Enhancing Recommender Systems via Data Augmentation

Lingzi Zhang

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2023

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

27/12/2023

Date

Lingzi Zhang

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

27/12/2023

Date

Prof. Chunyan Miao

Authorship Attribution Statement

This thesis contains materials from 3 papers published in the following peer-reviewed conference as well as 1 paper submitted to a peer-reviewed journal where I am the first author.

Chapter 3 is published as Lingzi Zhang, Yong Liu, Xin Zhou, Chunyan Miao, Guoxin Wang, and Haihong Tang. "Diffusion-based graph contrastive learning for recommendation with implicit feedback," in International Conference on Database Systems for Advanced Applications, pp. 232-247. Cham: Springer International Publishing, 2022.

The contributions of the co-authors are as follows:

- I proposed the key idea and problem setting.
- I co-designed the framework and experiments with Dr. Liu.
- I carried out experiments and analyzed the results.
- I wrote the manuscript. The manuscript was revised by Dr. Liu.
- Dr. Zhou, Prof. Miao, Mr. Wang, and Ms. Tang provided insightful comments and reviewed the manuscript.

Chapter 4 is published as Lingzi Zhang, Xin Zhou, Zhiwei Zeng, Zhiqi Shen. "Dual-view Whitening on Pre-trained Text Embeddings for Sequential Recommendation," in Proceedings of the AAAI Conference on Artificial Intelligence, 2024. and Lingzi Zhang, Xin Zhou, Zhiwei Zeng, Zhiqi Shen. "Are ID Embeddings Necessary? Whitening Pre-trained Text Embeddings for Effective Sequential Recommendation," in IEEE International Conference on Data Engineering (ICDE), 2024.

The contributions of the co-authors are as follows:

- I formulated the idea, developed the proposed frameworks, implemented the algorithms, carried out experiments, and analyzed the results.
- Dr. Zhou and I verified the theoretical proofs.
- I wrote the manuscript. The manuscript was revised by Dr. Zeng and Dr. Zhou.
- Dr. Shen reviewed the manuscript.

Chapter 5 is published as Lingzi Zhang, Xin Zhou, Zhiwei Zeng, Zhiqi Shen. "Multimodal Pre-training for Sequential Recommendation via Contrastive Learning.". in ACM Transactions on Recommender Systems, 2024.

The contributions of the co-authors are as follows:

- I formulated the idea, developed the proposed frameworks, implemented the algorithms, carried out experiments, and analyzed the results.
- I wrote the manuscript. The manuscript was revised by Dr. Zeng and Dr. Zhou.
- Dr. Shen reviewed the manuscript.

27/12/2023

Date

Lingzi Zhang

Acknowledgements

I wish to express my greatest gratitude to my advisor, Professor Chunyan Miao, for her steadfast support throughout my PhD journey. I would also like to extend my profound gratitude to Dr. Yong Liu, who has initiated my foray into academic research and mentored me every step of the way. His insightful discussions, generous provision of research resources, and consistent guidance have been invaluable. My sincere thanks go to my Thesis Advisory Committee (TAC) members, Prof. Chen Change Loy and Prof. Meng Hiot Lim, for their invaluable advice.

I would like to express my deep gratitude to Dr. Xin Zhou for his invaluable suggestions and expert guidance. Additionally, I am immensely thankful to Dr. Zhiwei Zeng for her enduring patience and exceptional skill in mentoring me through the intricacies of research paper writing.

I want to thank all of my friends, Nan Song, Yingchen Yu, Hongyu Zhou, Yinan Zhang, and Xin Lan, whom I have learned so much and shared many memories with during the past few years.

I am very grateful to my parents, Mr. Jijun Zhang and Ms. Xuan Wei, who have always supported me in pursuing what I want.

Lastly, to my husband, Mr. Qijia Wang, your unconditional love and unwavering support have been the pillars that carried me to this point. Thank you for being by my side every step of the way.

Contents

Summary

Recommender systems play an essential role in enhancing user experiences by providing personalized content and suggestions, thereby improving user engagement and satisfaction. However, a major challenge faced by recommender systems is data sparsity, where real-world datasets often lack comprehensive user-item interaction data, resulting in suboptimal performance. Additionally, their dependence on limited user-item interaction data makes them susceptible to over-fitting and poor generalization, further compromising their effectiveness.

A promising solution to mitigate data sparsity and improve the generalization capabilities of recommender systems involves the integration of data augmentation techniques. Data augmentation artificially expands datasets by creating new or modified copies of existing data. In model training, augmented data can be utilized in two primary ways: it can either be directly fed into the neural network or used in conjunction with contrastive learning. Both approaches are aimed at refining the model parameters in response to new variations introduced by the augmented data. This process is crucial for enabling the model to learn rich and discriminative representations, particularly under the constraint of sparse data.

However, generating effective data augmentations in recommender systems is challenging. Firstly, the inherent issue of data sparsity in recommender systems makes generating meaningful augmented data without introducing noise or bias a complex task. Second, the complexity of user-item interactions, influenced by user/item features, temporal dynamics of preferences, and specific contextual scenarios, further complicates this task.

This dissertation aims to develop effective data augmentation methods with contrastive learning to enhance model generalization and mitigate the data sparsity issue in recommender systems. Firstly, we analyze the original user-item interaction graph and propose a data augmentation method that minimizes the introduction of noise or bias, ensuring the integrity of the underlying data structure. Secondly, we

explore the incorporation of side information by designing a systematic data augmentation method based on item features, thus leveraging additional contextual information to improve recommendations. Lastly, we integrate multimodal features and the sequential order of user interactions into our data augmentation strategy, providing a robust solution that captures the complexity and richness of real-world user behavior. By addressing these aspects collectively, this dissertation demonstrates how a multifaceted data augmentation strategy can significantly enhance the performance of recommender systems. Specifically, the research is divided into three chapters, each addressing a specific research problem:

Research problem 1 Existing data augmentation methods predominantly focus on randomly removing edges from the user-item interaction graph, which overlooks the importance of differentiating between informative and irrelevant or noisy edges in the augmented graph. We present an advanced method utilizing graph diffusion, which smooths neighborhood interactions across the graph and transforms the original unweighted graph into a weighted one. The weights, based on the structural importance of each edge, facilitate the maintenance of an efficient neighborhood for each node in the diffusion graph. Particularly, we propose a Graph Diffusion Contrastive Learning (GDCL) framework for recommendation, where the diffusion graph is encoded to preserve heterogeneity, and a symmetric contrastive learning objective contrasts local node representations of the diffusion graph with the user-item interaction graph.

Research problem 2 Current feature-based data augmentation methods in recommender systems generally involve random alterations of features, such as dropout, shuffling, or perturbing embeddings with random noise. However, these methods mostly rely on arbitrary data augmentations, chosen through a process of trial-and-error. This reliance on non-systematic methods may constrain their generalizability and adversely affect their overall performance. This study first examines a sequential recommendation framework based on item text features, finding that anisotropy in pre-trained text embeddings can impair performance. To address this, a whitening transformation is applied to reconfigure the pre-trained text embedding distribution into an isotropic form, significantly enhancing model performance. However, an empirical analysis indicates that the whitening may adversely affect the manifold of items with similar textual semantics. To mitigate this, we first introduce an ensemble framework WhitenRec+, which combines fully

and partially whitened representations via a simple summation. Then, we refine its architecture by proposing a Dual-view Whitening method for Sequential Recommendation (DWSRec). DWSRec utilizes diverse views of whitened embeddings to alternately update the attention heads within the transformer model, effectively acting as data augmentation and improving overall performance.

Research problem 3 Most existing methods focus on augmenting a single type of feature and are often unable to explore augmentations with user behavior sequences across different types of features. In this work, we propose a novel Multimodal Pre-training for Sequential Recommendation (MP4SR) framework, which utilizes contrastive losses to capture the correlation among different modality sequences of users and different modality sequences of users and items. MP4SR employs a sequence mixup strategy for fusing different modality sequences and leverages contrastive learning at the sequence-to-sequence and sequence-to-item levels. This multimodal pre-training approach serves as an effective regularizer, optimizing the parameter space for recommendation tasks.

In conclusion, this dissertation proposes methods that develop different data augmentations across various data structures to enhance the performance of recommender systems. Extensive experiments on real-world datasets validate the effectiveness of these methods.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Challenges

Recommender systems have emerged as a critical solution to the prevalent problem of information overload in modern society, primarily driven by the widespread use of the internet. Recommender systems, leveraging advanced artificial intelligence and data-driven algorithms, effectively curate personalized content, guiding users through an overwhelming array of choices across various domains such as e-commerce, social media, news portals, and digital libraries. By selectively presenting items that align with individual preferences, recommender systems significantly streamline the decision-making process for time saving [1–4].

