recommendation through: learning robust behavioral representations from canonical interaction data without relying on fixed user identifiers, pioneering the application of neural soft clustering to individual user recommendation, integrating contrastive learning with clustering, and achieving efficient cluster based inference that avoids exhaustive item

scoring. Second, GIFT4Rec and MaskSimGCL (Chapter 3) achieve robust canonical and auxiliary fusion through: an attention based Weight Generated module that dynamically computes user's specific fusion weights controlling the relative contribution of canonical behavioral embeddings and auxiliary side information features, dual module fusion with local and global side information fusion employing meta learning optimization, and combining learnable masks that adaptively weight embedding dimensions, identifying task relevant parameters encoding canonical versus auxiliary signals. The framework has been validated through academic metrics and industrial deployment on the TV360 platform. This contribution was published in the ACIIDS 2022 [P1], ACIIDS 2023 [P2], and KSE 2024 [P3].

Contribution 2 (Chapter 4): A continual learning framework for adaptive multi-domain recommendation is developed that preserves domain specific knowledge while enabling adaptation across domains with varying canonical and auxiliary data characteristics. CNL4Rec achieves this through enabling multi directional knowledge transfer of canonical interaction patterns and auxiliary semantic structures rather than unidirectional source to target transfer, employing soft constraint mechanisms through domain masking that modulate rather than eliminate gradient updates, allowing parameters important for each domain to continue adapting, implementing fairness oriented optimization that evaluates and optimizes overall performance across all domains rather than solely target domain metrics. This contribution was published in the ACIIDS 2024 [P4].

Contribution 3 (Chapter 5): A hybrid conversational recommendation framework bridging canonical and auxiliary data is deployed that integrates structured preference modeling with language based semantic reasoning. This achieves contextual integration through combining past behavioral signals from GNN based modeling of canonical interaction data with real time conversational preferences from LLMs processing auxiliary linguistic inputs, employing retrieval augmented generation that grounds LLM responses in actual canonical user histories and item information, implementing ensemble learning to combine canonical data driven recommendation and auxiliary context driven conversational engines, achieving real time inference suitable for chatbot deployment while maintaining recommendation accuracy. This contribution has been submitted to the Journal of IEEE Access (2025) [P5].

Together, these contributions establish a deep learning foundation for robust and adaptive recommendation systems that effectively model both canonical data and auxiliary information, achieving scalability, domain adaptability, and semantic enrichment across diverse interaction modalities.

Scope of the Dissertation

The scope of this dissertation is centered on developing deep learning methods for robust and adaptive recommendation through deep modeling of canonical and auxiliary data. The research is organized around four technical chapters:

Chapters 2 and 3: These chapters focus on robust deep modeling of canonical user item interactions and effective fusion with auxiliary side information. Chapter 2 advances scalable modeling through ID free representation learning, soft clustering, and contrastive learning. Chapter 3 investigates attention based fusion mechanisms, meta learning for generalization, and masked contrastive learning.

Chapter 4: Adaptive continual learning for multi-domain recommendation with varying canonical and auxiliary data characteristics is explored, with emphasis on domain masking, domain specialization, and parameter based continual learning.

Chapter 5: Hybrid conversational recommendation combining GNN based modeling of canonical data with LLM based processing of auxiliary conversational context is integrated. Fully end to end LLM recommenders and open domain conversational agents are outside the scope.

Dissertation Outline

The Dissertation outline is illustrated in Figure 1, which contains a Preamble, five chapters, and a Conclusion. The related publications are marked to their corresponding Chapter:

Chapter 1: LITERATURE REVIEW OF BACKGROUND AND METHODS provides an overview of key concepts in recommendation systems, including problem formulation, canonical and auxiliary data types, traditional approaches, deep learning architectures, and evaluation metrics.

Chapter 2: ROBUST LARGE-SCALE RECOMMENDATION VIA INTERACTION EMBEDDING AND SOFT CLUSTERING introduces EfficientRec for scalable deep modeling of canonical interaction data through ID free user representations, neural soft clustering, and contrastive learning. [P1] (ACIIDS, 2022)

Chapter 3: BOOSTING RECOMMENDATION VIA GRAPH BASED FUSION OF CANONICAL INTERACTIONS AND AUXILIARY SIDE INFORMATION presents GIFT4Rec for attention based canonical auxiliary fusion with meta learning, and MaskSimGCL for masked graph contrastive learning. [P2] (ACIIDS, 2023), [P3] (KSE, 2024)

Chapter 4: ENHANCING MULTI-DOMAIN RECOMMENDATION WITH CON-

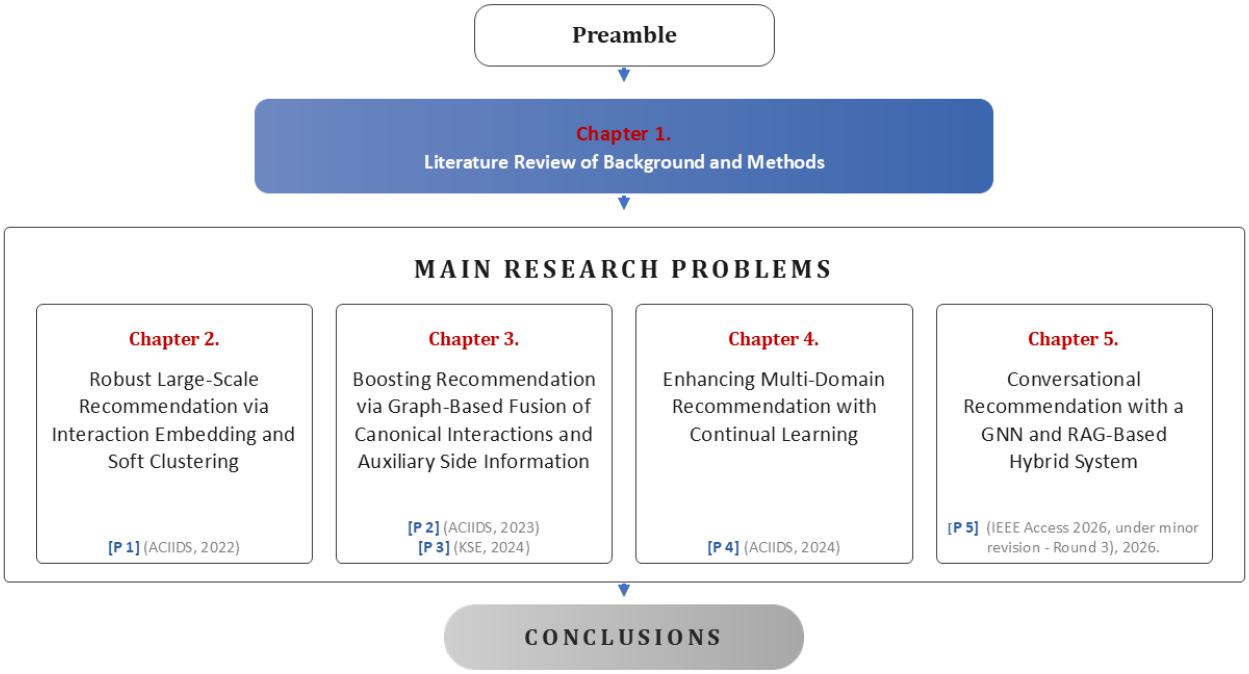

Figure 1: Outline of the dissertation. Each technical chapter addresses one facet of robust and adaptive recommendation over canonical and auxiliary data. Chapter 2 (EfficientRec): scalable modeling of canonical interactions; Chapter 3 (GIFT4Rec, MaskSimGCL): canonical-auxiliary fusion via meta-learning and masked graph contrastive learning; Chapter 4 (CNL4Rec): adaptive multi-domain continual learning; Chapter 5 (CG-RAG): conversational recommendation with GNN, LLM, and RAG. Corresponding publications are indicated for each chapter.

TINUAL LEARNING introduces CNL4Rec for adaptive multi-domain recommendation with domain masking and specialization mechanisms. [P4] (ACIIDS, 2024)

Chapter 5: CONVERSATIONAL RECOMMENDATION WITH A GNN AND RAG-BASED HYBRID SYSTEM proposes a hybrid framework bridging GNN based canonical data modeling with LLM and RAG powered auxiliary conversational processing. [P5] (Journal of Big Data, 2025, In peer reviewing)

Conclusions summarizes the dissertation's main contributions, discusses limitations, and provides an outlook for future work.

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 [78]. These systems have become essential components of modern digital platforms, powering personalized experiences across e-commerce, streaming services, social networks, and content platforms [48, 124]. 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 [84].

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 [70]. From a machine learning perspective, recommendation is a prediction task that aims to estimate missing values in a partially observed user item interaction matrix [51]. From an optimization standpoint, the objective is to maximize a utility function that captures both user satisfaction and business objectives.

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 an interaction matrix $R \in \mathbb{R}^{M \times N}$ between them [78]. 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. The recommendation task thus becomes:

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

Interaction History Representation

User item interactions history are conventionally represented as a rating matrix $\mathbf{R} \in \mathbb{R}^{M \times N}$, where $M = |\mathcal{U}|$ denotes the number of users and $N = |\mathcal{V}|$ denotes the number of items [51, 76]. Each entry $r_{u,v}$ in this matrix represents the observed interaction between user u and item v.

For explicit feedback systems, $r_{u,v}$ captures a user's rating that directly reflects their preference, such as a 5 star rating on a movie or a numerical score for a product [50]. For implicit feedback systems, $r_{u,v}$ encodes behavioral signals such as clicks, views, or purchases which indirectly indicate user interest without explicit preference statements [38].

A critical characteristic of the rating matrix is that only a small fraction of entries are observed. Let $\Omega \subseteq \mathcal{U} \times \mathcal{V}$ denote the set of observed user item pairs. The recommendation task can then be formulated as matrix completion: given the partially observed matrix $\mathbf{R}_\Omega$, predict the missing entries $\mathbf{R}_{\bar{\Omega}}$ where $\bar{\Omega} = (\mathcal{U} \times \mathcal{V}) \setminus \Omega$ represents unobserved interactions [51].

$$\hat{\mathbf{R}} = \mathcal{F}(\mathbf{R}_\Omega; \Theta) \tag{1.4}$$

where $\mathcal{F}$ represents the recommendation model parameterized by Θ, and $\hat{\mathbf{R}}$ denotes the

predicted complete rating matrix.

The density of the rating matrix, defined as $|\Omega|/(M \times N)$, is typically extremely low in real-world systems often less than 1% for large scale platforms [124]. This extreme sparsity fundamentally shapes the design of recommendation algorithms and motivates many of the advanced techniques discussed in this dissertation.

1.1.2 Research Scope and Objectives

This dissertation develops deep learning based solutions that address the interconnected challenges of sparsity, cold-start, and scalability. The proposed approaches share a common objective: learning robust, transferable representations that capture meaningful user preferences without requiring exhaustive interaction histories or expensive computational resources.

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				✓

1.2 Recommendation Data Types

The effectiveness of any recommender system fundamentally depends on the quality and diversity of data it can leverage [78, 89]. In this dissertation, we adopt a principled categorization that distinguishes between two complementary types of data: canonical data and auxiliary data.

Canonical data refers to the primary user item interaction signals that directly capture user preferences and behavioral patterns [51].

Auxiliary data encompasses all supplementary information sources that provide additional context about users, items, or their relationships beyond direct interactions [89].

The central thesis of this dissertation is that robust and adaptive recommendation requires the complementary integration of both canonical and auxiliary data through

deep learning architectures. Whereas canonical data provides the behavioral signals essential for preference learning, auxiliary data provides the semantic and contextual richness needed for generalization beyond observed interactions.

1.2.1 Canonical Data

Canonical data captures the direct interactions between users and items, serving as the primary source of preference signals in recommender systems [38, 51]. These interactions can be categorized based on how user preferences are expressed.

Explicit feedback

Explicit feedback represents direct expressions of user preferences, typically in the form of numerical ratings or categorical evaluations [50]. Common examples include: star ratings (1–5) on Amazon, like/dislike buttons on YouTube, and review scores on Yelp.

Implicit feedback

Implicit feedback encompasses behavioral signals that indirectly indicate user preferences without explicit preference statements [38, 76]. Examples include: user behaviors such as click throughs, purchases, viewing or listening history, search and browsing activities, and add-to-cart or wish list actions.

Interaction matrix characteristics

Regardless of feedback type, canonical data is represented as a user item interaction matrix $\mathbf{R} \in \mathbb{R}^{M \times N}$, where $M = |\mathcal{U}|$ and $N = |\mathcal{V}|$. The critical characteristic of this matrix is its extreme sparsity [124]:

$$\text{Density} = \frac{|\Omega|}{M \times N} \ll 1\% \tag{1.5}$$

Table 1.2 presents sparsity statistics from representative benchmark datasets, illustrating the severity of this challenge.

Table 1.2: Sparsity Statistics of Representative Recommendation Datasets

Dataset	Users	Items	Interactions	Density
MovieLens-1M	6,040	3,706	1,000,209	4.47%
MovieLens-20M	138,493	27,278	20,000,263	0.53%
Amazon-Books	294,739	1,477,922	8,654,619	0.002%
Yelp2018	31,668	38,048	1,561,406	0.13%
Gowalla	29,858	40,981	1,027,370	0.08%

This extreme sparsity fundamentally limits what can be learned from canonical data alone and motivates the integration of auxiliary information [89].

1.2.2 Auxiliary Data

Auxiliary data provides supplementary signals that enrich the semantic understanding of users, items, and their relationships [89]. Unlike canonical interaction data, auxiliary data does not directly encode preferences but offers contextual and descriptive information that supports more robust preference inference.

User-Side information

User auxiliary data describes characteristics of users beyond their interaction history [9, 74]:

- Demographic attributes: Age, gender, location, occupation, education level, and income provide simple, high level preference signals that reflect users' consumption patterns [74].
- Psychographic profiles: Interests, lifestyle preferences, personality traits, and value orientations offer deeper preference signals when available through surveys or inferred from behavior [129].
- Account metadata: Registration date, subscription tier, device information, and platform engagement metrics may correlate with behavioral patterns [17].
- Social connections: Friend lists, follower relationships, and group memberships encode social influence pathways [20].

Formally, user auxiliary data can be represented as a feature matrix $\mathbf{X}^{(u)} \in \mathbb{R}^{M \times d_u}$, where d_u is the dimensionality of user features. User's side information is particularly valuable for addressing user cold-start, enabling the system to make reasonable initial recommendations based on demographic similarity to existing users even before behavioral data accumulates [124].

Item-Side information

Item auxiliary data encompasses descriptive attributes and content features associated with items [74]:

- Structured metadata: Categorical attributes (genre, brand, category), numerical properties (price, duration, release year), and relational tags (director, author, manufacturer) provide explicit item characterization.

- Textual content: Titles, descriptions, reviews, and documentation capture semantic information through natural language [103]. Pre trained language models can extract rich representations from such text.
- Visual content: Product images, movie posters, and video thumbnails provide perceptual features that influence user decisions [31]. Convolutional neural networks enable automatic visual feature extraction.
- Multimedia signals: For audio and video items, acoustic features and video frame analysis offer additional modalities for content understanding.

Item auxiliary data is represented as $\mathbf{X}^{(i)} \in \mathbb{R}^{N \times d_i}$, where d_i is the item feature dimensionality. Item side information directly addresses item cold-start by enabling content based similarity computation for items lacking interaction history [83, 85]. It also supports explainable recommendations by providing interpretable features that justify suggestions.

Contextual information

Contextual auxiliary data captures situational factors that influence user preferences at the time of interaction:

- Temporal context: Time of day, day of week, season, and proximity to holidays or events affect consumption patterns (e.g., movie preferences differ between weekday evenings and weekend afternoons).
- Spatial context: User location, venue type, and geographic preferences influence recommendations, particularly for location based services.
- Device context: Access medium, screen size, and interface constraints shape content consumption (e.g., mobile users may prefer shorter content).
- Session context: Within session behavioral sequences and navigation patterns reveal short term intent that may differ from long term preference.

Context aware recommendation leverages these signals to adapt suggestions to the user's current situation, recognizing that preferences are not static but vary based on circumstances.

1.2.3 The Fusion of Canonical and Auxiliary Data

The central argument of this dissertation is that robust and adaptive recommendation requires the deep integration of canonical and auxiliary data, as each addresses limitations inherent in the other [89, 124].

Canonical data limitations addressed by auxiliary data:

- Sparsity: Auxiliary features enable generalization beyond observed interactions by providing dense representations even when interactions are sparse [103].
- cold-start: Side information bootstraps preferences for new users and items that lack interaction history [83, 85].
- Noise: Auxiliary signals provide regularization against noisy behavioral data by grounding predictions in stable content features.
- Interpretability: Item and user attributes support explainable recommendations with human understandable justifications.

Auxiliary data limitations addressed by canonical data:

- Preference ambiguity: Interactions ground abstract features in actual user choices, revealing which content characteristics translate to preference [51].
- Feature relevance: Behavioral patterns reveal which attributes matter for preferences, enabling feature selection and attention mechanisms [32].
- Personalization: Side information alone cannot capture individual taste variations; interactions provide user specific preference signals [76].
- Dynamic preferences: Interactions track evolving preferences over time, while static attributes cannot reflect temporal dynamics.

Deep integration framework:

The deep learning approaches developed in this dissertation are specifically designed to model the combination of canonical and auxiliary data.

- Graph neural networks: Propagate information across both interaction edges and auxiliary relationships, learning unified representations that capture collaborative and content signals [107, 112].
- Contrastive learning: Align representations learned from different data views, ensuring consistency across canonical and auxiliary perspectives [111, 120].
- Attention mechanisms: Learn to weight canonical and auxiliary signals adaptively based on their relevance to specific user item pairs [32, 114].

- Hybrid architectures: Combine structured preference modeling from canonical data with semantic understanding from auxiliary textual information through retrieval-augmented generation [43].

Table 1.3: Comparison of Canonical and Auxiliary Data along their defining dimensions.

Aspect	**Canonical data**	**Auxiliary data**
Definition	Primary user-item interaction signals that directly capture preferences and behavior	Supplementary sources giving context about users, items, or relationships beyond direct interactions
Relation to preference	Direct behavioral encodes preference itself	Indirect / descriptive does not encode preference, only context
Sub-types	Explicit feedback (ratings, reviews); implicit feedback (clicks, views, purchases)	User-side (demographics, social); item-side (metadata, text, visual); contextual (time, location, device, session); conversational (intents)
Availability	Requires accumulated interaction history	Often available before any interaction
Primary strength	Personalization, individual taste, temporal dynamics	Generalization, cold-start mitigation, interpret-ability

The resulting models achieve robustness through redundant information sources when interaction data is sparse, auxiliary features maintain prediction quality and adaptability through flexible integration mechanisms that adjust the contribution of each data source based on availability and relevance.

1.3 Traditional Recommendation Approaches

Traditional recommendation approaches form the foundational methodologies upon which modern deep learning based systems are built. This section reviews three fundamental paradigms: collaborative filtering, content based filtering, and hybrid methods.

1.3.1 Collaborative Filtering

Collaborative filtering (CF) is the most widely adopted recommendation paradigm, operating on the principle that users who agreed in the past will agree in the future [84, 92]. Unlike content-based approaches, CF leverages collective user behavior to infer preferences, making it applicable even when item content is unavailable [51].

Neighborhood-Based Methods

Neighborhood based CF directly exploits the user item interaction matrix to find

similar users or items [34, 82].

User based CF identifies users with similar rating patterns and predicts ratings based on how these neighbors rated target items. Similarity is typically computed using Pearson correlation or cosine similarity. This approach was pioneered by GroupLens [77] but faces scalability challenges with large user populations.

Item based CF shifts focus to item similarities, which are more stable over time [63, 82]. This approach gained notability at Amazon and enables efficient real time recommendations through precomputed item similarity matrices.

Despite their interpretability, neighborhood methods suffer from: data sparsity computing meaningful similarities requires sufficient rated items, cold-start new entities lack interaction history, and limited expressiveness they cannot capture complex non-linear patterns [92].

Model-Based Methods

Model based CF learns compact representations from interaction data rather than storing the entire matrix [51].

- Matrix Factorization (MF) decomposes the sparse rating matrix $\mathbf{R} \in \mathbb{R}^{M \times N}$ into user and item latent factor matrices: $\mathbf{R} \approx \mathbf{P}\mathbf{Q}^\top$, where $\mathbf{P} \in \mathbb{R}^{M \times d}$ and $\mathbf{Q} \in \mathbb{R}^{N \times d}$ [51]. The predicted rating is computed as $\hat{r}_{ui} = \mathbf{p}_u^\top \mathbf{q}_i$. MF achieved breakthrough performance in the Netflix Prize and has been extended with bias terms, temporal dynamics, and implicit feedback handling [38, 76].

- Factorization Machines (FM) generalize MF to incorporate arbitrary feature interactions, enabling learning from auxiliary information alongside interactions [75]. This bridges pure CF and content aware recommendation.

Despite effectiveness, model based methods have limitations: linear inner products may miss complex relationships, latent factors lack interpret ability, and cold-start remains challenging [124].

1.3.2 Content-Based Filtering

Content based filtering (CBF) recommends items similar to those a user has preferred, based on item features rather than collective behavior [74].

Items are represented using manually engineered features. For textual content, the Vector Space Model with TF-IDF weighting is common [81], along with topic models. For structured metadata, categorical attributes are encoded as one-hot encoder vectors.

User profiles are built by aggregating features of preferred items or training clas-

sifiers (naive Bayes, SVM) on rated items [74]. Overly narrow profiles can lead to filter bubbles with reduced serendipity.

1.3.3 Hybrid Methods

Hybrid recommender systems merge two or more techniques to enhance overall recommendation performance, typically addressing the cold-start problem. For instance, collaborative filtering based methods struggle with new items, while content based approaches face challenges with new users. Various types of hybrid recommendations exist, such as weighted, switching, mixed, and feature augmented recommender systems [2].

Table 1.4 assesses each approach along four dimensions. Cold-Start measures the ability to generate useful recommendations for new users or new items that lack interaction history; an approach rates well if it can leverage content or auxiliary signals to bootstrap recommendations without observed behaviour. Sparsity reflects robustness when the user item matrix is highly sparse (often below 1% density in real systems); a method rates well if it generalises from few observations rather than collapsing into noise. Explainability captures whether the system can provide a human-understandable rationale for its outputfor instance, citing similar users or matching item attributesrather than producing scores from an opaque computation. Scalability concerns computational feasibility as the numbers of users M and items N grow to millions; it rates well if time and space complexity permit real-time serving without prohibitive infrastructure.

Table 1.4: Comparison of Traditional Recommendation Approaches

Approach	**Cold-Start**	**Sparsity**	**Explainability**	**Scalability**
User-Based CF	Poor	Poor	Good	Poor
Item-Based CF	Poor	Moderate	Good	Moderate
Matrix Factorisation	Poor	Good	Poor	Good
Content-Based	Good (items)	Good	Good	Good
Hybrid Methods	Good	Good	Moderate	Moderate

User-Based CF: This approach identifies users with similar interaction histories and aggregates their ratings to generate recommendations. Cold-start performance is poor because a new user has no history from which similarity can be computed. Sparsity is equally problematic: when the matrix is highly sparse, the co-rated item overlap between users shrinks to the point where cosine or Pearson similarities become noise-dominated and neighbourhood search finds few reliable neighbours [82]. Explainability,

by contrast, is a natural strengththe system can directly state that users with similar tastes also liked a given item, which is intuitive and auditable. Scalability is poor because computing pairwise similarities requires $O(M^2)$ time and space; at millions of users this is computationally intractable without heavy approximation [82].

Item-Based CF: Item similarities can be pre-computed offline, making inference faster than in the user-based case and yielding moderate scalability. Sparsity tolerance is moderate rather than poor because popular items accumulate many ratings over time, producing more stable similarity estimates; however, long-tail items the majority in most catalogues still suffer from insufficient co-ratings. Cold-start remains poor for the same structural reason as user-based CF: a new item has no ratings and hence no computable similarity, and a new user has no history to look up. Explainability is good because a recommendation can be attributed to items the user previously interacted with.

Matrix Factorisation (MF): MF projects users and items into a shared low-dimensional latent space by decomposing the observed rating matrix $R \approx PQ^\top$. This design makes it robust to sparsityregularisation prevents overfitting to the few observed entries, and pairwise ranking variants such as BPR-MF [76] generalise effectively from implicit feedback. Inference reduces to a dot product $O(d)$, making MF highly scalable and the dominant industrial approach for over a decade [51]. Cold-start is poor because new users and items have no learned vectors; folding-in approximations exist but are disruptive in production. Explainability is poor because the latent factors carry no semantic meaning and the predicted score $\hat{r}_{ui} = \mathbf{p}_u^\top \mathbf{q}_i$ cannot be mapped to a human-interpretable rationale without additional post-hoc methods [51].

Content-Based Filtering (CBF): Because CBF derives recommendations from item feature vectors (e.g., genre tags, TF-IDF representations) and user profiles built from those features, it handles item cold-start wella new item is immediately representable through its content and can be matched to existing user profiles [74]. The rating notation "Good (items)" reflects that user cold-start remains unresolved: without any interaction, no user profile can be constructed. Sparsity tolerance is good because recommendations depend on feature similarity rather than co-occurrence counts. Explainability is good since recommendations are grounded in explicit attributes. Scalability is good because scoring at inference time is a vector operation over the feature dimension, independent of the total number of users or items.

Hybrid Methods: Hybrid approaches combine collaborative and content-based signals, allowing a graceful fallback to content when interaction data is absent and a transition to stronger collaborative signals as data accumulates [9]. This makes them the strongest traditional option for both cold-start and sparsity. Explainability is rated mod-

erate because the blending mechanismwhether a learned weight, a switching rule, or a stacking modelobscures which component drove a particular recommendation, reducing the transparency achievable by either pure method alone. Scalability is moderate because hybrids must maintain and update two separate model components, increasing storage and inference overhead compared to a single MF model [1].

The cross-row pattern in Table 1.4 reveals a fundamental tension: no traditional approach simultaneously resolves cold-start, sparsity, explainability, and scalability. CF methods handle sparsity and scalability (MF) or explainability (neighbourhood-based) but fail at cold-start. Content-based methods handle cold-start and explainability but cannot exploit collective behavioural signals. Hybrid methods alleviate cold-start and sparsity at the cost of reduced explainability and higher operational complexity. This multi-dimensional gap motivates the deep learning approaches developed in subsequent chapters, which jointly learn user and item representations from both interaction signals and auxiliary data, enabling end-to-end optimisation that is not achievable through rule-based combination of traditional components.

1.4 Deep Learning for Recommendation

The success of deep learning in computer vision, natural language processing, and speech recognition has caused a major change in recommender systems research [124]. Deep neural networks offer unprecedented capabilities for learning complex, non linear patterns from massive volumes of diverse data, addressing fundamental limitations of traditional recommendation approaches. This section provides an overview of deep learning techniques for recommendation, establishing the foundation for the advanced architectures developed in subsequent chapters.

1.4.1 Why Deep Learning for Recommendation?

Traditional recommendation methods face inherent limitations that constrain their effectiveness in modern large scale applications [124]. Collaborative filtering relies on linear latent factor models that may fail to capture complex user item interactions. Content based filtering requires extensive manual feature engineering. Deep learning addresses these limitations through several key advantages.

Non-linear Transformation and Representation Learning

The most fundamental advantage of deep learning is its ability to model highly non-linear relationships through hierarchical feature transformations [17]. Traditional matrix factorization captures user item interactions through linear inner products, implicitly assuming that preference can be decomposed into independent latent factors combined

additively. In reality, user preferences often exhibit complex, non-linear dependencies that linear models struggle to represent.

Deep neural networks overcome this limitation through compositions of non-linear activation functions across multiple layers [18]. Each layer transforms its input representation, progressively extracting higher-level abstractions. For recommendation, this enables learning intricate interaction patterns: which combinations of user characteristics and item features indicate strong preference, how contextual factors modulate baseline preferences, and what temporal patterns signal preference evolution. The universal approximation theorem guarantees that sufficiently deep networks can approximate arbitrarily complex functions.

Automatic Feature Learning

Traditional recommender systems require substantial feature engineering effort [124]. Content-based methods depend on carefully designed item representations extracting meaningful features from text, images, or structured metadata demands domain expertise and ongoing maintenance.

Deep learning fundamentally changes this paradigm by learning features directly from raw data [124]. Convolutional networks automatically extract visual features from product images. Recurrent networks learn sequential patterns from user behavior streams. Attention mechanisms identify which input elements are most relevant for prediction. This end-to-end learning approach reduces manual effort, enables discovery of unexpected predictive signals, and allows models to adapt as data distributions shift.

Heterogeneous Data Integration

Modern recommendation involves diverse data modalities: interaction logs, item metadata, textual descriptions, visual content, social connections, and temporal context [89]. Traditional methods process each modality separately through ad-hoc fusion strategies that cannot capture cross-modal interactions.

Deep learning provides natural mechanisms for multi modal integration through shared representation spaces. Different encoder networks process each modality, projecting heterogeneous inputs into compatible embedding spaces where they can be combined and jointly reasoned over. Attention mechanisms learn which modalities are most informative for specific predictions [100].

Scalability and Efficiency

Modern neural architectures offer significant efficiency advantages for large scale recommendation [17]. Once trained, neural networks perform inference through matrix operations highly optimized on GPUs and TPUs. Deep learning enables efficient two-

stage architectures: a lightweight retrieval model generates candidates from millions of items, followed by a sophisticated ranking model for the reduced set [17, 128]. This retrieve then rank paradigm achieves sub-millisecond latency while maintaining quality.

Flexibility and Modularity

Deep learning architectures are inherently modular components can be combined, replaced, and extended without redesigning the entire system [124]. Pretrained components can be finetuned for specific domains. The deep learning ecosystem provides mature frameworks, pretrained models, and active research communities, establishing deep learning as the dominant paradigm for industrial recommendation.

1.4.2 Neural Architectures for Recommendation

Various neural network architectures have been adapted for recommendation, each offering distinct inductive biases suited to different data characteristics [124].

Multilayer Perceptrons for Interaction Modeling

The multilayer perceptron (MLP) is the foundational architecture, consisting of fully connected layers with non linear activations [125]. For recommendation, MLPs model user item interactions by processing combined user and item representations through hidden layers.

Neural collaborative filtering (NCF) [32] replaces the linear inner product of matrix factorization with a learned non-linear interaction function. The architecture concatenates user and item embeddings, then passes them through hidden layers to predict interaction probability. Neural Matrix Factorization (NeuMF) combines MLP based interaction with traditional matrix factorization through a unified architecture.

Wide & deep learning [16], deployed at Google Play, combines a "wide" linear model capturing memorization with a "deep" MLP capturing generalization. DeepFM [27] refines this by replacing the wide component with factorization machines, enabling automatic feature interaction learning. Deep & cross network [106] introduces cross layers explicitly modeling feature interactions, while AutoInt [91] applies self-attention for adaptive feature interactions.

Convolutional Neural Networks for Content Understanding

Convolutional neural networks (CNNs) excel at processing grid-structured data through local receptive fields and parameter sharing.

Visual Recommendation: VBPR incorporates CNN to extract image features into BPR, enabling recommendations that account for visual aesthetics. Fashion recommendation particularly benefits from visual features capturing style, color patterns, and de-

sign aesthetics. Modern approaches leverage pretrained image models (e.g., ResNet, VGG) as feature extractors.

Textual Content: CNNs capture local n grams patterns through one dimensional convolutions over word embeddings, efficiently extracting key phrases from reviews and descriptions. This capability has been applied to review based recommendation, where sentiment understanding enhances preference prediction [130].

Sequential Patterns: Caser [96] applies horizontal and vertical convolutions to user interaction sequences, capturing both point level and union level sequential patterns.

Recurrent Neural Networks for Sequential Modeling

Recurrent neural networks (RNNs) and variants LSTM and GRU are designed for sequential data, maintaining hidden states capturing temporal dependencies.

Session-Based Recommendation: GRU4Rec [35] pioneered applying GRUs to model session click sequences, predicting the next item from evolving session state. Extensions incorporate attention mechanisms to weight historical items [54].

Personalized Sequential Recommendation: Recurrent recommender networks (RRN) [110] use LSTMs to capture both user preference dynamics and item popularity evolution. The hidden state encodes current preference profile, enabling modeling of preference drift alongside short term contextual effects.

While transformers have largely replaced RNNs for many tasks, RNN architectures remain relevant for streaming scenarios requiring computational efficiency and constant memory [124].

Autoencoders for Representation Learning

Autoencoders learn compressed representations by reconstructing inputs through a bottleneck layer.

AutoRec [87] applies autoencoders to collaborative filtering, taking partial rating vectors as input and reconstructing complete vectors. The bottleneck forces learning compressed representations capturing essential preference patterns.

Variational Autoencoders: Mult-VAE [59] extends with probabilistic modeling, learning distributions over latent representations providing regularization and uncertainty quantification.

Denoising Autoencoders: CDAE [113] corrupts inputs by masking interactions, training reconstruction of clean inputs. This forces robust representations generalizing beyond observed patterns.

Attention Mechanisms and Transformers

Attention mechanisms enable dynamic weighting of input elements based on prediction relevance [100].

SASRec [45] applies transformer self-attention to sequential recommendation, capturing dependencies between any sequence positions. Attention weights provide interpretable explanations of which past items influenced recommendations.

BERT4Rec [93] adapts BERT pre training using masked item prediction, learning bidirectional sequence representations enabling transfer learning across tasks.

Attentional Factorization Machines (AFM) [114] learns feature importance weights, improving upon uniform feature treatment.

Multi head attention captures diverse interaction patterns different heads can focus on recency, similarity, complementarity, or other relevance factors simultaneously [100].

1.4.3 Graph Neural Networks for Recommendation

Graph neural networks (GNNs) have emerged as a powerful paradigm for recommendation by modeling the inherent graph structure of user item interactions and auxiliary relationships [23, 112]. The user item interaction matrix naturally forms a bipartite graph where edges represent observed interactions. GNNs provide a principled framework for learning representations that capture both local interaction patterns and global structural properties through iterative message passing, where nodes aggregate information from neighbors to update representations [28, 46]. This enables high order collaborative signal propagation a user's representation incorporates not only directly interacted items but also items preferred by similar users, capturing collaborative filtering intuition within a neural architecture. Representative methods include LightGCN [33], which simplifies graph convolution by removing non-linear transformations, and NGCF [108], which explicitly models user item interaction during message passing. GNN based methods have demonstrated state of the art performance across diverse benchmarks.

1.4.4 Large Language Models for Recommendation

Large language models (LLMs) represent the frontier of deep learning for recommendation, bringing unprecedented capabilities in semantic understanding, reasoning, and natural language interaction [37]. Pretrained on large text datasets, LLMs encode rich world knowledge and can understand user preferences expressed in natural language. For recommendation, LLMs enable several transformative capabilities: semantic understanding of item descriptions and user reviews beyond keyword matching, natural language interfaces where users express preferences conversationally, explainable

recommendations with human readable justifications, and zero shot or few shot recommendation for new domains without task specific training [24]. Retrieval-Augmented Generation (RAG) addresses LLM limitations by grounding generation in retrieved factual information, combining knowledge retrieval with generative capabilities [21]. This hybrid approach is particularly valuable for recommendation, where accurate item information must be integrated with personalized preference modeling. LLM based conversational recommendation represents a paradigm shift from static ranked lists toward interactive, dialogue based preference elicitation [43]. The integration of LLMs with structured recommendation models combining semantic reasoning with collaborative signal modeling from graph neural networks is explored in Chapter 5, where a hybrid conversational recommendation framework is proposed.

The approaches surveyed above can be organized into six main research directions. Table 1.5 synthesizes them by their core idea, key limitation, and the chapter in which this dissertation addresses that limitation; detailed comparisons within each direction are provided in the corresponding chapters.

Table 1.5: Summary of Related Work by main Research Direction.

Direction	**Core idea (representative methods)**	**Key limitation**	**Addressed in**
CF & matrix factorization	Latent-factor, neighbourhood modeling of interactions (MF, BPR-MF, FCM-Rec, SCoC)	Sparsity, cold-start, no use of side information	Ch. 2
GNNs for recommendation	High-order message passing over the user–item graph (NGCF, LightGCN, GraphSAGE, GAT, LINKX)	ID-embedding memory cost, over-smoothing	Ch. 2
Contrastive & self-supervised	Self-supervised augmentation for robust representations (SGL, SSL4Rec, MixGCF, LightGCL)	Random augmentation distorts structure; noise sensitivity	Ch. 3
Cross-domain & continual learning	Transferring knowledge across domains / adapting over time (CDR surveys, CTNet, KEEP, DIIT, ECAT, MAML, DMAN)	Unidirectional transfer, catastrophic forgetting, no fairness	Ch. 4
LLM-based & conversational (RAG)	Natural-language interaction and generative recommendation (PALR, conversational LLM systems, RAG)	Hallucination, latency, no grounding in long-term behaviour	Ch. 5

1.5 Evaluation Metrics

Evaluation metrics play a critical role in assessing the effectiveness and practical capability of recommender systems. This section summarizes the metrics used throughout this dissertation, categorized into offline accuracy metrics and online performance metrics.

1.5.1 Offline Metrics: Accuracy and Quality

Offline metrics evaluate recommendation quality on test data, measuring how well a model predicts user preferences [34].

Hit Rate (HR@K) measures the proportion of users who interact with at least one recommended item in the top-K list. Precision@K quantifies the fraction of relevant items among the K recommendations. Recall@K measures the fraction of relevant items successfully retrieved in the top-K list. F1@K harmonically combines Precision and Recall into a single metric.

NDCG@K (Normalized Discounted Cumulative Gain) accounts for ranking position, assigning higher scores when relevant items appear earlier in the list. MAP@K (Mean Average Precision) calculates the average precision across all relevant items. AUC (Area Under the ROC Curve) measures the probability that the model ranks a positive item higher than a negative one.