Initially, general recommender systems predominantly rely on static modeling of user-item interactions to discern users' general preferences. Traditional methods [5, 6] calculate the similarity between users or items based on these interactions to predict user preferences for unrated items. Subsequently, more complex prediction models have emerged, ranging from shallow [7–13] to deep learning [4, 14–22] models. These models learn latent representations of users and items from observed interactions, with the prediction for an unobserved user-item pair being based on the similarity of these representations. More recently, graph neural networks (GNNs) have gained considerable traction in this field. Numerous graph-based recommendation models [23–35] have been developed, treating observed user-item interactions as a bipartite graph. GNNs are particularly effective in capturing high-order connection information within these graphs. However, these studies do

not consider the sequential nature of user behavior, restricting their ability to fully capture the dynamics and evolution of user preferences over time.

As an extension of general recommender systems, sequential recommender systems focus on predicting the next item a user might be interested in, based on their previous actions in a sequence [36]. This approach acknowledges that user preferences are dynamic and can evolve over time. Traditional methods for modeling sequential data leverage sequential pattern mining to find frequent patterns [37], and Markov Chain [38–40] to model transitions in user-item interaction sequences. With the rapid advancement of deep learning techniques, deep neural networks have become increasingly prevalent in sequential recommendations. Deep learning-based methods often employ neural network architecture, including Recurrent Neural Networks (RNNs) [41–43], Convolutional Neural Networks (CNNs) [44, 45], GNNs [32, 35, 46–48], and Transformer [49–53].

The proliferation of data in today's digital landscape offers a wealth of multimodal information. This abundance of diverse data types presents an invaluable opportunity for recommender systems to leverage varied data sources, thereby significantly improving the quality of recommendations. Modality-enhanced recommender systems, distinct from the above approaches, utilize a variety of data types including text, images, audio, and video to further enhance the recommendation performance [1]. Text-enhanced recommender systems enhance item representations through textual data, such as item descriptions, attributes, and brands. Existing text-enhanced methods propose building an auxiliary task by classifying text tokens [51, 54], employing pre-trained language models for feature extraction [55–61], or fine-tuning pre-trained language models specifically for recommendation tasks [58, 62–64]. Visual-enhanced recommender systems, alternatively, refine item representations through visual data, either by constructing item-specific visual representations [65–67] or incorporating visual modeling as auxiliary information [68–72]. Additionally, some studies combine different modalities of items together to build multimodal recommender systems [55, 59, 66, 73–84]. The integration of multimodal data enables modality-enhanced recommender systems to provide a more comprehensive understanding of user preferences, leading to more precise and accurate recommendations.

User-item interactions often exhibit significant sparsity relative to the entire interaction space, leading to the prevalent issue of data sparsity in recommender

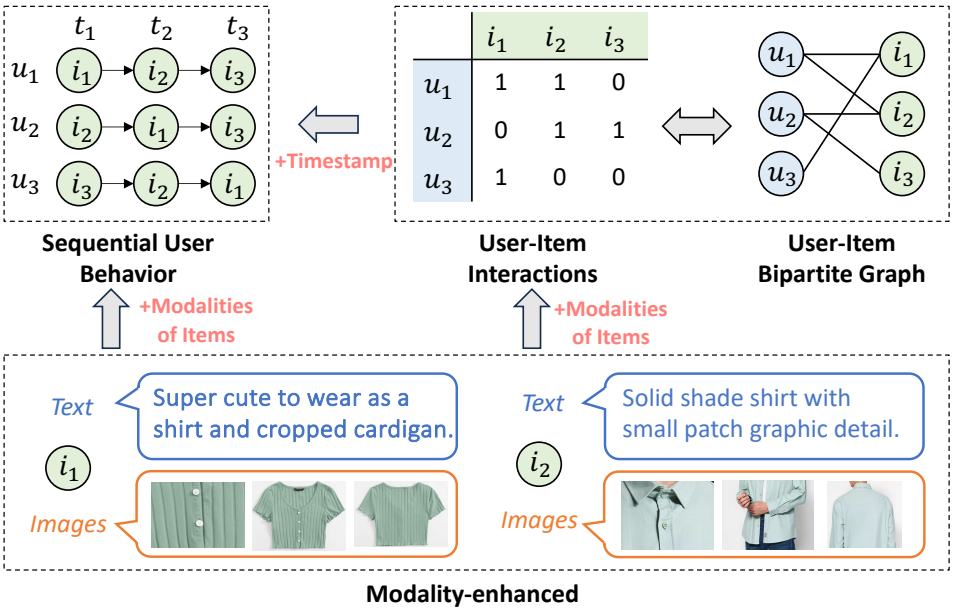


FIGURE 1.1: The illustration of different data structures in recommender systems.

systems. While the use of multimodal features can help mitigate data sparsity, these methods often face risks of overfitting and poor generalization on unseen data [85, 86]. Integrating data augmentation techniques presents a promising solution to address these issues.

Data augmentation, which artificially expands datasets by generating new or modified versions of existing data, enriches the dataset for more effective model training [86, 87]. Typical data types in the context of recommender systems, as shown in Figure 1.1, include user-item interactions represented in a bipartite graph, sequences of interactions with temporal ordering, and associated item features. Data augmentation methods typically revolve around these three data types. For graph-based augmentation, existing methods initially form graphs to delineate relationships among users and items, and exclusively between users or items. Subsequently, these graphs are augmented through techniques like edge perturbation [88–95], node dropping [88, 92, 96], subgraph sampling [88, 90, 92, 97–100], or employing global graph structures via graph diffusion and SVD algorithms [101–103]. Sequence-based augmentation modifies the original interaction sequence through various techniques, including item masking [51, 53, 104–106], cropping [51, 53, 104, 105], reordering [38, 107, 108], substitution [108, 109], and insertion [109,

110]. Lastly, feature-based augmentation targets augmentations on feature vectors. These vectors may consist of categorical features from categorical attributes or learned continuous feature embeddings. Techniques in this category encompass feature dropout [111, 112], shuffling [93, 100, 113], mixing[91, 114, 115], introducing feature noise [116–118].

In model training, the utilization of augmented data is bifurcated into two main approaches. The first involves directly feeding the augmented data into the neural network. This process plays a crucial role in the continuous evolution and adaptation of the model, as it adjusts and refines its parameters in response to the new patterns and variations introduced by the augmented data. The second approach incorporates augmented data within the framework of contrastive learning, which focuses on learning effective data representations by generating different views of the same data through augmentations and then seeks to maximize the agreement between positive pairs within these views [119]. This method enhances the performance of recommender systems by learning rich and discriminative representations from sparse user-item interactions in real-world scenarios.

However, the effective creation of data augmentations for recommender systems poses two significant challenges:

- Recommender systems often deal with extremely sparse data, where the majority of user-item interactions are unobserved. This sparsity makes it difficult to generate meaningful augmented data without introducing noise or bias.
- Users and items interact in complex ways and this interaction is influenced by a variety of factors, including user/item features and changes in preferences over time.

In this dissertation, the **research objective** is to *develop effective data augmentation methods and contrastive learning strategies tailored to different data structures in recommender systems*, to enhance model generalization and mitigate the data sparsity issue.

1.2 Research Objectives and Methodologies

In this section, we provide an overview of the research objectives and corresponding methodologies explored in this dissertation.

The objective of this research is to develop effective data augmentation methods and contrastive learning strategies tailored to different data structures in recommender systems. These methods aim to enhance model generalization and address the data sparsity issue. The research is divided into three chapters, each addressing a specific objective:

- How can we design a data augmentation method that minimizes the introduction of noise or bias when based on the interaction graph? Current methods largely rely on random edge/node dropout or subgraph sampling, which does not differentiate the importance of edges or nodes.
- How can we design a systematic data augmentation method based on item features? Existing methods in recommender systems often involve arbitrary feature alterations such as dropout, shuffling, mixing, or adding random noise to embeddings. These non-systematic, trial-and-error methods may limit generalizability and negatively impact performance.
- How can we design a data augmentation method that considers multimodal features of items and the sequential order of user interactions? Existing methods typically focus on augmenting a single type of feature and fail to integrate user behavior sequences across different feature types.

To achieve these research objectives, firstly, we introduce a graph-based augmentation method designed to generate augmented data while minimizing the introduction of noise or bias. Secondly, we propose a feature-based augmentation method aimed at developing systematic and effective approaches for feature augmentation. Thirdly, we present a method for feature augmentation on sequence data, focusing on the integration and utilization of data from various modalities. The overarching structure of these proposed methods is depicted in Figure 1.2.