Following standard practice in deep learning based recommendation research [33, 108], this dissertation primarily employs Recall@K and NDCG@K.

1.5.2 Online Metrics: Efficiency and Performance

Online evaluation measures how recommendations influence actual user behavior in real-world deployment.

A/B testing is a widely adopted online evaluation methodology in which the user population is randomly divided into two or more groups: a control group that is served by the existing (baseline) recommendation model, and one or more treatment groups that are served by the new (candidate) model under evaluation. By ensuring that the groups are statistically comparable in terms of user demographics and behavioral characteristics typically achieved through stratified random sampling any observed difference in outcome metrics can be causally attributed to the model change rather than to confounding factors. The experiment runs for a pre-determined period during which key performance indicators such as click-through rate (CTR), average content consumption per user (ACPU), average viewing duration per user (ADPU), conversion rate, and user

retention are continuously tracked and compared across groups. A/B testing provides the strongest form of causal evidence for evaluating recommendation models because it directly measures the impact of algorithmic changes on real user behavior under naturalistic conditions, thereby complementing offline accuracy metrics that cannot fully capture user satisfaction, engagement dynamics, or long-term retention effects.

This dissertation validates proposed methods through A/B testing on the TV360 [1] entertainment service which is from Viettel - the one of largest Corporation in Vietnam, measuring metrics such as average content viewing time and user interaction rates. Online evaluation complements offline metrics by capturing long term objectives (engagement, retention) and real-world trade offs among accuracy, diversity, and novelty that offline metrics cannot fully represent.

1.6 Datasets

Several large scale benchmark datasets are widely used for evaluating recommender systems across different application scenarios [2].

In this dissertation, the MovieLens datasets are used repeatedly as core benchmarks to validate the effectiveness, robustness, and generalization ability of the proposed recommendation frameworks. In particular, three representative versions are employed, namely MovieLens-100K (ML100K), MovieLens-1M (ML1M), and MovieLens-20M (ML20M). Information of MovieLens [3] datasets are described in Table 1.6 and Table 1.7 as below.

Table 1.6: Statistics of MovieLens datasets

Dataset	**Users**	**Movies**	**Ratings**
MovieLens-100K (ML100K)	943	1682	100,000
MovieLens-1M (ML1M)	6,040	3,900	1,000,209
MovieLens-20M (ML20M)	138,493	27,278	20,000,263

[1] Service overview available at: https://viettelecom.vn/vx/internet-truyenhinh/truyen-hinh

[2] https://github.com/caserec/Datasets-for-Recommender-Systems (Datasets-for-Recommender-Systems GitHub repo)

[3] https://grouplens.org/datasets/movielens/

Table 1.7: Structure of the MovieLens dataset

File	Field	Description
ratings.dat	UserID	Unique identifier of each user
	MovieID	Unique identifier of each movie
	Rating	Integer score on a 5-star scale
	Timestamp	Time when the rating was recorded,
users.dat	UserID	Unique identifier of each user
	Gender	User gender
	Age	User age group
	Occupation	Profession category
	Zip-code	Residential zip code provided voluntarily by users
movies.dat	MovieID	Unique identifier of each movie
	Title	Official movie title
	Genres	One or multiple genres with 18 genre categories

This dissertation employs six distinct datasets across its four experimental chapters: three versions of MovieLens (ML-100K, ML-1M, ML-20M), Yelp, Amazon, and TV360. This section explains the rationale for each choice and argues why together they constitute a representative experimental suite for the research problems addressed.

Table 1.8 maps each dataset to the core challenges it is selected to represent and the chapter in which it is used.

Table 1.8: Mapping of datasets to research challenges and chapters

Dataset	Primary Challenge Represented	Key Property	Chapter
MovieLens-1M	Scalability, cold-start, user diversity	Well-studied benchmark	2, 3, 4, 5
MovieLens-20M	Scalability stress test	0.53% density, $\times$20 scale	2
Yelp	Multi-domain (business categories)	Cross-domain heterogeneity	4
Amazon	Multi-domain (product categories)	Commercial item diversity	4
TV360	Industrial deployment, cold-start	Real-world OTT platform	2, 5

MovieLens Family (ML-100K, ML-1M, ML-20M)

Why MovieLens: MovieLens is the most widely used public benchmark in recommender systems research [29]. It provides explicit numerical ratings on a 5-star scale

alongside rich metadata user demographics (age, gender, occupation) and item attributes (genre, release year) that directly support experiments requiring both canonical interaction data and auxiliary side information. Because it has been used as the primary benchmark in hundreds of published works, results on MovieLens are directly comparable to prior literature, ensuring that performance claims in this dissertation can be contextualised within the broader field.

The three versions differ by roughly one order of magnitude in scale: ML-100K (943 users, 1,682 items, 100K ratings), ML-1M (6,040 users, 3,900 items, 1M ratings), and ML-20M (138,493 users, 27,278 items, 20M ratings). This progression is deliberate: ML-100K provides a tractable small-scale setting for ablation studies and hyperparameter sensitivity analysis; ML-1M is the standard at which most competing methods report results, enabling fair comparison; ML-20M is used specifically as a scalability stress test in Chapter 2.

Sparsity representativeness: The density of ML-1M (4.47%) and ML-20M (0.53%) brackets the range of sparsity commonly observed in real-world collaborative filtering benchmarks (see Table 1.2 of the dissertation). This range ensures that conclusions about sparsity robustness are not artefacts of a single operating point.

User diversity for cold-start evaluation: ML-1M contains sufficient users with sparse interaction histories to support the three-tier user stratification (cold: $\leq$20 interactions; warm: 21-50; active: $>$50) used in Chapters 2 and 3. This stratification is the primary mechanism through which the cold-start advantage of the proposed methods is measured, making ML-1M the natural vehicle for this evaluation.

Multi-domain partitioning. In Chapter 4, ML-1M is partitioned into five genre-based domains (Action, Comedy, Drama, Thriller, Sci-Fi). This is possible precisely because MovieLens records genre metadata for every item. The domain boundaries are semantically meaningful and well separated in terms of user preference patterns, making ML-1M an appropriate testbed for continual multi-domain recommendation even though it was originally designed as a single-domain dataset.

Yelp

Why Yelp: Yelp is a standard benchmark for multi-domain recommendation because its business taxonomy provides natural, non-overlapping domain labels (e.g., Restaurants, Shopping, Food, Beauty, Health). Unlike genre tags in MovieLens, which are assigned per item and may overlap, Yelp business categories represent genuinely heterogeneous domains with distinct item and user populations. This heterogeneity is important for Chapter 4: it tests whether the continual learning framework (CNL4Rec) can pre-

serve knowledge across domains whose statistical distributions differ substantially, not just across sub-genres of the same content type.

Scale and density: With 5,000 users, 3,000 items, and interactions distributed unevenly across five domains, Yelp exhibits moderate but uneven sparsity the distribution of interactions across domains is non-uniform, with some domains receiving far more reviews than others. This imbalance directly mirrors the domain-frequency mismatch encountered in real multi-domain platforms and is a realistic stress test for the domain fairness objective of CNL4Rec.

Amazon Product Reviews

Why Amazon: The Amazon dataset spans five product categories (Electronics, Books, Movies, Home, Sports) and is widely adopted for cross-domain recommendation research. It complements Yelp in a critical dimension: while Yelp domains share a geographical/social context (local business reviews), Amazon domains represent fundamentally different product ontologies. Electronics and Books, for instance, attract users with very different purchase motivations and browsing behaviours. Including Amazon alongside Yelp ensures that the multi-domain conclusions in Chapter 4 hold across both social-review and e-commerce interaction patterns, not just one type of platform.

Commercial relevance: E-commerce recommendation is one of the most economically significant application areas in the field. Validating CNL4Rec on Amazon directly establishes the practical applicability of the continual learning approach to commercial platforms undergoing frequent catalogue expansion across product lines.

TV360

Why TV360. TV360 is a large-scale OTT (over-the-top) streaming platform operated by Viettel, directly developed and maintained by the dissertation author in an industrial setting. It serves millions of users with live TV, VOD content, and movie/series recommendations, generating high-volume interaction logs under real operational constraints. Its inclusion in the experimental suite is motivated by three considerations.

First, it provides industrial validity: public benchmarks such as MovieLens are collected under controlled conditions with explicit ratings, whereas TV360 logs reflect implicit feedback (views, completion rates) under production constraints including cold-start users, rapidly changing content catalogues, and strict latency requirements. Demonstrating that the proposed methods generalise to this setting is evidence that the academic contributions are deployable in practice. Second, TV360 exhibits extreme sparsity and cold-start severity: the video subdataset has a density of approximately 0.002% com-

parable to Amazon-Books (Table 1.2) while the film subdataset has 0.039% density. These figures are substantially lower than the MovieLens benchmarks, confirming that the sparsity and cold-start findings are not confined to well-curated academic datasets.

Third, it enables online A/B validation: the industrial deployment allows direct measurement of user engagement metrics (Average Content Per User, Average Duration Per User) that offline benchmarks cannot provide.

1.7 Chapter Summary

This chapter established the theoretical foundations for deep learning based recommender systems. The recommendation problem was formalized as learning a utility function mapping user item pairs to preference scores, with three key challenges identified: data sparsity, cold-start, and scalability. The canonical and auxiliary data paradigm was introduced as the conceptual backbone canonical data (user item interactions) provides behavioral signals while auxiliary data (demographics, attributes, context, knowledge graphs) offers semantic richness; their deep integration is essential for robust recommendation. Traditional approaches including collaborative filtering, content based filtering, and hybrid methods were reviewed, followed by a survey of deep learning architectures (MLP, CNN, RNN, autoencoders, attention/transformers, graph neural networks, and large language models). Finally, evaluation metrics were summarized, covering offline metrics (Recall@K, NDCG@K) for accuracy assessment and online metrics (A/B testing) for real-world performance validation.

Chapter 2

Robust Recommendation via Interaction Embedding and Soft Clustering

2.1 Introduction

Modern recommender systems have become essential components of web scale applications, supporting millions of users across ecommerce platforms, streaming services, and social networks. Despite significant advances in deep learning based recommendation approaches, several fundamental challenges persist that limit the practical deployment and scalability of these systems. This section identifies the key scalability challenges in modern recommender systems and uses them as the primary motivation for the subsequent analysis in this chapter.

2.1.1 The Scalability Challenge in Recommendation

Modern recommender systems rely on learning unique embedding vectors for each user and item. While effective in academic benchmarks, this approach faces critical limitations in industrial deployment.

Memory Constraints: The memory footprint grows linearly with the number of entities. Chen et al. [14] calculated that serving 1 billion users with 64-dimensional embeddings requires approximately 238 GB solely for user embeddings. Combined with item embeddings and categorical features, total requirements can exceed commodity hardware capacity, forcing trade-offs between model expressiveness and computational resources.

Cold-start Problem: Users and items with insufficient interaction history struggle

to learn generalizable embeddings. New users cannot be effectively represented, leading to poor initial recommendations. New items lack feedback signals for embedding optimization, creating visibility problems where relevant content remains hidden from potential users.

Computational Overhead: The exhaustive item scoring paradigm creates misalignment between offline training and online serving. Production systems must respond within strict latency constraints (tens of milliseconds), yet most research optimizes for full matrix reconstruction objectives that are computationally infeasible in real-time scenarios.

2.1.2 Related Methodologies

Clustering-based Approaches: Clustering techniques provide natural solutions for scalability and cold-start by grouping similar users. Jiang et al. [44] presented user coresets via clustering to accelerate large-scale top-k systems.

Deep learning integration with clustering has opened new possibilities. Xie et al. [115] pioneered deep embedded clustering for joint learning of representations and cluster assignments. Nalavade et al. [73] combined deep embedded clustering with matrix factorization. Soft clustering approaches address hard clustering limitations: Bezdek [8] introduced Fuzzy C-Means allowing multiple cluster memberships, while Mao et al. [71] proposed Soft K-indicators for collaborative filtering.

The most sophisticated approach is Clustered Embedding Learning (CEL) by Chen et al. [14], enabling automatic clustering through top-down divisive partitioning with theoretical guarantees on solution identifiability.

ID-Independent Representation Learning: Ananyeva et al. [4] proposed replacing learned user embeddings with aggregated representations from interaction sequences. Hash-based methods offer another direction: PreHash by Shi et al. [88] learns hash functions for large-scale user modeling, while HashGNN by Tan et al. [94] combines graph neural networks with hashing. However, these approaches suffer from embedding collisions and reduced flexibility for incremental learning.

2.1.3 Limitations of Existing Approaches

Despite the significant progress reviewed in the preceding sections, several critical gaps remain unaddressed in the existing literature on clustering based and scalable recommendation systems. This section identifies four key gaps that motivate the proposed EfficientRec framework.

Absence of end to end ID-Free Frameworks

Current approaches to ID-independent recommendation focus primarily on modular replacement of user embeddings within existing architectures. The plug and play paradigm exemplified by the work of Ananyeva et al. [4] offers flexibility and compatibility with established model architectures, but this modularity comes at a cost.

Hash based methods such as PreHash [88] similarly operate as replaceable components within larger recommendation systems. While they successfully reduce memory requirements, they introduce their own limitations. The binding of hash buckets to warm users during the learning process reduces flexibility for handling new users, and the hash function itself is typically fixed after training, limiting adaptability to evolving user preferences. The CEL framework [14], despite its sophistication, still maintains entity specific cluster assignments that must be stored and updated, representing a partial rather than complete departure from the traditional embedding paradigm.

What is needed is a unified framework that integrates ID-free representation with task specific optimization in a truly end to end manner.

Soft Clustering for Individual Recommendation

The application of soft clustering to recommendation systems has followed two largely separate paths, neither of which addresses the specific needs of scalable individual recommendation with cluster based inference.

The first path involves fuzzy collaborative filtering approaches that generate partition matrices allowing users to belong to multiple groups. The Soft K-indicators alternative projection method of Mao et al. [71] demonstrated by this approach, producing sparse partition matrices that capture the multifaceted nature of user preferences. However, these methods typically lack integration with neural network learning, relying instead on traditional optimization procedures that may not scale effectively to the high dimensional feature spaces encountered in modern recommendation systems. The partition matrices are learned separately from the recommendation model, missing opportunities for end to end optimization that could improve both the clustering and recommendation components.

The second path involves neural soft clustering for group recommendation, as exemplified by Adaptive Similarity Driven Deep Embedded Clustering [90] and DeepGroup [25]. These methods successfully integrate deep learning with soft clustering, learning user representations that support probabilistic cluster membership. However, their objective is fundamentally different from individual recommendation. Group recommendation seeks to satisfy multiple users simultaneously, requiring aggregation strategies that balance potentially conflicting preferences. The techniques developed for group recommendation do not directly transfer to the individual recommendation setting, where

the goal is to optimize recommendations for a single user while leveraging cluster structure for efficiency and cold-start handling.

Integration of Contrastive Learning with Clustering based Recommendation

Contrastive learning has emerged as one of the most successful paradigms for self supervised representation learning, demonstrating remarkable effectiveness across computer vision, natural language processing, and increasingly, recommendation systems. Recent surveys document its extensive application in recommendation through methods such as SGL [111], SimGCL [119], and NCL, which use contrastive objectives to learn user and item representations that are robust to noise and capture meaningful similarity structure.

Separately, contrastive objectives have proven valuable for deep clustering, where they encourage learned representations to form well separated clusters. The DeepCluster approach of Caron et al. [11] demonstrated that alternating between clustering and contrastive representation learning can discover meaningful semantic categories in an unsupervised manner.

Despite the success of contrastive learning in both recommendation and clustering individually, the integration of contrastive objectives with cluster based recommendation systems remains limited. Existing contrastive recommendation methods focus on enhancing user and item representations without explicitly considering cluster structure. Existing deep clustering methods focus on learning representations suitable for clustering without considering recommendation objectives. The potential combining between these approaches where contrastive learning could improve cluster quality while cluster structure could inform contrastive pair selection remain unexplored.

Comprehensive Framework Combining Multiple Innovations

The preceding gaps highlight the absence of a comprehensive framework that integrates all four critical components for scalable recommendation: neural soft clustering, contrastive learning, cluster based inference, and user-ID-free design.

Existing methods address these components in isolation or in partial combinations. Deep clustering methods such as Deep Embedded Clustering [115] and Adaptive Similarity Driven Deep Embedded Clustering [90] integrate neural learning with clustering but have not been applied to individual recommendation with cluster based inference. Contrastive recommendation methods such as SGL [111] and SimGCL [119] enhance representation learning but do not incorporate clustering structure. cluster based inference methods such as Clustered Embedding Learning [14] and eTREE [3] enable efficient recommendation but rely on hard clustering that cannot capture the nuanced, multifaceted nature of user preferences. ID-free approaches such as the work of Ananyeva et

al. [4] provide scalable user representation but function as plug and play modules rather than end to end optimized systems.

The absence of a framework integrating all four components leaves significant potential unexploited. Each component addresses a distinct aspect of the scalable recommendation challenge, and their integration could yield benefits unavailable to approaches addressing these aspects independently. Neural soft clustering captures the multifaceted nature of preferences while enabling cluster based efficiency. Contrastive learning ensures robust, discriminative representations. cluster based inference provides computational scalability. user-ID-free design eliminates the memory bottleneck of embedding tables. A framework combining all four components would represent a significant advance toward truly scalable recommendation systems suitable for web scale deployment.

2.2 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.

2.2.1 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.

Contributions of EfficientRec

The EfficientRec framework makes several novel contributions to the field of scalable recommendation systems, addressing each of the research gaps identified in the preceding analysis.

First, EfficientRec is a novel ID-free user representation learning framework based on interaction embedding and neural soft clustering. Existing recommendation models represent each user through a dedicated embedding vector, a paradigm that ties memory requirements directly to the size of the user population and prevents the model from generalizing to unseen users without retraining. This dissertation proposes an alternative representation in which user-IDentity is entirely discarded in favour of behavioural signals: a user is characterized by an aggregated interaction embedding that is subsequently projected onto a set of probabilistic soft-cluster prototypes. Unlike hard-clustering approaches which force each user into a single preference group and thereby lose the nuanced, multifaceted nature of real-world tastes the proposed soft-assignment scheme allows users to simultaneously belong to multiple clusters with learned weights, yielding richer and more flexible representations. The resulting framework is end-to-end trainable, scales sub-linearly with user population size, and supports cold-start users without any modification to the learned parameters.

Second, EfficientRec is a contrastive learning strategy tailored to cluster-structured recommendation, improving representation quality and model stability. While contrastive learning has been shown to benefit recommendation and deep clustering individually, the two lines of work have not been integrated: existing contrastive recommendation methods do not exploit cluster structure when constructing training pairs, and existing contrastive clustering methods are not designed with individual recommendation objectives in mind. This dissertation bridges the gap by proposing a contrastive objective that operates at two complementary levels interaction-level and cluster-level to simultaneously sharpen intra-cluster cohesion and inter-cluster separation. By coupling contrastive supervision with the soft-clustering representation of the first contribution, the model learns user embeddings that are both discriminative and robust to the noise and sparsity inherent in real-world interaction data, leading to more stable recommendation performance across varying levels of user activity.

Comparative Analysis of EfficientRec Against Related Methods

To make the novelty of EfficientRec concrete, Table 2.1 provides a structured comparison against the most closely related methods across five design dimensions: user-ID elimination, clustering strategy, contrastive supervision, cluster-based inference, and end-to-end optimization. These dimensions correspond directly to the four research gaps identified in Section 2.1.3.

The memory complexity reported in Table 2.1 refers to the user-side representa-

Table 2.1: Comparative Analysis of EfficientRec Against Related Methods

Characteristic	**Ananyeva [4]**	**PreHash [88]**	**CEL [14]**	**SCoC [56]**	**EfficientRec (Ours)**
user-ID elimination	✓ (plug-in)	✓ (hash)	✗	✗	✓ **(end-to-end)**
Soft clustering	✗	✗	✗	✓	✓
Contrastive learning	✗	✗	✗	✗	✓
Cluster-based inference	✗	✗	✓ (hard)	✓ (soft)	✓ **(soft)**
End-to-end optimization	✗	✗	Partial	✗	✓
Memory complexity	$\mathcal{O}(M \times d)$	$\mathcal{O}(B \times d)$	$\mathcal{O}(K \times d)$	$\mathcal{O}(K \times d)$	$\mathcal{O}(K \times d),\ K \ll M$

M: number of users; B: number of hash buckets; K: number of clusters; d: embedding dimension.

tion footprint, i.e., the number of parameters required to represent the entire user base. ID-based models learn and store a separate d-dimensional embedding for every user, yielding $\mathcal{O}(Md)$ memory complexity, which grows linearly with the number of users M.

EfficientRec, by contrast, is ID-free and stores no per-user parameters. The user base is represented by a fixed set of K soft-cluster prototypes, each represented as a vector in $\mathbb{R}^d$. Consequently, the stored user-side representation requires only $\mathcal{O}(Kd)$ parameters.

An individual user representation is computed on the fly as a probabilistic mixture of these K prototypes, derived from the items in the user's interaction history, and is therefore never stored. Since the number of clusters K is a fixed constant selected independently of the size of the user base and is much smaller than the number of users, i.e., $K \ll M$, the user-side memory complexity is $\mathcal{O}(Kd)$ and, crucially, is independent of M. This is the origin of EfficientRec's scalability advantage.

Item embeddings, whose memory complexity is $\mathcal{O}(Nd)$, are shared by all compared methods and are therefore not the distinguishing factor in this comparison.

The comparison reveals two fundamental distinctions. First, while Ananyeva [4] and PreHash [88] eliminate user-ID embeddings, they do so as modular replacements within existing architectures, meaning their user representations are not optimized jointly with the recommendation objective. EfficientRec, by contrast, constructs user representations through end-to-end optimization, allowing the Interaction Embedding model to discover behavioral features specifically suited for recommendation rather than general purpose encoding.

Second, whereas CEL [14] and SCoC [56] incorporate cluster-based inference, neither employs contrastive supervision during clustering. Without contrastive learn-

ing, cluster boundaries are not explicitly encouraged to be discriminative, reducing the quality of preference groupings. EfficientRec's contrastive objective simultaneously improves cluster cohesion (similar users are pulled together) and separation (dissimilar users are pushed apart), directly benefiting both recommendation quality and scalability.

2.2.2 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 [13, 30] and soft clustering techniques [8]. 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.

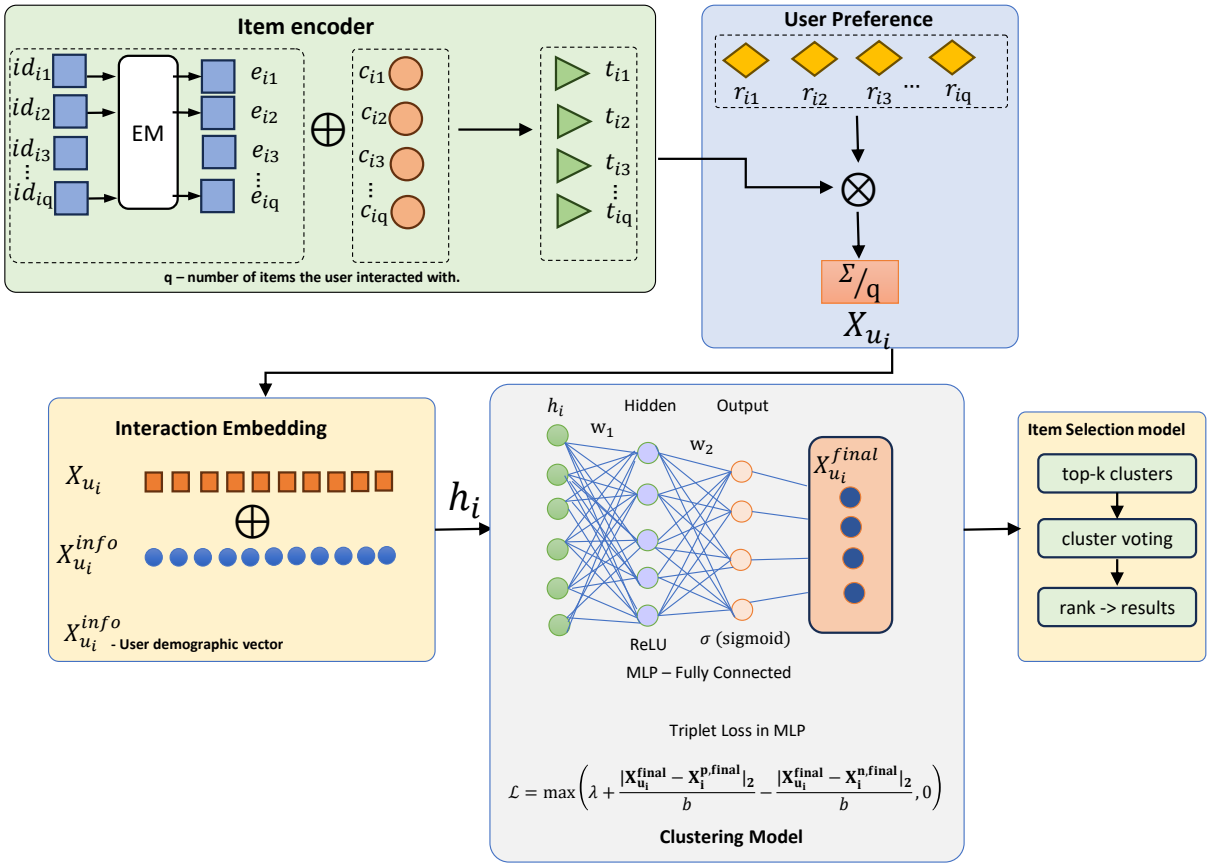

Figure 2.1: EfficientRec Overall Architecture

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.

Problem Formulation and Notation

Before describing the detailed architecture of each component, we first establish the mathematical notation and problem formulation that will be used throughout this section.

Consider a recommendation system operating over a set of items denoted as $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$, where N represents the total number of items available in the system. The item catalog may include products, movies, music tracks, articles, or any other type of content depending on the application domain. Each item in the catalog is associated

with a set of content features that describe its characteristics, such as genre, category, price, duration, or textual description.

The system serves a population of users denoted as $\mathcal{U} = \{u_1, u_2, \ldots, u_M\}$, where M represents the total number of users.

For each user u_i in the system, we observe an interaction history $S_i = \{v_{i,1}, v_{i,2}, \ldots, v_{i,q_i}\}$ consisting of the q items that the user has previously interacted with. The number of interactions q_i varies across users and is typically much smaller than the total number of items N, reflecting the inherent sparsity of user item interaction data. Associated with each interaction is a rating value, and the complete rating vector for user u_i is denoted as $R_i = \{r_{i1}, r_{i2}, \ldots, r_{iq}\}$, where r_{ij} represents the rating that user u_i assigned to item v_j.

In addition to interaction data, the system may have access to auxiliary information about users, such as age group, gender, geographic location, or device type. This auxiliary information is represented as a feature vector $\mathbf{X}_{u_i}^{info}$ for each user u_i. Auxiliary features can provide valuable signals for preference prediction, particularly for new users with limited interaction history.

The objective of EfficientRec is to learn a mapping function H that transforms the available user behavioral data into a compact preference vector that captures the user's interests across different preference dimensions:

$$\mathbf{X}_{u_i}^{final} = H(S_i, R_i, \mathbf{X}_{u_i}^{info}) \tag{2.1}$$

In this formulation, $\mathbf{X}_{u_i}^{final} \in \mathbb{R}^b$ is a b-dimensional final preference vector where each component $X_{u_i k}^{final}$ represents the user's degree of affinity toward preference cluster k. The preference vector serves as a compact summary of the user's interests that can be efficiently used for downstream recommendation tasks. The key insight of EfficientRec is that this preference vector is computed dynamically from the user's interactions rather than being retrieved from a stored embedding table, which enables the model to handle large user populations without increasing memory requirements.

Interaction Embedding Model

The Interaction embedding model is considered as the foundational component of the EfficientRec architecture. This component addresses the fundamental scalability limitation of traditional collaborative filtering approaches [50, 76], which require maintaining a separate embedding vector for each user in the system.

The key insight of the interaction embedding model is that a user's preferences can be inferred from the characteristics of the items they have interacted with and the ratings they have assigned to those items. Rather than storing a fixed embedding for each user,

the model dynamically computes a user representation by aggregating information from the user's interaction history. This approach offers several advantages: it eliminates the need for per user embedding storage, it naturally handles new users without requiring model updates, and it produces representations that are grounded in actual behavioral signals rather than abstract learned parameters.

In the interaction embedding model, the computation proceeds through five sequential steps, each of which serves a specific purpose in transforming raw interaction data into a meaningful user representation.

- Step 1: Rating Normalization

The first step in the interaction embedding computation is to normalize the raw rating values to a standardized scale. Different recommendation domains may use different rating scales, such as 1-5 stars for movie ratings, 1-10 scores for product reviews, or binary like/dislike signals for social media content. To enable consistent treatment across different rating systems and to distinguish between positive and negative preferences, all ratings are normalized to a bipolar scale ranging from -1 to $+1$.

The normalization transformation is defined as:

$$r_{ij} = \frac{2 \times (rating_{ij} - rating_{min})}{rating_{max} - rating_{min}} - 1 \tag{2.2}$$

This linear transformation maps the original rating range $[rating_{min}, rating_{max}]$ to the normalized range $[-1, +1]$. The choice of a bipolar scale centered at zero serves a critical purpose in the subsequent aggregation step. Positive normalized values indicate items that the user likes or prefers, with values closer to $+1$ indicating stronger positive preference. Negative normalized values indicate items that the user dislikes or rates poorly, with values closer to -1 indicating stronger negative preference. A normalized value of zero represents a neutral rating, indicating neither positive nor negative preference.

- Step 2: Item Representation

The second step constructs a comprehensive representation for each item in the user's interaction history. This representation combines two complementary sources of information: collaborative signals derived from the item's identity and content signals derived from the item's observable features. The combination of these two information sources enables the model to leverage both the hidden patterns discovered through collaborative filtering [84] and the explicit semantic information encoded in item metadata.

The collaborative component of the item representation is obtained by projecting the item identifier into a learned latent space through an embedding lookup operation

[72]:

$$\mathbf{e}_{ij} = EM(id_{ij}) \in \mathbb{R}^{d_e} \tag{2.3}$$

In this formulation, EM denotes the embedding lookup function and d_e represents the dimensionality of the collaborative embedding space. The embedding vectors are learned during training through backpropagation, and they capture latent collaborative patterns that are not directly observable from item content. Items that are frequently consumed together by similar users will develop similar embedding vectors, even if their observable features are quite different.

The content component of the item representation consists of a visible feature vector $\mathbf{c}_{ij} \in \mathbb{R}^{d_c}$ that encodes the observable attributes of the item. The specific features included depend on the application domain, such as genre, director, and cast for movies, or category, brand, and price for e-commerce products.

The collaborative and content representations are combined through concatenation:

$$\mathbf{t}_{ij} = \text{concat}(\mathbf{c}_{ij}, \mathbf{e}_{ij}) \in \mathbb{R}^{d_c+d_e} \tag{2.4}$$

This hybrid representation enables the model to make accurate predictions even when one source of information is incomplete or noisy, providing robustness for both cold-start items and items with sparse metadata.

- Step 3: Rating-Weighted Aggregation

The third step aggregates the individual item representations into a single user level embedding that summarizes the user's overall preferences. This aggregation is the core operation that enables user-ID independent representation learning. The aggregation is performed using a weighted average, where the normalized ratings serve as attention weights [5, 100] that modulate each item's contribution to the final user representation. The user representation $\mathbf{X}_{u_i}$ for user u_i is computed as:

$$\mathbf{X}_{u_i} = \frac{1}{q} \sum_{j=1}^{q} r_{ij} \times \mathbf{t}_{ij} \tag{2.5}$$

This formulation embodies several important design principles. First, the rating serves as an attention weight that modulates each item's influence on the user representation. When the normalized rating r_{ij} is positive, the corresponding item feature vector $\mathbf{t}_{ij}$ is added to the user representation, pulling it toward the characteristics of liked items. When the normalized rating is negative, the item feature vector is effectively subtracted, pushing the representation away from disliked items. This creates a push and pull dy-

namic that encodes both positive and negative preferences.

Second, division by q provides activity level normalization, ensuring that users with different numbers of interactions produce embeddings with comparable magnitudes. This is essential for fair comparison between highly active users and users with limited engagement.

Third, the dynamic computation from interaction history achieves user-ID independence, eliminating the need for per user parameter storage and enabling seamless handling of new users.

- Step 4: Auxiliary Information Integration

The fourth step incorporates auxiliary information that may provide additional signals about user preferences beyond what can be inferred from interaction history alone. The integration is performed through concatenation:

$$\mathbf{h}_i = \text{concat}(\mathbf{X}_{u_i}, \mathbf{X}_{u_i}^{info}) \tag{2.6}$$

where $\mathbf{X}_{u_i}^{info} \in \mathbb{R}^{d_d}$ represents the auxiliary information vector for user u_i. Common auxiliary features include age group, gender, geographic region, device type, account tenure, and subscription tier. The auxiliary features serve as prior information that can guide predictions when interaction data is sparse, particularly for new users. As users accumulate more interactions, the behavioral signals become more informative and gradually dominate the representation.

- Step 5: Final Preference Vector Computation

The fifth step transforms the combined feature vector into the final preference vector $\mathbf{X}^{final}$ through a multi layer perceptron (MLP) [79]:

$$\mathbf{X}_{u_i}^{final} = \sigma\left(W_2 \cdot \text{ReLU}(W_1 \cdot \mathbf{h}_i + \mathbf{b}_1) + \mathbf{b}_2\right) \tag{2.7}$$

The choice of sigmoid activation (rather than softmax) is a deliberate design decision that enables soft clustering behavior [19]. Unlike softmax, which forces preference values to sum to one, sigmoid allows each preference dimension to be activated independently. This means a user can have high affinity toward multiple preference clusters simultaneously, accurately reflecting the multi faceted nature of real-world preferences.

The resulting final preference vector $\mathbf{X}_{u_i}^{final} \in (0,1)^b$ has b dimensions, where each dimension $X_{u_i k}^{final}$ represents the user's degree of membership in preference cluster k. Values close to 1 indicate strong affinity, while values close to 0 indicate weak affinity.

Critically, $X_{u_i}^{final}$ serves a dual role: it is simultaneously the soft cluster membership vector used for recommendation inference (Eq. 2.10) and the anchor representation fed directly into the triplet loss (Eq. 2.8). The same MLP weights W_1, W_2 therefore receive gradients from both the clustering objective and the recommendation objective in a single backward pass. This co-optimization is the key distinction from prior work that trains clustering and recommendation modules separately: here, improving cluster geometry and improving recommendation quality are not competing objectives applied to separate parameters, but two views of the same learned transformation.

Clustering Model with Contrastive Learning

The Clustering model learns the parameters of the Interaction embedding model such that the resulting preference vectors accurately reflect user similarities and differences. Training is performed through contrastive learning [68], which learns representations by contrasting positive pairs against negative pairs. The training strategies are illustrated in Figure 2.2.

Triplet loss is selected over alternative contrastive objectives such as InfoNCE for two reasons. First, InfoNCE maximises a ratio across a batch of negatives, so its geometric effect on the embedding space depends heavily on batch size and negative sampling strategy both of which are difficult to control when user histories vary widely in length. Triplet loss, by contrast, imposes an explicit distance constraint on each individual triple, making the optimisation target independent of batch composition. Second, the margin λ provides a direct and interpretable geometric guarantee: once the anchor positive distance is smaller than the anchor negative distance by at least λ, the triple contributes zero gradient and training focuses on harder cases.

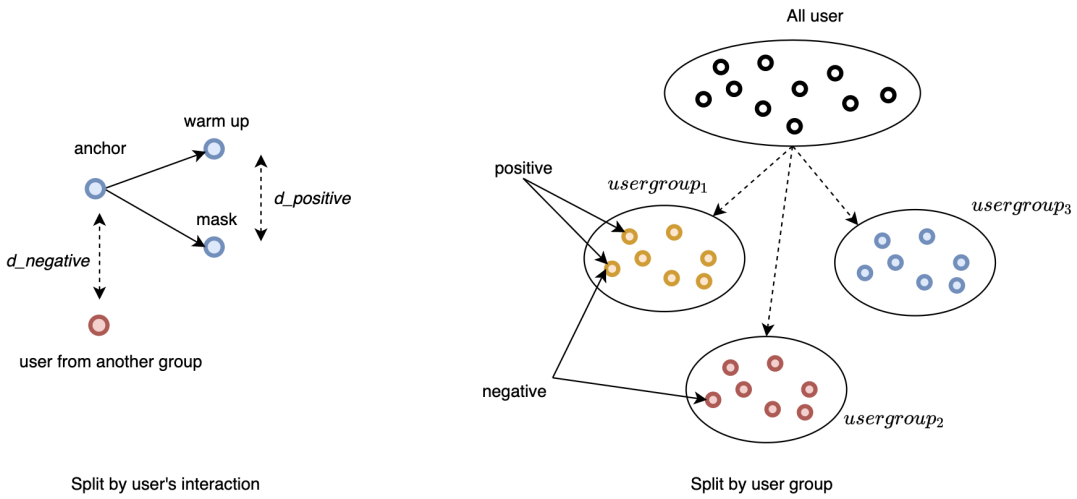

Figure 2.2: Two triplet construction strategies for contrastive learning.