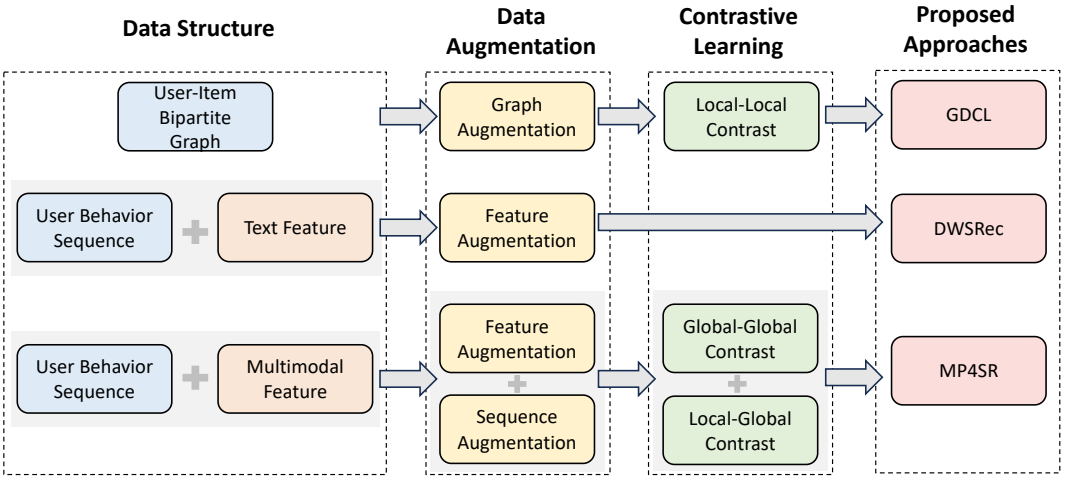


FIGURE 1.2: The overall structure of proposed methods in this dissertation.

1.2.1 Diffusion-based Graph Contrastive Learning for Recommendation

1.2.1.1 Motivation

The advent of deep learning has catalyzed the development of various neural network-based recommendation models [120, 121], with a notable subset utilizing GNNs [23] due to their ability to exploit high-order connectivity between users and items through iterative message propagation, enhancing recommendation performance. However, challenges persist, including limited supervision signals from sparse user-item interactions, unbalanced user/item node degrees, and noise in user feedback [88].

Data augmentation with contrastive learning has emerged as a promising approach to overcome these challenges and enhance robustness and generalization in GNNs-based representation learning [88, 99, 122, 123]. Crucial to this approach is the design of graph augmentation strategies, which are often confined to random edge/node dropout or subgraph sampling [88, 99, 122]. However, these methods cannot discern the importance of edges, leading to the desire for an augmented graph that maintains informative edges while discarding irrelevant or noisy ones.

1.2.1.2 Methodology

We propose using graph diffusion instead of random dropout to create an effective augmentation view. This process transforms the original unweighted graph into a weighted one, where weights indicate edge importance based on graph structure [124]. This allows for the design of sparsification methods that retain effective neighborhoods in the diffusion graph. Additionally, unlike traditional GNNs that incorporate high-order information through multiple convolutional layers, graph diffusion extends connections from one-hop to multi-hops without the noise typically introduced by iterative layer expansion.

Our study introduces the Graph Diffusion Contrastive Learning (GDCL) framework for recommendation. Unlike existing diffusion algorithms that focus on homogeneous graphs, GDCL performs graph diffusion considering different node types in the heterogeneous graph, *i.e.*, the user-item interaction graph. This heterogeneity is inadequately addressed by standard Graph Convolutional Network (GCN)-based encoders, which treat relations uniformly. GDCL extends GCN to model diffusion graph heterogeneity, maintaining distinct relation types and fusing them via a mean aggregator. The framework employs a multi-task training paradigm, optimizing both the recommendation and self-supervised learning tasks. For recommendation, we diverge from previous models by using representations from both the diffusion graph and the user-item graph, enhancing user and item representation learning. The self-supervised task contrasts node representations from both views using a symmetric mutual information maximization objective function, thereby enhancing the overall model performance.

1.2.2 Dual-view Whitening on Pre-trained Text Embeddings for Sequential Recommendation

1.2.2.1 Motivation

The field of sequential recommendation [51, 56, 57] has seen a growing interest in incorporating textual information about items, such as product attributes [54], descriptions [57], and reviews [98]. Existing sequential recommendation models often use text embeddings from pre-trained language models like BERT [57, 58, 125].

Recent research [58] suggests that superior performance in text-based models for sequential recommendation is achievable mainly with advanced pre-trained language models. However, we argue that these models do not optimally utilize pre-trained text embeddings for sequential recommendation. This suboptimal usage partly arises from the representation degeneration issue in BERT embeddings, leading to anisotropy in the vector space and limiting their effectiveness in downstream tasks [126, 127]. Furthermore, feature-based augmentation is vital for improving user representation learning in this context, as these representations rely on sequences of item text embeddings without distinct ID embeddings for users. Current methods in recommender systems typically involve arbitrary feature alterations, such as dropout [111, 112], shuffling [93, 100, 113], mixing [91, 114, 115], or adding random noise to embeddings [116–118]. However, the reliance on these non-systematic, trial-and-error methods may limit their generalizability and negatively impact performance, highlighting the need for more systematic approaches in the realm of text-based sequential recommendation methods.

1.2.2.2 Methodology

To address the anisotropy in pre-trained text embeddings and leverage systematic approaches for feature augmentation, our methodology focuses on using both full and partial whitening as data augmentation techniques. The whitening transformation reshapes the distribution of pre-trained text embeddings into an isotropic Gaussian distribution. This process effectively removes correlations among axes and has been shown to enhance the performance of sequential models when compared to those using ID embeddings or original text embeddings.

While over-whitening can be advantageous, it may adversely affect the manifold of items with similar textual semantics. To mitigate this, partial whitening can be implemented, where dimensions are only partially decorrelated, allowing for the retention of more original textual semantics. However, relying solely on partial whitening leads to inferior performance compared to full whitening. Thus, our proposed method employs both full and partial whitening for data augmentation: full whitening for improved performance through complete dimension decorrelation, and partial whitening to maintain more original textual semantics.

Incorporating these insights, we propose two sequential recommendation models to leverage both full and partial whitening for data augmentation. The first model, WhitenRec+, is an ensemble framework that combines fully whitened and partially whitened representations via summation. This simple method significantly enhances representation learning in sequential recommender systems. Our experimental results confirm WhitenRec+'s superiority over other baselines.

Further advancing this approach, we introduce the Dual-view Whitening method for Sequential Recommendation (DWSRec). DWSRec uses a dual-view item encoder with a shared projection head to generate both fully and partially whitened representations from pre-trained text features. Then, we employ a decoupled attention-based dual-view transformer for encoding sequences of both fully and partially whitened representations, treating them as separate data augmentation instances. This is followed by a decoupled attention-based dual-view transformer, which encodes sequences of both types of augmented representations. DWSRec's use of diverse whitened embeddings as data augmentation techniques enriches the training process, leading to improved performance. DWSRec also features a dual-view fusion module that adaptively combines these augmented, view-specific sequence embeddings and item embeddings for recommendation, using distinct weighted attention layers. Our comprehensive evaluations demonstrate DWSRec's effectiveness in improving user and item representation uniformity and alignment. Additionally, DWSRec improves the conditioning of the transformed item embedding matrix, which contributes to greater training stability. Notably, DWSRec is shown to be superior in preserving information, requiring less data for reconstructing training input.

1.2.3 Multimodal Contrastive Learning for Sequential Recommendation

1.2.3.1 Motivation

Currently, multimodal data is widely available, and as a result, a large amount of research has been conducted on multimodal recommender systems that leverage multimodal content (such as images and text descriptions) associated with items.

For instance, some works [66, 68] have used item multimodal content as a regularization factor in collaborative filtering frameworks. More recent studies [55, 80, 81] have employed GNNs to uncover connections between different modalities, thereby deepening the understanding of user preferences.

While recommender systems leveraging multimodal features are effective, they encounter risks of overfitting and poor generalization when applied to unseen data [85, 86]. The integration of data augmentation techniques emerges as a promising solution to address these issues. However, existing methods [93, 100, 111–113] generally focus on augmenting a single type of feature and are often unable to explore augmentations with user behavior sequences across different types of features.

1.2.3.2 Methodology

Our methodology aims to improve the fusion of multimodal information and its utilization in sequential recommender systems through data augmentation and contrastive learning. We introduce a multimodal sequence-based data augmentation method, incorporating a complementary sequence mixup approach. This method effectively fuses sequences of text and image representations in a manner that reduces the representation discrepancy between sequences of different modalities. Subsequently, we construct two contrastive loss functions. These functions utilize self-supervised signals to efficiently aggregate and align visual and textual information. By capturing the intrinsic correlations within the data, this approach significantly enhances the performance of downstream recommendation tasks.

In particular, we introduce a Multimodal Pre-training for Sequential Recommendation (MP4SR) framework, which employs contrastive losses to identify correlations among user and item behavior sequences across different modalities. MP4SR consists of three main components: multimodal feature extraction, the Multimodal Mixup Sequence Encoder (M^2SE) backbone network, and pre-training tasks. We first tokenize item images into text keywords using a language-image pre-trained model [128] and extract initial text and image features using Sentence-BERT [129]. This step harmonizes textual and visual modalities while retaining essential image information. M^2SE then integrates user modality sequences using a complementary sequence mixup strategy, processed by a Transformer to obtain mixed-modality sequence representations. Finally, contrastive learning is applied at the

sequence-to-sequence and sequence-to-item levels. We employ a modality-specific next item prediction loss and a cross-modality contrastive learning loss, minimizing self-supervised pre-training criteria to act as a regularizer on the parameter space, thereby enhancing recommendation performance.