- Triplet Loss Formulation:

The model is trained using a triplet loss [86]:

$$\mathcal{L} = \max\left(\lambda + \frac{\|\mathbf{X}_{u_i}^{final} - \mathbf{X}_i^{p,final}\|_2}{b} - \frac{\|\mathbf{X}_{u_i}^{final} - \mathbf{X}_i^{n,final}\|_2}{b}, 0\right) \tag{2.8}$$

where $\mathbf{X}_{u_i}^{final}$ is the anchor, $\mathbf{X}_i^{p,final}$ is a positive sample, $\mathbf{X}_i^{n,final}$ is a negative sample, λ is the margin, and b is the vector dimension. The loss encourages the anchor positive distance to be smaller than the anchor negative distance by margin λ.

- Strategy 1: User Group Split:

Users are categorized based on their dominant preference category. Positive pairs are formed from users in the same preference group, while negative pairs come from different groups. This strategy enforces discriminative representations across different user populations.

- Strategy 2: User Interaction Split:

The user's interaction history is randomly partitioned into warm up and mask sets [13]. Each subset is encoded separately, and the resulting representations form a positive pair. This strategy enforces intra user representation consistency, ensuring stable representations regardless of which specific items are observed.

The two strategies address orthogonal failure modes and are therefore both necessary. Strategy 1 (group split) prevents inter-user collapse: without it, users from different preference groups may converge to similar regions of the embedding space, making cluster boundaries indistinct. Strategy 2 (interaction split) prevents intra-user instability: without it, a user's representation may shift substantially depending on which subset of their history is observed a critical failure for cold-start users whose histories are small and unrepresentative. A model trained on Strategy 1 alone separates user groups globally but produces unstable representations under sparse observations; a model trained on Strategy 2 alone achieves per-user stability but may not enforce sufficient separation between distinct preference profiles. Together, the two strategies ensure that the learned embedding space is simultaneously well-separated across users and consistent within each user, satisfying both requirements for reliable soft cluster membership inference.

- Soft Clustering versus Hard Clustering:

The comparison is illustrated in Figure 2.3. Hard clustering algorithms like K-means [69] suffer from sparsity (some clusters have too few users) and hard boundary problems (similar users near boundaries are separated).

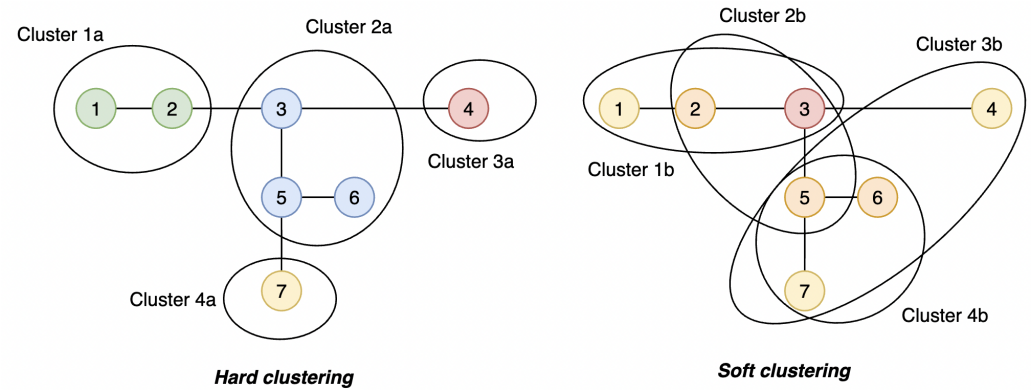

Figure 2.3: Comparison of hard clustering (left) versus soft clustering (right)

Soft clustering [8, 19] addresses these weaknesses by allowing each user to belong to multiple clusters with varying degrees of membership through sigmoid activation.

Item Selection Model

The Item selection model generates personalized recommendations efficiently through a two phase approach:

- Phase 1: Offline Shortlist Construction:

The model pre-computes shortlist scores:

$$\mathfrak{I}_{kj} = \frac{\sum_{i \in \mathcal{U}} r_{ij} \cdot X_{u_i k}^{final}}{\sum_{i \in \mathcal{U}} X_{u_i k}^{final}} \tag{2.9}$$

This represents the weighted average rating that users with membership in cluster k have given to item j. The offline computation has complexity $\mathcal{O}(M)$ and is performed once after training.

- Phase 2: Online Recommendation Generation:

For a target user:

(1) compute final preference vector $\mathbf{X}_{u_i}^{final} = H(S_i, R_i, \mathbf{X}_{u_i}^{info})$;

(2) select top-k clusters $CL = \text{argtop}_k(\mathbf{X}_{u_i}^{final})$;

(3) compute recommendation scores through cluster voting:

$$Score_{ij} = \frac{\sum_{k \in CL} \mathfrak{I}_{k_j} \cdot X_{u_i k}^{final}}{\sum_{k \in CL} X_{u_i k}^{final}} \tag{2.10}$$

(4) rank items by score and return top results.

Handling New and Cold-Start Users

A central advantage of the ID-free design is that EfficientRec handles new and low-interaction users without any architectural change, additional parameters, or retraining. Because no per-user embedding is stored, a user representation is never looked up; it is computed on demand from the user's observed interaction subset. Three properties of the architecture combine to make this effective in the cold-start regime.

Inductive, ID-free representation: As described in Step 3, the user vector is obtained by aggregating the embeddings of the items the user has interacted with, rather than from a learnable user-identifier table. Consequently, a previously unseen user is mapped into the same representation space as existing users directly at inference time, using only the interactions available at that moment. This is fundamentally different from ID-based collaborative or graph models, in which a user absent from the embedding table has no representation and the model must be retrained to incorporate them. EfficientRec is therefore inductive: it generalizes to users never seen during training at zero additional parameter cost.

Activity-level normalization for low-interaction users: The aggregated representation is normalized by the user's interaction count q (Step 2), so that users with very few interactions and highly active users produce embeddings of comparable magnitude. This normalization is what makes the inductive representation usable in the few-shot regime: a user with only a handful of interactions is still placed on the same scale as well-established users, ensuring that the soft cluster assignment that follows is not biased toward high-activity profiles. A new user with a small history is thus assigned softly to the preference clusters that best match their limited evidence, effectively borrowing statistical strength from behaviorally similar users.

Auxiliary information as a cold-start prior: When interaction evidence is scarce, the behavioral signal alone may be insufficient. EfficientRec addresses this within the same forward pass through the auxiliary-integration step in Eq. (2.6), where side features age group, gender, geographic region, device type, account tenure, and subscription tier are concatenated with the behavioral representation. For a new user, these attributes act as an informative prior that guides the prediction before behavioral evidence has accumulated. As the user interacts more, the behavioral component becomes increasingly informative and gradually dominates the representation, so the model transitions smoothly from a prior-driven to a behavior-driven prediction without any switch in architecture or objective. In the limiting case of a user with essentially no interactions, the prediction is governed by the auxiliary prior together with the cluster-level priors learned during training.

2.3 Experimental Settings and Results

2.3.1 Experimental Settings

A. Offline Experimental Settings

a) Dataset and User Categorization

We conduct experiments on the MovieLens-1M dataset containing 6,040 users, 3,706 items, and 1,000,209 interactions. Users are categorized into three groups based on interaction frequency: Cold ($\leq$20 interactions), Warm (21-50 interactions), and Active ($>$50 interactions). We use temporal split with 80% training and 20% testing data.

Table 2.2: Experimental Configuration

Component	Specification
GPU	NVIDIA A100-SXM4-40GB
Framework	PyTorch 2.1.0, CUDA 12.2
Platform	Google Colab Pro+

b) Evaluation Metrics

We evaluate recommendation quality using two primary metrics: Recall@K and NDCG@K

All experiments are evaluated at K=30 and reported over 5 random seeds.

c) Baseline Methods

We compare EfficientRec against state of the art of the time the model published methods from three categories:

- Clustering-based and Matrix Factorization Methods: Traditional approaches that leverage user clustering or matrix decomposition techniques.
  - FCM-Rec [49]: Fuzzy C-Means clustering for collaborative filtering that assigns users to multiple clusters with membership degrees.
  - FCCF [42]: Fuzzy Clustering-based Collaborative Filtering that combines fuzzy clustering with neighborhood-based recommendations.

  - SCoC [56]: Soft Clustering-based Collaborative filtering that uses soft cluster assignments for preference prediction.
  - BPR-MF [76]: Bayesian Personalized Ranking with Matrix Factorization, optimizing pairwise ranking loss for implicit feedback.

- Graph-based Methods: Neural network approaches that model user item interactions as graphs.
  - GraphSAGE [28]: Inductive representation learning that samples and aggregates features from local neighborhoods.
  - GAT [101]: Graph Attention Network that applies attention based neighbor aggregation with learnable weights.
  - NGCF [108]: Neural Graph Collaborative Filtering that captures collaborative signals through embedding propagation on user item graphs.
  - LightGCN [33]: Simplified graph convolution that removes feature transformation and nonlinear activation for efficiency.
  - LINKX [60]: Scalable link prediction method that separates ego and neighbor embeddings for heterophilic graphs.

- Contrastive Learning Methods: Self-supervised approaches that learn representations through contrastive objectives.
  - SGL [111]: Self-supervised Graph Learning that augments user-item graphs and maximizes agreement between different views.
  - SSL4Rec [117]: Self-Supervised Learning for Recommendation that incorporates auxiliary self supervised tasks.
  - MixGCF [40]: Mixes positive samples in graph collaborative filtering to generate harder negative samples for contrastive learning.

d) Optimal Hyperparameter Configuration

Table 2.3: Optimal Hyperparameter Configuration

Hyperparameter	**Optimal Value**
Embedding Dimension	64
Number of Layers (L)	3
Number of Clusters (K)	256
Contrastive Weight (μ)	0.1
Clustering Margin (m)	0.5
Learning Rate	1e-3
Weight Decay (γ)	1e-5
Batch Size	2048

The values reported in Table 2.3 are the optimal configuration obtained from systematic grid search experiments on the MovieLens-1M validation set, in which each hyperparameter was varied independently while all others were held fixed. The detailed sensitivity analysis for the two most critical parameters number of clusters K and contrastive weight μ is presented in Tables 2.8 and 2.9 respectively. The remaining parameters (embedding dimension, number of layers, learning rate, weight decay, and batch size) follow standard settings widely adopted in the recommendation literature and were confirmed to be optimal on the same validation set through the same grid search procedure.

B. Online Experimental Settings

We conduct online testing on the TV360 service to evaluate practical effectiveness. The platform serves two product groups: movies (7,000 series, relatively dense) and videos (200,000 items, very sparse), with 2 million active users.

Table 2.4: Statistics of Online Datasets

Dataset Info	**TV360 Films**	**TV360 Videos**
Active users	1,875,642	732,514
Number of items	7,251	185,324
Number of ratings	5,347,897	3,040,837
Sparsity ratio (%)	0.039	0.002

The TV360 dataset follows the same prediction objective as the offline MovieLens experiments: given a user's historical interaction records, the model is tasked with predicting a ranked list of top-K items most likely to be relevant to that user. User feedback is collected on an explicit 1–5 rating scale.

In the online experiment, user interactions are divided sequentially over time into warm up, mask, and target segments. We perform A/B testing by randomly dividing users into homogeneous groups through stratified sampling. We compare Average Content Per User (ACPU) and Average Duration Per User (ADPU).

Table 2.5: Overall Performance Comparison (All Users) @30, reported as mean ± standard deviation over 5 random seeds.

Model	Recall@30	NDCG@30	Category
Proposed Method			
EfficientRec	$\mathbf{0.1994 \pm 0.0028}^{\dagger}$	$\mathbf{0.1178 \pm 0.0027}^{\text{ns}}$	Proposed
Graph-based Methods			
NGCF	$\underline{0.1958 \pm 0.0013}$	$\underline{0.1174 \pm 0.0007}$	Graph
LINKX	0.1928 ± 0.0022	0.1160 ± 0.0012	Graph
GraphSAGE	0.1658 ± 0.0006	0.0970 ± 0.0004	Graph
GAT	0.1639 ± 0.0001	0.0946 ± 0.0001	Graph
LightGCN	0.1636 ± 0.0003	0.0945 ± 0.0001	Graph
Contrastive Learning Methods			
SSL4Rec	0.1644 ± 0.0001	0.0950 ± 0.0012	SSL
MixGCF	0.1591 ± 0.0001	0.0910 ± 0.0001	CL
SGL	0.0192 ± 0.0015	0.0565 ± 0.0011	CL
Clustering-based and MF Methods			
SCoC	0.1584 ± 0.0001	0.1107 ± 0.0004	Clustering
FCCF	0.1090 ± 0.0001	0.1072 ± 0.0012	Clustering
BPR-MF	0.1074 ± 0.0003	0.0912 ± 0.0001	MF
FCM-Rec	0.1067 ± 0.0001	0.1068 ± 0.0001	Clustering

Bold indicates the best result and underline 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 improves by +1.8% in Recall@30. Each value is the mean ± standard deviation over 5 random seeds, † denotes a statistically significant improvement of EfficientRec over the second-best baseline (NGCF) under a two-sided Welch's t-test ($p = 0.043 < 0.05$); ns indicates the difference is not statistically significant ($p = 0.76$). On Recall@30 the proposed method significantly improves the strongest baseline, whereas on NDCG@30 it achieves the best average but is statistically on par with NGCF.

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.

b) Performance on Cold, Warm, and Active Users

Table 2.6: Performance Comparison on Cold, Warm, and Active Users @30

Model	Cold Users		Warm Users		Active Users	
	R@30	N@30	R@30	N@30	R@30	N@30
EfficientRec	**0.2085**	**0.0910**	**0.1840**	**0.1145**	**0.1689**	0.1264
NGCF	<u>0.2008</u>	<u>0.0902</u>	<u>0.1747</u>	<u>0.1104</u>	<u>0.1647</u>	<u>0.1275</u>
LINKX	0.1866	0.0867	0.1717	0.1070	0.1651	**0.1291**
GraphSAGE	0.1443	0.0628	0.1290	0.0792	0.1506	0.1161
SSL4Rec	0.1444	0.0618	0.1262	0.0772	0.1495	0.1139
LightGCN	0.1411	0.0605	0.1256	0.0766	0.1494	0.1137
MixGCF	0.1358	0.0581	0.1196	0.0733	0.1460	0.1097

R@30 = Recall@30, N@30 = NDCG@30. **Bold** = Best, <u>Underline</u> = Second Best.

EfficientRec achieves the best Recall@30 across all three user scenarios:

- Cold Users: EfficientRec achieves Recall@30 of 0.2085, ahead of the second best method (NGCF: 0.2008) by +3.8%. This demonstrates the effectiveness of the soft clustering mechanism in transferring knowledge from similar user groups to address data sparsity in cold-start scenarios.
- Warm Users: EfficientRec achieves Recall@30 of 0.1840, outperforming NGCF (0.1747) by +5.3%. The improvement is the largest among all scenarios, indicating that the model effectively leverages moderate interaction history combined with cluster based knowledge transfer.
- Active Users: EfficientRec achieves Recall@30 of 0.1689, outperforming NGCF (0.1647) by +2.6%. The relatively smaller improvement suggests that when abundant interaction data is available, the advantage of clustering based knowledge transfer is partially offset by the rich behavioral signals available to all methods.

EfficientRec maintains consistent superiority across all activity levels, demonstrating robustness across diverse user segments.

c) Component Contribution Analysis

To understand the contribution of each component in EfficientRec, we conduct comprehensive ablation studies by systematically removing or replacing key compo-

nents. The central argument of this chapter is that scalable and robust recommendation can be achieved through the synergy of three innovations: ID-free interaction embedding, neural soft clustering, and contrastive learning. To validate this argument, we design ablation experiments that isolate the contribution of each component. Each configuration removes or replaces exactly one component while keeping the rest unchanged, allowing us to assess whether the observed improvements stem from the proposed innovations rather than from the combination of standard techniques. Specifically, the ablation addresses three questions: Does dynamic user representation from interaction history outperform static ID-based embedding? Does soft clustering provide measurable benefits over hard clustering for individual recommendation? Does contrastive learning enhance representation quality beyond the primary recommendation objective?

Table 2.7 presents the performance of each configuration that illustrates the percentage drop compared to the full model.

Table 2.7: Component Ablation Study on EfficientRec

Configuration	**Performance**		**Drop from Full Model**	
	Recall@30	**NDCG@30**	**ΔRecall**	**ΔNDCG**
Full Model	0.1994±0.0028	0.1174±0.0027	–	–
w/o Interaction Embedding	0.1934±0.0023	0.1118±0.0025	-3.0%	-4.8%
Hard Clustering (vs Soft)	0.1954±0.0027	0.1136±0.0033	-2.0%	-3.2%
w/o Contrastive Loss	0.1964±0.0023	0.1166±0.0030	-1.5%	-0.7%

Note: ΔRecall and ΔNDCG represent relative difference compared to full model.

The results provide direct evidence for all three claims. The removal of interaction embedding causes the largest degradation (-3.0% Recall, -4.8% NDCG), confirming that behavior-driven user representations are not merely a memory-saving substitute for ID embeddings but actively improve recommendation quality through end-to-end optimization. Replacing soft clustering with hard clustering degrades performance by -2.0% Recall, validating that allowing users to belong to multiple preference groups with probabilistic weights captures preference structure more faithfully than binary assignment. Removing the contrastive loss results in -1.5% Recall, demonstrating that the self-supervised signal provides meaningful regularization, particularly by encouraging discriminative cluster boundaries. Notably, the three components contribute complementarily rather than redundantly: interaction embedding addresses what to represent, soft clustering addresses how to organize, and contrastive learning addresses how to train each targeting a distinct aspect of the scalable recommendation problem.

d) Hyperparameter Sensitivity Analysis

We analyze the sensitivity of EfficientRec to key hyperparameters. Each experiment varies one hyperparameter while keeping others at their optimal values.

(1) Number of Clusters (K):

Table 2.8: Impact of Number of Clusters

K	**Recall@30**	**NDCG@30**
8	0.1924±0.0035	0.1133±0.0024
32	0.1954±0.0034	0.1148±0.0029
64	0.1968±0.0024	0.1158±0.0025
128	0.1982±0.0030	0.1168±0.0030
256	**0.1994±0.0027**	**0.1174±0.0022**
384	0.1988±0.0025	0.1170±0.0024
512	0.1980±0.0026	0.1165±0.0023
640	0.1972±0.0028	0.1158±0.0025

Performance improves with the number of clusters up to K=256, achieving Recall@30 of 0.1994 and NDCG@30 of 0.1174. Beyond K=256, performance decreases as clusters become too small to capture meaningful preference patterns, leading to over fragmentation. The optimal K=256 represents a balance between cluster granularity and statistical reliability.

(2) Contrastive Learning Weight (μ):

Table 2.9: Impact of Contrastive Learning Weight

μ	**Recall@30**	**NDCG@30**
0.0	0.1970±0.0028	0.1160±0.0025
0.01	0.1974±0.0027	0.1162±0.0028
0.05	0.1984±0.0024	0.1168±0.0024
0.1	**0.1994±0.0026**	**0.1174±0.0022**
0.2	0.1986±0.0032	0.1169±0.0031
0.5	0.1978±0.0029	0.1165±0.0034

The optimal contrastive weight is μ=0.1, achieving +1.2% improvement over μ=0.0

(no contrastive loss). Values larger than 0.2 cause the contrastive objective to dominate the training process, degrading recommendation performance. The optimal μ=0.1 provides effective regularization without overwhelming the primary recommendation objective.

d) Scalability Experiment

To evaluate the scalability of EfficientRec, we conduct experiments on MovieLens-20M, which contains approximately 20 million ratings from 138,000 users on 27,000 movies, a 20$\times$ increase in data size compared to MovieLens-1M. This experiment assesses how well different methods maintain their performance when scaling to larger datasets with increased sparsity.

The results reveal notable performance degradation across all baseline methods when scaling from 1M to 20M. NGCF, the best-performing graph-based method on MovieLens-1M, experiences a -20.9% drop in Recall@30. Self-supervised learning method SSL4Rec shows a -22.4% decrease, while contrastive learning method MixGCF suffers the largest degradation at -23.8%. The clustering-based method SCoC demonstrates a -21.5% drop, indicating that traditional clustering approaches also struggle with increased scale.

In contrast, EfficientRec demonstrates superior scalability with only -16.7% performance drop significantly smaller than all baseline methods. On MovieLens-20M, EfficientRec achieves R@30 of 0.1661 and N@30 of 0.0978, ahead of the second-best method NGCF (R@30 = 0.1549) by +7.2%. This superior scalability can be attributed to three key design choices: (1) the ID-free interaction embedding eliminates the memory bottleneck of user/item embedding tables that grow linearly with dataset size, (2) soft clustering provides efficient approximate retrieval that scales sub-linearly with the number of items, and (3) contrastive learning enhances representation robustness under increased sparsity conditions.

These results confirm that EfficientRec's architecture is particularly well-suited for web-scale recommendation scenarios where maintaining both accuracy and computational efficiency is critical.

Table 2.10: Scalability Comparison: Performance on MovieLens-1M vs MovieLens-20M

Method	Type	MovieLens-1M		MovieLens-20M		Drop (%)
		R@30	N@30	R@30	N@30	
NGCF	Graph	0.1958	0.1168	0.1549	0.0898	-20.9%
SSL4Rec	SSL	0.1644	0.0950	0.1276	0.0723	-22.4%
MixGCF	CL	0.1591	0.0910	0.1213	0.0679	-23.8%
SCoC	Clustering	0.1584	0.1107	0.1243	0.0856	-21.5%
EfficientRec	**Proposed**	**0.1994**	**0.1174**	**0.1661**	**0.0978**	$\mathbf{-16.7\%}$

Note: Drop (%) represents performance degradation in R@30 when scaling from 1M to 20M.

EfficientRec achieves the highest Recall@30 on both scales, outperforming the second-best method NGCF by $+1.8\%$ on MovieLens-1M and $+7.2\%$ on MovieLens-20M. The widening gap across scales is notable: while NGCF degrades by -20.9% when data volume increases $20\times$, EfficientRec degrades by only -16.7%, a difference of 4.2 percentage points. This advantage is attributable to the ID-free design, which eliminates the embedding table bottleneck that causes graph-based methods to scale poorly, and to the soft clustering mechanism, which remains reliable under the increased sparsity of ML-20M (0.53% density). The NDCG@30 pattern mirrors Recall@30, confirming that the scalability advantage holds for ranking quality as well as retrieval coverage.

Memory footprint and scaling in the number of users

Because EfficientRec represents users through K shared soft-cluster prototypes rather than a per-user embedding table, its user-side parameter memory is $O(K \cdot d)$ and therefore independent of the user population M, in contrast to the $O(M \cdot d)$ growth of ID-based models. With $d = 64$ and $K = 256$, the prototypes occupy a constant ≈ 64 KB, whereas an ID-based table requires ≈ 1.47 MB on MovieLens-1M (6,040 users) and ≈ 33.8 MB on MovieLens-20M (138,493 users) reductions of roughly $24\times$ and $541\times$. Extrapolated to the industrial scale of 10^9 users (the ≈ 238 GB case noted in Section 2.1), the same 64 KB footprint represents a reduction of over six orders of magnitude. This constant user-side cost, rather than data volume alone, explains why EfficientRec degrades by only -16.7% in Recall@30 from 1M to 20M (Table 2.10), while ID- and graph-based baselines degrade by -20.9% to -23.8%.

Cluster stability and interpretability

The behaviour of the soft-clustering mechanism can be read directly from the sen-

sitivity and scaling results. With respect to stability, Table 2.8 shows that Recall@30 varies smoothly and unimodally with the number of clusters, peaking at $K = 256$ and changing by less than two points across the wide range $K \in [32, 640]$; the absence of any abrupt collapse indicates that the learned partition is robust to the choice of granularity, with the gentle decline beyond $K = 256$ reflecting expected over-fragmentation rather than instability. The consistently small per-seed standard deviation (± 0.002-0.003) reported across all configurations further indicates that cluster formation is stable across random initialisations rather than seed-dependent. With respect to semantic meaning, the contrastive objective (Table 2.9) explicitly pulls together users with similar behaviour and separates dissimilar ones, so the prototypes are encouraged to act as discriminative preference groups rather than arbitrary partitions; the $+1.2\%$ gain of $\mu = 0.1$ over the contrastive-free variant confirms that this structure contributes to recommendation quality. The probabilistic (soft) membership additionally allows each user to load onto several prototypes simultaneously, which is consistent with the multifaceted nature of real preferences. Finally, the fact that the clusters remain reliable under the increased sparsity of MovieLens-20M (Table 2.10) and yield the largest online gains on the sparsest catalog (Table 2.11) is itself evidence that the prototypes capture genuine, transferable preference semantics: knowledge can only be borrowed across users sharing a cluster if that cluster encodes a meaningful taste pattern.

2.3.2 Online Experimental Results

To measure the real-world impact of the proposed recommendation model, we employ two complementary online metrics. 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. It captures the breadth of user engagement: a higher ACPU indicates that the model successfully surfaces more relevant items, encouraging broader content exploration. 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. It captures the depth of engagement: a higher ADPU indicates that users find the recommended content sufficiently interesting to sustain prolonged viewing sessions. Together, these two metrics provide a comprehensive view of online recommendation quality ACPU reflects content discovery effectiveness while ADPU reflects content satisfaction. A model that improves both metrics simultaneously demonstrates genuine gains in recommendation relevance rather than superficial improvements in a single dimension. Both metrics are computed exclusively over items surfaced by the recommendation algorithm to isolate its contribution from organic user browsing behavior.

Table 2.11: 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.

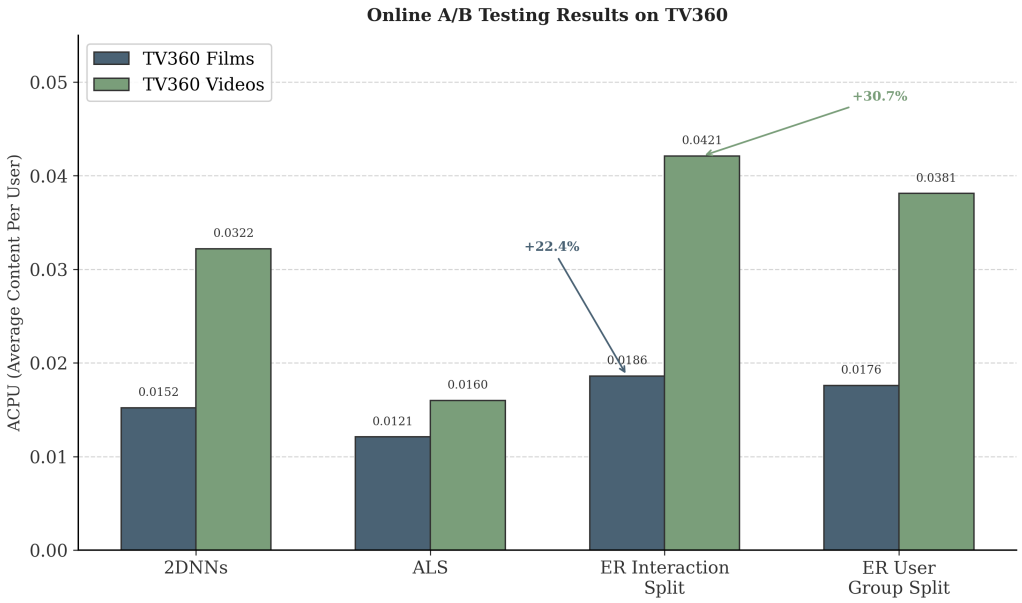

Figure 2.4: Online A/B testing results: Average Content Per User (ACPU) comparison across methods. EfficientRec (Interaction Split) achieves +22.4% improvement for Films and +30.7% for Videos compared to 2DNNs baseline.

EfficientRec (Interaction Split) achieves the best overall results, ahead of 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.

2.4 Chapter Summary

This chapter introduced EfficientRec, a scalable, ID-free recommendation framework that replaces per-user embeddings with behavior-driven representations learned from interaction subsets through deep interaction modeling, soft clustering, and contrastive learning. By computing each user vector on demand rather than storing it, the design lowers the user-side memory footprint from $O(M \cdot d)$ to $O(K \cdot d)$, scales sublinearly with the user population, and extends naturally to new and low-interaction users without retraining. Offline experiments on public benchmarks together with online A/B testing on a production streaming platform confirm that EfficientRec matches or exceeds strong graph-based baselines while substantially reducing resource cost.

The framework nonetheless has boundaries worth making explicit. Because the representation is driven primarily by behavior, users with almost no interactions must lean on the auxiliary prior and on the cluster-level priors learned during training, whose usefulness in turn depends on how rich and reliable the available side features are. The number of clusters is also fixed through grid search rather than learned, so it may drift away from the optimum as the user distribution evolves, and the evaluation so far has centered on movie and streaming data, leaving behavior in other domains to be confirmed.

These observations point to several natural extensions. Learning the number of clusters, or updating clusters incrementally as new interactions stream in, would let the model track a shifting population more closely, while incorporating richer multimodal side information would give a stronger prior precisely where behavioral evidence is thinnest. Finally, compressing the model through quantization, pruning, and distillation would bring inference within the latency and memory budgets needed for on-device deployment.

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).

3.1.1 Graph Neural Networks for Recommendation Systems

Graph Neural Networks (GNNs) have emerged as a powerful paradigm for collaborative filtering, fundamentally transforming how recommendation systems model user item interactions [107, 112]. Unlike traditional matrix factorization approaches that treat user item interactions as independent entries in a sparse matrix, GNN-based methods explicitly model the relational structure inherent in recommendation data through graph-based representations. This structural perspective enables the capture of high order collaborative signals that propagate through the user item bipartite graph, revealing preference patterns that cannot be discovered through direct interaction analysis alone.

The foundational insight underlying GNN-based recommendation is that user item interactions naturally form a bipartite graph structure $\mathcal{G} = (\mathcal{U}, \mathcal{V}, \mathcal{E})$, where users $\mathcal{U}$ and items $\mathcal{V}$ constitute two disjoint node sets, and edges $\mathcal{E}$ represent observed interactions. This graph structure encodes rich collaborative information: users who interact with similar items are implicitly connected through shared item neighbors, and items consumed by similar users share user driven relational connections. GNNs exploit this structure through iterative message passing mechanisms, where each node aggregates information from its neighbors to update its representation, progressively incorporating higher order connectivity patterns.

LightGCN (Light Graph Convolutional Network) [33] constitutes an important contribution that simplified the GNN architecture specifically for collaborative filtering. Through systematic ablation studies, He et al. demonstrated that the feature transformation and nonlinear activation components inherited from general purpose GCNs are not only unnecessary but potentially harming for recommendation tasks. The resulting architecture retains only the essential neighborhood aggregation operation:

$$\mathbf{e}_u^{(l+1)} = \sum_{i \in \mathcal{N}(u)} \frac{1}{\sqrt{|\mathcal{N}(u)||\mathcal{N}(i)|}} \mathbf{e}_i^{(l)} \tag{3.1}$$

where $\mathbf{e}_u^{(l)}$ denotes the user embedding at layer l, and $\mathcal{N}(\cdot)$ represents the neighborhood function. The final representation is obtained by averaging embeddings across all layers:

$$\mathbf{e}_u = \frac{1}{L+1} \sum_{l=0}^{L} \mathbf{e}_u^{(l)} \tag{3.2}$$

This simplified design achieves state of the art performance while significantly reducing computational complexity, establishing LightGCN as the backbone for subsequent GNN-based recommendation research.

NGCF (Neural Graph Collaborative Filtering) [108] pioneered the explicit modeling of collaborative signals through graph neural networks by embedding the user item interaction graph structure into the embedding process. NGCF propagates embeddings on the bipartite graph to capture the collaborative filtering effect, explicitly encoding high order connectivity in user and item representations. The model introduces feature transformation and nonlinear activation during message passing:

$$\mathbf{e}_u^{(l+1)} = \sigma\left(\mathbf{W}_1\mathbf{e}_u^{(l)} + \sum_{i\in\mathcal{N}(u)} \frac{1}{\sqrt{|\mathcal{N}(u)||\mathcal{N}(i)|}}\left(\mathbf{W}_1\mathbf{e}_i^{(l)} + \mathbf{W}_2(\mathbf{e}_i^{(l)} \odot \mathbf{e}_u^{(l)})\right)\right) \tag{3.3}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are trainable weight matrices, $\sigma(\cdot)$ is a nonlinear activation function, and $\odot$ denotes element-wise product that captures feature interactions.

GAT (Graph Attention Networks) [101] introduce attention mechanisms into graph neural networks, enabling nodes to differentially weight the importance of their neighbors during message aggregation. In the recommendation context, attention-based GNNs learn to prioritize interactions that are most informative for preference prediction, naturally handling the heterogeneous importance of different user item connections. The attention mechanism computes importance weights α_{ij} that modulate the contribution of each neighbor:

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}^T[\mathbf{We}_i\|\mathbf{We}_j]\right)\right)}{\sum_{k\in\mathcal{N}(i)}\exp\left(\text{LeakyReLU}\left(\mathbf{a}^T[\mathbf{We}_i\|\mathbf{We}_k]\right)\right)} \tag{3.4}$$

where $\mathbf{a}$ is a learnable attention vector, $\mathbf{W}$ is a shared weight matrix, and $\|$ denotes concatenation.

Despite these advances, GNN-based recommendation methods face persistent challenges that limit their effectiveness in real-world deployment scenarios. The fundamental limitation stems from the inherent sparsity of user item interaction data, which constrains the quality of learned representations and the effectiveness of message propagation. When users have limited interaction histories, the neighborhood aggregation mechanism cannot capture sufficient collaborative signals, leading to suboptimal representations that fail to generalize beyond observed patterns.

3.1.2 The Role of Auxiliary Data in Enhancing GNN-based Recommendation

To address the limitations of canonical data from interaction only GNN models, recent research has increasingly focused on integrating auxiliary data that is additional

information sources beyond direct user item interactions into graph-based recommendation frameworks [89, 124]. Auxiliary data encompasses diverse information types that provide complementary signals for preference modeling, including user demographics, item attributes, textual descriptions, visual features, social relationships, and knowledge graph entities.

The integration of auxiliary data into GNN-based recommendation serves multiple critical purposes:

- Semantic grounding: Auxiliary data provides semantic context for learned representations, enabling the model to understand why users prefer certain items beyond mere co-occurrence patterns.
- Cold-start mitigation: Auxiliary information provides informative signals for users and items lacking sufficient interaction histories, addressing one of the most persistent challenges in recommendation systems.
- Representation robustness: Redundant information sources can compensate when interaction data is noisy or incomplete, enhancing the reliability of preference predictions.

Auxiliary Information Fusion represents a fundamental approach to incorporating auxiliary data into recommendation systems. Side information includes user-side attributes (demographics, preferences, behavioral patterns) and item-side attributes (categories, descriptions, visual features). Formally, let $\mathbf{X}_u^{\text{info}} \in \mathbb{R}^{d_s}$ denote the side information vector for user u, and $\mathbf{X}_i^{\text{side}} \in \mathbb{R}^{d_s}$ for item i. The challenge lies in effectively fusing these heterogeneous information sources with interaction-derived collaborative signals. Early approaches employed simple concatenation:

$$\mathbf{h}_u = [\mathbf{e}_u \| \mathbf{X}_u^{\text{info}}] \tag{3.5}$$

or weighted combination:

$$\mathbf{h}_u = \alpha \cdot \mathbf{e}_u + (1 - \alpha) \cdot \mathbf{W}_s \mathbf{X}_u^{\text{info}} \tag{3.6}$$

KGAT (Knowledge Graph-Enhanced Recommendation) extends the graph structure beyond user item interactions to incorporate external knowledge bases that encode rich semantic relationships among entities [104, 107]. The Knowledge Graph Attention Network (KGAT) [107] constructs a unified graph $\mathcal{G} = \{(h, r, t) | h, t \in \mathcal{E}, r \in \mathcal{R}\}$ that integrates user item interactions with knowledge graph triples, where h and t are head and tail entities, and r represents the relation type. The attention mechanism learns to

weight different relation types-based on their relevance for preference prediction:

$$\pi(h, r, t) = (\mathbf{W}_r \mathbf{e}_t)^T \tanh(\mathbf{W}_r \mathbf{e}_h + \mathbf{e}_r) \tag{3.7}$$

This enables effective knowledge transfer from external sources to recommendation tasks.

HINs (Heterogeneous Information Networks) provide another framework for auxiliary data integration by modeling multiple types of nodes and edges within a unified graph structure [67, 123]. HIN recommendation methods leverage meta paths sequences of node and edge types to capture complex semantic relationships that cannot be expressed in homogeneous graphs. For example, a meta path "User-Movie-Director-Movie" captures the preference pattern where users who like movies by a particular director may also appreciate other works by the same director.

3.1.3 Self-Supervised Learning for Robust graph-based Recommendation

Beyond supervised learning from interaction data and auxiliary information, self-supervised learning has emerged as a powerful paradigm for enhancing the robustness of GNN-based recommendation [111, 118]. Self-supervised methods construct auxiliary training signals from the data itself, providing additional supervision that reinforces representation learning without requiring external labels.

Contrastive Learning has proven particularly effective for graph-based recommendation, learning representations by contrasting positive pairs (similar or related nodes) against negative pairs (dissimilar or unrelated nodes). The contrastive objective, typically formulated as the InfoNCE loss, encourages the model to capture meaningful similarity structures:

$$\mathcal{L}_{CL} = \sum_{i \in \mathcal{B}} -\log \frac{\exp\left(\mathbf{z}_i'^T \mathbf{z}_i'' / \tau\right)}{\sum_{j \in \mathcal{B}} \exp\left(\mathbf{z}_i'^T \mathbf{z}_j'' / \tau\right)} \tag{3.8}$$

where $\mathbf{z}_i'$ and $\mathbf{z}_i''$ are two augmented views of node i, $\mathcal{B}$ is a sampled batch, and τ is the temperature parameter.