1.3 Major Contributions

In this dissertation, novel data augmentation methods across different data structures for recommender systems are proposed to mitigate the data sparsity problem and enhance the model generalization capabilities. The main research contributions of this dissertation are three-fold and can be summarized as follows:

- We propose a Graph Diffusion Contrastive Learning (GDCL) framework for recommendation, which leverages the diffusion graph as a key augmentation view for contrastive learning. The framework can effectively identify important edges, thereby retaining effective neighborhoods in the graph and enhancing graph representation learning. Specifically, we perform graph diffusion on the user-item interaction graph. Then, the diffusion graph is encoded to preserve its heterogeneity by learning a dedicated representation for every type of relation. A symmetric contrastive learning objective is used to contrast local node representations of the diffusion graph with those of the user-item interaction graph for learning better user and item representations. Extensive experiments on real datasets demonstrate that GDCL consistently outperforms state-of-the-art recommendation methods.

- We show that anisotropy in pre-trained text embeddings restricts the performance of text-based sequential recommendation models. To resolve this issue, we employ whitening transformation to transform pre-trained text embedding distribution into an isotropic form, which can significantly improve the performance of text-based sequential recommendation models. Our empirical analysis of the whitening process reveals that it may hurt the manifold of items exhibiting similar textual semantics. To this end, we propose two models, WhitenRec+ and DWSRec (*i.e.*, Dual-view Whitening method for Sequential Recommendation), which leverage different degrees of whitening

transformations as data augmentation to reap the benefits of full whitening while preserving some of the inherent semantics in the original text features. Extensive experiments are conducted on three benchmark datasets to evaluate the performance of the proposed methods for the sequential recommendation. Notably, WhitenRec+ and DWSRec outperform state-of-the-art models across all metrics for all three datasets.

- We propose a novel Multimodal Pre-training for Sequential Recommendation (MP4SR) framework, which effectively fuses text and image sequences and utilizes contrastive losses to capture the correlation among different modality sequences of users and items. MP4SR consists of three key components: 1) multimodal feature extraction, 2) a backbone network, Multimodal Mixup Sequence Encoder (M^2SE), and 3) pre-training tasks. After utilizing pre-trained encoders to generate initial multimodal features of items, M^2SE adopts a complementary sequence mixup strategy to fuse different modality sequences, and leverages contrastive learning to capture modality interactions at the sequence-to-sequence and sequence-to-item levels. Extensive experiments on three real-world datasets demonstrate that MP4SR outperforms state-of-the-art approaches in both normal and cold-start settings. We further highlight the efficacy of incorporating multimodal pre-training in sequential recommendation representation learning, serving as an effective regularizer and optimizing the parameter space for the recommendation task.

1.4 Thesis Organization

The remainder of this thesis is organized as follows:

- Chapter 1 provides background information and an overview of the thesis.
- Chapter 2 offers a comprehensive review of recommender systems, including topics such as general recommendation, sequential recommendation, multimodal recommendation, and data augmentation techniques enhancing model generalization.

- Chapter 4 presents our research on graph diffusion and symmetric contrastive learning, which improves user and item representations in general recommender systems.
- Chapter 5 demonstrates a multi-view whitening-based model, applying varying degrees of whitening transformations on pre-trained text features for sequential recommendation.
- Chapter 6 introduces a multimodal pre-training framework for sequential recommendation, employing contrastive losses to capture correlations among different modalities of sequences and items.
- Chapter 7 concludes this thesis and discusses future research directions.

Chapter 2

Literature Review

In this chapter, we present a comprehensive overview of research related to recommender systems and strategies to enhance model generalization. Initially, we examine interaction-based recommender systems, which primarily focus on depicting user-item interactions. Then, we shift our attention to recommender systems that utilize content information of items across various modalities. Finally, we delve into strategies designed to enhance model generalization, specifically through the application of data augmentation and contrastive learning techniques. The overall structure of the literature review is presented in Figure 2.1.

2.1 Interaction-based Recommender Systems

Recommender systems infer users' preferences and items' properties from their attributes or user-item interactions and further recommend items that users might be interested in [130]. A considerable amount of literature has been published on the recommendation-related topic, as it can be applied to many business applications for solving the information overload problem and enhancing user experiences. In this section, our focus is specifically on recommendation methods that model the interactions between users and items. Interaction-based recommender systems can be categorized based on whether they consider the order of items. Specifically, they can be classified into two primary tasks: general recommender systems and sequential recommender systems [130, 131].

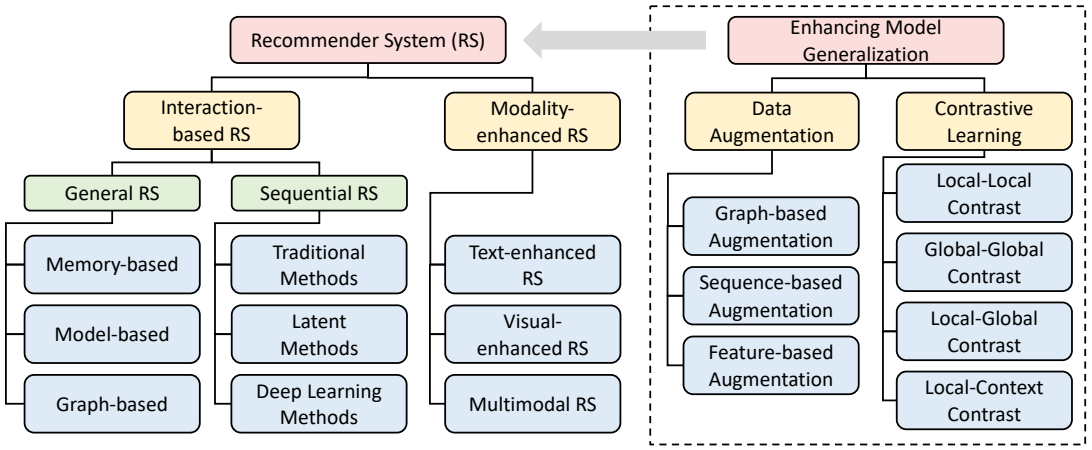


FIGURE 2.1: The overall structure of the literature review [1, 2, 4, 86, 87, 131].

2.1.1 General Recommender Systems

The general recommender systems model the user-item interactions in a static way and can only capture the users' general preferences. In general, general recommender systems can be further categorized into three categories: memory-based methods, model-based methods, and graph-based methods [4, 130, 131].

2.1.1.1 Memory-based Methods.

Memory-based methods typically make predictions about a user's interests based on the preferences of similar users. This approach does not involve building a model but relies on the historical data of user-item interactions. There are two primary categories of memory-based methods: user-based and item-based.

User-based methods. In a standard user-based method, a user's rating for a particular item is predicted by measuring the ratings that similar users have designated to that item [5, 6]. Typically, the top K users exhibiting the greatest similarities to the reference user are chosen. Their ratings for the specified item are then consolidated to produce a predicted rating for that user. The predicted rating of user i to item j is as follows:

$$\hat{R}_{ij} = \frac{1}{C} \sum_{k \in Z_i} sim(i,k) R_{kj}, \tag{2.1}$$

where Z_i is the set of K neighboring users of user i, C is a normalizing constant, and $sim(i,k)$ denotes the similarity between user i and user k. The R_{kj} indicates user k's preference for item j, with $R_{kj} > 0$ signifying a positive preference.

Item-based methods. In contrast to the user-based method, item-based methods recommend items on the basis of information about other items that a user has previously rated [132, 133]. The recommended items for the given user are ranked by aggregating the similarities between each candidate item and the items that the user has rated. The predicted rating of user i to item j according to the item-based method is as follows:

$$\hat{R}_{ij} = \frac{1}{C} \sum_{k \in Z_j} sim(j,k) R_{ik}, \tag{2.2}$$

where Z_j is the set of K neighboring items of item j, C is a normalizing constant, and $sim(j,k)$ denotes the similarity between item j and item k.

To ascertain similar users or items, similarity metrics, such as the Pearson correlation, cosine similarity, or Jaccard similarity, are employed on rating vectors of users or items. Rating vectors of users consist of item ratings allocated by an individual user, whereas rating vectors of items represent each item by the scores assigned by users.

Despite the simplicity and widespread use of memory-based approaches, they exhibit three primary limitations [3, 130]. Firstly, calculating similarities between all user or item pairs incurs a significant computational cost due to its quadratic time complexity. Secondly, the accuracy of recommendations is contingent upon the chosen similarity measure, which often relies on a suboptimal relationship either between users or items. Lastly, these approaches frequently grapple with the challenge posed by the sparsity of the interaction matrix. A sparse interaction matrix, characterized by a limited number of non-zero values, renders the prediction of recommendations more prone to errors.