SGL (Self-supervised Graph Learning for Recommendation)

Self-supervised Graph Learning (SGL) [111] pioneers the application of contrastive self-supervised learning to graph-based recommendation. The core motivation of SGL is to address the data sparsity problem that fundamentally constrains collaborative filtering performance. By introducing auxiliary self-supervised signals derived from graph structure, SGL reinforces node representation learning through self-discrimination, enabling

the model to learn more robust and generalizable representations without requiring additional supervision.

SGL proposes a joint learning framework that combines the traditional supervised recommendation loss with an auxiliary contrastive loss:

$$\mathcal{L}_{\text{joint}} = \mathcal{L}_{\text{rec}} + \lambda \mathcal{L}_{\text{cl}} \tag{3.9}$$

where $\mathcal{L}_{\text{rec}}$ is the recommendation loss (typically BPR loss), $\mathcal{L}_{\text{cl}}$ is the contrastive loss, and λ controls the balance between the two objectives. To generate diverse views for contrastive learning, SGL explores three graph augmentation strategies: node dropout, edge dropout, and random walk sampling. These structural perturbations create variations of the original graph that preserve essential collaborative patterns while introducing controlled diversity for contrastive learning.

SimGCL (Simple Graph Contrastive Learning for Recommendation)

SimGCL [119] challenges the conventional wisdom that graph augmentations are necessary for effective contrastive learning in recommendation. Through systematic investigation, SimGCL reveals that the performance improvements attributed to graph-based contrastive learning primarily stem from the uniformity regularization effect of the InfoNCE loss rather than the specific augmentation strategies employed. This finding suggests that the computational overhead associated with graph augmentations may not be justified by their contribution to representation quality.

Based on this insight, SimGCL proposes a remarkably simple yet highly effective alternative: instead of perturbing the graph structure, SimGCL directly adds random uniform noise to node embeddings to create contrastive views. The noise perturbation is formulated as:

$$e_i' = e_i + \Delta_i, \quad \|\Delta_i\|_2 = \epsilon \tag{3.10}$$

where e_i is the original node embedding, Δ_i is the noise vector, and ϵ controls the perturbation magnitude. This noise-based augmentation strategy eliminates the computational cost of repeatedly constructing and processing augmented graphs while achieving superior recommendation performance.

XSimGCL (Cross-Layer Contrastive Learning)

XSimGCL [121] extends SimGCL by introducing cross-layer contrastive learning as a more principled approach to generating contrastive views. The key innovation of XSimGCL is the recognition that representations at different GNN layers encode information at different granularities: earlier layers capture local neighborhood patterns while deeper layers incorporate broader structural context. By contrasting representa-

tions across layers rather than at only the final layer, XSimGCL enables the model to learn multi-scale invariances that enhance representation robustness.

Furthermore, XSimGCL introduces a unified design where the same perturbed representations serve both the recommendation task and the contrastive learning objective. This design eliminates the need for separate forward passes for different views, further improving computational efficiency. XSimGCL demonstrates that thoughtful architectural choices in how contrastive objectives are integrated can yield substantial improvements in both effectiveness and efficiency.

3.1.4 Research Gaps and Contributions

Despite the significant progress reviewed above, several critical research gaps remain that motivate the contributions presented in this chapter.

Gap 1: Adaptive Side Information Fusion. Existing methods for integrating auxiliary data with GNN-based recommendation typically employ static fusion strategies that apply uniform weighting across all users and contexts. However, the informativeness of different data sources varies significantly for some users, demographic information may be highly predictive, while for others, behavioral signals provide more accurate preference indicators. There is a need for adaptive fusion mechanisms that can dynamically adjust the contribution of auxiliary data-based on user specific characteristics and data quality.

Gap 2: Cold-Start Handling with Auxiliary Data. While auxiliary data theoretically provides signals for cold-start users and items, existing approaches lack principled mechanisms for leveraging side information when interaction data is severely limited or absent. The challenge is to develop models that can gracefully transition from auxiliary data driven prediction for cold entities to interaction driven prediction for warm entities, without requiring separate model architectures or manual intervention.

Gap 3: Dimension Aware Contrastive Learning. Current contrastive learning methods for recommendation treat all embedding dimensions uniformly during augmentation and learning. This uniform treatment fails to account for the heterogeneous informativeness of different dimensions some dimensions may encode critical preference signals while others capture noise. There is a need for adaptive mechanisms that can identify and differentially treat embedding dimensions-based on their task relevance.

Gap 4: Integration of Masking with Graph Contrastive Learning. While learnable masking has proven effective in various deep learning domains for identifying task-relevant features, its integration with graph-based contrastive learning for recommendation remains unexplored. The potential synergy between masking and contrastive learn-

ing where masks can inform view generation and contrastive objectives can guide mask learning offers compelling opportunities for more effective self-supervised recommendation.

To address these gaps, this chapter presents two complementary contributions:

- Section 3.2: GIFT4Rec introduces a novel framework for side information fusion in GNN-based recommendation, specifically designed for cold-start scenarios. GIFT4Rec employs an attention-based "Weight Generated" module that dynamically computes user's specific fusion weights, controlling the relative contribution of behavioral embeddings and auxiliary side information. The framework incorporates both local and global fusion modules optimized through meta learning principles, achieving robust recommendation across warm-start and cold-start conditions.
- Section 3.3: MaskSimGCL proposes a masked contrastive learning framework that extends SimGCL with learnable dimension masks. The masks adaptively weight embedding dimensions-based on their task relevance, providing implicit regularization against overfitting while enabling mask informed contrastive view generation. By applying differential perturbations-based on learned importance scores, MaskSimGCL generates semantically consistent contrastive views that enhance representation learning in sparse data environments.

Together, these contributions advance the state of the art in GNN-based recommendation by addressing the complementary challenges of auxiliary data integration and self-supervised representation learning, providing principled solutions for robust recommendation under data sparsity and cold-start conditions. Table 3.1 summarizes the mapping between research gaps and the proposed solutions.

Table 3.1: Mapping of Research Gaps to Chapter Contributions

Research Gap	**GIFT4Rec**	**MaskSimGCL**
Adaptive side information fusion	✓	
Cold-start handling	✓	
Dimension-aware learning		✓
Masking + contrastive integration		✓
Data sparsity robustness	✓	✓

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).

The Cold-Start Challenge in Recommendation Cold-start recommendation represents one of the most fundamental and persistent challenges in recommender systems, arising when models must generate predictions for users or items with insufficient historical interaction data [102]. This challenge arises in two primary scenarios:

- User cold-start: Newly registered users lack sufficient behavioral history to support personalized recommendations. Without prior interaction records, collaborative filtering techniques cannot identify similar users or infer preferences from historical patterns.
- Item's cold-start: Newly introduced items have no interaction records and cannot be effectively incorporated into collaborative filtering pipelines. This prevents the system from learning item characteristics through user feedback.

Traditional collaborative filtering approaches, which rely exclusively on user item interaction matrices, inherently struggle in these scenarios because their prediction mechanisms depend on patterns extracted from historical interactions that are simply unavailable for cold-start entities [51].

The severity of the cold-start problem is increased in modern digital ecosystems where user bases grow continuously, item catalogs expand rapidly, and new users and products are constantly introduced. In such dynamic environments, a significant portion of recommendation requests may involve cold-start scenarios, making robust cold-start handling essential for industrial systems. For instance, e-commerce platforms regularly

introduce thousands of new products daily, while streaming services must recommend newly released content to users who have not yet developed viewing histories.

Furthermore, the inability to provide accurate recommendations for new users risks losing potential engagement during the critical onboarding phase, while poor cold-start item recommendations may result in valuable new products remaining undiscovered. Research indicates that users who receive poor initial recommendations are significantly less likely to continue engaging with the platform [52].

Related Methodologies To address the cold-start problem, researchers have explored various methodologies that can be categorized into three main directions: side information fusion, attention mechanisms for adaptive learning, and meta learning for rapid adaptation.

Side Information Fusion for Cold-Start Mitigation

Side information fusion has emerged as a promising direction for alleviating cold-start challenges by leveraging auxiliary data associated with users and items to compensate for missing interaction histories [126]. Typical forms of side information include:

- User attributes: Demographics (age, gender, location), social relationships, and contextual signals
- Item attributes: Textual descriptions, content features (genres, categories), visual information, and knowledge graph entities

Recent studies have demonstrated that integrating side information with user item interaction data enables recommender systems to construct more expressive representations that can infer user preferences even when direct behavioral evidence is unavailable. For example, in movie recommendation, combining user demographic information with movie attributes such as genres, directors, and actors allows the system to generate meaningful recommendations for users who have not yet established a viewing history.

DropoutNet [102] introduced a neural network approach that uses content features as input and applies dropout during training to simulate cold-start conditions, enabling the model to learn robust representations that generalize to users with limited interactions. The Heterogeneous Information Network approach [123] constructs heterogeneous graphs that integrate multiple types of side information to learn comprehensive user and item representations for cold-start scenarios.

Attention Mechanisms for Adaptive Information Weighting

Attention mechanisms have revolutionized deep learning by enabling models to focus selectively on relevant components of input data [100]. In the context of recommender systems, attention-based approaches have demonstrated significant improvements in capturing nuanced user item relationships.

Graph Attention Networks (GAT) [101] apply attention mechanisms to graph-structured data, enabling nodes to differentially weight the importance of their neighbors during message passing. This selective aggregation leads to more expressive representations compared to uniform neighborhood averaging. The Knowledge Graph Attention Network (KGAT) [107] extends this concept by applying attention weighted aggregation over knowledge graph relations, effectively capturing semantic relationships between entities.

In the context of side information fusion, attention mechanisms offer a principled approach to dynamically assess and weight the contribution of heterogeneous information sources. Rather than treating all auxiliary attributes equally, attention-based fusion can learn to prioritize information sources that are most relevant for specific users or recommendation contexts.

Meta-Learning for Rapid Adaptation

Meta-learning, often characterized as "learning to learn," provides a paradigm for training models that can quickly adapt to new tasks with minimal data. The Model-Agnostic Meta-Learning (MAML) [26] framework demonstrates that neural networks can be explicitly trained to learn initialization parameters that facilitate rapid adaptation to new tasks.

In the recommendation domain, meta-learning approaches have been successfully applied to address cold-start challenges. MeLU [52] applies MAML to learn user preference estimators that can quickly adapt to new users with few interactions. The warm-up approach [99] uses meta-learning to generate effective embeddings for cold-start advertisements. More recently, AMeLU [66] combines attention mechanisms with meta-learning to capture diverse user preferences during the adaptation process, recognizing that users may have varying interests across different item categories.

These meta-learning approaches demonstrate the potential for learning transferable knowledge that generalizes beyond the training distribution. However, existing methods have not fully explored the integration of meta-learning with side information fusion in a unified framework.

Limitations of Existing Approaches Despite the progress made by existing methodologies, several limitations remain that impact their effectiveness in real-world cold-start

recommendation scenarios.

Limitation 1: Systematic Bias from Uncontrolled Side Information

Side information may introduce systematic bias if the auxiliary data are incomplete, noisy, or reflect societal biases. For example, demographic attributes such as age, gender, or location may inadvertently cause models to learn biased patterns or produce unfair treatment across different user groups. When side information is directly incorporated without careful control mechanisms, these biases can propagate through the recommendation pipeline and result in unfair outcomes.

Furthermore, side information quality varies significantly across users and items. Some users may have complete demographic profiles while others provide minimal information. Similarly, item attributes may be inconsistently populated across the catalog. Existing approaches that treat side information uniformly fail to account for this heterogeneity in data quality and completeness.

Limitation 2: Static and Uniform Fusion Mechanisms

Most existing side information fusion methods employ static combination rules that cannot adapt to the varying informativeness of different data sources across diverse user populations and recommendation contexts. These approaches typically use fixed weighted combinations or simple concatenation strategies that treat all information sources equally for all users.

However, the relevance and reliability of different information types vary significantly. For some users, demographic information may be highly predictive of preferences, while for others, behavioral signals provide more accurate indicators. A user's age might strongly predict music preferences but have little relevance for technical book recommendations. Static fusion mechanisms cannot capture these context dependent relationships.

DropoutNet [102], while effective in simulating cold-start conditions, applies uniform dropout without considering the relative importance of different information sources. This can lead to suboptimal representations when certain side information is more valuable than interaction derived features for specific user segments.

Limitation 3: Overfitting Risk from Rich Auxiliary Data

Incorporating rich side information increases the risk of overfitting, especially when models become overly dependent on auxiliary attributes that happen to correlate with training data but do not generalize to new scenarios. Deep neural networks

with high capacity can easily memorize spurious correlations between side information features and user preferences observed in training data.

In cold-start scenarios, this overfitting problem is particularly severe because the model must make predictions for users or items that differ from the training distribution. A model that has overfit to demographic patterns in the training set may fail catastrophically when encountering users with unusual demographic combinations or items with novel attribute configurations.

Existing approaches lack mechanisms to explicitly regularize against overfitting to auxiliary attributes while maintaining the ability to leverage useful side information for cold-start prediction.

Limitation 4: Disconnection Between Meta-Learning and Side Information

While meta-learning approaches like MeLU [52] have shown promise for cold-start recommendation, they primarily focus on learning good initializations for user embeddings without explicitly considering how to optimally balance behavioral signals and auxiliary information. The meta-learning objective is typically defined over interaction prediction performance without accounting for the fusion of heterogeneous information sources.

Similarly, attention-based fusion methods operate independently of meta-learning principles, missing the opportunity to learn fusion strategies that generalize to unseen users and items. The AMeLU [66] approach begins to address this limitation but does not provide a comprehensive framework for side information fusion in graph-based recommendation.

Limitation 5: Limited Exploitation of Graph Structure for Side Information

Graph Neural Networks have demonstrated powerful capabilities for learning user item representations through message passing over interaction graphs. However, existing GNN-based recommender systems often treat side information as secondary features that are simply concatenated with learned embeddings, rather than deeply integrating auxiliary information into the graph learning process.

This weak integration fails to fully exploit the relational structure that connects users, items, and their attributes. A more principled approach would learn to fuse side information in a manner that complements and enhances the collaborative signals captured through graph-based message passing.

Table 3.2: Summary of Limitations and Research Limitations

Limitation	**Research Limitation**
Systematic bias from uncontrolled side information	Need for adaptive mechanisms that control the influence of potentially biased auxiliary data
Static and uniform fusion mechanisms	Need for dynamic fusion that adapts to user-specific information relevance
Overfitting risk from rich auxiliary data	Need for regularization strategies that prevent over-reliance on training correlations
Disconnection between meta-learning and side information	Need for unified frameworks that combine meta-learning with side information fusion
Limited exploitation of graph structure	Need for deep integration of side information with GNN-based collaborative filtering

Comparative Positioning of GIFT4Rec Against Related Methods

Table 3.3 summarizes the key design differences between GIFT4Rec and the most closely related cold-start and side-information fusion methods across six dimensions that directly correspond to the five limitations identified above.

Table 3.3: Design Comparison of GIFT4Rec Against Related Methods

Characteristic	**DropoutNet [102]**	**MeLU [52]**	**AMeLU [66]**	**KGAT [107]**	**LightGCN [33]**	**GIFT4Rec (Ours)**
GNN-based backbone	✗	✗	✗	✓	✓	✓
Side information fusion	✓ (fixed)	✗	✗	✓ (fixed)	✗	✓ **(adaptive)**
Per-user adaptive fusion weight	✗	✗	Partial	✗	✗	✓
Meta-learning for generalization	✗	✓	✓	✗	✗	✓
Joint meta-learning + side fusion	✗	✗	✗	✗	✗	✓
Cold-start handling	✓	✓	✓	Partial	✗	✓

Fixed: uniform or static weights applied identically across all users. *Partial*: limited cold-start support requiring at least some interaction data.

The comparison highlights one gap that no existing method resolves: the *joint* integration of meta-learning with per-user adaptive side information fusion. DropoutNet [102] and KGAT [107] incorporate side information but apply fixed, user-agnostic fusion rules that cannot adapt to individual data quality. MeLU [52] and AMeLU [66] leverage meta-learning for rapid user adaptation, yet neither explicitly models the dynamic balance between behavioral signals and auxiliary features side information is either absent or treated as a secondary input without learned weighting. GIFT4Rec closes this gap through the Weight Generated module, which produces a per-user fusion coef-

ficient optimized via a two-level strategy: a local objective (LSIF) that adapts to each user's interaction density, and a global meta-learning objective (GSIF) that ensures the learned weights generalize to unseen users and items.

3.2.2 Gift4Rec: Model Architecture and Components

To address the limitations identified above, we propose GIFT4Rec (GNN-based Side Information Fusion Technique for Recommendation), a novel architecture explicitly designed for robust cold-start recommendation through the synergistic combination of attention-based fusion and meta-learning principles.

GIFT4Rec introduces a unified framework that addresses all five limitations through three key innovations:

- Attention-based Weight Generation: A learnable Weight Generated module that dynamically computes user's specific fusion weights, controlling the relative contribution of behavioral embeddings and side information features. This addresses the limitations of static fusion and enables fairness aware recommendation by preventing over reliance on potentially biased attributes.
- Dual Module Side Information Fusion: The framework comprises two complementary modules:
  - Local Side Information Fusion (LSIF): Optimizes fusion weights-based on recommendation performance during standard training, learning to balance information sources for accurate prediction.
  - Global Side Information Fusion (GSIF):Employs meta-learning-inspired optimization to learn fusion strategies that generalize to unseen data, reducing overfitting risk.
- GNN-Integrated Architecture: Deep integration with Graph Neural Networks enables the model to leverage both collaborative signals from the interaction graph and semantic information from user and item's attributes in a unified representation learning framework.

For cold-start users who lack behavioral history, this mechanism naturally shifts toward relying more heavily on side information, while for active users with rich interaction data, the model can leverage the more informative behavioral embeddings.

Contributions of GIFT4Rec

The main contributions of GIFT4Rec in addressing the identified research Limitations are summarized as follows:

Novel Side Information-Driven Cold-Start Technique

We propose an effective approach to infer the interests of cold-start users and recommend suitable items under extremely sparse interaction settings. By leveraging auxiliary user attributes in combination with graph-based interaction modeling, GIFT4Rec can generate meaningful recommendations even for users with no prior interactions. The model learns to extract predictive signals from demographic information, contextual features, and other auxiliary attributes when behavioral data is unavailable.

This contribution directly addresses Limitation 1 by providing controlled mechanisms for incorporating side information while mitigating bias propagation.

Attention-Based Adaptive Fusion Mechanism

We introduce a novel attention-based fusion mechanism, implemented through the weight generated module, that dynamically controls and estimates the relative importance of heterogeneous user information sources. Unlike static fusion approaches, our mechanism learns user's specific weights that adapt to the informativeness of different data sources for individual users.

The weight generated module takes the concatenation of behavioral and side information embeddings as input and outputs a fusion coefficient through an MLP with sigmoid activation. This enables the model to learn complex, non linear relationships between information sources and their relevance for specific users.

This contribution directly addresses Limitation 2 by replacing static fusion rules with learned, adaptive weighting strategies.

Meta-Learning Integration for Enhanced Generalization

We integrate a meta-learning inspired strategy through the global side information fusion module (GSIF) to reduce the risk of overfitting and enhance the model's generalization capability on previously unseen users and items. The GSIF module implements a form of two levels optimization:

- Inner loop: Standard recommendation training optimizes the GNN parameters and behavioral embeddings using the cross entropy loss $\mathcal{L}_{CF}$.

- Outer loop: The Weight Generated module parameters are optimized using a meta-objective $\mathcal{L}_{\text{global}}$ that compares model performance when using behavioral versus side information embeddings on validation data.

This two levels structure ensures that the learned fusion weights generalize beyond the training distribution, addressing the overfitting concerns raised in Limitation 3.

This contribution also directly addresses Limitation 4 by providing an explicit connection between meta-learning principles and side information fusion.

Unified GNN-based Architecture

GIFT4Rec presents a consistent end to end architecture that deeply integrates side information fusion with GNN-based collaborative filtering. Rather than treating side information as an afterthought, the framework learns to optimally combine graph derived behavioral signals with auxiliary features through the shared weight generated mechanism.

The architecture ensures that the local and global fusion modules operate through shared parameters, enabling consistent and complementary learning objectives. This unified design allows the model to simultaneously optimize for accurate recommendation through LSIF and robust generalization through GSIF.

This contribution directly addresses Limitation 5 by providing deep integration rather than shallow concatenation of side information with graph-based representations.

Table 3.4: Mapping of GIFT4Rec Contributions to Research Limitations

Research Limitation	**GIFT4Rec Solution**
Adaptive control of side information	Weight Generated module with learned fusion coefficients
Dynamic user-specific fusion	Attention-based mechanism computing per-user weights
Overfitting prevention	Global Side Information Fusion with meta-learning optimization
Meta-learning + side information integration	Two -level optimization jointly learning fusion and generalization
Deep GNN integration	end to end architecture with shared Weight Generated parameters

Model Architecture

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).

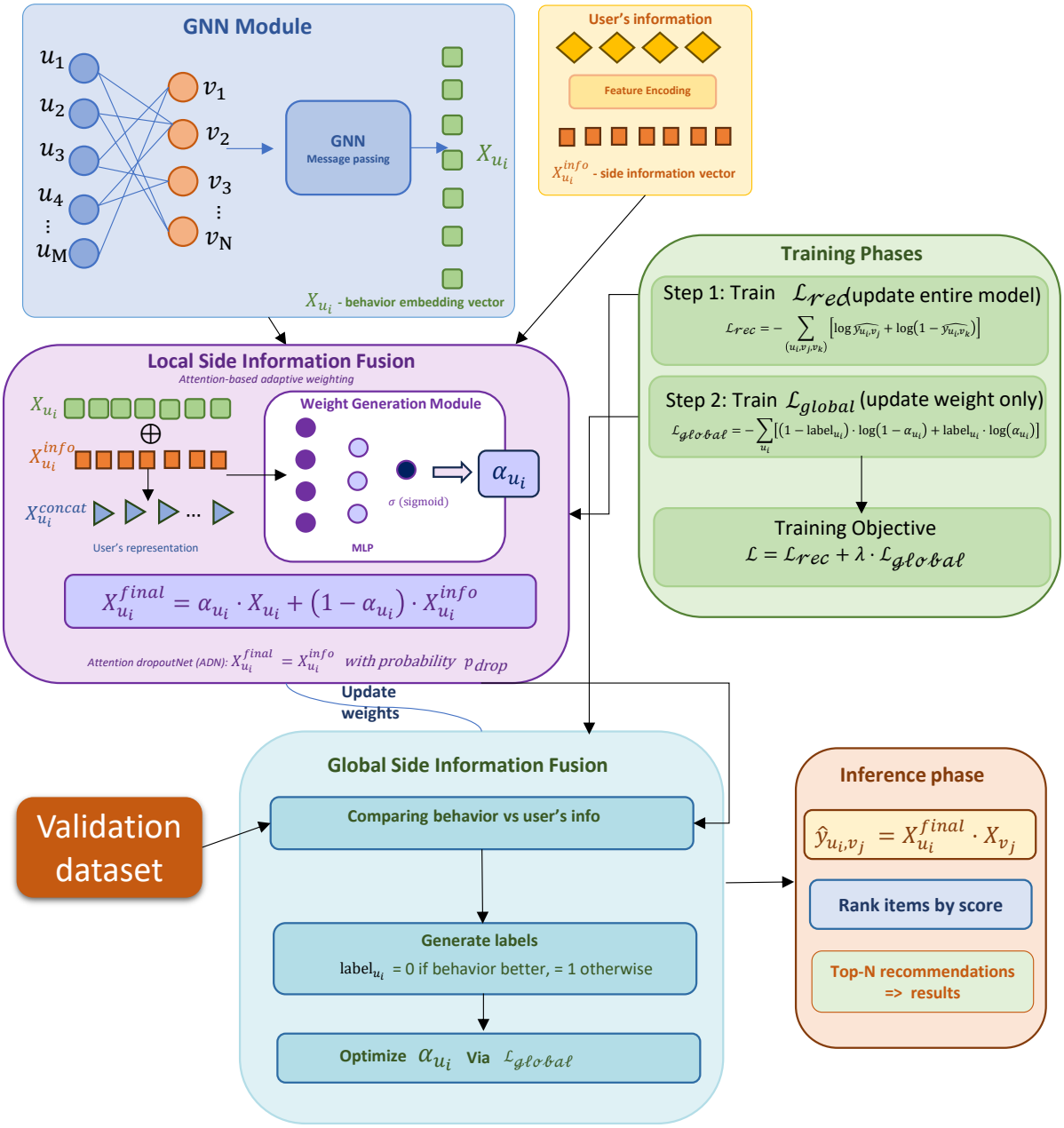

Figure 3.1: GIFT4Rec overall architecture

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.

Problem Formulation and Notation

Before describing the detailed architecture of each component, we establish the

mathematical notation consistent with the formulation in Chapter 2. Consider a recommendation system with users $\mathcal{U} = \{u_1, u_2, \ldots, u_M\}$ and items $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$. The user item interaction graph is defined as $\mathcal{G} = (\mathcal{U}, \mathcal{V}, \mathcal{E})$, where an edge $(u_i, v_j) \in \mathcal{E}$ indicates an observed interaction between user u_i and item v_j.

For each user u_i, we observe:

- An interaction history $S_i = \{v_{i1}, v_{i2}, \ldots, v_{iq}\}$ consisting of q items the user has interacted with.
- A side information vector $\mathbf{X}_{u_i}^{\text{info}} \in \mathbb{R}^{d_s}$ containing auxiliary attributes such as demographics, preferences, or contextual features.

The objective of GIFT4Rec is to learn a mapping function $\mathcal{H}$ that produces a final user representation by adaptively fusing behavioral signals with side information:

$$\mathbf{X}_{u_i}^{\text{final}} = \mathcal{H}\left(\mathcal{G}, S_i, \mathbf{X}_{u_i}^{\text{info}}\right) \tag{3.11}$$

The key insight of GIFT4Rec is that this fusion is personalized each user receives a different weight balancing behavioral and side information-based on the informativeness of their interaction history.

Training Procedure

The training of GIFT4Rec follows a two-phase procedure that alternates within each epoch, ensuring that the recommendation objective and the generalization objective receive appropriately decoupled gradient signals.

Phase 1 Local optimization. In the first phase, all model parameters are updated jointly. For each mini-batch of training triples (u_i, v_j^+, v_j^-), the GNN encoder performs L layers of message passing to produce the behavioural embeddings $\mathbf{X}_{u_i}$ and $\mathbf{X}_{v_j}$. The Weight Generated module computes the per-user fusion coefficient α_{u_i} from the concatenation of $\mathbf{X}_{u_i}$ and $\mathbf{X}_{u_i}^{\text{info}}$, and the Attention DropoutNet (ADN) applies stochastic masking of the behavioural embedding with probability p_{drop} to simulate cold-start conditions during training. The resulting fused representation $\mathbf{X}_{u_i}^{\text{final}}$ is used to compute the recommendation loss $\mathcal{L}_{\text{rec}}$ (Eq. 3.24), and gradients are back-propagated through the entire network.

Phase 2 Global fine-tuning. In the second phase, the GNN encoder and all embedding tables are frozen. For each user in the validation set, the model independently scores candidate items using only the behavioural embedding ($\hat{y}^{\text{beh}}$) and only the side

information ($\hat{y}^{\text{info}}$). A binary label is derived from whichever source achieves higher Recall, and only the Weight Generated module is updated via $\mathcal{L}_{\text{global}}$ (Eq. 3.22). Freezing the GNN during this phase prevents the meta-objective from distorting the collaborative representations already learned in Phase 1.

The two phases are summarized in the following steps:

1. Initialize GNN encoder, side information encoder, and Weight Generated module.
2. **For each epoch**:
   a. *Phase 1:* For each training batch (u_i, v_j^+, v_j^-), compute $\mathbf{X}_{u_i}$, $\mathbf{X}_{v_j}$ via GNN; compute α_{u_i} via Weight Generated; apply ADN; compute $\mathbf{X}_{u_i}^{\text{final}}$; update all parameters using $\mathcal{L}_{\text{rec}}$.
   b. *Phase 2:* Freeze GNN; for each validation user compute $\hat{y}^{\text{beh}}$ and $\hat{y}^{\text{info}}$; assign binary label; update Weight Generated module using $\mathcal{L}_{\text{global}}$; unfreeze GNN.
3. Repeat until convergence

GNN Interaction Module

The GNN in the GIFT4Rec architecture is responsible for learning user and item representations from the interaction graph structure. This component captures high order collaborative signals by propagating information through the graph, enabling the model to discover preference patterns that emerge from collective user behavior.

Unlike methods that treat each user item pair independently, the GNN module leverages the global graph structure to learn representations that encode both direct interactions and indirect relationships through multi hops neighbors. This enables the model to make accurate predictions even for user item pairs with no direct interaction history.

The module employs an iterative message passing scheme where each node aggregates information from its neighbors:

$$\mathbf{h}_{\cdot}^{(\ell+1)} = \text{Update}^{(\ell)}\left(\mathbf{h}_{\cdot}^{(\ell)}, \text{Aggregate}^{(\ell)}\left(\left\{\mathbf{h}_{n}^{(\ell)} : n \in \mathcal{N}(\cdot)\right\}\right)\right) \tag{3.12}$$

where $\mathbf{h}_{\cdot}^{(\ell)}$ denotes the representation at layer ℓ, and $\mathcal{N}(\cdot)$ represents the neighborhood of the target node. Following the LightGCN design [33], we adopt a simplified aggregation that removes non-linear transformations:

$$\mathbf{h}_{u}^{(\ell+1)} = \sum_{v \in \mathcal{N}(u)} \frac{1}{\sqrt{|\mathcal{N}(u)||\mathcal{N}(v)|}} \mathbf{h}_{v}^{(\ell)} \tag{3.13}$$

After L layers of propagation, the final user and item embeddings are obtained by averaging across all layers:

$$\mathbf{X}_{u_i} = \frac{1}{L+1} \sum_{\ell=0}^{L} \mathbf{h}_{u_i}^{(\ell)}, \quad \mathbf{X}_{v_j} = \frac{1}{L+1} \sum_{\ell=0}^{L} \mathbf{h}_{v_j}^{(\ell)} \tag{3.14}$$

The layer combination captures collaborative signals at different semantic levels lower layers encode local neighborhood patterns while higher layers capture global structural properties.

Local Side Information Fusion Module

The local side information fusion module addresses a fundamental challenge in recommendation: how to optimally combine behavioral signals with side information when their relative informativeness varies across users. Users with rich interaction histories may have behavioral embeddings that accurately reflect their preferences, while users with limited interactions may benefit more from side information.

Rather than applying a fixed fusion strategy uniformly across all users, this component learns a personalized fusion weight $\alpha_{u_i} \in [0, 1]$ for each user that balances the contribution of behavioral and side information embeddings.

The module consists of four components:

Feature concatenation: The behavioral embedding $\mathbf{X}_{u_i}$ from the GNN module and the side information embedding $\mathbf{X}_{u_i}^{\text{info}}$ are concatenated to form a joint representation:

$$\mathbf{X}_{u_i}^{\text{concat}} = \left[\mathbf{X}_{u_i} \mathbf{X}_{u_i}^{\text{info}}\right] \tag{3.15}$$

Weight generated module: An MLP with sigmoid activation learns to predict the optimal fusion weight-based on the joint representation:

$$\alpha_{u_i} = \sigma\left(\text{MLP}\left(\mathbf{X}_{u_i}^{\text{concat}}\right)\right) \tag{3.16}$$

where $\sigma(\cdot)$ denotes the sigmoid function ensuring $\alpha_{u_i} \in [0, 1]$.

The weight generated module learns to assess the quality of each information source by examining their characteristics jointly. Users with distinctive behavioral patterns will receive higher α values, while users whose side information is more predictive will receive lower values.

Adaptive Fusion: The final user representation is computed as a weighted combi-

nation:

$$\mathbf{X}_{u_i}^{\text{final}} = \alpha_{u_i} \cdot \mathbf{X}_{u_i} + (1 - \alpha_{u_i}) \cdot \mathbf{X}_{u_i}^{\text{info}} \tag{3.17}$$

Attention dropoutNet (ADN): To encourage the model to leverage side information more effectively, we introduce a dropout mechanism during training. With probability p_{drop}, the behavioral embedding is masked, forcing the model to rely solely on side information:

$$\mathbf{X}_{u_i}^{\text{final}} = \begin{cases} \mathbf{X}_{u_i}^{\text{info}} & \text{with probability } p_{\text{drop}} \\ \alpha_{u_i} \cdot \mathbf{X}_{u_i} + (1 - \alpha_{u_i}) \cdot \mathbf{X}_{u_i}^{\text{info}} & \text{otherwise} \end{cases} \tag{3.18}$$

ADN prevents the model from over relying on behavioral signals and ensures that side information pathways remain effective, which is critical for cold-start users who lack interaction history.

Global Side Information Fusion Module

The global side information fusion module provides meta-level supervision for the weight generation process. While the local module learns from training interactions, the global module evaluates which information source yields better recommendation performance on validation data, providing an additional learning signal.

This component addresses the distribution mismatch between training and evaluation weights optimized solely on training data may not generalize well to held-out users and items. The global module provides a corrective signal that aligns the fusion strategy with actual recommendation performance.

For each user u_i in the validation set, we compute recommendation performance using behavioral embedding only and side information only:

$$\hat{y}_{u_i,v_j}^{\text{behavior}} = \mathbf{X}_{u_i} \cdot \mathbf{X}_{v_j} \tag{3.19}$$

$$\hat{y}_{u_i,v_j}^{\text{info}} = \mathbf{X}_{u_i}^{\text{info}} \cdot \mathbf{X}_{v_j} \tag{3.20}$$

A binary label is assigned-based on which source performs better:

$$\text{label}_{u_i} = \begin{cases} 0 & \text{if Metric}(\hat{y}^{\text{behavior}}) > \text{Metric}(\hat{y}^{\text{info}}) \\ 1 & \text{otherwise} \end{cases} \tag{3.21}$$

The weight generated module is then trained with a cross entropy loss to align the

learned weights with the performance-based labels:

$$\mathcal{L}_{\text{global}} = -\sum_{u_i} \left[(1 - \text{label}_{u_i}) \cdot \log(1 - \alpha_{u_i}) + \text{label}_{u_i} \cdot \log(\alpha_{u_i}) \right] \tag{3.22}$$

During global training, all parameters except the weight generated module are frozen, enabling efficient fine tuning without disrupting the learned GNN representations.

Training Objective

The complete training objective combines the recommendation loss with the global supervision:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda \cdot \mathcal{L}_{\text{global}} \tag{3.23}$$

where $\mathcal{L}_{\text{rec}}$ is the standard cross entropy loss for recommendation:

$$\mathcal{L}_{\text{rec}} = -\sum_{(u_i, v_j, v_k)} \left[\log \hat{y}_{u_i, v_j} + \log\left(1 - \hat{y}_{u_i, v_k}\right) \right] \tag{3.24}$$

with v_j being a positive (interacted) item and v_k being a negative (non-interacted) item.

Cold-Start Handling

GIFT4Rec addresses the cold-start problem through three complementary mechanisms that collectively ensure meaningful recommendations can be generated even when a user has no prior interaction history.

First, the GNN Interaction Module initialises the behavioural embedding of a cold-start user as the mean of all warm-start user embeddings:

$$\mathbf{X}_{u_i}^{\text{cold}} = \frac{1}{|\mathcal{U}_{\text{warm}}|} \sum_{u_j \in \mathcal{U}_{\text{warm}}} \mathbf{X}_{u_j} \tag{3.25}$$

This mean embedding carries no user-specific collaborative signal and is therefore a deliberately weak representation. Its low informativeness causes the Weight Generated module to produce a small fusion coefficient $\alpha_{u_i} \approx 0$, which automatically shifts the final representation almost entirely toward the side information $\mathbf{X}_{u_i}^{\text{info}}$:

$$\mathbf{X}_{u_i}^{\text{final}} \approx \mathbf{X}_{u_i}^{\text{info}} \quad \text{when } \alpha_{u_i} \approx 0 \tag{3.26}$$

In this way, auxiliary attributes such as user demographics and item features serve as the primary source of personalisation for new users, bypassing the absence of interaction

data entirely.

Second, the Attention DropoutNet (ADN) mechanism applied during training is designed specifically to prepare the model for cold-start inference. By stochastically masking the behavioural embedding with probability p_{drop} during training, ADN forces the model to learn to produce accurate recommendations using side information alone:

$$\mathbf{X}_{u_i}^{\text{final}} = \mathbf{X}_{u_i}^{\text{info}} \quad \text{with probability } p_{\text{drop}} \tag{3.27}$$

Without this training-time simulation, the model would never encounter the cold-start condition during training and would therefore fail to generalise to it at inference time. ADN closes this train-test gap by explicitly training the side-information pathway to be independently sufficient for recommendation.

Third, the Global Side Information Fusion (GSIF) module reinforces cold-start robustness at the level of the fusion strategy itself. By evaluating, on held-out validation users, whether side information or behavioural embeddings yield better performance and using this signal to supervise the Weight Generated module, GSIF ensures that the learned fusion weights correctly down-weight unreliable behavioural embeddings not only for users seen during training, but also for the cold-start distribution encountered at deployment.

Together, these three mechanisms form a coherent cold-start strategy: the mean embedding initialisation provides a well-defined fallback for the GNN encoder; ADN trains the side-information pathway to function independently; and GSIF calibrates the fusion coefficient to generalise to unseen users. As user interactions accumulate over time, α_{u_i} naturally increases as the behavioural embedding becomes more discriminative, allowing the model to progressively transition from side-information-driven to interaction-driven personalisation without any retraining.

Fairness and Demographic Bias

Since GIFT4Rec fuses user demographics age, gender, and location as side information, it is worth examining how the design treats these sensitive attributes with respect to fairness and bias. A key property of the framework is that demographic attributes never act as a fixed or dominant basis for recommendation. Rather than combining behavioural and auxiliary signals with a static rule, the attention-based Weight-Generated module produces a user-specific fusion weight α that controls the relative contribution of the two sources for each user individually.

Because α is learned per user, the influence of demographic information is adaptive

rather than constant. It is largest only when behavioral evidence is scarce the cold-start regime, where some signal is genuinely needed to make a reasonable recommendation and it decreases automatically as interactions accumulate, at which point α shifts the representation toward the behavioral component. In effect, the model leans on demographics only as long as it has nothing better, and stops doing so as soon as personalized behavior becomes available. This adaptive balancing structurally limits the model's reliance on sensitive attributes and, with it, the risk of amplifying demographic bias or producing systematically different treatment across demographic groups. Fairness is therefore promoted as an intrinsic consequence of the fusion design, not as a post-hoc correction applied on top of the model.

3.2.3 Experimental Settings and Results

A. Experimental Settings

a) Dataset and User Categorization We conduct experiments on the MovieLens-1M dataset containing 6,040 users, 3,706 items, and 1,000,209 interactions. Users are categorized into three groups-based on interaction frequency: Cold ($\leq$20 interactions), Warm (21-50 interactions), and Active ($>$50 interactions). We use temporal split with 80% training and 20% testing data.

Table 3.5: GIFT4Rec -Experimental Configuration

Component	Specification
GPU	NVIDIA A100-SXM4-40GB
Framework	PyTorch 2.1.0, CUDA 12.2
Platform	Google Colab Pro+

b) Evaluation Metrics We evaluate recommendation quality using two primary metrics:

- Recall@K: Measures the fraction of relevant items that appear in the top-K recommendations.
- NDCG@K: Normalized Discounted Cumulative Gain, accounts for ranking position by giving higher weights to items ranked at top positions.

All experiments are evaluated at K=30 and reported over 5 random seeds.

c) Baseline Methods We compare GIFT4Rec against state-of-the-art methods from four categories:

- Auxiliary Information & Side Fusion Methods: KGAT, GAT, KGAT DropoutNet, LINKX
- Basic Graph Neural Network Methods: NGCF, LightGCN
- Self-supervised Learning Methods: SSL4Rec
- Clustering-based Cold-start Methods: EfficientRec (Chapter 2)

B. Experimental Results

a) Overall Performance

Table 3.6: GIFT4Rec – Overall Performance Comparison (All Users) @30, reported as mean ± standard deviation over 5 random seeds.

Model	Recall@30	NDCG@30	Category
GIFT4Rec	$\mathbf{0.2162 \pm 0.0001}^{\dagger}$	$\mathbf{0.1263 \pm 0.0001}^{\dagger}$	Proposed
KGAT	0.1793 ± 0.0008	0.1067 ± 0.0007	Aux+KG
LINKX	0.1928 ± 0.0022	0.1160 ± 0.0012	Aux+Graph
GAT	0.1639 ± 0.0001	0.0946 ± 0.0001	Aux+Attn
KGAT DropoutNet	0.0908 ± 0.0003	0.0359 ± 0.0008	Aux+KG
NGCF	0.1958 ± 0.0013	$\underline{0.1178 \pm 0.0007}$	GNN
LightGCN	0.1636 ± 0.0003	0.0945 ± 0.0001	GNN
SSL4Rec	0.1644 ± 0.0001	0.0950 ± 0.0012	SSL
EfficientRec (Ch.2)	$\underline{0.1994 \pm 0.0028}$	0.1174 ± 0.0027	Proposed

Bold = best, underline = second best.

Table 3.6 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 improves on 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. Each value is the mean ±

standard deviation over 5 random seeds. † denotes a statistically significant improvement of GIFT4Rec over the second-best baseline under a two-sided Welch's t-test ($p < 0.001$ for both metrics). GIFT4Rec significantly outperforms all baselines on both Recall@30 (+8.4% over the second-best) and NDCG@30 (+7.2% over the second-best.

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.

b) Performance on Cold, Warm, and Active Users

Table 3.7: GIFT4Rec - Performance Comparison on Cold, Warm, and Active Users @30

Model	**Cold Users**		**Warm Users**		**Active Users**	
	R@30	N@30	R@30	N@30	R@30	N@30
GIFT4Rec	**0.2511**	**0.1152**	**0.2148**	**0.1335**	<u>0.1663</u>	0.1257
KGAT	0.1690	0.0805	0.1507	0.0948	0.1566	0.1206
LINKX	0.1866	0.0867	0.1717	0.1070	0.1651	**0.1291**
GAT	0.1418	0.0609	0.1261	0.0769	0.1494	0.1137
NGCF	0.2008	0.0942	0.1747	0.1104	0.1647	<u>0.1285</u>
LightGCN	0.1411	0.0605	0.1256	0.0766	0.1494	0.1137
SSL4Rec	0.1444	0.0618	0.1262	0.0772	0.1495	0.1139
EfficientRec (Ch.2)	<u>0.2085</u>	<u>0.0910</u>	<u>0.1840</u>	<u>0.1145</u>	**0.1689**	0.1264

R@30 = Recall@30, N@30 = NDCG@30. **Bold** = Best, <u>Underline</u> = Second Best.

GIFT4Rec achieves the best Recall@30 for Cold and Warm users:

Cold Users: GIFT4Rec achieves Recall@30 of 0.2511, ahead of the second-best method (EfficientRec: 0.2085) by +20.4% (relative difference). This demonstrates that attention-based side information fusion effectively addresses cold-start by leveraging auxiliary user features when interaction data is sparse.

Warm Users: GIFT4Rec achieves Recall@30 of 0.2148, outperforming EfficientRec (0.1840) by +16.7% (relative difference). The largest improvement among all scenarios indicates that the model effectively combines moderate interaction history with

side information.

Active Users: EfficientRec achieves the best Recall@30 (0.1689), while GIFT4Rec (0.1663) shows competitive performance. The smaller performance gaps among top methods for active users confirm that side information fusion provides greater benefits when interaction data is limited.

c) Component Contribution Analysis

To understand the contribution of each component in GIFT4Rec, we conduct comprehensive ablation studies by systematically removing or replacing key components. The central argument of GIFT4Rec is that cold-start recommendation benefits from adaptive fusion of behavioral and auxiliary signals, where the fusion weight is personalized per user rather than fixed globally. The ablation study is designed to test this argument at three levels of specificity: Is auxiliary side information itself beneficial, independent of how it is fused? Does learned adaptive fusion improve over static fusion strategies? Does the depth of the weight generation network matter, i.e., does non-linear weight learning capture relationships that linear weighting cannot?

Table 3.8 presents the performance of each configuration.

Table 3.8: Component Ablation Study on GIFT4Rec

Configuration	**Recall@30**	**ΔRecall**	**NDCG@30**	**ΔNDCG**
GIFT4Rec-Full	0.2162$\pm$0.003		0.1263$\pm$0.003	
w/o GNN Module	0.1816$\pm$0.004	-16.0%	0.1073$\pm$0.004	-15.0%
w/o Side Information	0.1892$\pm$0.003	-12.5%	0.1063$\pm$0.003	-15.8%
w/o Weight Generator	0.1989$\pm$0.003	-8.0%	0.1142$\pm$0.003	-9.6%
w/o Deep WG	0.2021$\pm$0.004	-6.5%	0.1156$\pm$0.003	-8.5%
Fixed α=0.5	0.2038$\pm$0.003	-5.7%	0.1212$\pm$0.003	-4.0%

Note: ΔRecall and ΔNDCG represent relative difference compared to full model.

The results confirm the argument in a layered manner. Removing side information entirely causes -12.5% Recall, establishing that auxiliary data provides substantial value beyond interaction signals alone. However, how this information is fused matters significantly: replacing the learned Weight Generated module with a fixed weight degrades performance by -5.7% Recall, while removing the Weight Generated module altogether causes -8.0%. This gap demonstrates that adaptive, user-specific fusion is essential not all users benefit equally from side information, and a static rule cannot

capture this variation. Furthermore, replacing the deep weight generator with a shallow (linear) version results in -6.5% versus -8.0% for full removal, indicating that non-linear weight learning captures complex relationships between information sources that linear weighting misses. Together, these results establish that GIFT4Rec's contribution lies not in the individual components (GNN, side information) but in the adaptive fusion mechanism that controls their interaction.

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 [119] 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.

A. Related Methodologies

Graph Neural Network and Graph Contrastive Learning

This section employs Graph Neural Networks and Graph Contrastive Learning as core techniques for the proposed models. Since these approaches (e.g., LightGCN, SimGCL) have been introduced in the section 3.1 of this chapter so they are not repeated here.

Learnable Masks in Deep Learning

The concept of learnable masks has proven highly effective across various deep

learning domains for identifying and emphasizing task relevant features while suppressing less informative components. In natural language processing, differentiable masks have been successfully applied to extract informative text spans and improve model interpretability. The Dynamic Mask Attention Network (DMAN) [95] introduces learnable mask matrices that adaptively model localness in sequence learning, demonstrating that learned masks can effectively capture task-specific importance patterns.

In the context of neural networks, masking mechanisms serve multiple purposes: they enable selective attention to important features, provide regularization against overfitting, and facilitate efficient computation by focusing resources on relevant dimensions. The success of masking techniques in diverse applications motivates their adaptation to graph-based recommendation, where the challenge of identifying task relevant embedding dimensions is very noticeable due to the high dimensional nature of learned representations and the sparsity of supervision signals.

B. Limitations of Existing Approaches

Despite the significant progress achieved by existing graph-based contrastive learning methods for recommendation, several critical limitations remain that constrain their effectiveness and practical applicability. This section identifies four key research Limitations that motivate the development of MaskSimGCL.

Limitation 1: Uniform Treatment of Embedding Dimensions

Existing contrastive learning methods for recommendation, including SGL, SimGCL, and XSimGCL, treat all dimensions of node embeddings uniformly during both augmentation and learning. When adding noise for contrastive view generation, the same perturbation magnitude is applied across all embedding dimensions regardless of their relative importance for the recommendation task. This uniform treatment is weak in accounting for the heterogeneous informativeness of different embedding dimensions.

In practice, learned representations typically exhibit varying levels of task relevance across dimensions. Some dimensions may encode critical preference signals that are essential for accurate recommendation, while others may capture noise or less discriminative patterns. Applying uniform perturbations risks either reducing contrastive diversity or destroying essential preference information. This limitation suggests the need for adaptive mechanisms that can identify and differentially treat embedding dimensions-based on their importance.

Limitation 2: Overfitting in Sparse Data Environments

Graph-based recommender systems are inherently prone to overfitting when trained

on sparse interaction data. The high dimensionality of learned embeddings combined with the limited number of observed interactions creates conditions where models can easily memorize training patterns without learning generalizable representations. While contrastive learning provides implicit regularization through the uniformity objective, existing methods lack explicit mechanisms to prevent the model from fitting to noise in the data.

Standard regularization techniques such as weight decay and dropout provide general purpose constraints but do not specifically address the unique challenges of graph-based collaborative filtering. The absence of targeted regularization mechanisms that can identify and suppress less informative embedding parameters leaves existing methods vulnerable to overfitting, particularly in scenarios with extreme data sparsity.

Limitation 3: Suboptimal Contrastive View Generation

The quality of contrastive learning fundamentally depends on the properties of generated views. Effective contrastive views should be sufficiently diverse to provide meaningful learning signals while remaining semantically consistent to preserve essential information. Existing noise-based augmentation strategies, while computationally efficient, generate views through random perturbations that do not consider the semantic structure of the embedding space.

Random uniform noise treats the embedding space as isotropic, ignoring the fact that different dimensions may have vastly different semantic significance. This approach may inadvertently introduce excessive noise to critical dimensions that encode core preference patterns, degrading the quality of learned representations. Conversely, less informative dimensions may receive insufficient perturbation, limiting the diversity of contrastive views.

Limitation 4: Lack of Integration of Masking and Contrastive Learning

While learnable masks have demonstrated effectiveness in various deep learning applications, their integration with graph-based contrastive learning for recommendation remains unexplored. Existing work on masking techniques in NLP and computer vision has not been adapted to the unique requirements of collaborative filtering, where the challenge is to identify task relevant dimensions in the context of sparse user item interactions and graph structured data.

The combination of learnable masking and contrastive learning offers convincing opportunities. Masks can inform the contrastive view generation process by indicating which dimensions are important and should be preserved versus which dimensions can tolerate larger perturbations. Conversely, the contrastive learning objective can provide training signals for learning effective masks. This bidirectional relationship suggests that

a unified framework integrating masking and contrastive learning could achieve benefits that exceed the sum of its parts.

Table 3.9: Summary of Research Limitations and MaskSimGCL Solutions

Research Limitation	**Limitation**	**MaskSimGCL Solution**
Uniform Dimension Treatment	Same perturbation applied to all embedding dimensions regardless of importance	Learnable masks adaptively weight dimension importance
Overfitting Risk	No explicit regularization mechanism for sparse data	Masking serves as implicit regularizer by focusing on informative parameters
Suboptimal Views	Random noise ignores semantic structure of embedding space	Informed perturbations-based on learned crucial scores
Missing Integration	Masking and contrastive learning built separately	Unified framework jointly optimizes both objectives

Comparative Positioning of MaskSimGCL Against Related Methods

Table 3.10 provides a structured comparison of MaskSimGCL against the most closely related graph contrastive learning methods across five design dimensions that correspond directly to the four limitations identified in Section 3.3.1.

Table 3.10: Design Comparison of MaskSimGCL Against Related Methods

Characteristic	**SGL [111]**	**SimGCL [119]**	**XSimGCL [121]**	**LightGCL [10]**	**MaskSimGCL (Ours)**
Graph contrastive learning	✓	✓	✓	✓	✓
Learnable dimension mask	✗	✗	✗	✗	✓
Dimension-aware perturbation	✗	✗	✗	✗	✓
Implicit regularization via mask	✗	✗	✗	✗	✓
Augmentation strategy	Stochastic drop	Uniform noise	Cross-layer noise	SVD structure	**Mask-informed noise**

All methods use LightGCN as the GNN backbone. *Stochastic drop*: random edge/node dropout; *Uniform noise*: same perturbation magnitude across all dimensions; *Cross-layer noise*: contrasts representations at different GNN layers; *SVD structure*: leverages singular value decomposition of the interaction graph.

The comparison reveals that all four baseline methods share the same critical weakness: uniform or structure-only augmentation that treats every embedding dimension

identically. SGL [111] perturbs the graph topology through random edge and node dropout, producing coarse augmentations that lose fine-grained dimension-level signals. SimGCL [119] improves efficiency by replacing graph dropout with additive uniform noise, but the noise magnitude remains identical across all dimensions. XSimGCL [121] refines this by contrasting representations across GNN layers, yet still applies the same noise to each dimension regardless of its semantic importance. LightGCL [10] introduces structural awareness through SVD-based augmentation but provides no mechanism to prioritize or suppress individual embedding dimensions.

MaskSimGCL introduces a fundamentally different design principle: learnable masks explicitly assign an importance score to each embedding dimension, enabling mask-informed perturbation that applies stronger noise to less informative dimensions while preserving critical ones. This bidirectional relationship masks guide view generation, and the contrastive objective refines mask quality is unique among all existing methods.

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.

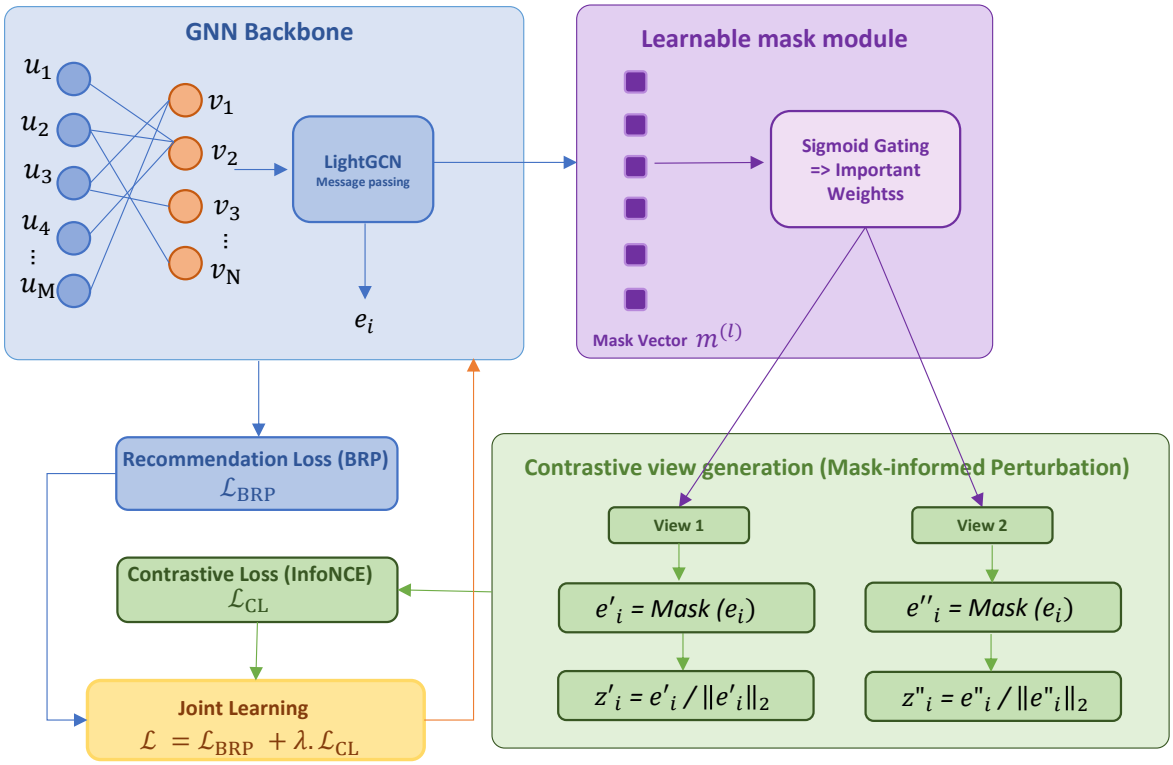

Figure 3.2: MaskSimGCL overall architecture

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.

a. Problem Formulation and Notation

Consider a recommendation system operating over a bipartite graph $\mathcal{G} = (\mathcal{U}, \mathcal{V}, \mathcal{E})$, where $\mathcal{U} = \{u_1, u_2, \ldots, u_M\}$ denotes the set of M users, $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ denotes the set of N items, and $\mathcal{E}$ represents the observed user item interactions. An edge $(u_i, v_j) \in \mathcal{E}$ indicates that user u_i has interacted with item v_j.

The interaction data can be represented as a binary adjacency matrix $\mathbf{R} \in \mathbb{R}^{M \times N}$, where $r_{ij} = 1$ if user u_i has interacted with item v_j, and $r_{ij} = 0$ otherwise. The neighborhood of a node is denoted as $\mathcal{N}(\cdot)$. For user u_i, the neighborhood $\mathcal{N}(u_i)$ consists of all items that u_i has interacted with. Similarly, for item v_j, the neighborhood $\mathcal{N}(v_j)$ consists of all users who have interacted with v_j.

Each user and item is associated with a d-dimensional embedding vector. The user embedding matrix is denoted as $\mathbf{E}^{(0)} \in \mathbb{R}^{M \times d}$, where the i-th row $\boldsymbol{e}_{u_i}^{(0)}$ represents the initial embedding of user u_i. Similarly, the item embedding matrix is $\mathbf{E}^{(0)} \in \mathbb{R}^{N \times d}$, where $\boldsymbol{e}_{v_j}^{(0)}$ represents the initial embedding of item v_j. These embeddings are randomly initialized and learned through end to end training.

The learnable mask vectors are denoted as $\boldsymbol{m}^{(\ell)} \in \mathbb{R}^d$ for layer ℓ, where $\ell = 0, 1, \ldots, L$ and L is the total number of GNN layers. Each mask vector has the same dimensionality as the node embeddings, enabling element-wise importance weighting. The normalized adjacency matrix of the bipartite graph is denoted as $\tilde{\mathbf{A}} \in \mathbb{R}^{(M+N) \times (M+N)}$, which incorporates symmetric normalization to ensure stable message propagation.

b. Graph Neural Network Backbone

The graph neural network backbone is the main part of the MaskSimGCL and is responsible for learning user and item representations through message passing operations on the user item bipartite graph. This component follows the LightGCN design architecture, which demonstrates that simplified graph convolutions without feature transformation and nonlinear activation are not only sufficient but actually preferable for collaborative filtering tasks.

The key insight motivating this design choice is that the recommendation task fundamentally differs from general graph learning tasks. In collaborative filtering, the primary goal is to propagate and aggregate collaborative signals across the graph structure, rather than to perform complex feature transformations. The user and item embeddings serve as the only input features, and their identity information is already captured in the

initial embeddings. Therefore, the heavy transformation operations inherited from general GCNs are unnecessary and may even introduce noise that degrades recommendation performance.

Message Propagation Mechanism

The computation proceeds through L sequential propagation layers, where each layer aggregates information from neighboring nodes. At layer ℓ, the representation of user u_i is updated by aggregating the representations of items in its neighborhood:

$$\boldsymbol{h}_{u_i}^{(\ell+1)} = \sum_{v_j \in \mathcal{N}(u_i)} \frac{1}{\sqrt{|\mathcal{N}(u_i)|} \cdot \sqrt{|\mathcal{N}(v_j)|}} \cdot \boldsymbol{h}_{v_j}^{(\ell)} \tag{3.28}$$

Similarly, the representation of item v_j is updated as:

$$\boldsymbol{h}_{v_j}^{(\ell+1)} = \sum_{u_i \in \mathcal{N}(v_j)} \frac{1}{\sqrt{|\mathcal{N}(v_j)|} \cdot \sqrt{|\mathcal{N}(u_i)|}} \cdot \boldsymbol{h}_{u_i}^{(\ell)} \tag{3.29}$$

The symmetric normalization factor $\frac{1}{\sqrt{|\mathcal{N}(u_i)|} \cdot \sqrt{|\mathcal{N}(v_j)|}}$ serves a critical purpose in the propagation process. This normalization prevents nodes with many connections from dominating the aggregated representation and ensures that the message magnitudes remain stable across layers. Without this normalization, the embeddings of high degree nodes would grow as propagation depth increases, leading to numerical instability and degraded model performance.

The propagation can be expressed in compact matrix form as:

$$\mathbf{E}^{(\ell+1)} = \tilde{\mathbf{A}} \cdot \mathbf{E}^{(\ell)} \tag{3.30}$$

where $\mathbf{E}^{(\ell)} \in \mathbb{R}^{(M+N) \times d}$ is the concatenated embedding matrix of all users and items at layer ℓ, and $\tilde{\mathbf{A}}$ is the symmetrically normalized adjacency matrix.

Multi-Scale Representation Aggregation

After L layers of propagation, the final user and item embeddings are obtained by averaging the representations across all layers:

$$\boldsymbol{e}_{u_i} = \frac{1}{L+1} \sum_{\ell=0}^{L} \boldsymbol{h}_{u_i}^{(\ell)}, \quad \boldsymbol{e}_{v_j} = \frac{1}{L+1} \sum_{\ell=0}^{L} \boldsymbol{h}_{v_j}^{(\ell)} \tag{3.31}$$

This layer combination strategy captures collaborative signals at different semantic levels. The representation at layer 0 ($\boldsymbol{h}^{(0)}$) contains only the node's own information

encoded in its initial embedding. Layer 1 representations ($\boldsymbol{h}^{(1)}$) incorporate first-order neighborhood information, representing direct user item interactions. Deeper layers capture progressively higher order collaborative patterns: layer 2 encodes second order relationships (user item-user paths), and so forth. By averaging across all layers, the final representation integrates multi scale structural information, providing a comprehensive encoding of the node's position and relationships within the collaborative graph.

c. Learnable Mask Module

The Learnable Mask Module represents the core innovation of MaskSimGCL, introducing trainable mask vectors that adaptively weight the importance of different embedding dimensions. This mechanism addresses a fundamental limitation of existing graph-based contrastive learning methods, which treat all embedding dimensions uniformly regardless of their contribution to the recommendation task.

The key insight motivating this design is that learned representations typically show different levels of importance across dimensions. Some dimensions may encode critical preference signals essential for accurate recommendation, while others may capture noise or less separated patterns. By introducing learnable masks, the model can explicitly identify and emphasize the most informative dimensions while suppressing irrelevant or noisy components.

Mask-Filtered Embedding Computation

At each layer ℓ of the GNN, a learnable mask vector $\boldsymbol{m}^{(\ell)} \in \mathbb{R}^d$ is applied to filter the node embeddings. The mask is first passed through a sigmoid activation function to produce importance weights in the range $(0, 1)$:

$$\boldsymbol{w}^{(\ell)} = \sigma(\boldsymbol{m}^{(\ell)}) \tag{3.32}$$

where $\sigma(\cdot)$ denotes the sigmoid activation function $\sigma(x) = \frac{1}{1+e^{-x}}$. The choice of sigmoid activation ensures that the importance weights remain bounded while allowing gradients to flow through during backpropagation.

The mask filtered representation of node i at layer ℓ is computed through element-wise multiplication:

$$\boldsymbol{e}_{i,\ell}^{\text{new}} = \boldsymbol{e}_{i,\ell} \odot \boldsymbol{w}^{(\ell)} = \boldsymbol{e}_{i,\ell} \odot \sigma(\boldsymbol{m}^{(\ell)}) \tag{3.33}$$

where $\odot$ denotes element-wise product. This formulation reflects several important design principles. First, dimensions with high importance weights (close to 1) are

preserved with minimal attenuation, ensuring that critical preference signals are maintained. Second, dimensions with low importance weights (close to 0) are effectively suppressed, reducing their contribution to the final representation. Third, the element-wise multiplication maintains the dimensionality of the original embeddings, enabling seamless integration with downstream components.

Regularization Through Importance Filtering

The learnable mask module provides implicit regularization through a dimension-selection mechanism that is specifically tailored to the sparse interaction data characteristic of recommendation systems. Unlike generic regularization techniques such as weight decay or dropout which apply uniform constraints to all parameters the masking mechanism learns which dimensions are informative and suppresses those that are not, acting as a form of learned feature selection.

Concretely, each embedding dimension k is multiplied by the sigmoid-activated mask weight $\sigma(m_k^{(\ell)}) \in (0, 1)$. Dimensions with high importance weights (close to 1) are preserved with minimal attenuation, ensuring that critical preference signals are retained; dimensions with low weights (close to 0) are effectively suppressed, preventing the model from fitting noise patterns in sparse interaction data. The gradient through the masking operation makes this selection process end-to-end trainable:

$$\frac{\partial \mathcal{L}}{\partial m^{(\ell)}} = \frac{\partial \mathcal{L}}{\partial \mathbf{e}_{i,\ell}^{\text{new}}} \odot \mathbf{e}_{i,\ell} \odot \sigma(m^{(\ell)}) \odot (1 - \sigma(m^{(\ell)})) \tag{3.34}$$

This gradient formulation reveals a self-reinforcing dynamic in the importance filtering mechanism. The update to each mask logit is modulated by the sigmoid derivative factor $\sigma\big(m^{(\ell)}\big)\big(1 - \sigma\big(m^{(\ell)}\big)\big)$, which is maximal when the gate is uncertain ($\sigma(m^{(\ell)}) = 0.5$) and vanishes as the gate saturates toward 0 or 1. Consequently, a dimension that consistently helps reduce the loss is driven toward a high mask weight, and an uninformative one toward zero, but the magnitude of these updates shrinks as the weights approach either extreme. Rather than an unbounded amplification, the masks therefore follow a bounded, logistic trajectory with stable fixed points at 0 and 1: the $\sigma(1 - \sigma)$ factor acts as an implicit stabilizer that prevents runaway growth while still yielding a near-binary separation between informative and uninformative dimensions.

d. Contrastive View Generation Module

The contrastive view generation module is responsible for creating augmented representations that serve as positive pairs in the contrastive learning framework. This component builds upon the noise-based augmentation strategy introduced by SimGCL while

incorporating mask informed perturbations that improve the quality of generated views.

Background: SimGCL's noise-based Augmentation

Before describing the MaskSimGCL approach, we first review the contrastive view generation mechanism in SimGCL. Traditional graph-based contrastive learning methods, such as SGL, generate contrastive views through graph augmentations including node dropout, edge dropout, and random walk sampling. However, SimGCL demonstrates that the performance improvements attributed to graph-based contrastive learning primarily stem from the uniformity regularization effect of the InfoNCE loss rather than the specific augmentation strategies employed.

Based on this insight, SimGCL proposes a simplified yet effective alternative: instead of perturbing the graph structure, it directly adds random uniform noise to node embeddings. For each node i, the augmented embedding is computed as:

$$\boldsymbol{e}_i' = \boldsymbol{e}_i + \boldsymbol{\Delta}_i, \quad \|\boldsymbol{\Delta}_i\|_2 = \epsilon \tag{3.35}$$

where $\boldsymbol{\Delta}_i$ is the noise vector and ϵ controls the perturbation magnitude. The noise vector is constructed by first sampling from a uniform distribution $\mathcal{U}(0,1)$ and then scaling to achieve the desired L2 norm:

$$\boldsymbol{\Delta}_i = \text{sign}(\boldsymbol{e}_i) \odot \boldsymbol{\Delta}_i', \quad \boldsymbol{\Delta}_i' \sim \mathcal{U}(0,1) \tag{3.36}$$

This noise-based strategy eliminates the computational overhead of repeatedly constructing and processing augmented graphs while maintaining the uniformity regularization that drives performance improvements.

Mask-Informed Perturbation Strategy

While SimGCL's uniform noise addition is computationally efficient, it treats all embedding dimensions equally during augmentation. This uniform treatment fails to account for the heterogeneous informativeness of different dimensions. MaskSimGCL addresses this limitation by leveraging the learned importance scores from the mask module to guide the perturbation process.

The core idea is to apply larger perturbations to less important dimensions while preserving important dimensions with smaller noise. This differential treatment generates contrastive views that maintain semantic consistency in the critical dimensions while introducing diversity in the less informative dimensions. The mask informed noise vector is computed as:

$$\boldsymbol{e}'_{i,\ell} = \boldsymbol{e}_{i,\ell} + f(\boldsymbol{\Delta}_{i,\ell}) \tag{3.37}$$

where $f(\cdot)$ is a masking function that amplifies perturbations on less important dimensions. Specifically, the function amplifies the value at position k of $\boldsymbol{\Delta}_{i,\ell}$ by a factor of α if the corresponding mask weight $w_k^{(\ell)}$ is smaller than a predefined threshold β:

$$f(\boldsymbol{\Delta}_{i,\ell})_k = \begin{cases} \alpha \cdot \Delta_{i,\ell,k} & \text{if } \sigma(\boldsymbol{m}_k^{(\ell)}) < \beta \\ \Delta_{i,\ell,k} & \text{otherwise} \end{cases} \tag{3.38}$$

where $\alpha > 1$ is the amplification factor and $\beta \in (0, 1)$ is the importance threshold. This mechanism ensures that dimensions identified as less important receive larger perturbations, encouraging the contrastive learning to focus on invariances in the important dimensions.

Dual View Construction

For contrastive learning, two augmented views are generated for each node using independent noise samples. Let $\boldsymbol{e}'_i$ and $\boldsymbol{e}''_i$ denote the two augmented representations of node i. These representations are L2-normalized before computing the contrastive loss:

$$\boldsymbol{z}'_i = \frac{\boldsymbol{e}'_i}{\|\boldsymbol{e}'_i\|_2}, \quad \boldsymbol{z}''_i = \frac{\boldsymbol{e}''_i}{\|\boldsymbol{e}''_i\|_2} \tag{3.39}$$

The normalization projects the embeddings onto a unit hypersphere, ensuring that the cosine similarity used in the contrastive loss is equivalent to the dot product of the normalized representations.

e. Joint Optimization Framework

The joint optimization framework combines the supervised recommendation objective with the self-supervised contrastive learning objective in a multitask learning formulation. This integrated approach enables the model to simultaneously optimize for accurate preference prediction and robust representation learning.

Recommendation Loss

The recommendation task is optimized using the Bayesian Personalized Ranking (BPR) loss [76], which is specifically designed for implicit feedback scenarios. The BPR loss encourages the model to rank positive (interacted) items higher than negative (non interacted) items for each user.

The predicted preference score between user u_i and item v_j is computed as the inner product of their final embeddings:

$$\hat{y}_{u_i,v_j} = \boldsymbol{e}_{u_i}^{\top} \cdot \boldsymbol{e}_{v_j} \tag{3.40}$$

The BPR loss is then formulated as:

$$\mathcal{L}_{\text{BPR}} = - \sum_{(u_i,v_j,v_k)\in\mathcal{O}} \log \sigma(\hat{y}_{u_i,v_j} - \hat{y}_{u_i,v_k}) \tag{3.41}$$

where $\mathcal{O} = \{(u_i, v_j, v_k) \mid (u_i, v_j) \in \mathcal{E}, (u_i, v_k) \notin \mathcal{E}\}$ is the set of training triplets, with v_j being a positive item and v_k being a randomly sampled negative item. The sigmoid function ensures that the loss is bounded and provides smooth gradients for optimization.

Contrastive Loss

The contrastive learning objective employs the InfoNCE loss [80], which maximizes the agreement between the two augmented views of the same node while minimizing the agreement with views from different nodes:

$$\mathcal{L}_{\text{CL}} = \sum_{i\in\mathcal{B}} -\log \frac{\exp\left(\boldsymbol{z}_i'^{\top} \boldsymbol{z}_i''/\tau\right)}{\sum_{j\in\mathcal{B}} \exp\left(\boldsymbol{z}_i'^{\top} \boldsymbol{z}_j''/\tau\right)} \tag{3.42}$$

where $\mathcal{B}$ is a sampled batch of nodes, $\tau > 0$ is the temperature parameter that controls the sharpness of the distribution. Lower temperatures make the contrastive learning more sensitive to differences between representations.

The InfoNCE loss serves two complementary purposes. First, it encourages the two augmented views of the same node (positive pairs) to have high similarity, promoting invariance to the applied perturbations. Second, it pushes apart the representations of different nodes (negative pairs), leading to a more uniform distribution of embeddings in the representation space. This uniformity regularization helps mitigate the popularity bias that often affects collaborative filtering, where popular items dominate the learned representations.

Combined Training Objective

The final training objective of MaskSimGCL integrates two terms with distinct and complementary roles:

$$\mathcal{L} = \mathcal{L}_{\text{BPR}} + \lambda \mathcal{L}_{\text{CL}} \tag{3.43}$$

Term 1: $\mathcal{L}_{\text{BPR}}$ recommendation accuracy. The Bayesian Personalised Ranking loss is the primary supervised objective. It optimises the model to rank positive (interacted) items above negative (non-interacted) items for each user, directly targeting the ranking quality measured by Recall@K and NDCG@K:

$$\mathcal{L}_{\text{BPR}} = -\sum_{(u_i, v_j^+, v_k^-) \in \mathcal{O}} \log \sigma\left(\hat{y}_{u_i, v_j^+} - \hat{y}_{u_i, v_k^-}\right) \tag{3.44}$$

Without this term, the model would have no direct signal about which items a user actually prefers, making accurate ranking impossible.

Term 2: $\lambda\, \mathcal{L}_{\text{CL}}$ representation robustness via contrastive learning. The InfoNCE contrastive loss serves two functions simultaneously. First, it maximises agreement between the two mask-informed augmented views of the same node (positive pairs), encouraging the model to learn representations that are invariant to which specific dimensions are perturbed:

$$\mathcal{L}_{\text{CL}} = \sum_{i \in \mathcal{B}} -\log \frac{\exp\left(\mathbf{z}_i'^{\top} \mathbf{z}_i'' / \tau\right)}{\sum_{j \in \mathcal{B}} \exp\left(\mathbf{z}_i'^{\top} \mathbf{z}_j'' / \tau\right)} \tag{3.45}$$

Second, it pushes apart representations of different nodes (negative pairs), inducing a uniformity effect in the embedding space that counters popularity bias a well-known failure mode in collaborative filtering where embeddings of popular items cluster together and crowd out long-tail items. The coefficient λ balances the contribution of this self-supervised signal against the supervised BPR objective; too large a λ can cause the contrastive term to dominate and degrade ranking accuracy, while too small a λ provides insufficient regularization against sparsity.

3.3.3 Experimental Setting and Results

A. Experimental Settings

a. Dataset and User Categorization

We conduct experiments on the MovieLens-1M dataset containing 6,040 users, 3,706 items, and 1,000,209 interactions. Users are categorized into three groups-based on interaction frequency: Cold ($\leq$20 interactions), Warm (21-50 interactions), and Active ($>$50 interactions). We use temporal split with 80% training and 20% testing data.

Table 3.11: Experimental Configuration

Component	Specification
GPU	NVIDIA A100-SXM4-40GB
Framework	PyTorch 2.1.0, CUDA 12.2
Platform	Google Colab Pro+

b. Evaluation Metrics

We evaluate recommendation quality using two primary metrics: Recall@K (measures the fraction of relevant items in top-K recommendations) and NDCG@K (Normalized Discounted Cumulative Gain, accounts for ranking position). All experiments are evaluated at K=30 and reported over 5 random seeds.

c. Baseline Methods

We compare MaskSimGCL against state-of-the-art methods from three categories:

- Graph Contrastive Learning Methods: XSimGCL, SimGCL, LightGCL, DirectAU, SGL
- Basic Graph Neural Network Methods: LightGCN, SSL4Rec
- Proposed Methods: GIFT4Rec (Section 3.2), EfficientRec (Chapter 2)

Table 3.12: Optimal Hyperparameter Configuration

Hyperparameter	Optimal Value
Embedding Dimension (d)	64
GNN Layers (L)	3
SSL Temperature (τ)	0.2
SSL Weight (λ)	0.5
Noise Magnitude (ϵ)	0.1
Mask Alpha (α)	2.0
Mask Beta (β)	0.5
Learning Rate	1e-3
Weight Decay (γ)	1e-5
Batch Size	2048

B. Experimental Results

a. Overall Performance

Table 3.13: MaskSimGCL - Overall Performance Comparison (All Users) @30, reported as mean ± standard deviation over 5 random seeds.