2.1.1.2 Model-based Methods.

The model-based methods aim to build a prediction model based on interactions between users and items. In this way, the trained prediction model can be used to predict the unknown ratings of users for new items. Model-based methods can better adapt and scale up to large-scale datasets with significant performance

improvements when compared with memory-based ones. Based on the type of models used, model-based methods can be classified into latent factor models and deep learning models.

Latent factor models. Latent factor models decompose the high-dimensional user-item rating matrix into low-dimensional user and item latent vectors. The basic idea of latent factor models is that both users and items can be characterized by a few latent features, and thus the prediction can be computed as the inner product of user-feature and item-feature vectors. One representative group of methods leverages matrix factorization (MF) techniques, which have attracted considerable attention due to their advantages with respect to scalability and accuracy, as witnessed by the algorithms developed within the Netflix contest [7]. We briefly introduce the most common formulation of the MF model. Firstly, each user and item are modeled as a vector of latent factors and the predicted rating of user i given to item j is calculated as follows:

$$\hat{R}_{ij} = \sum_{k=1}^{d} U_{ik} V_{jk} = \boldsymbol{U}_i \boldsymbol{V}_j^\top, \tag{2.3}$$

where $\boldsymbol{U}_i \in \mathbb{R}^d$ and $\boldsymbol{V}_j \in \mathbb{R}^d$ denote the latent vector of user i and item j. d is the number of dimensions. Next, the latent vectors of all users $\boldsymbol{U}$ and items $\boldsymbol{V}$ can be obtained by solving

$$\boldsymbol{U}^*, \boldsymbol{V}^* = \underset{\boldsymbol{U},\boldsymbol{V}}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{N} I_{ij} (R_{ij} - \boldsymbol{U}_i \boldsymbol{V}_j^\top)^2 + \frac{\lambda_U}{2} \|\boldsymbol{U}\|_F^2 + \frac{\lambda_V}{2} \|\boldsymbol{V}\|_F^2, \tag{2.4}$$

where $\boldsymbol{U} \in \mathbf{R}^{M \times d}$ and $\boldsymbol{V} \in \mathbf{R}^{N \times d}$ are two matrices of latent vectors representing all users and items, and $\boldsymbol{U}^*$ and $\boldsymbol{V}^*$ stand for their optimal values obtained from the minimization. I_{ij} is an indicator function that is equal to 1 if $R_{ij} > 0$, otherwise 0. $\|\boldsymbol{U}\|_F$ denotes the Frobenius norm of the matrix, and λ_U and λ_V are regularization parameters that are usually set to alleviate model overfitting.

MF is also formulated from a probabilistic perspective—that is, as a probabilistic matrix factorization (PMF) problem [8]. The PMF framework models the conditional probability of latent factors given the observed ratings and includes priors that handle complexity regularization. Hu et al. [9] introduce a weighted regularized matrix factorization (WRMF) model, treating all missing data as negative samples and heuristically assigning confidence weights to positive samples.

Building on this, He et al. [11] suggest weighting the missing data based on item popularity, an approach that exhibited enhanced performance relative to WRMF. Alternatively, Rendle et al. [10] propose a pairwise ranking objective, known as Bayesian Personalized Ranking, which models the relationships between positive and negative items for each user, with negative samples being randomly drawn from unobserved feedback. The recent work by Van Balen and Goethals [12] introduces a model that employs a full-rank factorization of the inverse Gram matrix. This approach allows the generation of high-dimensional embeddings that can achieve sparsity without compromising the factorization of a dense affinity matrix. Their findings demonstrate that these embeddings amalgamate the benefits of latent representations with the efficacy of high-dimensional linear models. To this end, MF-based methods only consider the linear interaction between a user and an item, by assuming that their latent factors are independent of each other. In contrast, Rendle [13] proposes the factorization machine, which can model all interactions in any kind of real-valued feature vector.

Deep learning models. Over the past few decades, deep learning has achieved remarkable success in fields like computer vision and natural language processing. Recent developments in deep learning-based recommender systems have surpassed traditional models, delivering superior recommendation quality [4]. Deep learning can effectively identify nonlinear and intricate user/item interactions, facilitating the encoding of complex abstractions in the higher layers of data representation. Furthermore, it can learn underlying explanatory factors and derive useful representations from input data, obviating the need for manual feature design and seamlessly integrating diverse content information, including text, images, and video. Depending on what types of deep learning models are used, we classify them into Multilayer Perceptron (MLP), Autoencoder, CNNs, and Neural Attention-based methods.

MLP can be used to add nonlinear transformation and enhance the representation learning of features for users and items. DeepFM [14] is an integrated end-to-end model that effectively combines factorization machines with MLP layers. This model is able to capture high-order feature interactions using deep neural networks, while simultaneously addressing low-order interactions through factorization machines. The Wide & Deep model [15] can handle both regression and classification tasks. The "wide" component, a single-layer perceptron, functions as a generalized

linear model and excels at memorizing explicit historical data features. In contrast, the "deep" component, an MLP layer, facilitates generalization by generating abstract representations.

Autoencoder-based models, leveraging the concept of input reconstruction for enhanced representation learning, accept the incomplete user-item matrix as input. They subsequently learn a hidden representation for each instance using an encoder. This is then followed by a decoder segment that reconstructs the input derived from the derived hidden representation [1]. Initially, AutoRec [16] employs the traditional autoencoder, which directly processes user or item rating vectors as input and reconstructs these rating vectors at the output layer. In contrast, the collaborative filtering neural network [17] leverages stacked denoising autoencoders (DAE) to enhance its robustness. The collaborative denoising autoencoder [134] integrates DAE with latent user vectors. MultVAE [135], a variant of the variational autoencoder, utilizes the multinomial likelihood for recommendations with implicit data and proposes a principled Bayesian inference method for parameter estimation. RecVAE [136] advances MultVAE by adopting a composite prior distribution for latent codes within the β-VAE framework. Most recently, VAE++ has been introduced by Ma et al. [18], adeptly leveraging heterogeneous feedback, namely purchase feedback, examination feedback, and a combination of both, to enhance recommendation performance.

CNNs have demonstrated efficacy in capturing user-item interaction patterns [19, 20] and in extracting features from diverse multimedia data [65, 137]. Firstly, for modeling user-item interactions, He et al. [19] introduce the ConvNCF, which employs the outer product to represent user/item interactions. By applying CNNs to the outcome of the outer product, the model effectively discerns high-order correlations among embedding dimensions. Tang and Wang [20] showcase the use of CNNs in sequential recommendation, incorporating both hierarchical and vertical CNNs to address union-level sequential patterns and skip behaviors, thus enabling sequence-aware recommendations. Secondly, CNNs have also been employed in feature representation learning from diverse sources, including images, text, audio, and video. For instance, DeepCoNN [137] employs two parallel CNNs to interpret user behaviors and item attributes from review texts. This architecture not only mitigates the data sparsity challenge but also enhances model interpretability by harnessing the rich semantic representations of review texts. Moreover, Lei et al.

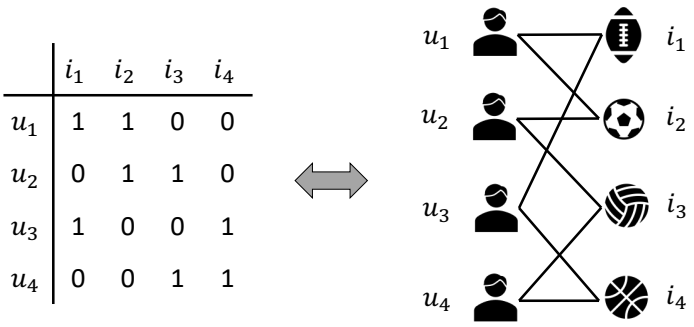


FIGURE 2.2: An illustration of the user-item bipartite graph.

[65] introduce a comparative deep learning model for image recommendation, integrating two CNNs for image representation alongside an MLP for modeling user preferences.

Different historical items contribute variably to the modeling of a user's preference. Consequently, the neural attention mechanism, adept at filtering irrelevant features from raw inputs and mitigating the adverse effects of noisy data, has gained significant attention in recent years [1, 4]. A notable example is the Attentive Collaborative Filtering model [138], which attributes each interacted item with a user-aware attentive weight, signifying its relevance to user representation. DeepICF [21] emphasizes item-level attention. It begins with a pairwise interaction layer that models the relationship between each historical item and the target item. This is followed by an attention-based pooling layer that assigns varied weights to the outputs of the pairwise interaction layer from different historical layers. Subsequently, the combined output from these layers is processed through MLP layers, to model the high-order interactions between items. Cheng et al. [22] introduce a general feature-level attention method tailored for item-based collaborative filtering models. They craft a light attention neural network that seamlessly integrates both item-level and feature-level attention for neural item-based collaborative filtering models. This approach is model-agnostic and easy to implement.