Model	Recall@30	NDCG@30	Category
MaskSimGCL	$\mathbf{0.2404 \pm 0.0008}^{\dagger}$	$\mathbf{0.1322 \pm 0.0004}^{\dagger}$	Proposed
XSimGCL	$\underline{0.2301 \pm 0.0004}$	0.1250 ± 0.0002	GCL
SimGCL	0.2292 ± 0.0012	0.1249 ± 0.0008	GCL
LightGCL	0.2128 ± 0.0003	0.1103 ± 0.0001	GCL
DirectAU	0.2110 ± 0.0009	0.1226 ± 0.0003	GCL
SGL	0.0192 ± 0.0015	0.0565 ± 0.0011	GCL
SSL4Rec	0.1644 ± 0.0001	0.0950 ± 0.0012	GNN
LightGCN	0.1636 ± 0.0003	0.0945 ± 0.0001	GNN
GIFT4Rec (Section 3.2)	0.2162 ± 0.0001	$\underline{0.1263 \pm 0.0001}$	Proposed
EfficientRec (Chapter 2)	0.1994 ± 0.0028	0.1174 ± 0.0027	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 improves on 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. The relatively moderate margin reflects that XSimGCL is already a strong contrastive baseline; the improvement nonetheless demonstrates that dimension-aware augmentation provides a consistent and principled gain over uniform-noise approaches. Each value is the mean ± standard deviation over 5 random seeds. † denotes a statistically significant improvement of MaskSimGCL over the second-best baseline under a two-sided Welch's t-test ($p < 0.001$ for both metrics). MaskSimGCL significantly outperforms all baselines on Recall@30 (+4.5% over the second-best) and NDCG@30 (+4.7% over the second-best).

b. Performance on Cold, Warm, and Active Users

Table 3.14: Performance Comparison on Cold, Warm, and Active Users @30

Model	Cold Users		Warm Users		Active Users	
	R@30	N@30	R@30	N@30	R@30	N@30
MaskSimGCL	**0.3599**	**0.1668**	**0.2555**	**0.1509**	**0.1948**	**0.1483**
XSimGCL	<u>0.3479</u>	<u>0.1633</u>	<u>0.2402</u>	<u>0.1428</u>	<u>0.1870</u>	<u>0.1385</u>
SimGCL	0.3454	0.1623	0.2394	0.1427	0.1785	0.1310
LightGCL	0.3419	0.1592	0.2160	0.1276	0.1705	0.1268
DirectAU	0.2510	0.1160	0.2005	0.1223	0.1535	0.1045
SSL4Rec	0.1444	0.0618	0.1262	0.0772	0.1495	0.1139
LightGCN	0.1411	0.0605	0.1256	0.0766	0.1494	0.1137
GIFT4Rec	0.2511	0.1152	0.2148	0.1335	0.1663	0.1257
EfficientRec	0.2085	0.0910	0.1840	0.1145	0.1689	0.1264

R@30 = Recall@30, N@30 = NDCG@30. Bold = Best, Underline = Second Best.

MaskSimGCL achieves the best Recall@30 across all three user segments, with XSimGCL as the second-best in each case.

Cold users: (Recall@30 = 0.3599 vs. XSimGCL = 0.3479, $+3.4\%$). Cold users have at most 20 interactions, so their embeddings carry limited collaborative signal. The mask mechanism concentrates this scarce signal onto the most informative dimensions and discards noisy ones that would otherwise dominate under sparse data, resulting in more stable representations for users with limited history.

Warm users: (Recall@30 = 0.2555 vs. XSimGCL = 0.2402, $+6.4\%$). The largest relative improvement across all segments is observed for warm users (21-50 interactions). This reflects the sweet spot of the mask mechanism: warm users provide enough interactions for the mask to reliably distinguish important from unimportant dimensions, while the remaining sparsity still leaves room for contrastive regularization to contribute meaningfully.

Active users: (Recall@30 = 0.1948 vs. XSimGCL = 0.1870, $+4.2\%$). For active users with more than 50 interactions, GNN embeddings are already well-informed by rich behavioral data, reducing the relative benefit of contrastive regularization. The mask still provides consistent gains by filtering out redundant dimensions, but the improvement is smaller than for warm users, which is consistent with the expectation that mask-informed augmentation is most decisive when data is moderately sparse.

c. Component Contribution Analysis

To understand the contribution of each component in MaskSimGCL, we conduct comprehensive ablation studies by systematically removing key components. The central argument of MaskSimGCL is that contrastive learning for recommendation can be improved by treating embedding dimensions non-uniformly: a learnable mask identifies which dimensions encode critical preference signals and which capture noise, and this information is used to generate semantically consistent contrastive views. The ablation study isolates three levels of this argument: Is contrastive learning itself essential for robust representation? Does the learnable mask provide value as an implicit regularizer, independent of its use in noise generation? Does using the mask to inform the noise perturbation provide additional benefit beyond the mask's regularization effect?

Table 3.15: Component Ablation Study on MaskSimGCL

Configuration	**Recall@30**	**ΔRecall**	**NDCG@30**	**ΔNDCG**
MaskSimGCL (Full)	0.2404$\pm$0.003		0.1322$\pm$0.003	
w/o Contrastive Learning	0.1731$\pm$0.003	-28.0%	0.0923$\pm$0.003	-30.2%
w/o Mask-Informed Noise	0.2212$\pm$0.004	-8.0%	0.1239$\pm$0.004	-6.3%
w/o Learnable Mask (SimGCL)	0.2260$\pm$0.003	-6.0%	0.1219$\pm$0.003	-7.8%

Note: ΔRecall and ΔNDCG represent relative difference compared to full model.

The evidence supports all three claims with a clear hierarchy. Removing contrastive learning entirely causes the most severe degradation (-28.0% Recall, -30.2% NDCG), confirming that self-supervised contrastive signals are foundational to robust graph-based recommendation. Removing the learnable mask while keeping uniform noise (i.e., reverting to the SimGCL baseline) degrades Recall by -6.0% but NDCG by -7.8%, revealing that the mask's dimension aware filtering particularly improves ranking precision by suppressing noisy dimensions. Critically, removing only the mask informed noise strategy keeping the mask for regularization but applying uniform perturbations causes -8.0% Recall, which is larger than removing the mask itself. This confirms that the primary contribution of MaskSimGCL is not the mask as a standalone regularizer but the integration of learned importance scores into the contrastive view generation process. The mask and the noise strategy are designed to work together: the mask identifies what matters, and the informed noise ensures that contrastive views preserve what matters while diversifying what does not.

3.4 Chapter Summary

This chapter addressed sparsity and cold-start through two complementary contributions. GIFT4Rec performs adaptive side-information fusion, using an attention-based weight-generation module to compute user-specific fusion weights and to combine behavioural embeddings with auxiliary signals through local and global modules optimized under meta-learning principles. MaskSimGCL complements it by strengthening representation robustness with masked graph contrastive learning. Together they yield consistent gains over graph-based baselines in both warm-start and cold-start settings, showing that principled fusion and contrastive regularization improve recommendation under data scarcity.

Several limitations temper these results. The benefit of fusion is only as good as the side information it draws on, so noisy, incomplete, or biased attributes can weaken or even mislead the learned weights. The masking ratio and augmentation settings of the contrastive objective are still tuned by hand and may not carry over cleanly between datasets, and the contrastive component itself adds training cost that can become a bottleneck on very large interaction graphs.

Addressing these issues suggests a clear path forward. Denoising objectives and uncertainty-aware weighting would let the model discount unreliable attributes rather than trust them uniformly, and automating the choice of masking and augmentation strategies would remove a fragile manual step. Extending the fusion framework to multimodal side information, such as textual and visual item content, while improving the scalability of contrastive training on ultra-large graphs, would broaden its applicability to richer and larger settings.

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 [122], 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 The Multi-domain Recommendation Challenge

In industrial recommendation systems, platforms such as Taobao, Alibaba, and streaming services simultaneously serve users across multiple business domains where each domain exhibits distinct user behavior patterns, interaction distributions, and semantic characteristics [64, 125]. The conventional approach of training separate models for each domain fails to exploit the potential synergies and shared knowledge that could benefit all domains collectively. Conversely, naively training a unified model across all domains often results in negative transfer, where the optimization for one domain degrades performance in others, and catastrophic forgetting, where knowledge acquired from earlier domains is overwritten during subsequent training phases.

Recent industrial deployments have highlighted the practical importance of this challenge. CTNet [65], deployed at Taobao, addresses the problem of Continual Transfer Learning (CTL) where knowledge must be transferred from time evolving source domains to time evolving target domains. The KEEP framework [127] proposes a two-stage approach that extracts knowledge from a super domain and plugs it into downstream models. DIIT [39] focuses on extracting domain invariant information for industrial cross domain recommendation. ECAT [36] introduces a comprehensive framework for entire space continual and adaptive transfer learning. While these methods have demonstrated significant improvements in their respective target domains, they share a common limitation: the unidirectional nature of knowledge transfer that prioritizes target domain improvement potentially at the cost of source domain performance.

4.1.2 Terminological Clarification

Before presenting the contributions of this chapter, it is necessary to clarify three closely related but distinct concepts that are frequently conflated in the recommendation literature.

Multi-domain recommendation refers to the setting where a single platform serves users across multiple content categories or business verticals such as movies, music, e-commerce, and news simultaneously. The goal is to build a unified system that delivers accurate recommendations in all participating domains, rather than optimizing for any single domain in isolation. The key challenge is that different domains exhibit distinct user behavior patterns, item distributions, and semantic characteristics, making joint modeling non-trivial.

Cross-domain recommendation is a special case that focuses on transferring knowledge from one or more data-rich source domains to a data-sparse target domain. Unlike multi-domain recommendation, cross-domain methods are inherently asymmetric: they designate explicit source and target roles, and their optimization objective is typically formulated in terms of target domain performance. Representative methods such as CTNet [65], KEEP[127], and DIIT[39] follow this unidirectional transfer paradigm.

Continual learning (also known as lifelong learning) is a training paradigm rather than a problem formulation. It addresses the challenge of learning new tasks sequentially without forgetting previously acquired knowledge known as catastrophic forgetting. In the recommendation context, continual learning provides a principled mechanism for updating a model as new domains are introduced over time, without retraining from scratch on all historical data.

The problem addressed in this chapter lies at the intersection of these three con-

cepts. We formulate the problem as multi-domain recommendation, where the objective is to achieve balanced, high-quality performance across all domains rather than improving a single target domain. We adopt continual learning as the training paradigm to enable sequential domain adaptation, and we explicitly depart from the cross-domain paradigm by rejecting its unidirectional, target-centric transfer assumption. Concretely, the proposed CNL4Rec framework trains on domains sequentially (a continual learning setup), but unlike cross-domain methods, it enables multi-directional knowledge transfer where all domains can both contribute to and benefit from shared representations, and it evaluates success by the aggregate performance across all domains (a multi-domain objective).

4.1.3 Limitations of Existing Cross-Domain and Multi-Domain Approaches

A systematic analysis of recent cross domain and multi-domain recommendation methods reveals several critical gaps that motivate the proposed CNL4Rec framework.

Limitation 1: Unidirectional Transfer Paradigm

The predominant paradigm in existing cross domain recommendation follows a unidirectional transfer approach where knowledge flows exclusively from source domains to target domains. CTNet [65] treats source domain representations as external knowledge for target domain CTR prediction, preserving source and target domain parameters during transfer but fundamentally optimizing for target domain improvement. KEEP [127] extracts knowledge from a super domain and plugs it into downstream models, creating a one way knowledge pipeline. DIIT [39] designs extractors to transfer domain invariant information from source domain models to target domain models. ECAT [36] proposes sample transfer and representation transfer mechanisms, both oriented toward enhancing target domain performance.

This one way design works well for improving sparse target domains, but it can cause an imbalance. Over time, the system focuses too much on the target domain, which may slow down or reduce performance in the source domains.

Limitation 2: Hard Constraint Parameter Isolation

Existing continual learning approaches for cross domain recommendation typically employ hard constraints that completely freeze parameters seen as important for previous domains. While this strategy effectively prevents catastrophic forgetting, it fundamentally limits the capacity for bidirectional knowledge sharing. When parameters are frozen entirely, they cannot receive gradient updates from subsequent domain training, eliminating the possibility of reverse knowledge transfer where insights from newer do-

mains could improve representations for earlier domains. This strict parameter separation creates barriers between domains and prevents shared representations from naturally forming to benefit all domains at the same time.

Limitation 3: Absence of Fairness Considerations

More importantly, existing cross domain methods lack explicit mechanisms for ensuring fair performance across all participating domains. The optimization objectives in CTNet, KEEP, DIIT, and ECAT are formulated exclusively in terms of target domain metrics, with source domain performance treated as, at best, a secondary consideration. This design choice reflects an implicit assumption that source domains, being typically larger and more data rich, can tolerate some performance degradation in service of improving smaller target domains. However, in real-world multi-domain platforms where all domains contribute to the overall user experience and business value, this asymmetric treatment creates systemic bias that accumulates over time, potentially resulting in significant performance disparities across domains.

4.1.4 Research Gaps and Challenges

Based on the analysis of existing approaches, this research identifies four critical gaps that the proposed framework aims to address:

Gap 1: Multi-Directional Knowledge Transfer. Current methods exclusively support unidirectional transfer from source to target domains. There exists no principled framework for enabling multi directional knowledge flow where all domains can both contribute to and benefit from shared representations. A truly unified multi-domain recommendation system should allow knowledge to propagate in all directions, with each domain serving simultaneously as both a source and a target of transferable knowledge.

Gap 2: Soft Constraint Continual Learning. Existing continual learning mechanisms rely on hard parameter freezing that prevents any modification to protected parameters. This binary approach fails to recognize the gradual nature of parameter importance and prevents the potential for incremental knowledge refinement. A more nuanced approach would employ soft constraints that modulate rather than eliminate gradient updates, allowing less important parameters to continue adapting while still protecting domain critical knowledge.

Gap 3: Domain Fairness Optimization. No existing framework explicitly optimizes for balanced performance across all domains. The sole focus on target domain improvement creates a systematic bias that disadvantages source domains over time. A fair multi-domain recommendation framework should incorporate explicit fairness objectives that ensure all domains benefit from the knowledge sharing process, preventing the emer-

gence of performance disparities.

4.2 CNL4Rec: multi-domain Recommendation Model based on Continual Learning

4.2.1 Problem Statement

To address the identified gaps, this chapter proposes the "Continual Learning for multi-domain Recommendation (CNL4Rec)" framework, which introduces several novel contributions that distinguish it from existing approaches.

First, CNL4Rec employs a domain masking mechanism that learns task specific masks to identify embedding dimensions essential for each domain. Unlike hard freezing approaches, the domain masks operate through a soft thresholding mechanism that allows gradual modulation of parameter importance. Parameters identified as highly important for previous domains receive suppressed but non zero gradient updates, enabling knowledge preservation while still permitting incremental refinement based on new domain information.

Second, the domain specialization module implements a gradient regulation strategy that protects domain's important parameters while enabling adaptive reuse of less important parameters for new domain learning. By computing the union of importance masks across all previous domains and modulating gradients accordingly, this mechanism achieves a principled balance between stability by preserving past knowledge and plasticity as accommodating new knowledge.

Third, the framework is designed with explicit consideration for domain fairness. Rather than optimizing exclusively for target domain performance, CNL4Rec evaluates and optimizes overall performance across all domains, ensuring that knowledge transfer benefits all participating domains rather than improving one at the expense of others. The experimental evaluation explicitly measures performance on all domains and computes aggregate metrics that penalize performance imbalances.

Comparative positioning of CNL4Rec against related methods

Table 4.1 maps each of the three research gaps identified above onto the corresponding design dimension of CNL4Rec and its four closest competitors, making explicit which limitations each method addresses and which it leaves open.

The table reveals a clear pattern: all four baselines share the same three structural gaps simultaneously. Regarding Gap 1, CTNet [65], KEEP [127], DIIT [39], and ECAT [36] all route knowledge in one direction only from a richer source toward a

Table 4.1: Comparison of CNL4Rec with Related Cross-Domain Methods

Characteristic	**CTNet [65]**	**KEEP [127]**	**DIIT [39]**	**ECAT [36]**	**CNL4Rec (Ours)**
Transfer direction	Uni (src→tgt)	Uni (super→tgt)	Uni (src→tgt)	Uni (space→tgt)	**Multi-directional**
Constraint type	Hard freeze	Hard (2-stage)	Hard	Hard	**Soft thresholding**
Reverse knowledge transfer	✗	✗	✗	✗	✓
Soft gradient modulation	✗	✗	✗	✗	✓
Explicit fairness objective	✗	✗	✗	✗	✓
All domains optimized jointly	✗	✗	✗	✗	✓

Uni: unidirectional transfer; *Hard freeze*: parameters fully blocked from gradient updates once deemed important for a domain; *Soft thresholding*: gradient magnitude modulated proportionally to learned importance scores rather than hard-blocked.

designated target so no mechanism exists for later domains to refine representations of earlier ones. Regarding Gap 2, each method relies on hard parameter freezing, which eliminates gradient flow to protected parameters entirely and prevents the gradual, incremental knowledge sharing that soft constraints can achieve. Regarding Gap 3, every baseline formulates its optimization objective exclusively in terms of target domain metrics, leaving source domain performance to degrade silently as an accepted side-effect of the transfer process.

4.2.2 Model Architecture and Components

This section presents the complete architecture of the proposed CNL4Rec (Continual learning for multi-domain recommendation) framework. The model is designed to address the fundamental challenges of multi-domain recommendation systems, including catastrophic forgetting, negative transfer between domains, and performance degradation under sequential domain learning scenarios. The architecture consists of three tightly integrated modules: the Domain Masking Module, the Domain Specialization Module, and the Behavior Extraction Module. Together, these components form a unified continual learning framework that preserves domain specific knowledge while enabling effective knowledge transfer across heterogeneous recommendation domains.

a. Notation and Problem Setup

Before describing the detailed architecture of each component, we first establish the mathematical notation that will be used throughout this section.

Consider a multi-domain recommendation scenario operating across K distinct domains. Let $\mathcal{U} = \{u_1, u_2, \ldots, u_M\}$ denote the set of M users and $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ denote the set of N items. The domain set is represented as $\mathcal{K} = \{1, 2, \ldots, K\}$, where

each domain $k \in \mathcal{K}$ contains a subset of users $\mathcal{U}_k \subseteq \mathcal{U}$ and items $\mathcal{V}_k \subseteq \mathcal{V}$. The number of users and items in domain k are denoted as $N_{U_k} = |\mathcal{U}_k|$ and $N_{I_k} = |\mathcal{V}_k|$, respectively.

Each user u_i and item v_j is associated with a learnable embedding vector. The user embedding matrix is denoted as $\mathbf{X}_U \in \mathbb{R}^{N_U \times d_U}$, where the i-th row $\mathbf{X}_{U_i} \in \mathbb{R}^{d_U}$ represents the embedding of user u_i with dimensionality d_U. Similarly, the item embedding matrix is $\mathbf{X}_I \in \mathbb{R}^{N_I \times d_I}$, where $\mathbf{X}_{I_j} \in \mathbb{R}^{d_I}$ represents the embedding of item v_j with dimensionality d_I. These embedding matrices are randomly initialized and learned through end to end training across sequential domain tasks.

The core challenge addressed by CNL4Rec is to learn effective recommendation models that can adapt to new domains while preserving knowledge acquired from previously learned domains. This is formalized as a continual learning problem where domains arrive sequentially, and the model must maintain stable performance across all encountered domains without access to historical training data during subsequent domain training phases.

b. Architecture Overview

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.

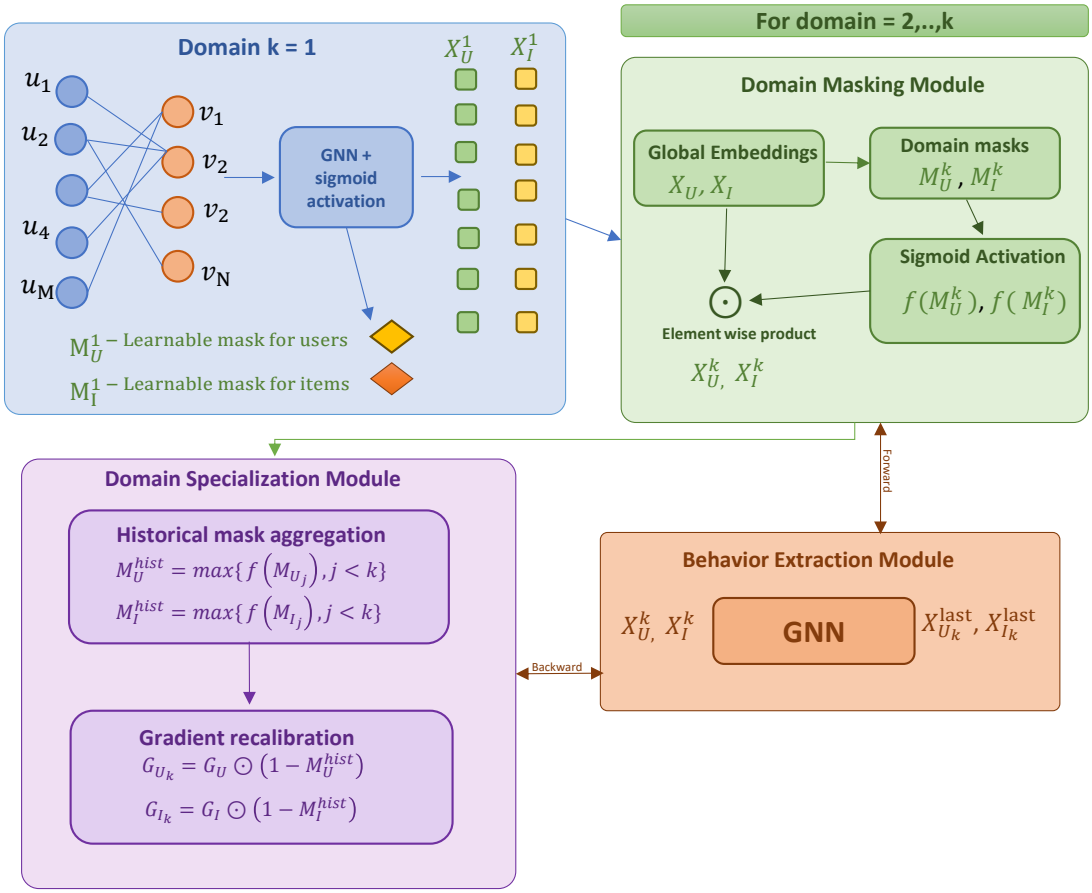

Figure 4.1: CNL4Rec Overall Architecture

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.

c. Domain Masking Module

The Domain Masking Module constitutes the first core component of the CNL4Rec architecture. This module is responsible for learning domain specific importance weights that identify which embedding dimensions are critical for accurate recommendation within each domain. The key insight motivating this design is that different domains may rely on different subsets of latent features to capture user preferences, and explicitly modeling this domain feature relationship enables more effective knowledge isolation and transfer.

Domain specific mask initialization

For each domain k, the module maintains two learnable mask matrices that correspond to the user and item embedding spaces:

- A user domain mask $\mathbf{M}_{U_k} \in \mathbb{R}^{N_U \times d_U}$ that weights the importance of each dimension in the user embedding space for domain k.
- An item- domain mask $\mathbf{M}_{I_k} \in \mathbb{R}^{N_I \times d_I}$ that weights the importance of each dimension in the item embedding space for domain k.

These mask matrices are initialized with small random values and are optimized jointly with the recommendation objective. The mask dimensions match the corresponding embedding dimensions, enabling fine grained control over which features are activated for each domain.

Mask Activation

The raw mask values are transformed through a nonlinear activation function $f(\cdot)$ to produce importance weights within a fixed range that ensure stable optimization and interpretable importance scores. The activation function is defined using a thresholded sigmoid formulation:

$$f(x) = \begin{cases} 1, & \text{if } \sigma(x) > \text{threshold} \\ 0, & \text{if } \sigma(x) \leq \text{threshold} \end{cases} \tag{4.1}$$

where $\sigma(\cdot)$ denotes the sigmoid activation function $\sigma(x) = \frac{1}{1+e^{-x}}$, and threshold is a predefined hyperparameter that controls the sparsity of the resulting masks. This hard gating mechanism transforms continuous importance scores into binary decisions, enforcing explicit parameter isolation between domains.

The raw mask values are transformed through a thresholded sigmoid function (Eq 4.1) to produce binary importance weights. The sigmoid function $\sigma(\cdot)$ is chosen for three reasons. First, it maps any real-valued mask parameter continuously to $(0, 1)$, providing a smooth, differentiable intermediate representation that allows gradients to flow through the mask during end-to-end training a property that hard step functions do not possess. Second, applying a fixed threshold to the sigmoid output converts these continuous scores into binary decisions, enforcing explicit parameter isolation between domains: dimensions above the threshold are fully activated for the current domain, while those below are completely suppressed. Third, the sigmoid's smooth gradient particularly its non-zero derivative near the threshold boundary enables the mask parameters to be learned efficiently via standard gradient-based optimisation, unlike alternatives such as ReLU whose gradient is zero for negative inputs and would prevent suppressed dimensions from ever being reactivated in subsequent domains.

Domain specific embedding computation

The activated masks are applied to the global embedding matrices through element-wise multiplication to obtain domain specific user and item representations:

$$\mathbf{X}_{U_k} = f(\mathbf{M}_{U_k}) \odot \mathbf{X}_U \tag{4.2}$$

$$\mathbf{X}_{I_k} = f(\mathbf{M}_{I_k}) \odot \mathbf{X}_I \tag{4.3}$$

where $\odot$ denotes element wise product. Through this operation, only the embedding dimensions that are considered important for domain k are activated, while less relevant dimensions are suppressed. This selective activation achieves several important objectives:

- Parameter isolation: Dimensions with zero mask values are effectively frozen for domain k, protecting parameters that may be critical for other domains.
- Gradient control: During backpropagation, gradients flow only through activated dimensions, preventing updates to suppressed parameters.
- Implicit regularization: The sparsity induced by masking reduces the effective model capacity, mitigating overfitting in sparse domain data.

The domain specific embeddings $\mathbf{X}_{U_k}$ and $\mathbf{X}_{I_k}$ are subsequently fed into the behavior extraction module for final preference prediction. This design ensures that each domain operates on a customized subset of the latent space while sharing the underlying embedding parameters.

d. Domain Specialization Module

While the domain masking module identifies domain relevant latent dimensions, the domain specialization module is responsible for protecting previously learned knowledge and regulating gradient updates during continual multi-domain learning. This module is central to continual learning: the model must be stable enough to preserve past knowledge while remaining plastic enough to acquire new knowledge.

Historical Mask Aggregation.

When training on domain k, the module first computes a cumulative importance mask that summarizes which embedding dimensions have been identified as important in any of the previously learned domains $\{1, 2, \ldots, k-1\}$. This aggregation is performed using an element wise maximum operation:

$$\mathbf{M}_U^{\text{hist}} = \max\{f(\mathbf{M}_{U_j}) \mid j \in \mathbb{N}, 0 < j < k\} \tag{4.4}$$

$$\mathbf{M}_I^{\text{hist}} = \max\{f(\mathbf{M}_{I_j}) \mid j \in \mathbb{N}, 0 < j < k\} \tag{4.5}$$

where the $\max$ operator is applied element wise across all previously learned domain masks. The resulting historical mask $\mathbf{M}^{\text{hist}}$ highlights dimensions that have been considered important by any prior domain, providing a unified view of which parameters require protection.

The choice of maximum aggregation, rather than mean or sum, ensures that any dimension identified as important by any previous domain receives full protection. This conservative approach prioritizes knowledge preservation over parameter efficiency, which is appropriate for scenarios where catastrophic forgetting poses a significant risk.

Gradient Recalibration

During backpropagation for domain k, the original gradients of the loss function with respect to the embedding matrices are recalibrated to prevent updates to historically important parameters. Let $\mathbf{G}_U$ and $\mathbf{G}_I$ denote the original gradients computed through standard backpropagation. The recalibrated gradients are computed as:

$$\mathbf{G}_{U_k} = \mathbf{G}_U \odot (1 - \mathbf{M}_U^{\text{hist}}) \tag{4.6}$$

$$\mathbf{G}_{I_k} = \mathbf{G}_I \odot (1 - \mathbf{M}_I^{\text{hist}}) \tag{4.7}$$

This formulation achieves selective gradient suppression through a complementary mask operation. For each embedding dimension:

- If the dimension was important in any prior domain ($\mathbf{M}^{\text{hist}} = 1$), the recalibrated gradient becomes zero, completely freezing that parameter.
- If the dimension was not important in any prior domain ($\mathbf{M}^{\text{hist}} = 0$), the original gradient is preserved, allowing full adaptation for the current domain.

This binary protection mechanism ensures that parameters critical to past domains receive zero effective gradients, maintaining their learned values unchanged. Simultaneously, parameters that have not been claimed by previous domains remain available for learning new domain specific patterns.

Knowledge Transfer and Parameter Reallocation

Beyond preventing catastrophic forgetting, the domain specialization module enables controlled knowledge transfer across domains. The gradient recalibration mechanism implicitly facilitates positive transfer by:

- Preserving shared Patterns: Parameters that encode domain invariant user preferences or item characteristics are protected once learned, making them available for all subsequent domains without relearning.
- Allocating fresh capacity: Unimportant parameters from previous domains are reallocated to encode new domain specific patterns, ensuring that the model can continue learning without running out of representational capacity.
- Reducing negative transfer: By isolating domain specific parameters, the module prevents conflicting gradient signals from different domains from interfering with each other.

This design addresses both the stability and plasticity requirements of continual learning: stability is achieved through gradient suppression on important parameters, while plasticity is maintained by allowing adaptation on unimportant parameters.

Mask Saturation and Capacity Analysis

Since the cumulative mask M_{hist} accumulates importance across domains, the set of protected dimensions is non-decreasing: after many domains it could approach saturation, leaving little free capacity for new ones. In CNL4Rec this is mitigated because the masking is soft ($M_{\text{hist}} \in [0, 1]$, with gradients scaled by $(1 - M_{\text{hist}})$), so protected dimensions keep residual plasticity instead of being frozen; moreover, per-domain masks are sparse and important dimensions overlap across related domains, so M_{hist} grows slowly in K. For the embedding size and the number of domains used here, the protected fraction stays well below saturation.

For very long task sequences, however, saturation becomes a fundamental limit. In that regime, preserving plasticity requires *capacity expansion* rather than reallocating a fixed budget: the embedding dimensionality must be increased, either globally or by appending a small block of fresh dimensions per new domain. Validating this calls for dimension-expansion experiments that measure the trade-off between the added capacity, the extra memory cost, and the resulting continual-learning performance, which we leave for future work.

e. Behavior Extraction Module

The Behavior Extraction Module constitutes the final component of the CNL4Rec architecture, serving as the interface between the continual learning mechanism and the downstream recommendation task. This module is intentionally designed so that the continual learning capabilities of CNL4Rec can be applied to any graph neural network backbone.

Domain Adaptive Embedding Reception

The behavior extraction module receives the domain adaptive embeddings $\mathbf{X}_{U_k}$ and $\mathbf{X}_{I_k}$ produced by the domain masking module. These embeddings have already been filtered to contain only the domain relevant features, providing a clean input signal for the recommendation backbone.

Graph neural network backbone

The backbone is instantiated as a graph neural network, the final user and item embeddings are computed through iterative message passing on the interaction graph:

$$\mathbf{X}_{U_k}^{\text{last}} = \text{GNN}(\mathbf{X}_{U_k}) \tag{4.8}$$

$$\mathbf{X}_{I_k}^{\text{last}} = \text{GNN}(\mathbf{X}_{I_k}) \tag{4.9}$$

where $\text{GNN}(\cdot)$ denotes a generic graph based propagation operator such as GCN, GAT, or LightGCN. The graph propagation aggregates both structural and semantic neighborhood information, enabling the extraction of high order collaborative behavioral patterns while respecting the domain specific feature restrictions imposed by the masking mechanism.

Preference Prediction

The final behavioral embeddings are used to compute preference scores through inner product computation:

$$\hat{y}_{ij} = (\mathbf{X}_{U_{ik}}^{\text{last}})^{\top} \cdot \mathbf{X}_{I_{jk}}^{\text{last}} \tag{4.10}$$

where $\hat{y}_{ij}$ represents the predicted preference score between user u_i and item v_j in domain k. Higher scores indicate stronger predicted preferences, which are used for ranking items in the recommendation list.

4.3 Experimental Settings and Results

4.3.1 Experimental Settings

a) Datasets

We conduct experiments on three benchmark datasets representing diverse recommendation scenarios with multiple domains:

MovieLens-1M: Contains 1,000,209 ratings from 6,040 users on 3,952 movies. We partition the dataset into 5 domains based on movie genres: Action (138,766 interactions), Comedy (196,945 interactions), Drama (228,440 interactions), Thriller (108,216 interactions), and Sci-Fi (83,197 interactions). Only positive interactions (rating ≥ 4) are retained.

Yelp: Contains user reviews for local businesses, organized into 5 business category domains: Restaurants (35,750 interactions), Shopping (37,058 interactions), Food (31,062 interactions), Beauty (32,563 interactions), and Health (36,099 interactions). The dataset contains 5,000 users and 3,000 items.

Amazon: Constructed from Amazon product reviews spanning 5 product categories: Electronics (39,118 interactions), Books (35,340 interactions), Movies (46,470 interactions), Home (49,644 interactions), and Sports (29,100 interactions). The dataset comprises 6,000 users and 4,000 items.

Table 4.2: Statistics of Multi-Domain Datasets

Dataset	**Users**	**Items**	**Domains**	**Total Inter.**	**Density**
MovieLens-1M	6,038	2,972	5	755,564	4.21%
Yelp	5,000	3,000	5	172,532	1.15%
Amazon	6,000	4,000	5	199,672	0.83%

b) Evaluation Metrics

We evaluate recommendation quality using Recall@30 as the primary metric, following the standard practice in multi-domain recommendation literature. All experi-

ments are evaluated over 5 random seeds to ensure statistical reliability. We adopt temporal split with 80% training data and 20% testing data.

c) Baseline Methods

We compare CNL4Rec against five state of the art methods representing different paradigms in cross domain and multi-domain recommendation:

- MF (Matrix Factorization): Classic collaborative filtering baseline trained independently for each domain.
- KEEP: Knowledge extraction and plugging framework from super domain to downstream models.
- DIIT: Domain Invariant Information Transfer framework for cross domain recommendation.
- ECAT: Enhanced Cross domain recommendation with Adaptive Transfer mechanisms.
- CTNet: Continual Transfer Network for time evolving cross domain scenarios.

d) Optimal Hyperparameter Configuration

Table 4.3: Optimal Hyperparameter Configuration

Hyperparameter	**Optimal Value**
Embedding Dimension	64
GNN Layers (L)	3
Dropout Rate	0.2
Learning Rate	1e-3
Weight Decay (γ)	1e-5
Batch Size	2048

The values reported in Table 4.3 are the optimal configuration obtained from systematic grid search experiments on the validation splits of all three datasets (MovieLens-1M, Yelp, Amazon), in which each hyperparameter was varied independently while all others were held fixed. The detailed component contribution analysis for the two most critical architectural choices the MLP block and the domain specialization module is

presented in Table 4.7. The remaining parameters (embedding dimension, number of GNN layers, dropout rate, learning rate, weight decay, and batch size) follow standard settings widely adopted in the continual and multi-domain recommendation literature and were confirmed to be optimal on the same validation splits through the same grid search procedure.

4.3.2 Experimental Results

A. Overall results

a. Results on MovieLens-1M

On MovieLens-1M (density 4.21%), CNL4Rec achieves the best mean Recall@30 of 0.2288, ahead of 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 4.4: Performance Comparison on MovieLens-1M (Recall@30). CV(%)=coefficient of variation across genres (lower is more balanced); Min=worst-domain Recall@30 (higher is better). **Bold**=Best, Underline=Second Best.

Method	Action	Comedy	Sci-Fi	Thriller	Drama	Mean	Std	CV(%)	Min
CNL4Rec	**0.2481**	**0.2583**	**0.1865**	**0.2349**	0.2163	**0.2288**	0.0254	**11.1**	**0.1865**
CTNet	0.1600	0.0838	0.1342	0.2034	**0.2268**	0.1616	0.0506	31.3	0.0838
DIIT	0.1246	0.0632	0.1106	0.1181	0.1061	0.1045	0.0216	20.7	0.0632
KEEP	0.0598	0.0060	0.0841	0.1005	0.1796	0.0860	0.0567	65.9	0.0060
MF	0.0744	0.0060	0.0286	0.0485	0.2061	0.0727	0.0704	96.8	0.0060
ECAT	0.0063	0.0273	0.0173	0.0744	0.1665	0.0583	0.0588	100.8	0.0063

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 ahead of 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.

Beyond the mean, we assess how evenly each method performs across genres us-

ing the coefficient of variation ($\mathrm{CV} = \mathrm{std}/\mathrm{mean}$, lower is more balanced) and the worst domain Recall@30 (Min, higher is better). On MovieLens-1M, CNL4Rec is the most balanced method by a wide margin, attaining the lowest dispersion ($\mathrm{CV} = 11.1\%$) and the highest worst domain score (0.1865) alongside the best mean (0.2288). Several baselines instead obtain a non-trivial mean by excelling in a single genre while collapsing elsewhere: MF and ECAT perform reasonably on Drama but fall to near-zero on Comedy/Action, yielding extreme dispersion ($\mathrm{CV} = 96.8\%$ and 100.8%), and even CTNet the strongest baseline by mean is far less uniform ($\mathrm{CV} = 31.3\%$). This shows that CNL4Rec's advantage does not come from sacrificing weaker genres; the domain masking and gradient recalibration raise the floor (worst-domain) rather than only the average. The sole exception is Drama, where CTNet marginally leads (0.2268 vs. 0.2163), yet CNL4Rec's weakest genre still exceeds the best worst-domain score of every baseline, confirming balanced, fair performance across the multi-domain setting.

b. Results on Yelp

On the Yelp dataset (interaction density 1.15%), CNL4Rec achieves the best mean Recall@30 of 0.0174, ahead of 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 4.5: Performance Comparison on Yelp (Recall@30). CV(%)=coefficient of variation across categories (lower is more balanced); Min=worst-domain Recall@30 (higher is better). **Bold**=Best, Underline=Second Best.

Method	Restaurant	Shopping	Food	Beauty	Health	Mean	Std	CV(%)	Min
CNL4Rec	**0.0194**	0.0150	0.0184	0.0166	**0.0182**	**0.0174**	0.0015	8.8	0.0150
ECAT	0.0164	0.0172	0.0162	**0.0172**	0.0154	0.0165	0.0007	**4.1**	0.0154
CTNet	0.0146	0.0152	**0.0192**	0.0148	0.0179	0.0164	0.0019	11.4	0.0146
KEEP	0.0158	**0.0182**	0.0156	0.0156	0.0164	0.0165	0.0010	6.0	**0.0156**
MF	0.0150	0.0158	0.0179	0.0162	0.0156	0.0163	0.0010	6.1	0.0150
DIIT	0.0138	0.0154	0.0150	0.0156	0.0178	0.0155	0.0013	8.4	0.0138

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.

On Yelp, all methods are tightly clustered (means within 0.0155–0.0174), so the dispersion measures reveal smaller differences than on MovieLens-1M. CNL4Rec attains the best mean (0.0174) and a competitive worst-domain score (0.0150), but it is not the most balanced method here: ECAT achieves the lowest dispersion ($\mathrm{CV} = 4.1\%$) and KEEP the highest worst-domain (0.0156), whereas CNL4Rec's CV (8.8%) is mid-range, driven mainly by its strong Restaurant result (0.0194). This is consistent with the role of the domain-balancing mechanism: when domain boundaries are sharp (as on MovieLens-1M and Amazon) the masking and gradient recalibration substantially raise the floor, but on a denser, more homogeneous benchmark such as Yelp where all categories already perform similarly the balancing effect is less pronounced. CNL4Rec nonetheless remains competitive across all five categories, with no domain collapsing, indicating that its best-mean performance is not achieved at the expense of any single category.

c. Results on Amazon

On the Amazon dataset (interaction density 0.83%, the sparsest of the three benchmarks), CNL4Rec achieves a mean Recall@30 of 0.0132, marginally ahead of 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,

Table 4.6: Performance Comparison on Amazon (Recall@30). CV(%)=coefficient of variation across categories (lower is more balanced); Min=worst-domain Recall@30 (higher is better). **Bold**=Best, Underline=Second Best.

Method	Electronics	Books	Movies	Home	Sports	Mean	Std	CV(%)	Min
CNL4Rec	0.0139	0.0125	0.0135	**0.0133**	0.0126	**0.0132**	0.0005	**4.1**	**0.0125**
ECAT	**0.0147**	0.0130	0.0128	0.0132	0.0116	0.0131	0.0010	7.6	0.0116
MF	0.0129	0.0109	**0.0170**	0.0125	0.0124	0.0131	0.0020	15.6	0.0109
KEEP	0.0114	**0.0145**	0.0142	0.0118	0.0127	0.0129	0.0012	9.6	0.0114
CTNet	0.0115	0.0120	0.0137	0.0113	**0.0136**	0.0124	0.0010	8.3	0.0113
DIIT	0.0134	0.0135	0.0105	0.0103	0.0116	0.0119	0.0014	11.6	0.0103

$+0.8\%$). The fact that MF performs better than 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 perform better 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. Amazon is the sparsest benchmark (density 0.83%), and the methods are tightly clustered by mean (0.0119-0.0132). Here CNL4Rec is again the most balanced: it attains the lowest dispersion ($\mathrm{CV} = 4.1\%$) and the highest worst-domain score (0.0125), together with the best mean (0.0132). Interestingly, CNL4Rec wins outright in only one category (Home), while the per-category best is split across ECAT (Electronics), MF (Movies), KEEP (Books), and CTNet (Sports); yet none of these baselines maintains a high floor each has a weak category (e.g., DIIT 0.0103 on Home, MF 0.0109 on Books) which inflates their dispersion (CV up to 15.6%). CNL4Rec's advantage therefore stems not from dominating any single category but from being consistently competitive across all of them, never collapsing in any domain. This confirms that the domain masking and gradient recalibration deliver balanced, fair performance, with the strongest effect on benchmarks where domain boundaries are sharp. 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. This narrow margin should be interpreted as a boundary condition of the proposed mechanism rather than a failure. The domain mask in CNL4Rec is learned from the gradient-based importance signal of each domain; when interaction density falls below 1 % (as in Amazon), this signal becomes too weak to reliably separate domain-critical dimensions from noisy ones, so the learned mask tends toward a near-uniform pattern and its discriminative benefit shrinks. The soft constraint is precisely what keeps performance stable in this regime: because it only modulates rather than freezes gradients, an imperfectly estimated mask under extreme sparsity does not lock the model into wrong dimensions, but instead allows continued adaptation that degrades gracefully toward the baseline. This is why CNL4Rec still attains the best mean Recall@30 (+0.8%) without negative transfer, whereas a hard-constraint design would risk protecting unreliable dimensions and harming later domains. We therefore regard +0.8% as the expected behaviour of a soft, protection-oriented mechanism at the sparsity limit, where the headroom for any continual-learning method is inherently small; richer auxiliary signals or density-adaptive masking are promising directions to widen this margin and are left for future work.

B. Component Contribution Analysis

a) Component Ablation Study

To understand the contribution of each component in CNL4Rec, we conduct comprehensive ablation studies by systematically removing or replacing key components.

The central argument of CNL4Rec is that effective multi-domain recommendation requires two complementary mechanisms: domain masking to identify which embedding dimensions are critical for each domain, and domain specialization to protect those dimensions during subsequent domain training via gradient modulation. The ablation study tests whether each mechanism contributes independently and whether their combination is necessary: Does domain masking improve performance by enabling domain-specific feature selection? Does domain specialization reduce catastrophic forgetting by protecting previously learned parameters? Are the two mechanisms complementary, i.e., does removing either one cause degradation that cannot be compensated by the other?

Table 4.7: Component Ablation Study on CNL4Rec (MovieLens-1M, Mean Recall@30)

Configuration	**Mean Recall@30**	**ΔRecall**
CNL4Rec (Full)	0.2288	
w/o MLP Block	0.1876	-18.0%
w/o Domain Specialization	0.1968	-14.0%
w/o Residual Connections	0.1991	-13.0%
w/o Domain Masking	0.2013	-12.0%
w/o LayerNorm	0.2059	-10.0%
ReLU (vs GELU)	0.2105	-8.0%

Note: ΔRecall represents relative difference compared to full model.

The results confirm the complementary nature of both mechanisms. Removing domain masking degrades mean Recall by -12.0%, demonstrating that explicit identification of domain-relevant dimensions is essential for preventing inter-domain interference. Removing domain specialization causes an even larger degradation of -14.0%, indicating that protecting important parameters through gradient regulation is critical for preserving previously acquired domain knowledge. The fact that both removals cause substantial and comparable degradation rather than one dominating confirms that the two mechanisms address distinct failure modes: domain masking prevents negative transfer, while domain specialization prevents catastrophic forgetting. Neither mechanism alone is sufficient; their combination provides the stability plasticity balance required for con-

tinual multi-domain learning.

C. Domain Order Impact

Table 4.8: Training Strategy Comparison

Strategy	Mean Recall@30	vs Full	Ranking
CNL4Rec Sequential	**0.2288 ± 0.006**		1st
Pretrain + Fine-tune	0.2196 ± 0.004	-4.0%	2nd
Joint + Masking	0.2013 ± 0.005	-12.0%	3rd
Joint (All domains)	0.1945 ± 0.005	-15.0%	4th
Sequential (no CL)	0.1899 ± 0.005	-17.0%	5th

Our results indicate that the model is generally robust to changes in domain training order, as most ordering strategies yield comparable performance. In contrast, organizing domains by increasing size severely harms performance, leading to the largest degradation of approximately 20%.

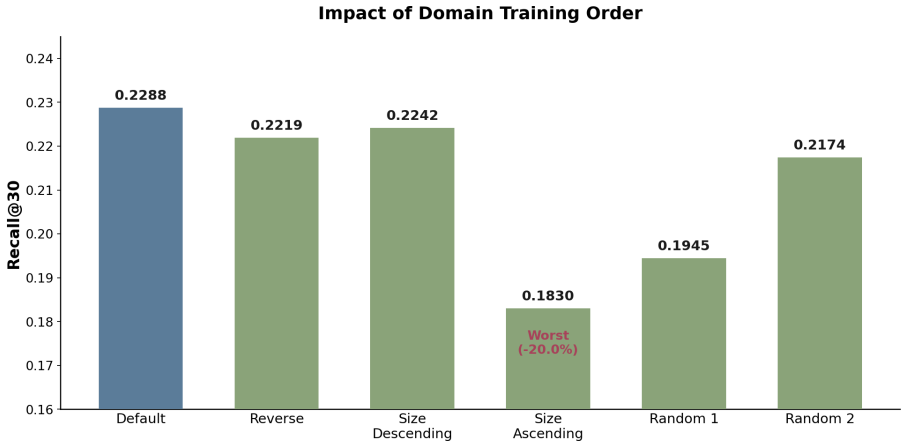

Figure 4.2: Impact of domain training order on CNL4Rec performance. The default training order achieves the best performance, while the size ascending strategy performs the worst, resulting in a relative degradation of 20.0%.

Quantifying catastrophic forgetting

The degree to which CNL4Rec mitigates catastrophic forgetting can be isolated from the training-strategy comparison in Table 4.8. Naive sequential training without

the continual-learning mechanism (Sequential, no CL) attains only 0.1899 mean Recall@30, whereas the full CNL4Rec sequential model reaches 0.2288 a $+20.5\%$ relative gain that is directly attributable to the masking and specialization machinery, since the two configurations differ only in whether parameters are protected during sequential updates. The ablation in Table 4.7 corroborates this: removing the Domain Specialization module, whose sole purpose is to shield previously important parameters, degrades mean Recall@30 by -14.0%, the largest drop among the continual-learning components. Together these results express forgetting not merely as a claim but as the measurable gap that the stability-preserving components recover.

Cross-domain knowledge transfer

Beyond retention, Table 4.8 provides evidence of positive multi-directional transfer. CNL4Rec trained sequentially (0.2288) outperforms a joint model trained on all domains simultaneously (Joint, 0.1945) by $+17.6\%$, and also surpasses a pretrain-then-fine-tune transfer baseline (0.2196) by $+4.2\%$. That sequential continual training exceeds joint multitask training indicates that knowledge acquired on earlier domains is reused beneficially rather than merely preserved: domains reinforce one another rather than competing for a shared parameter budget. This reading is consistent with the per-domain results in Tables 4.4–4.6, where CNL4Rec is the strongest method across most domains simultaneously rather than winning one domain at the expense of others the behaviour expected of fairness-oriented, multi-directional transfer. Finally, the robustness to domain ordering in Table 4.8 (most orderings yielding comparable accuracy, only the size-ascending order degrading by about 20%) shows that the transferred knowledge is not fragile to the training sequence, a desirable property for continual deployment.

4.4 Chapter Summary

This chapter presented CNL4Rec, a continual-learning framework for adaptive multi-domain recommendation. Domain masking identifies the parameters important to each domain, while domain specialization modulates gradient updates so that knowledge from previously seen domains is preserved as new domains are learned, mitigating catastrophic forgetting. A fairness objective further encourages balanced performance across domains rather than optimizing a single target. Across multiple multi-domain benchmarks, CNL4Rec attains the best mean and worst-domain performance with low cross-domain variance, outperforming representative cross-domain baselines.

The approach rests on assumptions that also delimit its scope. It works best when domain boundaries are reasonably well defined, and is therefore less suited to highly

overlapping or rapidly evolving domains; its accuracy is also sensitive to the order in which domains arrive, degrading when they are presented from smallest to largest. Because the accumulated union of importance masks grows with each new domain, memory and complexity rise as domains continue to accumulate.

These limitations motivate several directions. Relaxing the boundary assumption would allow the model to cope with fuzzy and non-stationary domains, and developing order-robust training, together with a bound on the growth of the mask union, would keep the method stable as the number of domains scales.

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.

A key factor in effective personalization lies in accurately capturing user intent. User intent can manifest in two primary forms: explicit intent, which is directly communicated through user input (e.g., natural language queries), and implicit intent, which is inferred from user behavior (e.g., browsing history, viewing patterns, past interactions). While chatbots are particularly effective at handling explicit intent through conversational understanding, recommendation engines are better suited to uncover implicit intent based on historical user data. Integrating these two capabilities allows for a more comprehensive interpretation of user needs. By embedding an ensemble-based recommendation engine into the existing chatbot framework, the system can provide real time adaptability to user preferences with improved personalization while also enriching contextual awareness through the combination of NLP driven intent recognition and recommendation models. These methods enable the chatbot to analyze complex user item

interactions, capture nuanced relationships among movies, genres, and user preferences, and generate highly personalized and diverse recommendations beyond simple content matching or collaborative filtering.

Chatbots are growing as an innovative approach to recommendation tasks, especially since they can effectively capture real time contextual data, including user preferences. Using conversational interfaces, the system can efficiently provide more personalized, dynamic, and interactive recommendations. Conversational Recommendation Systems (CRSs) typically involve multi turn interactions and can be broadly categorized by their initiative, depending on the conversation initiator and questioner. Different proposals have been raised to improve the performance of this module, including the supplementation of past user preferences, the reception of user feedback on each recommendation round, or the provision of context information through a memory network.

To be effectively integrated into a chatbot system, recommendation models must operate in real time to ensure seamless and immediate user interaction. Traditional approaches such as matrix factorization or sequential deep learning models often fall short due to high computational costs and batch oriented inference, making them unsuitable for low latency environments. In contrast, graph based deep learning models have been widely applied in recommendation scenarios owing to their ability to produce expressive node representations and extract meaningful relationships between users, items, and interactions through structured graph connections. From the first development of GCN [46], various adjustments surrounding that prototype have been proposed and achieved remarkable results [33, 101]. However, nowadays, LLMs have also received numerous notable performance wise innovations, especially in retrieval and generation tasks. Regarding this field, many state of the art LLMs display remarkable information retrieval, representation, and generation capabilities, which immensely aid the recommendation tasks [6, 15].

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.

5.1.2 Related Methodologies

This section reviews the key methodological foundations for the proposed conversational recommendation system. Since Graph Neural Network architectures have been extensively discussed in previous chapters, this section focuses on three interconnected areas: Conversational Recommender Systems, Large Language Models in Recommendation, and Retrieval-Augmented Generation.

a. Conversational Recommender Systems

Conversational Recommender Systems (CRS) represent a paradigm shift from traditional one shot recommendation toward interactive, dialogue based preference elicitation and item suggestion. Unlike conventional recommender systems that estimate user preferences solely from historical behavior, CRS enable dynamic, multi turn interactions where users can express preferences, ask questions, and provide feedback through natural language [43]. This interactive approach allows systems to actively clarify ambiguous preferences, explain recommendations, and adapt suggestions based on real time user feedback.

Recent CRS architectures increasingly leverage knowledge graphs to enhance recommendation reasoning and provide more informative responses. Zhou et al. [131] propose semantic fusion techniques that integrate knowledge graph embeddings with dialogue context to improve both recommendation accuracy and response generation quality. The knowledge graph provides structured information about item attributes, relationships, and categories that can guide both preference elicitation and recommendation explanation. Li et al. [57] address the cold-start problem in CRS by seamlessly unifying item attributes and user preferences in the conversational context, enabling systems to recommend items to new users by leveraging attribute level preferences expressed during conversation. Wang et al. [109] introduce knowledge enhanced prompt learning for unified conversational recommendation, demonstrating that incorporating structured knowledge into language model prompts significantly improves both recommendation relevance and dialogue coherence.

b. Large Language Models in Recommendation

Large Language Models (LLMs) have introduced transformative capabilities to recommendation systems, enabling sophisticated semantic understanding, natural language interaction, and few-shot adaptation to new domains without extensive task specific training [61]. The emergence of models such as GPT, LLaMA, and their variants has opened new possibilities for building more intelligent and flexible recommendation systems that can understand and generate natural language at human like levels.

Lin et al. [61] provide a comprehensive survey categorizing LLM applications in recommendation into three main paradigms. The first paradigm uses LLMs as feature encoders, where models extract rich semantic representations from textual item descriptions, user reviews, and conversational context. Unlike traditional bag of words or TF-IDF methods, LLM-based encoders capture contextual semantics, synonymy, polysemy, and complex linguistic patterns that significantly enhance item and user representation quality. The second paradigm employs LLMs as scoring functions, directly prompting models to rank or rate items based on user preferences expressed in natural language. TALLRec [6] demonstrates effective tuning frameworks for aligning LLMs with recommendation objectives through instruction tuning, while zero shot approaches [105] show that pre trained LLMs can provide reasonable recommendations without task specific training by leveraging their world knowledge and reasoning capabilities. The third paradigm positions LLMs as conversational agents, where models serve as the dialogue backbone for CRS, generating contextually appropriate responses while incorporating recommendation logic.

For conversational recommendation specifically, LLMs offer several compelling advantages. Friedman et al. [22] explore leveraging LLMs in conversational recommender systems, highlighting opportunities in generating natural, engaging dialogue that can explain recommendations, handle follow up questions, and adapt to user feedback in real time. LLMs can understand diverse ways users express preferences, from explicit statements to implicit hints, and can generate responses that feel natural and personalized. However, ensuring recommendation relevance and factual accuracy remains challenging, as LLMs may hallucinate item attributes or recommend non existent items without proper grounding mechanisms.

Key considerations when deploying LLMs for conversational recommendation include several technical and practical challenges. Computational cost and inference latency require significant attention, as LLMs demand substantial computational resources and inference latency must be managed to maintain responsive conversational interactions suitable for real time chatbot deployment. Hallucination mitigation is critical be-

cause LLMs may generate plausible sounding but factually incorrect information about items, which is particularly problematic when users rely on accurate descriptions for decision making. Personalization without fine tuning requires effective prompt engineering and context management strategies to adapt responses to individual preferences without expensive per user model training.

Effective prompt design is crucial for eliciting high quality recommendations from LLMs. Li et al. [58] propose prompt based recommendation systems demonstrating how carefully crafted prompts can guide LLMs to generate relevant suggestions. Key strategies include providing clear task instructions specifying the recommendation objective and desired output format, including relevant context such as user preferences and conversational history, using few shot examples to demonstrate expected behavior, and incorporating constraints to ensure recommendations are drawn from valid item catalogs.

c. Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) has emerged as a powerful paradigm for combining the generative capabilities of LLMs with the precision of information retrieval systems, effectively addressing key limitations of pure LLM-based approaches such as hallucination and outdated knowledge [53]. By grounding language model generation in retrieved factual information, RAG systems can produce more accurate, verifiable, and up to date responses.

For conversational recommendation applications, RAG addresses several critical challenges that limit pure LLM-based approaches. First, grounding recommendations in factual information by retrieving actual item descriptions from the catalog prevents hallucination about item attributes and ensures recommendations correspond to real, available items. Second, dynamic knowledge integration enables systems to incorporate up to date item information, pricing, and availability without retraining, supporting scenarios where item catalogs change frequently. Third, personalization through retrieval allows systems to retrieve relevant user history and past preferences as context, enabling personalized recommendations without fine-tuning the underlying language model for each user. Fourth, explainable recommendations become possible as retrieved item information can be directly referenced to generate explanations that cite specific item attributes, improving transparency and user trust in the recommendation rationale.

Hybrid retrieval strategies combining semantic similarity with structured metadata filtering have proven particularly effective for recommendation queries. For example, a query about "romantic comedies from the 2000s starring Tom Hanks" can combine embedding based retrieval of semantically similar movie descriptions with structured

filtering on genre, release year, and cast attributes, ensuring both semantic relevance and factual constraint satisfaction.

5.1.3 Contributions of Conversational Graph-based Recommendation Method

The proposed hybrid framework in this chapter integrates these complementary capabilities with the graph based recommendation models detailed in previous chapters. GNN-based models capture long term user preferences from historical interaction patterns, LLMs enable natural conversational interaction and real time preference elicitation, and RAG grounds recommendations in accurate, up to date item information. This integration creates a comprehensive conversational recommendation system that combines the strengths of structured preference learning with flexible natural language interaction. The contribution of this chapter is outlined as follows:

- We propose an innovative film recommendation chatbot that leverages not only previous interaction data but also instantaneous information.
- A graph-based deep learning approach is adopted for the past interaction information in order to determine the unique representation of each user.
- An LLM model is deployed to initiate conversations and gather current user preferences to provide final recommendations.
- Intensive experiments are conducted to challenge the efficiency of the proposed framework. The results of the models are demonstrated, and additional tests are provided to evaluate the contribution of each component separately in depth.

This work was published in "Improving Retrieval-Augmented Generation for Scalable Movie Chatbots via Graph Based Recommendation Models" 2025 [P5].

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 overall workflow of our method is illustrated in Figure 5.1. In particular, the system comprises two main components: the conversational generator and the recommendation generator. The conversational generator is responsible for interpreting user input, maintaining dialogue context, and generating coherent natural language responses. It leverages both historical interactions and real time user queries to

ensure conversational relevance and fluency. Meanwhile, the recommendation generator focuses on retrieving and ranking personalized content based on the inferred user intent and preferences. It integrates contextual signals extracted from the conversation with recommendation algorithms to provide accurate and adaptive suggestions. Together, these components form an end to end framework capable of delivering both interactive dialogue and effective recommendations in a unified manner.

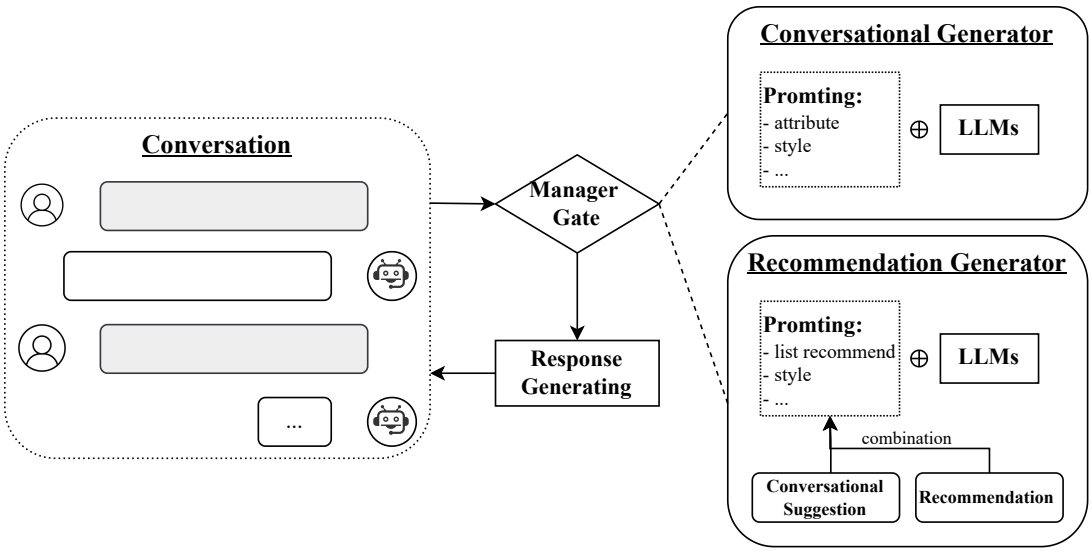

Figure 5.1: Overall workflow of the proposed CG-RAG model. The recommendation engine (GNN-based) produces behavioral ranking scores from historical user-item interactions, while the conversational engine (LLM with retrieval-augmented generation) produces context-aware relevance scores from the current dialogue. In the ensemble (fusion) module, the two score sets are normalized and combined through a weighted convex combination, yielding the final recommendation list.

In the conversational generator, we developed an LLM pipeline with a few-shot learning mechanism to simulate controlled user conversations. In this approach, conversation data is generated by providing the LLMs with a small number of example dialogues designed to elicit user intent information. The simulated dialogues are not restricted to a fixed length; instead, each conversation continues until sufficient information has been collected to make a recommendation. However, most generated conversations naturally fall within 1 to 5 turns, with each turn consisting of a user query followed by the chatbot's response. The user intent collected in each turn varies from simple, ambiguous inputs to specific requirements on multiple metadata tags.

For the simulation of user conversations, we employed three medium sized LLMs that offer a favorable balance between computational efficiency and expressive capability. These models were selected for their suitability in production level server environments while still ensuring the generation of diverse and realistic dialogue data that

effectively captures a wide range of user intents. Qwen 3B [116] is a 3 billion parameter language model trained on a multi-domain corpus developed by Alibaba Cloud. It exhibits strong conversational fluency and contextual reasoning abilities, making it suitable for simulating user chatbot interactions that require varied levels of intent complexity. Meta AI's LLaMA 3B model [98] is designed for efficiency and performance at a small scale. Its compact size allows for fast generation while maintaining enough reasoning capability to simulate coherent and goal oriented dialogue flows across short-/medium length exchanges. Gemma 2B [97], a lightweight transformer model developed by Google DeepMind, balances the speed of generation with the consistency of dialogue. It performs well in intent inference tasks and is particularly effective in producing concise user prompts with implicit or evolving preferences. The evaluation result for each model will be discussed in the following section.

Beyond gathering user intent and preferences, the system also incorporates user demographic information (such as age, gender, or other relevant attributes) to guide the chatbot in adopting a conversational style that aligns with the simulated user's profile. This demographic information enables chatbots to fine-tune their conversational style, including language, tone, role, and even cultural references, to better align with user expectations, foster rapport, and improve the overall user experience. This enables the chatbot to generate responses that not only extract intent but also reflect the user's likely communication style.

5.2.1 Overall Architecture of CG-RAG

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.

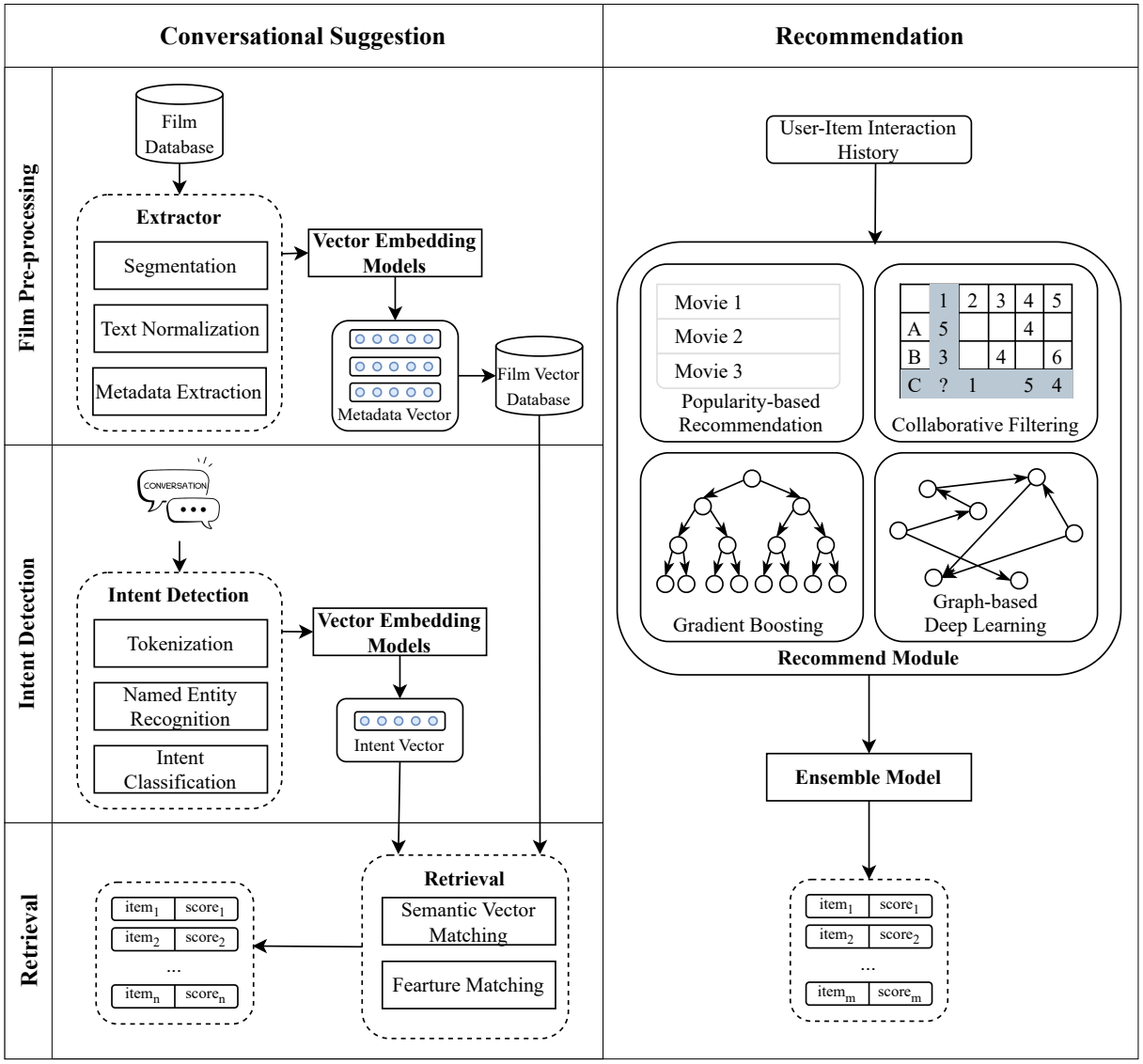

Figure 5.2: Architecture of the conversational suggestion and recommendation generator, comprising a conversational engine (intent detection producing an intent vector), a graph-based recommendation engine, and a feature-matching and retrieval layer, combined through a shared fusion layer to generate the final conversational recommendations.

Figure 5.2 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 data and is built upon a systematic empirical evaluation of representative deep recommendation methods. Specifically, we benchmark twelve models spanning graph propagation, graph contrastive learning, and embedding-alignment/self-supervised 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.

5.2.2 Conversational Suggestion Process

The conversational suggestion module is based on the retrieval method. This module processes user conversation and retrieves relevant information from the data source, which can be achieved by capturing semantic details from both the query and item descriptions.

Before retrieving relevant information for the target user, preprocessing is required for efficient intent detection. The submitted query text is normalized, which includes tokenization and stop word removal. For more targeted searches, text segments are arranged into sentences or paragraphs to identify sections that contain relevant keywords. Query manipulation is also taken into consideration, since ambiguous phrasing should be effectively addressed to ensure the accuracy of the query request to the item list. This procedure involves query expansion, reformulation, and few shot learning rewriting to accommodate the existing dataset.

The retrieval phase in modern RAG systems begins with metadata extraction, a process critical to aligning external knowledge with user intent. Contemporary approaches leverage multimodal semantic parsing to deconstruct queries into six core dimensions: plot semantics, quality indicators, genre taxonomies, cast signatures, directorial styles, and geographical contexts. This metadata ecosystem feeds into dynamic indexing strategies that maintain separate vector stores for cast filmographies, directorial trademarks, and cross lingual plot embeddings.

- Plot analysis extracts keywords matching through neural topic modeling, identifying either latent narrative structures like "nonlinear timelines" or "moral ambigu-

ity" that conventional keyword searches might miss or specific elements directly relating to what is shown in the description, such as "romantic comedy set in New York" or "suspense".

- In addition, the entire film description is also embedded sentence by sentence. This allows the model to capture fine-grained semantic nuances and contextual variations across the narrative. This sentence level embedding approach improves the detection of subtle thematic shifts and narrative structures, enabling the extraction of more precise and meaningful keywords or topics related to the film's plot.
- Rating filtration incorporates any specific rating criteria mentioned by the user, which includes but is not limited to IMDb, Rotten Tomatoes, and specialized critic circles, dynamically adjusting thresholds based on user sophistication levels.
- Genre recognition involves hierarchical taxonomies utilization, enabling granular distinctions between subgenres like "neo-noir thriller" versus "psychological horror". This provides a clearer and more distinguished result between the existing elements.
- Cast and director recognition identifies names mentioned in the query and employs knowledge graph embeddings that capture indirect relationships, for instance, recommending films featuring frequent Nolan collaborators beyond just credited roles.
- Geographic filtering includes the recognition of the country of origin for films specified in the query. This is useful for users looking for international films, such as "French thrillers" or "Japanese anime movies".

The metadata extracted from the user inquiry is the basis for the search phase. The aforementioned criteria, namely plot, rating, genre, cast, and geographic information, are incorporated for named entity recognition to filter a large corpus of candidate items. As a result, intent classification is achieved, and an intent vector aggregating user intention is formulated by a vector embedding model. This metadata driven approach enhances accuracy and personalization in movie recommendations.

Regarding the information stored in the film database, the related data are processed by an extractor, which handles description segmentation, text normalization, and metadata extraction. Segmentation and normalization ensure the accuracy of the extractor, while the extracted metadata follow the same patterns as the user intent detection module. Similarly, a vector embedding model is utilized, and each movie will have a unique metadata vector, which will be stored in the film vector database.

Once the metadata are extracted from both the user query and the film database, we proceed to filter the movie database based on these extracted attributes. The filtering process ensures that only the most relevant movies are considered before ranking. The filtering process is implemented through hybrid retrieval mechanisms combining sparse retrieval techniques and dense retrieval models. On the one hand, sparse retrieval techniques, specifically BM25, efficiently match explicit keywords and metadata tags within movie descriptions or structured fields. On the other hand, dense retrieval models are based on pretrained transformer encoders that embed both the query and movie metadata into a shared semantic vector space, enabling retrieval of candidates with latent semantic similarity beyond exact keyword overlap. This dual approach balances precision and recall, ensuring that the candidate set includes both exact matches and semantically related movies that align with nuanced user preferences.

After filtering, we obtain a refined subset of movies that align with the user's specified criteria. However, since multiple movies may match the filtering constraints, a ranking mechanism is employed to determine the most relevant results. In the proposed model, vector embeddings are generated for both the user query and the metadata rich movie descriptions. These embeddings capture semantic relationships beyond simple keyword matching, ensuring more accurate ranking. Subsequently, similarity calculation is applied, where the cosine similarity between the user query and the available movie descriptions is quantified, and the ranking is determined based on the calculated score. A higher similarity score indicates a closer match between the movie and the user's intent; thus, the items are ranked in descending order, and the top ranked items are presented to the user as the most relevant recommendations.

5.2.3 Prediction, Optimization, and Generation

The score between user u and item i in the ensemble module is computed as a weighted score fusion (convex combination) of the two engines:

$$y_{u,i} = \alpha y^R_{u,i} + (1 - \alpha) y^C_{u,i}, \tag{5.1}$$

where $\alpha \in [0, 1]$ is the hyperparameter to control the importance between two engines. The recommendation pipeline is optimized using Bayesian Personalized Ranking loss, which is defined as:

$$\mathcal{L}^R = -\sum ln(\sigma(y^R_{u,i} - y^R_{u,i'}) \tag{5.2}$$

where σ is the sigmoid function, u, i is a positive pair that has been observed, and u, i' is an unobserved interaction.

The answer generator produces natural language responses based on the set of ag-

gregated candidate items and the current dialogue context. It follows a structured generation process that incorporates four key elements: acknowledgment of the user's query, concise highlighting of relevant suggestions, explanation of recommendation rationale through item attributes, and follow up prompts to elicit further user preferences. To ensure factual grounding and linguistic fluency, the generation module applies retrieval aware decoding strategies minimizing randomness for factual metadata and allowing greater flexibility for thematic elaboration. Additionally, the system adapts to each user via stylistic adjustments aligned with individual communication styles. The three aforementioned LLM models, including Qwen-3B [116], LLaMA-3B [98], and Gemma-2B [97], are also utilized to achieve the desirable outcome.

5.3 Experimental Setting and Results

5.3.1 Experimental Settings

Dataset

This research uses the Movielens-1M dataset released by the GroupLens Research team at the University of Minnesota [29]. This dataset is described in Chapter 1.

Additionally, this research is grounded in a real-world application scenario, utilizing a proprietary dataset derived from user interaction logs on TV360, a commercial streaming platform developed by Viettel Group. While the raw data cannot be released publicly due to confidentiality constraints, relevant experimental outcomes associated with this study can be disclosed and are presented in the subsequent section. Key statistics of the dataset are summarized as shown in Table 5.1:

Table 5.1: TV360 dataset division statistic

	User	Movie	Interaction
Train	9,932	2,316	923,800
Test	9,477	1,923	263,943
Validation	9,548	1,912	131,972
Total	10,000	2,433	1,319,715

State of the art approaches utilize advanced LLMs to generate diverse conversation templates that ensure naturalness and variety in dialogues. The quality and authenticity of generated conversations is ensured by a human annotator, who evaluates conversations at both the turn level and dialogue level granularities. Additionally, human review-

ers serve as a secondary quality control layer, providing more comprehensive evaluation of annotated conversations. Their role involves validating annotation consistency, identifying systematic biases, and ensuring adherence to established quality standards. To ensure the quality and authenticity of the conversational dataset, we employed a rigorous human validation protocol. A team of five trained annotators with prior experience in dialogue annotation was recruited. The annotators first manually authored 100 seed conversations (10% of the target dataset) from sampled user interaction histories to establish high-quality demonstrations. Using these seed dialogues, we applied few-shot prompting with LLM to automatically generate the remaining 900 conversations. All 900 LLM-generated conversations were then reviewed by the five annotators at both the turn level and the dialogue level; a total of 124 conversations (13.8%) were revised or regenerated owing to semantic inconsistencies or hallucinated film titles. This procedure yields a dataset whose quality is human-verified rather than purely model-generated.