2.1.1.3 Graph-based Methods.

As the data for recommendation tasks naturally forms a user-item bipartite graph, graph-based models have been widely applied for learning user preferences to solve the recommendation problem. An example of the user-item bipartite graph is

shown in Figure 2.2. Approaches to graph-based recommender systems can be categorized into random walk-based methods, graph embedding methods, and GNNs-based methods [23].

Random walk methods. Random walk methods rely on the random walk algorithm, which visits nodes in a graph at random or based on a pre-defined probability distribution. The probability that the walker lands on each node at an equilibrium state is used for ranking items for recommendation. For example, Bagci and Karagoz [139] applies random walk with restart to rank users for friend recommendations. Jiang et al. [24] utilizes a generalized random walk with restart model to learn user preferences and uses a Bayesian Personalized Ranking loss [10] to learn weights of links in the network.

Graph embedding methods. Graph embedding methods encode graph structure information by mapping graph nodes into low-dimensional latent embeddings. Some works [25, 26] generate distributed representations by first using a random walk to obtain a sequence of nodes based on one meta-path and then employing the skip-gram for learning node representations. For instance, Shi et al. [140] proposes a heterogeneous information network-based recommendation model, which adapts node embeddings learned from the heterogeneous information network for recommendation tasks. Some works [25, 26, 141] utilize neural networks, like MLP, autoencoder, which are then integrated with neural recommendation models to learn users or items embedding. For example, Han et al. [25] proposes a Neural network-based Aspect-level Collaborative Filtering model, which uses a heterogeneous information network to model objects and relations in the recommender system and extracts aspect-level latent factors for users and items.

GNNs methods. GNNs apply neural network techniques to graph data. Some works [27–30] leverage Graph Attention Networks [142], which differentiate the importance of neighbors with an attention mechanism. For example, KGAT [30] employs an attention mechanism to discriminate the importance of neighbors and recursively propagates embeddings of neighbors to the center node to refine its embedding. Some works [31, 143–146] leverage GCN [147], which iteratively aggregate information from local graph neighbor nodes by leveraging graph structure and/or node feature information. For example, Ying et al. [143] integrates an efficient random walk algorithm with GCN to learn embeddings for nodes in

web-scale interaction graphs. Wang et al. [148] proposes a neural graph collaborative filtering (NGCF) model, which explicitly models higher-order connectivity in a user-item graph and effectively injects collaborative signals into the embedding learning process. Moreover, He et al. [31] proposes to simplify NGCF by removing unnecessary components of GCN, *e.g.*, non-linear activation and feature transformation. The model performance is further improved. Some works [32–35] leverage Gated GNNs, which introduce Gated Recurrent Unit (GRU) into GNNs to model complex transitions in a sequence of items. For example, [35] proposes a graph contextualized self-attention model, which models the local graph structure of a session using GNNs and non-local contextualized representations using a multi-layer self-attention network.

The common paradigm of GNNs-based recommendation models is formulated as follows. Specifically, for a node t in a graph $\mathcal{G}$, its representation in a GNNs-based model is generally derived as,

$$\begin{aligned} \mathbf{e}_t^{(\ell+1)} &= f_{combine}\big(\mathbf{e}_t^{(\ell)}, f_{aggregate}\big(\{\mathbf{e}_i^{(\ell)} : i \in \mathcal{N}_t\}\big)\big), \\ \mathbf{e}_t &= f_{pooling}\big(\mathbf{e}_t^{(0)}, \mathbf{e}_t^{(1)}, \cdots, \mathbf{e}_t^{(L)}\big), \end{aligned} \tag{2.5}$$

where $\mathbf{e}_t^{(\ell)}$ denote the representation of t at ℓ-th GNN layer, and $\mathbf{e}_t$ is the final representation of t after L layers. Here, $f_{combine}(\cdot)$, $f_{aggregate}(\cdot)$, and $f_{pooling}(\cdot)$ denote combination, aggregation, and pooling functions, respectively.

In the literature, different *aggregation* strategies have been proposed to propagate neighborhood information to the center node to refine its embedding. For example, GC-MC [149] first applies a linear transformation to features of direct neighbors and then uses mean pooling for neighborhood aggregation. PinSage [143] shows that assigning different weights to neighbors during aggregation gains better performance. NGCF [150] additionally encodes the user-item interaction via the element-wise product of the user embedding and the item embedding into messages being passed. The *combination* seeks to integrate representations of the center node and its neighbors. The common operation includes concatenation [143] and summation [145, 148]. The *pooling* operation aims to combine outputs from multiple layers of GNNs and generate the final representation for each node. For example, PinSage [143] uses the representation from the last layer, whereas NGCF [148] and IGMC [151] concatenates representations from all layers.

2.1.2 Sequential Recommender Systems

Sequential recommendation is a subfield of recommender systems that aims to provide personalized item recommendations to users over time. It considers the order in which items are consumed by users to predict the next item the user is likely to interact with. Sequential recommender systems can be classified into traditional methods, latent representation methods, and deep learning methods [4, 36].

2.1.2.1 Traditional Methods

Traditional methods for modeling sequential data, such as sequential pattern mining [36, 37] and Markov chain models [38–40], serve as intuitive solutions for sequential recommender systems, leveraging their inherent proficiency in capturing sequential dependencies within user-item interaction sequences.

Sequential pattern mining. Sequential pattern-based recommender systems mine frequent patterns from sequence data and then leverage these patterns to inform subsequent recommendations. For example, Yap et al. [37] present a personalized recommendation framework based on sequential pattern mining. By introducing a novel Competence Score measure, this framework effectively discerns user-specific sequence significance and harnesses this insight to enhance the accuracy of personalized recommendations. While this approach is straightforward, sequential pattern mining frequently yields a plethora of redundant patterns, thereby incurring unnecessary computational costs in terms of time and space. Another notable limitation is its tendency to overlook infrequent patterns and items due to frequency constraints, thereby biasing recommendations towards popular items [36].

Markov chain. Markov chain-based methods employ Markov chain (MC) models to model transitions in user-item interaction sequences, aiming to predict the next interaction. For instance, FPMC [38] integrates both an MF term and an item-item transition term to capture long-term preferences and short-term transitions, respectively. While the transitions captured by such models typically employ a first-order MC, higher-order MCs contend that the next action depends on multiple preceding actions. Some methods leverage high-order MCs to account for more previous items [39, 40]. As an example, He and McAuley [40] introduce a fusion of similarity-based methods and Higher-order MC methods to address sparsity

in real-world datasets exhibiting sequential dynamics. Yet, MC-based methods present discernible limitations. Specifically, they are constrained by their ability to only recognize short-term dependencies due to the inherent Markov property, which posits that the current interaction is influenced solely by the most recent interactions. Additionally, they only discern point-wise dependencies, overlooking the collective interdependencies present in user-item interactions [36].

2.1.2.2 Latent Representation Methods

Latent representation methods initially derive a latent representation for each user or item. Subsequently, these representations are employed to predict potential user-item interactions. Through this approach, intricate and implicit dependencies are discerned within a latent space, substantially enhancing recommendation outcomes. These methods predominantly bifurcate into factorization-based and embedding-based methods.

Factorization-based methods. Factorization-based methods typically employ matrix factorization or tensor factorization to decompose observed user-item interactions into latent factors, pertaining to both users and items, for enhanced recommendations [38, 152]. However, these models are susceptible to the sparsity inherent in observed data, often resulting in suboptimal recommendations.

Embedding-based methods. Embedding-based methods obtain latent representations for each user and item to inform subsequent recommendations, effectuating this by encoding all user-item interactions in a sequence into a latent space. Specifically, certain studies [153, 154] utilize the learned latent representations as the input to a network, with the aim of further computing an interaction score between users and items. For example, Wang et al. [154] propose a shallow wide-in-wide-out network, which is an attention-based model that learns an attentive context embedding that intensifies relevant items but downplays those irrelevant to the next choice. Other work [155] directly employs the learned latent representations to calculate a metric, such as the Euclidean distance, as the interaction score. Specifically, He et al. [155] propose to embed items into a 'transition space' where users are modeled as translation vectors operating on item sequences. The model is optimized to consider the three-order relationships between users, candidate items, and previous behaviors.

2.1.2.3 Deep Learning Methods

With the rapid advancement of deep learning techniques, deep neural networks have become increasingly prevalent in sequential recommendations. These networks possess the ability to discern intricate, non-linear patterns in data—an essential trait for capturing nuanced user-item interactions in sequences. Moreover, they are adept at modeling sequential data, effectively capturing temporal dynamics. Methods based on deep learning can be categorized into four groups, depending on the specific deep learning technique employed: Recurrent Neural Networks (RNNs), CNNs, GNNs, and Transformer.