The conversation dataset statistics (5.2) reveal that both the MovieLens-1M (ML1M) and TV360 datasets contain an equal number of conversations, each of which has 1000 conversations. However, the TV360 dataset exhibits slightly more interaction per conversation, with an average of 5.6 rounds compared to 4.5 in MovieLens-1M. Additionally, conversations in TV360 are longer on average, containing 513 tokens per conversation versus 445 tokens in MovieLens-1M.

Table 5.2: Conversation Dataset Statistics. There are 1000 conversations generated for 1000 distinct users for both datasets, each of which varies in length and conversation rounds. The table showcases the average statistics of those conversations.

	Movielens-1M	TV360
Number of Conversations	1000	1000
Average Rounds per Conversation	4.5	5.6
Average Length of Conversation (tokens/words)	445	513

Environment and Evaluation Metrics

The experiments are conducted on an Intel Xeon E5-2698 v4 processor, a 20-core/40-thread CPU with a base frequency of 2.2 GHz and turbo boost up to 3.6 GHz, fabricated on Intel's 14 nm Broadwell-EP architecture. The system is powered by four NVIDIA V100 GPUs, leveraging Volta architecture with 5,120 CUDA cores and 32GB HBM2 memory per GPU.

Regarding evaluation metrics, we utilize Recall and Precision top-K to evaluate model performance. In this study, the value of K is selected in $\{30, 50\}$

Hyper-parameters

All models are implemented using PyTorch, with the embedding size of the user, item, and hidden layer adjusted in $\{16, 32, 64\}$. The learning rate is set to $1e-3$ while the number of propagation layers using graph-based techniques is set in the range of $\{1, 2, 3\}$, γ is set to $1e-5$, we conduct experiments within 5 different seeds and 50 epochs.

5.3.2 Experimental Results

Recommendation Engine Result

The running results of all models are demonstrate in Table 5.3.

Table 5.3: 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

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 improve over standard graph propagation approaches 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.

Performance on MovieLens-1M

On MovieLens-1M, NCL achieves the best results across all four metrics, recording R@30 = 0.2497, P@30 = 0.1990, R@50 = 0.3351, and P@50 = 0.1685. Compared to the second-ranked model XSimGCL (R@30 = 0.2209), NCL delivers a relative gain of approximately $+13.0\%$ on R@30 and $+15.3\%$ on R@50, with the margin holding consistently across both Recall and Precision metrics. This indicates that NCL produces well-calibrated representations that balance coverage and targeting quality simultaneously. XSimGCL and SimGCL rank second and third respectively (R@30 of 0.2209 and 0.2125), confirming that cross-layer and standard graph contrastive learning are the next most effective strategies, though the gap between them is modest at roughly $+4\%$. At the bottom, MixGCF and SGL both record R@30 = 0.0192, the lowest scores in the table, suggesting that hard-negative mixing and edge-dropout augmentation are poorly calibrated for the dense interaction structure of MovieLens-1M.

Performance on TV360

The ranking changes considerably on TV360. DirectAU emerges as the best performing model with R@30 = 0.1932 and R@50 = 0.2714, ahead of the second-ranked XSimGCL (R@30 = 0.1725) by approximately $+12.0\%$ on R@30. DirectAU's alignment-and-uniformity objective optimizes the geometry of the embedding space directly, without relying on graph augmentation, which proves advantageous on TV360 where the interaction graph is sparser and less community-structured than MovieLens-1M. In contrast, NCL which dominated MovieLens-1M collapses to R@30 = 0.0915 on TV360, a reduction of over 60% in absolute recall relative to its MovieLens-1M score. This sharp drop reveals that NCL's neighborhood contrastive objective is tightly coupled to graph density and strong community structure, properties present in MovieLens-1M but not in TV360. Standard graph propagation models (LightGCN, GAT, GraphSAGE) also decline markedly on TV360, clustering between R@30 = 0.032 and 0.042, compared to 0.148 0.165 on MovieLens-1M, further confirming that plain message-passing is insufficient when the interaction graph provides weaker structural signal.

A deeper analysis attributes this cross-dataset gap to differences in graph topology rather than raw data volume. Although TV360 originates from a large-scale platform, its conversational interaction graph is structurally sparser and far less community structured than MovieLens-1M: it exhibits a lower effective node degree and a flatter, more popularity-skewed degree distribution, together with a weaker clustering tendency. Message passing and graph-contrastive models such as NCL, LightGCN, and GAT implicitly assume homophily and smooth community structure they propagate and align embed-

dings over local neighborhoods so when these structural properties are weak, neighborhood aggregation amplifies noise rather than signal, which explains why NCL collapses by over 60% in absolute Recall and the plain propagation models cluster at R@30 = 0.032–0.042 on TV360. In contrast, DirectAU performs best on TV360 precisely because its geometry-based alignment optimizes embedding uniformity and alignment directly in the representation space and does not rely on rich neighborhood structure, while XSimGCL remains robust on both datasets because its cross-layer contrastive objective regularizes representations without overfitting to any single graph topology. This behavioral contrast is consistent with the nature of the two domains: MovieLens-1M consists of explicit ratings from engaged users, producing stable and tightly clustered taste communities, whereas TV360 reflects broad, transient, popularity-driven OTT viewing in which per-user preferences are less consistent and community boundaries are blurred. The gap is therefore not an artifact but an informative finding: it demonstrates that the superiority of any single model is contingent on the underlying graph structure, which is exactly what motivates the ensemble design combining a community exploiting model (NCL), a geometry aligned model (DirectAU), and a topology-robust backbone (XSimGCL) to maintain quality across both dense and sparse regimes.

Best Model Selection

Based on the analysis of Table 5.3, the selection of the optimal model depends on whether the criterion is peak single-dataset performance or cross-dataset robustness.

For peak performance, NCL is the optimal choice on MovieLens-1M and DirectAU on TV360, each dominating its respective dataset by a clear margin. However, their strong dataset-specificity limits their reliability as standalone models in a general-purpose system.

XSimGCL stands out as the most robust single model, ranking second on both datasets with competitive absolute scores (R@30 = 0.2209 on ML-1M and 0.1725 on TV360). Its cross-layer contrastive objective provides representational stability across varying data distributions without overfitting to the structural characteristics of any single graph.

Taken together, the results of Table 5.3 provide a clear empirical basis for selecting NCL, DirectAU, and XSimGCL as the three models for ensemble integration. NCL contributes peak performance on dense, community-rich datasets; DirectAU contributes robustness on sparse datasets through geometry-based alignment; and XSimGCL provides consistent cross-dataset performance as a stable backbone.

Conversational Retrieval Engine Result

Qwen-3B consistently ahead of the other models in all Recall@K metrics, achieving the highest recall scores at every level from *k=10* (0.207) to *k=200* (0.888), indicating its superior ability to retrieve relevant movie recommendations in the K outputs. LLaMA-3B follows closely behind, with slightly lower recall values (for example, 0.204 at *k=10* and 0.869 at *k=200*), while Gemma-2B lags behind the two, particularly at lower *k* values. Despite Qwen-3B's superior performance, it also incurs the highest response time at 2.42 seconds, suggesting a trade off between accuracy and latency. Overall, the data suggest that Qwen-3B offers the best retrieval quality, albeit at a marginally higher computational cost, but is still adaptable to real life scenarios.

Table 5.4: 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	<u>0.204</u>	<u>0.227</u>	<u>0.239</u>	<u>0.291</u>	<u>0.428</u>	<u>0.869</u>	<u>2.28</u>
Qwen-3B	**0.207**	**0.232**	**0.246**	**0.299**	**0.435**	**0.888**	**2.42**

Bold = Best, <u>Underline</u> = 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 5.5: 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	<u>0.072</u>	**0.086**	<u>0.093</u>	<u>0.112</u>	<u>0.226</u>	<u>0.731</u>	<u>2.13</u>
Qwen-3B	**0.073**	<u>0.083</u>	**0.095**	**0.118**	**0.233**	**0.740**	**2.18**

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

Ensemble Engine Result

Table 5.6: 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	<u>0.2046</u>	<u>0.2353</u>	<u>0.2503</u>	<u>0.3012</u>	**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	<u>0.1106</u>	<u>0.1582</u>	<u>0.1808</u>	<u>0.2606</u>

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

5.4 Chapter Summary

This chapter introduced CG-RAG, a hybrid conversational recommendation architecture that bridges long-term user behavior with real-time conversational intent. A graph neural network models structured long-term preferences, a large language model handles intent extraction and natural-language dialogue, and a hybrid retrieval stage combining sparse (BM25) and dense retrieval grounds recommendations in the actual item catalog. Because candidate items are constrained to retrieved catalog entries and the generator is conditioned on retrieved metadata, the system produces context-aware, grounded recommendations through interactive dialogue rather than static ranked lists.

A few limitations should be acknowledged. The evaluation rests largely on conversations that were drafted automatically and then revised by human annotators, rather than collected from real users at scale, and the language-model component introduces inference latency that may constrain real-time deployment. The dissertation also stops short of reporting dedicated quantitative metrics for hallucination and cross-turn consistency.

Each of these gaps suggests a way forward. Evaluating the system on genuine user conversations, and benchmarking it against LLM-based recommendation while reporting computational cost alongside accuracy, would place its performance on firmer ground. Latency could be reduced through LLM compression, approximate nearest neighbor retrieval, and caching, and the reliability of the dialogue could be quantified with explicit grounding metrics, such as the fraction of recommended items matching the retrieved catalog, together with measures of multi-turn consistency.

Conclusions

Summary of Contributions

This dissertation develops deep models for modern recommendation systems, organized around three major contributions that correspond to the three fundamental challenges identified in the introduction: robust and scalable modeling of canonical and auxiliary data, adaptive multi-domain recommendation, and conversational recommendation.

The first contribution is a collection of models for robust and scalable deep modeling of canonical and auxiliary data, addressing scalability, data sparsity, and cold-start. EfficientRec replaces explicit user identifiers with behavior-driven representations through deep interaction modeling, soft clustering, and contrastive learning, substantially reducing model size and enabling efficient large-scale deployment while remaining robust under extreme sparsity; its effectiveness is confirmed by online A/B testing on a production streaming platform. Complementing it, GIFT4Rec fuses collaborative interaction signals with heterogeneous side information through graph-based relational learning to alleviate sparsity and cold-start, while MaskSimGCL introduces masked graph contrastive learning that strengthens representation robustness, yielding consistent gains in both warm-start and cold-start settings.

The second contribution is an adaptive multi-domain recommendation model based on continual learning. CNL4Rec employs domain masking and domain specialization to regulate parameter updates across domains, preserving previously acquired knowledge while adapting to new domains and thereby mitigating catastrophic forgetting. Unlike conventional cross-domain methods that optimize exclusively for a target domain, it delivers balanced performance across all participating domains, incorporating fairness into the multi-domain objective.

The third contribution is a hybrid conversational recommendation architecture that bridges long-term user behavior with real-time, temporally evolving user intent. The proposed CG-RAG architecture integrates GNN-based structured preference modeling with large language models and retrieval-augmented generation, combining hybrid sparse-dense retrieval to ground recommendations in the actual item catalog. This design unifies symbolic interaction knowledge with semantic language understanding to enhance recommendation accuracy, interpretability, and context-aware personalized dialogue.

Although deep recommendations are often regarded as black boxes, the models proposed in this dissertation each expose an intrinsic source of explanation. EfficientRec assigns users to soft preference clusters, so a recommendation can be justified in terms of cluster membership and behavioral similar users. GIFT4Rec is explainable at the attribute level: its attention-based Weight-Generated module yields a per-user fusion weight α that reveals how much the behavioral and auxiliary signals contributed, while the attention distribution indicates which side-information attributes mattered. The mask-based mechanisms of MaskSimGCL and CNL4Rec expose which representation dimensions and, for CNL4Rec, which domain-specific parameters drive a prediction. CG-RAG offers the most transparent form of explanation: because every recommendation is grounded in retrieved catalog items with explicit metadata, its provenance is traceable, and the conversational interface can surface a natural-language justification directly to the user.

Overall, the proposed models achieve well performance across multiple public benchmarks and a real-world industrial dataset, establishing both theoretical insights and practical solutions that bridge the gap between academic research and large-scale industrial deployment.

Limitations

Despite its contributions, this dissertation has several limitations that point to future research.

Computational Resources: Training graph neural networks, contrastive modules, and continual learning updates remains resource intensive at large scale, even though EfficientRec already lowers the user's side memory footprint to $O(K \cdot d)$. This can be mitigated through model compression (quantization, pruning, distillation), distributed and mixed precision training, and inference time techniques such as approximate nearest neighbour retrieval, LoRA, and caching.

Auxiliary Information Quality, Robustness, and Fairness: Side information is often noisy, incomplete, or biased, and biased attributes can degrade fairness. The dissertation partially mitigates this (noise via MaskSimGCL, missing data via GIFT4Rec fusion, fairness via the CNL4Rec objective), and further solutions include denoising and uncertainty aware weighting for noise, imputation and self-supervised pretraining for missing data, and inverse propensity or counterfactual debiasing with exposure-parity constraints for bias and fairness.

Domain Boundary Assumptions: The continual-learning framework assumes reasonably clear domain boundaries and is less effective for highly overlapping or rapidly evolv-

ing domains, or for scenarios involving task reordering and domain recurrence.

Conversational System Reliability: Integrating GNNs, large language models, and retrieval-augmented generation enhances reasoning and interaction but introduces inference latency, hallucination risk, and controllability, privacy, and safety concerns that are not yet fully resolved.

Evaluation Scope: Although experiments span multiple public benchmarks and a real-world industrial dataset, they cannot cover the full diversity of practical scenarios; generalization to domains such as finance, healthcare, and education requires further empirical validation and domain-specific adaptation.

Future Work

Future research can extend the findings of this dissertation in several promising directions.

Production Oriented Optimization: Developing fine tuning strategies that continuously adapt models based on real user interactions, system logs, and behavioral drift. This includes incremental model updates under strict latency constraints, automated hyperparameter tuning in live environments, and robust monitoring pipelines capable of detecting performance degradation or distributional shifts. Incorporating online learning loops and real time feedback mechanisms would enhance system adaptability to evolving user preferences.

Multimodal Personalization: Expanding the recommendation framework beyond structured interaction and textual signals to incorporate multimodal information including images, audio, video, and device level behavioral cues. Multimodal fusion techniques such as vision language modeling, cross modal contrastive learning, and unified embedding spaces can significantly enrich user and item representations, leading to more expressive semantic understanding and improved personalization across diverse content formats.

Efficient Deployment: Exploring lightweight architectures for on device inference, cost efficient LLM compression, and privacy preserving multimodal learning for deploying recommendation systems at scale. Context aware adaptive recommendation agents that operate seamlessly across channels, devices, and service domains represent an important direction for practical deployment.

Fairness and Sustainability: Incorporating explicit fairness constraints and bias mitigation techniques into the optimization objectives. Additionally, considering energy efficiency, environmental sustainability, and long term user satisfaction as essential fac-

tors for future large scale artificial intelligence systems.

Cross-Domain Generalization: The components proposed in this dissertation are domain-agnostic in principle: any application with user-item interactions plus auxiliary context can reuse the canonical-auxiliary modeling, the ID-free scalable representation (EfficientRec), the robustness mechanisms (MaskSimGCL), continual adaptation (CNL4Rec), and the conversational RAG interface (CG-RAG). Generalizing to high-stakes domains, however, requires domain-specific adaptation along three axes objectives, constraints, and validation summarized in Table 5.7.

Table 5.7: Generalization of the proposed system to other application domains.

Domain	**What transfers**	**Domain-specific challenge**	**Recommended adaptation**
Finance	Cold-start and sparsity handling; continual learning for non-stationary markets; conversational advisory	Regulatory suitability, auditability, concept drift, risk constraints	Constraint-aware ranking; strong interpretability and audit trails; fairness in exposure/lending
Healthcare	Auxiliary fusion for sparse patient data; robustness to noise; grounded conversational decision support	Safety-critical decisions, privacy (HIPAA/GDPR), clinician interpretability, label scarcity	Privacy-preserving/federated training; uncertainty quantification; human-in-the-loop; strict anti-hallucination grounding; clinical validation
Education	Temporal/sequential preference modeling; cold-start for new learners; personalized tutoring dialogue	Long-term learning gains vs. engagement; pedagogical and equity constraints	Optimize for learning outcomes (not only clicks); knowledge-tracing signals; fairness across socioeconomic groups

Across all three domains, three cross-cutting recommendations apply: (i) replace or augment auxiliary features with domain-appropriate signals; (ii) treat fairness, privacy, and interpretability as first-class design constraints rather than post-hoc additions; and (iii) re-validate empirically on domain benchmarks before deployment, since accuracy gains observed on recommendation datasets may not transfer directly to outcome-oriented domains.

Together, these directions point toward a future in which recommendation systems are deeply integrated into operational environments, continuously optimized through real-world signals, and enhanced by rich multimodal understanding to deliver more adaptive, personalized, and intelligent user experiences.

List of Publications

[P 1] Vu Hong Quan, Le Hoang Ngan, Le Minh Duc, **Nguyen Tran Ngoc Linh**, Le Hoang Quynh. "EfficientRec: An Unlimited User Scale Recommendation System Based on Clustering and User's Interaction Embedding Profile." *In Asian Conference on Intelligent Information and Database Systems (ACIIDS)*, pp. 681-696. Springer Nature Singapore, 2022. ***(Scopus Conference)***

[P 2] **Nguyen Tran Ngoc Linh**, Vu Chi Dung, Le Hoang Ngan, Hoang Anh Dung, Phan Xuan Hieu, Ha Quang Thuy, Le Hoang Quynh, Tran Mai Vu. "GIFT4Rec: An Effective Side Information Fusion Technique Apply to Graph Neural Network for Cold-Start Recommendation." *In Asian Conference on Intelligent Information and Database Systems (ACIIDS)*, pp. 334-345. Springer Nature Singapore, 2023. ***(Scopus Conference)***

[P 3] **Nguyen Tran Ngoc Linh**, Le Hoang Ngan, Hoang Anh Dung, Phan Xuan Hieu, Ha Quang Thuy, Le Hoang Quynh, Tran Mai Vu. "The Masked Simple Graph Contrastive Learning for Recommendation." *In 2024 16th International Conference on Knowledge and System Engineering (KSE)*, pp. 156–160. IEEE, 2024.***(Scopus Conference)***

[P 4] **Nguyen Tran Ngoc Linh**, Vu Chi Dung, Le Hoang Ngan, Hoang Anh Dung, Phan Xuan Hieu, Ha Quang Thuy, Le Hoang Quynh, Tran Mai Vu. "Continual Learning Based on Task Masking for Multi-domain Recommendation." *In Asian Conference on Intelligent Information and Database Systems (ACIIDS)*, pp. 257-266. Springer Nature Singapore, 2024.***(Scopus Conference)***

[P 5] **Nguyen Tran Ngoc Linh**, Hoang Anh Dung, Tran Manh Cuong, Vu Minh Thanh, Vu The Anh, Nguyen Xuan Bach, Bui Tuan Nghia, Le Hoang Quynh, Vuong Thi Hai Yen, Tran Mai Vu. "Improving Retrieval-Augmented Generation for Scalable Movie Chatbots via Graph Based Recommendation Models" Submitted to IEEE Access- under minor revision (Round 3), 2026. ***(Q1 Journal)***

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] F. M. Almutairi, Y. Wang, D. Wang, E. Zhao, and N. D. Sidiropoulos, "etree: Learning tree-structured embeddings," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI '21)*, vol. 35, no. 8. AAAI Press, 2021, pp. 6609–6617.

[4] M. Ananyeva, O. Lashinin, V. Ivanova, S. Kolesnikov, and D. I. Ignatov, "Towards interaction-based user embeddings in sequential recommender models," in *Proceedings of the 5th Workshop on Online Recommender Systems and User Modeling (ORSUM@RecSys 2022)*, ser. CEUR Workshop Proceedings, vol. 3303. Seattle, WA, USA: CEUR-WS.org, 2022.

[5] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *International Conference on Learning Representations*, 2015.

[6] K. Bao, J. Zhang, Y. Zhang, W. Wang, F. Feng, and X. He, "Tallrec: An effective and efficient tuning framework to align large language model with recommendation," in *Proceedings of the 17th ACM Conference on Recommender Systems*. ACM, 2023, pp. 1007–1014.

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

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

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

[10] X. Cai, C. Huang, L. Xia, and X. Ren, "Lightgcl: Simple yet effective graph contrastive learning for recommendation," *arXiv preprint arXiv:2302.08191*, 2023.

[11] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proceedings of the European Conference on Computer Vision (ECCV '18)*. Springer, 2018, pp. 132–149.

[12] J. Chen, H. Dong, X. Wang, F. Feng, M. Wang, and X. He, "Bias and debias in recommender system: A survey and future directions," *ACM Transactions on Information Systems*, vol. 41, no. 3, pp. 1–39, 2023.

[13] 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.

[14] Y. Chen, G. Huzhang, A. Zeng, Q. Yu, H. Sun, H.-Y. Li, J. Li, Y. Ni, H. Yu, and Z. Zhou, "Clustered embedding learning for recommender systems," in *Proceedings of the ACM Web Conference 2023 (WWW '23)*. Austin, TX, USA: ACM, 2023, pp. 631–641.

[15] Z. Chen, "Palr: Personalization aware llms for recommendation," *arXiv preprint arXiv:2305.07622*, 2023.

[16] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir *et al.*, "Wide & deep learning for recommender systems," in *Proceedings of the 1st workshop on deep learning for recommender systems*, 2016, pp. 7–10.

[17] 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.

[18] DeepRec, "Adaptive embedding," https://github.com/alibaba/DeepRec/blob/main/docs/docs_en/Adaptive-Embedding.md, 2021, alibaba DeepRec Project.

[19] J. C. Dunn, "A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters," *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, 1973.

[20] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," in *The world wide web conference*, 2019, pp. 417–426.

[21] W. Fan, Y. Ding, L. Ning, S. Wang, H. Li, D. Yin, T.-S. Chua, and Q. Li, "A survey on rag meeting llms: Towards retrieval-augmented large language models," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and*

Data Mining, ser. KDD '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 6491–6501.

[22] L. Friedman, S. Ahuja, D. Allen, T. Tan, H. Sidahmed, C. Long, J. Xie, G. Schubiner, A. Patel, H. Lara *et al.*, "Leveraging large language models in conversational recommender systems," *arXiv preprint arXiv:2305.07961*, 2023.

[23] C. Gao, Y. Zheng, N. Li, Y. Li, Y. Qin, J. Piao, Y. Quan, J. Chang, D. Jin, X. He *et al.*, "A survey of graph neural networks for recommender systems: Challenges, methods, and directions," *ACM Transactions on Recommender Systems*, vol. 1, no. 1, pp. 1–51, 2023.

[24] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, H. Wang, and H. Wang, "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997*, vol. 2, no. 1, 2023.

[25] S. S. Ghaemmaghami and A. Salehi-Abari, "Deepgroup: Group recommendation with implicit feedback," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM '21)*. ACM, 2021, pp. 3408–3412.

[26] A. I. Griva, A. D. Boursianis, L. A. Iliadis, P. Sarigiannidis, G. Karagiannidis, and S. K. Goudos, "Model-agnostic meta-learning techniques: A state-of-the-art short review," in *2023 12th International Conference on Modern Circuits and Systems Technologies (MOCAST)*. IEEE, 2023, pp. 1–4.

[27] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "Deepfm: A factorization-machine based neural network for ctr prediction," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI '17)*. Melbourne, Australia: AAAI Press, 2017, pp. 1725–1731.

[28] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[29] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 5, no. 4, p. 19, Dec. 2015.

[30] 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.

[31] R. He and J. McAuley, "Vbpr: visual bayesian personalized ranking from implicit feedback," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.

[32] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 173–182.

[33] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 639–648.

[34] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 5–53, 2004.

[35] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in *International Conference on Learning Representations (ICLR)*, 2016, arXiv:1511.06939. [Online]. Available: https://arxiv.org/abs/1511.06939

[36] C. Hou, Y. Zhou, Y. Cao, and T.-Y. Liu, "Ecat: An entire space continual and adaptive transfer learning framework for cross-domain recommendation," in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '24. ACM, 2024, pp. 2885–2889.

[37] M. Hou, L. Wu, Y. Liao, Y. Yang, Z. Zhang, Y. Wang, C. Zheng, H. Wu, and R. Hong, "A survey on generative recommendation: Data, model, and tasks," *AI Open*, 2026.

[38] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *IEEE International Conference on Data Mining*. IEEE, 2008, pp. 263–272.

[39] H. Huang, X. Lou *et al.*, "Diit: A domain-invariant information transfer method for industrial cross-domain recommendation," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, ser. CIKM '24. ACM, 2024.

[40] T. Huang, Y. Dong, M. Ding, Z. Yang, W. Feng, X. Wang, and J. Tang, "Mixgcf: An improved training method for graph neural network-based recommender systems," in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021, pp. 665–674.

[41] A. Iovine, F. Narducci, and G. Semeraro, "Conversational recommender systems and natural language:: A study through the converse framework," *Decision Support Systems*, vol. 131, p. 113250, 2020.

[42] E. Ivannikova, S. A. Khan, W. Oyomno, Q. Fu, K. Tan, A. Flanagan *et al.*, "Federated collaborative filtering for privacy-preserving personalized recommendation system," *CoRR*, 2019.

[43] D. Jannach, A. Manzoor, W. Cai, and L. Chen, "A survey on conversational recommender systems," *ACM Computing Surveys*, vol. 54, no. 5, pp. 1–36, 2021.

[44] J.-Y. Jiang, P. H. Chen, C.-J. Hsieh, and W. Wang, "Clustering and constructing user coresets to accelerate large-scale top-k recommender systems," in *Proceedings of The Web Conference 2020 (WWW '20)*. ACM, 2020, pp. 2177–2187.

[45] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *2018 IEEE international conference on data mining (ICDM)*. IEEE, 2018, pp. 197–206.

[46] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations*, 2017.

[47] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[48] 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.

[49] H. Koohi and K. Kiani, "User based collaborative filtering using fuzzy c-means," *Measurement*, vol. 91, pp. 134–139, 2016.

[50] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 426–434.

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

[52] 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.

[53] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 9459–9474.

[54] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, "Neural attentive session-based recommendation," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 1419–1428.

[55] L. Li, Y. Zhang, and L. Chen, "Prompt distillation for efficient llm-based recommendation," in *Proceedings of the 32nd ACM international conference on information and knowledge management*, 2023, pp. 1348–1357.

[56] M. Li, L. Wen, and F. Chen, "A novel collaborative filtering recommendation approach based on soft co-clustering," *Physica A: Statistical Mechanics and its Applications*, vol. 561, p. 125140, 2021.

[57] S. Li, W. Lei, Q. Wu, X. He, P. Jiang, and T.-S. Chua, "Seamlessly unifying attributes and items: Conversational recommendation for cold-start users," *ACM Transactions on Information Systems*, vol. 39, no. 4, pp. 1–29, 2021.

[58] X. Li, Y. Zhang, and E. C. Malthouse, "Pbnr: Prompt-based news recommender system," *arXiv preprint arXiv:2304.07862*, 2023.

[59] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, "Variational autoencoders for collaborative filtering," in *Proceedings of the 2018 world wide web conference*, 2018, pp. 689–698.

[60] D. Lim, F. Hohne, X. Li, S. L. Huang, V. Gupta, O. Bhalerao, and S. N. Lim, "Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 20 887–20 902.

[61] J. Lin, X. Dai, Y. Xi, W. Liu, B. Chen, H. Zhang, Y. Liu, C. Wu, X. Li, C. Zhu, H. Guo, Y. Yu, R. Tang, and W. Zhang, "How can recommender systems benefit from large language models: A survey," *ACM Transactions on Information Systems*, vol. 42, no. 6, pp. 1–47, 2024.

[62] J. Lin, X. Dai, Y. Xi, W. Liu, B. Chen, H. Zhang, Y. Liu, C. Wu, X. Li, C. Zhu *et al.*, "How can recommender systems benefit from large language models: A survey," *ACM Transactions on Information Systems*, vol. 43, no. 2, pp. 1–47, 2025.

[63] 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.

[64] A. Liu, "Title of the paper," *Journal Name*, vol. 1, pp. 1–10, 2023.

[65] L. Liu, Y. Wang, T. Wang, D. Guan, J. Wu, J. Chen, R. Xiao, W. Zhu, and F. Fang, "Continual transfer learning for cross-domain click-through rate prediction at taobao," in *Companion Proceedings of the ACM Web Conference 2023*, 2023, pp. 346–350.

[66] S. Liu, Y. Liu, X. Zhang, C. Xu, J. He, and Y. Qi, "Meta-learned user preference estimator with attention network for cold-start recommendation," in *Journal of Physics: Conference Series*, vol. 2504, no. 1. IOP Publishing, 2023, p. 012028.

[67] S. Liu, I. Ounis, C. Macdonald, and Z. Meng, "A heterogeneous graph neural model for cold-start recommendation," in *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, 2020, pp. 2029–2032.

[68] Z. Liu, Y. Ma, Y. Ouyang, and Z. Xiong, "Contrastive learning for recommender system," *arXiv preprint arXiv:2101.01317*, 2021.

[69] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

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

[71] M. Mao, J. Lu, G. Zhang, and J. Zhang, "A novel collaborative filtering recommendation approach based on soft co-clustering," *Physica A: Statistical Mechanics and its Applications*, vol. 561, p. 125230, 2021.

[72] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[73] J. E. Nalavade, C. S. Kolli, and S. N. P. Kumar, "Deep embedded clustering with matrix factorization based user rating prediction for collaborative recommendation," *Multiagent and Grid Systems*, vol. 19, no. 2, pp. 169–185, 2023.

[74] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The adaptive web: methods and strategies of web personalization*. Springer, 2007, pp. 325–341.

[75] S. Rendle, "Factorization machines," in *2010 IEEE International conference on data mining*. IEEE, 2010, pp. 995–1000.

[76] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2009, pp. 452–461.

[77] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: An open architecture for collaborative filtering of netnews," in *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, 1994, pp. 175–186.

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

[79] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[80] E. Rusak, P. Reizinger, A. Juhos, O. Bringmann, R. S. Zimmermann, and W. Brendel, "Infonce: Identifying the gap between theory and practice, june 2024," *URL http://arxiv. org/abs/2407.00143*, 2024.

[81] G. Salton, A. Wong, and C.-S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[82] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, 2001, pp. 285–295.

[83] 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.

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

[85] 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.

[86] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823.

[87] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "Autorec: Autoencoders meet collaborative filtering," in *Proceedings of the 24th international conference on World Wide Web*, 2015, pp. 111–112.

[88] S. Shi, W. Ma, M. Zhang, Y. Zhang, X. Yu, H. Shan, Y. Liu, and S. Ma, "Beyond user embedding matrix: Learning to hash for modeling large-scale users in recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. Virtual Event, China: ACM, 2020, pp. 319–328.

[89] 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.

[90] S. Singhal and K. Kaur, "Group formation technique based on deep embedded clustering and similarity for group recommendation system," *Knowledge and Information Systems*, 2025.

[91] W. Song, C. Shi, Z. Xiao, Z. Duan, Y. Xu, M. Zhang, and J. Tang, "Autoint: Automatic feature interaction learning via self-attentive neural networks," in *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019, pp. 1161–1170.

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

[93] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer," in *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019, pp. 1441–1450.

[94] Q. Tan, N. Liu, X. Zhao, H. Yang, J. Zhou, and X. Hu, "Learning to hash with graph neural networks for recommender systems," in *Proceedings of The Web Conference 2020 (WWW '20)*. Taipei, Taiwan: ACM, 2020, pp. 1988–1998.

[95] Q. Tan, J. Zhang, N. Liu, X. Huang, H. Yang, J. Zhou, and X. Hu, "Dynamic memory based attention network for sequential recommendation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 5, 2021, pp. 4384–4392.

[96] J. Tang and K. Wang, "Personalized top-n sequential recommendation via convolutional sequence embedding," in *Proceedings of the eleventh ACM international conference on web search and data mining*, 2018, pp. 565–573.

[97] G. Team, T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, S. Pathak, L. Sifre, M. Rivière, M. S. Kale, J. Love *et al.*, "Gemma: Open models based on gemini research and technology," *arXiv preprint arXiv:2403.08295*, 2024.

[98] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[99] M. Vartak, A. Thiagarajan, C. Miranda, J. Bratman, and H. Larochelle, "A meta-learning perspective on cold-start recommendations for items," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[100] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 5998–6008.

[101] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations*, 2018.

[102] M. Volkovs, G. Yu, and T. Poutanen, "Dropoutnet: Addressing cold start in recommender systems," *Advances in neural information processing systems*, vol. 30, 2017.

[103] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative deep learning for recommender systems," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 1235–1244.

[104] H. Wang, F. Zhang, M. Zhang, J. Leskovec, M. Zhao, W. Li, and Z. Wang, "Knowledge-aware graph neural networks with label smoothness regularization for recommender systems," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 968–977.

[105] L. Wang and E.-P. Lim, "Zero-shot next-item recommendation using large pretrained language models," *arXiv preprint arXiv:2304.03153*, 2023.

[106] R. Wang, B. Fu, G. Fu, and M. Wang, "Deep & cross network for ad click predictions," in *Proceedings of the ADKDD'17*, 2017, pp. 1–7.

[107] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "Kgat: Knowledge graph attention network for recommendation," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 950–958.

[108] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 165–174.

[109] X. Wang, K. Zhou, J.-R. Wen, and W. X. Zhao, "Towards unified conversational recommender systems via knowledge-enhanced prompt learning," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 2022, pp. 1929–1937.

[110] C.-Y. Wu, A. Ahmed, A. Beutel, A. J. Smola, and H. Jing, "Recurrent recommender networks," in *Proceedings of the tenth ACM international conference on web search and data mining*, 2017, pp. 495–503.

[111] 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.

[112] 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.

[113] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, "Collaborative denoising auto-encoders for top-n recommender systems," in *Proceedings of the ninth ACM international conference on web search and data mining*, 2016, pp. 153–162.

[114] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T.-S. Chua, "Attentional factorization machines: Learning the weight of feature interactions via attention networks," *arXiv preprint arXiv:1708.04617*, 2017.

[115] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proceedings of the 33rd International Conference on Machine Learning (ICML '16)*. PMLR, 2016, pp. 478–487.

[116] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv *et al.*, "Qwen3 technical report," *arXiv preprint arXiv:2505.09388*, 2025.

[117] T. Yao, X. Yi, D. Z. Cheng, F. Yu, T. Chen, A. Menon, L. Hong, E. H. Chi, S. Tjoa, J. Kang *et al.*, "Self-supervised learning for large-scale item recommendations," in *Proceedings of the 30th ACM international conference on information & knowledge management*, 2021, pp. 4321–4330.

[118] J. Yu, H. Yin, J. Li, Q. Wang, N. Q. V. Hung, and X. Zhang, "Self-supervised multi-channel hypergraph convolutional network for social recommendation," in *Proceedings of the web conference 2021*, 2021, pp. 413–424.

[119] J. Yu, H. Yin, X. Xia, T. Chen, L. Cui, and Q. V. H. Nguyen, "Are graph augmentations necessary? simple graph contrastive learning for recommendation," in *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, 2022, pp. 1294–1303.

[120] 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.

[121] J. Yu, X. Xia, T. Chen, L. Cui, N. Q. V. Hung, and H. Yin, "Xsimgcl: Towards extremely simple graph contrastive learning for recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 2, pp. 913–926, 2024.

[122] 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.

[123] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, "Heterogeneous graph neural network," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 793–803.

[124] 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.

[125] W. Zhang, T. Du, and J. Wang, "Deep learning over multi-field categorical data: – a case study on user response prediction," in *European conference on information retrieval*. Springer, 2016, pp. 45–57.

[126] X. Zhang, B. Xu, C. Li, Y. Zhou, L. Li, and H. Lin, "Side information-driven session-based recommendation: A survey," *arXiv preprint arXiv:2402.17129*, 2024.

[127] Y. Zhang, Z. Chan, S. Xu, W. Bian, S. Han, H. Deng, and B. Zheng, "Keep: An industrial pre-training framework for online recommendation via knowledge extraction and plugging," in *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*, ser. CIKM '22. ACM, 2022, pp. 3684–3693.

[128] 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.

[129] H. Zheng, K. Wu, J.-H. Park, W. Zhu, and J. Luo, "Personalized fashion recommendation from personal social media data: An item-to-set metric learning approach," in *2021 IEEE International conference on big data (big data)*. IEEE, 2021, pp. 5014–5023.

[130] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," in *Proceedings of the tenth ACM international conference on web search and data mining*, 2017, pp. 425–434.

[131] K. Zhou, W. X. Zhao, S. Bian, Y. Zhou, J.-R. Wen, and J. Yu, "Improving conversational recommender systems via knowledge graph based semantic fusion," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2020, pp. 1006–1014.

[132] F. Zhu, Y. Wang, C. Chen, J. Zhou, L. Li, and G. Liu, "Cross-domain recommendation: challenges, progress, and prospects," *arXiv preprint arXiv:2103.01696*, 2021.

[133] K. Zou and A. Sun, "A survey of real-world recommender systems: Challenges, constraints, and industrial perspectives," *arXiv preprint arXiv:2509.06002*, 2025.