RNNs. RNNs-based recommendation methods have shown exceptional capabilities in harnessing sequential information for recommendations [41, 43]. Beyond the basic RNNs, additional variants, such as Long-Short-Term-Memory (LSTM) [42] and Gated Recurrent Unit (GRU) [43], have been developed to adeptly capture long-term dependencies within a sequence. For instance, Wu et al. [42] introduce a recurrent recommender network, a non-parametric recommendation model grounded in LSTM technology. Utilizing two LSTM networks, the model constructs dynamic user and item states. Concurrently, it incorporates stationary latent attributes of both users and items, taking into account stable properties such as users' long-term interests and static item features. Donkers et al. [43] present GRU4Rec, a session-based recommendation model employing the GRU. This model interprets user behavioral sequences as time-series data and uses a multi-layered GRU structure to encapsulate sequential nuances. To optimize the training of this architecture, the authors introduce modifications to the conventional GRU. These adjustments include the deployment of session-parallel mini-batches, mini-batch-based output sampling, and a ranking loss function to enhance its adaptability to the recommendation task. However, methods based on RNNs have two notable limitations: Firstly, they can inadvertently introduce spurious dependencies owing to the strong assumption that any adjacent interactions in a sequence are inherently dependent. This assumption does not always hold in real-world scenarios, as sequences often contain irrelevant or noisy interactions. Secondly, such methods tend to capture only point-wise dependencies and overlook collective dependencies where multiple interactions collaboratively influence the subsequent interaction [36].

CNNs. CNNs-based recommendation methods interpret sequences in a way that is analogous to image processing. Specifically, a CNN begins by organizing the

embeddings of these interactions into a matrix, which is then interpreted as an "image" across both time and latent spaces. Subsequently, CNNs identify sequential patterns as local features of this "image" using convolutional filters, which are integral for subsequent recommendations. While RNNs face challenges in processing long sequences due to their inherent structure and substantial computational expenses, CNNs can partially mitigate these limitations. For example, Caser conceptualizes the embedding matrix of prior items as an "image." It employs both horizontal and vertical convolutional layers to detect point-level and union-level sequential patterns, respectively. Through convolution, it becomes feasible to perceive relevant skip behaviors and encapsulate long-term user preferences via user embedding. NextItNet [44] represents a generative CNNs model that incorporates masked filters with 1D dilated convolutions, expanding the receptive fields for sequential recommendations. This facilitates the capture of both long-term and short-term item dependencies. GRec [45] builds on NextItNet by implementing a gap-filling encoder-decoder framework combined with masked-convolution operations. This allows for a comprehensive evaluation of both past and future contexts without the challenge of data leakage. However, CNNs-based models encounter challenges in capturing long-term dependencies. This limitation primarily stems from the constrained sizes of the filters. As a result, their applicability in scenarios requiring the understanding of long-term sequential patterns is restricted.

GNNs. GNNs-based recommendation methods typically construct diverse graph forms to encapsulate the item transition patterns within user sequences. A notable body of work [32, 35, 46–48] constructs a directed graph derived from data sequences, leveraging GNNs to ascertain item transition patterns. For instance, SR-GNN [32] conceptualizes each sequence as an unweighted, directed graph, with the resulting sequence representation achieved through a gated GNN. GC-SAN [35], an offshoot of SR-GNN, integrates a self-attention mechanism, enhancing the capture of contextual information between historical items. SURGE [48] first represents each user's interaction history as a graph. It then harnesses the robust capabilities of GNNs to fuse and distill users' core interests from noisy behavior sequences, restructuring these loose item sequences into cohesive item-item interest graphs via metric learning. Nonetheless, sequence graphs derived from isolated, short sequences often comprise minimal nodes and connections, presenting an insufficient knowledge base to mirror users' dynamic preferences and failing to harness GNNs to their fullest in graph learning.

To augment the inherent sequence graph structure, another set of works introduces additional sequences, capturing further details from analogous user sequences or all user sequences. This inclusion aids in understanding the transition patterns of the current sequence. These supplementary sequences might represent different behavioral patterns. For instance, HetGNN [156] incorporates all behavior sequences, constructing edges between consecutive items within the same sequence, and labeling these with their respective behavioral types. Alternatively, the sequences might represent a subset or the entirety of sequences within a dataset. For instance, RetaGNN [157] initially forms a global tripartite graph encapsulating relationships among users, items, and item attributes from all sequences. It then derives a local subgraph from this global structure for each user-item pair in a sequence, employing self-attention to encode both long-term and short-term temporal patterns. TASRec [158] designs a dynamic graph prioritizing recent transitions and formulates a specialized graph neural network to discern temporal augmented item representations. These are then integrated into standard sequential neural networks to optimize recommendations. GCL4SR [159] constructs a weighted item transition graph grounded in interaction sequences across all users, providing a holistic context for each interaction and minimizing sequence data noise. Augmented sequence representations are then derived from this transition graph, serving a contrastive learning objective.

Additionally, certain studies [113, 160] strive to amplify recommendations by formulating intricate graphs that adeptly model high-order relations in sequential data. Taking cues from hypergraphs' capacity to model relations beyond the pairwise realm, these structures have been exploited to comprehend the high-order relations amongst items and cross-session data. For example, SHARE [160] employs sliding windows to grasp contextual information, connecting items within the same window through a hyperedge. Conversely, DHCN [113] views each session as a hyperedge where all items interlink, and varying hyperedges, linked through shared items, form a hypergraph encapsulating item-level high-order correlations.

In a departure from directly translating temporal sequences into graph's directed edges, some research [161, 162] contemplates timestamps within sequences during graph construction. Within these graphs, each edge symbolizes a user-item

interaction, complemented by the pertinent time attribute. Convolution operations are then conducted on these temporal graphs to glean user and item representations. Specifically, Fan et al. [162] introduce a Continuous Time Bipartite Graph (CTBG), encompassing user/item nodes and interaction edges annotated with timestamps. This approach facilitates the dissemination of temporal collaborative insights around each node to adjacent nodes within the CTBG, harmoniously melding sequential patterns with temporal collaborative signals.

Transformer. Recently, transformer-based recommendation methods have exhibited notable efficacy in identifying long-range dependencies within sequences [49–53, 163]. For instance, SASRec [49] employs a self-attention mechanism coupled with a positional encoding embedding to encode a sequence of items. BERT4Rec [50] builds upon SASRec by introducing a bi-directional self-attention module. Moreover, CL4SRec [53] innovates with three data augmentation strategies—item cropping, masking, and reordering—to bolster contrastive tasks and facilitate self-supervised signal extraction. HPM [164] introduces a dual-transformer module coupled with a dual contrastive learning framework. This design is tailored to discerningly capture both low- and high-level user preferences to augment the learning of these preferences at both granularity levels.

2.2 Modality-enhanced Recommender Systems

Modality-enhanced recommender systems, apart from utilizing user interaction data, integrate various data types like text, images, audio, and video. This incorporation of diverse data modalities aids in forming a more comprehensive and accurate understanding of user preferences. Based on the types of content modalities, modality-enhanced recommender systems can be classified into three categories: text-enhanced recommender systems, visual-enhanced recommender systems, and multimodal recommender systems [1, 2].

2.2.1 Text-enhanced Recommender Systems

Recent works [51, 54, 56–59, 80] have attempted to leverage textual data of items, such as descriptions, attributes, or brands of products, to improve item representations for recommendations. Text-enhanced recommender systems have gained increasing attention due to the explosion of text data and the need for more personalized and informative recommendations. Existing text-enhanced methods propose building an auxiliary task by classifying text tokens, employing pre-trained language models for feature extraction, or fine-tuning pre-trained language models specifically for recommendation tasks.

Text tokens classification. Some works [51, 54] focus on modeling item texts as tokens (*i.e.*, attributes or brands) and optimizing the model with the token classification task. For example, S^3-Rec [51] adopts a pre-training strategy that leverages intrinsic data correlations among attributes, items, subsequences, and sequences. This approach generates self-supervision signals, thereby improving the quality of data representations. In DIF-SR [54], the modeling of item attributes is moved from the input to the attention layer. The attention calculation of auxiliary information and item representation is decoupled to improve the modeling capability of item representations.

Feature extraction using pre-trained language models. Other works [55–61] leverage the pre-trained language model as a feature extractor. Texts of items are fed into the language model and the corresponding item feature embeddings are output. A recommender model can utilize these knowledge-aware embeddings for various recommendation tasks. For instance, FDSA is proposed by [56], where different item features are first aggregated using a vanilla attention layer, followed by a feature-based self-attention block to learn how features transit among items in a sequence. Wu et al. [165] explore to model news with pre-trained language models and finetune them with the news recommendation task. Hou et al. [57] propose a framework UniSRec, utilizing item texts with an MoE-based adaptor and employing contrastive learning tasks to derive more transferable representations for sequential recommendations. It further involves a linear transformation of the original text representations to mitigate their anisotropy problem. Hou et al. [60] propose to transform text encodings into discrete codes, followed by utilizing the embedding lookup for refining item textual representation from pre-trained language models. Li et al. [61] explore the limits of text-based collaborative filtering

(TCF), finding that it hasn't reached its full potential and could improve with advancements in NLP models. However, they highlight a significant challenge: even highly complex item encoders require re-tuning for new data, and current TCF models lack the expected transferability, indicating that building foundational recommender models is more complex than in NLP and CV fields.

Fine-tuning pre-trained language models. Another line of work [58, 62–64] aims to directly finetune the pre-trained language models with the recommendation task to combine the learning from texts and sequential patterns of users. NRMS [62] is a neural news recommendation method utilizing multi-head self-attention. It comprises a news encoder, which learns from news titles by analyzing word interactions, and a user encoder, which derives user profiles from their news browsing patterns, emphasizing the relationships between news items. The model also employs additive attention to enhance the selection of keywords and news, thereby refining news and user representations. Geng et al. [63] introduce the "Pretrain, Personalized Prompt, and Predict Paradig" (P5), a unified text-to-text framework for recommendations. P5 converts all relevant data, including user-item interactions, user descriptions, item metadata, and user reviews, into natural language sequences, enabling deeper semantic understanding for personalization. It uses a consistent language modeling objective in pretraining, making it a foundational model for various recommendation tasks. P5's structure also allows easy integration with other data modalities and supports instruction-based recommendations through personalized prompts. Li et al. [64] present Recformer, a framework for learning language representations in sequential recommendation. It represents items as key-value attribute pairs and employs a novel bi-directional Transformer model to understand natural language and sequential patterns. The framework also includes a pretraining and finetuning learning structure, enhancing its ability to make language-based recommendations and adapt to different recommendation contexts.

2.2.2 Visual-enhanced Recommender Systems

Visual-enhanced recommender systems, depending on the type of visual information used, are classified into two categories: image and video recommender systems.

Image recommender systems. Current image recommendation approaches fall into two primary categories [1]: content-based [65–67] and hybrid models [68–70]. Content-based [68] models utilize visual signals to construct visual representations of items, representing user preferences within this visual space. Lei et al. [65] propose a dual-net deep network model for learning hybrid representations that capture both visual information and user preferences regarding images. This model, aimed at personalized image recommendations, is trained using a comparative deep learning method. This method employs triplets of users and positive/negative images, focusing on learning the relative distances between these elements. Yang et al. [67] employ a two-way architecture, extracting decision rules from a boosted tree model and learning rule embeddings with attention to attribute interactions. Additionally, it integrates visual and rule-based information in a shared embedding space, facilitating mutual enhancement between visual and rule spaces.

Conversely, hybrid recommendation models address the data sparsity issue common in collaborative filtering by incorporating item visual modeling. VBPR [68] integrates visual features from product images into the matrix factorization recommendation framework by introducing visual embeddings for users and items. Item embeddings are extracted using a pre-trained convolutional neural network. The recommendation score is computed by summing the inner products of both collaborative and visual embeddings. He and McAuley [69] develop a scalable model that uses product images and user feedback to track the changing trends in fashion and personal preferences over time. In this model, the visual content of items is utilized as a regularization term in matrix factorization models, aligning each item's latent vector closely with its visual image representation derived from CNNs. Li et al. [70] suggest the creation of a heterogeneous graph comprising users, outfits, and items, utilizing hierarchical GNNs for personalized outfit recommendations.

Video recommender systems. In the domain of video recommender systems, several studies [71, 72] develop content-based systems enriched with visual and audio information. These models first extract features from video and audio content, subsequently utilizing neural networks for the integration of these features through either early or late fusion techniques. A key characteristic of these content-based models is their independence from user-video interactions, which enables their application to recommend new videos without the need for historical user behavior data. In contrast, studies such as [138] have introduced an attentive collaborative

filtering model which integrates user-video interactions for multimedia recommendations. This model effectively combines an attention mechanism with visual inputs to accurately calculate weights, capturing users' historical preferences and the specific characteristics of the items.

2.2.3 Multimodal Recommender Systems

Multimodal recommender systems [166] utilize various data types like text, images, audio, and video to create more accurate and personalized user suggestions. These systems can be categorized based on their neural network architecture: traditional MF, attention networks, and GNNs.

MF-based methods. Early works [66, 68, 73, 74] incorporate multimodal information into MF-based recommendation frameworks. An example is DeepStyle, proposed by Liu et al. [73], which integrates style feature modeling into visual recommender systems using the Bayesian Personalized Ranking (BPR) framework, focusing on learning item style features to better align with user preferences. AMR [74] underscores the vulnerability of multimedia recommender systems and aims to enhance the robustness of multimodal recommender systems. It introduces adversarial learning to train the model to defend an adversary, creating a more robust system.

Attention-based methods. Attention networks [75–79] have also been employed to merge multimodal features for improved user and item representation learning. For instance, Liu et al. [75] learns the multimodal representation for both users and micro-videos to provide personalized micro-video recommendations. Rather than focusing solely on video attention, UVCAN employs a co-attention mechanism between items and micro-videos for better joint attention performance. A stacked attention network is utilized to process user profiles and micro-video multimodal features, taking the multimodal features as input queries and obtaining video attention through multi-step reasoning. The learned video representation is then used as the input query to capture user attention via multi-step reasoning. Chen et al. [77] propose a visually explainable collaborative filtering model that combines image region-level features with user review data, using a multimodal attention network for effective integration of these features. Additionally, Tran and Lauw [79] propose a method to transform uninterpreted dimensions in user

representations into words. This is achieved by regularizing factors derived from user-item interactions with those learned from textual content. They introduce an attention-based alignment technique to enhance and align hidden factor representations.

GNNs-based methods. Recent advancements leverage GNNs to utilize multimodal item information [55, 59, 80–84, 167–170]. MMGCN [55], for example, uses a modality-specific user-item bipartite graph with GCNs' message passing mechanism, capturing multi-hop neighbor information to enhance user and item representations. DualGNN [82] further expands this approach by adding a user co-occurrence graph and a model preference learning module. MVGAE [83] introduces a multi-modal variational graph auto-encoder, fusing node embeddings per the product-of-experts principle. LATTICE [81] presents a modality-aware graph structure learning layer for incorporating high-order item affinities into item representations. In addition, BM3 [59] eliminates the need for randomly sampled negative examples in user-item interaction modeling, using latent embedding dropout for self-supervised learning and a novel multi-modal contrastive loss. Yu et al. [84] present a multi-view graph convolutional network, MGCN, for multimedia recommendations, which purifies modality features using item behavior information to reduce noise. These features are then enriched in separate user-item and item-item views, enhancing feature distinguishability. A behavior-aware fusion module is also developed to model user preferences by adaptively prioritizing different modality features.

2.3 Enhancing Model Generalization

In recommender systems, employing data augmentation and contrastive learning methods is crucial for enhancing model generalization, particularly given the prevalent issue of data sparsity. Data augmentation in this realm involves artificially expanding the training dataset by generating modified versions of existing data points, thereby enabling the model to learn more robust and generalizable representations. Contrastive learning, on the other hand, involves differentiating between similar (positive) and dissimilar (negative) example pairs, often created through data augmentation. By forcing the model to focus on the subtle differences and similarities in user interactions, contrastive learning encourages the development of

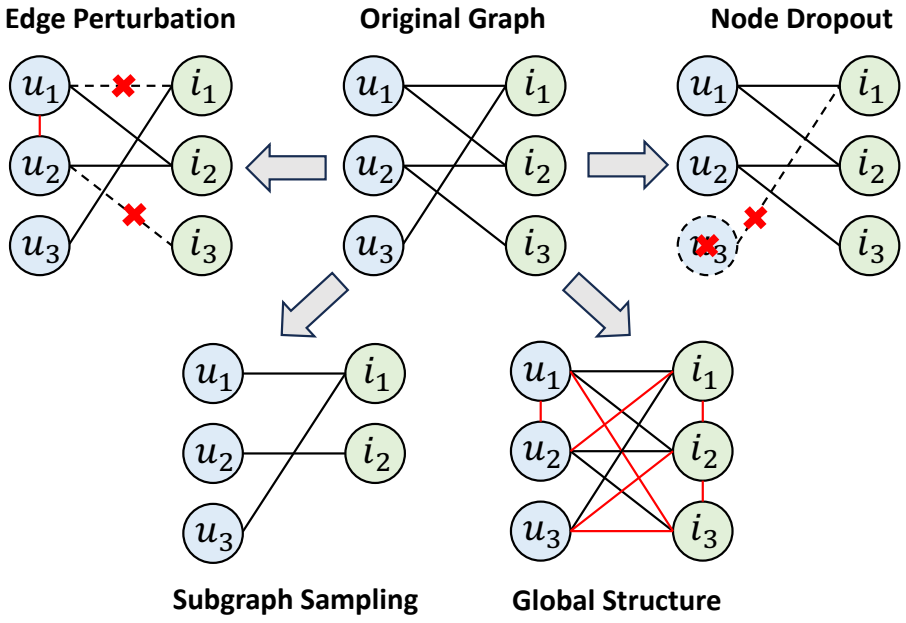


FIGURE 2.3: Graph-based augmentation [86, 87].

a more nuanced and generalizable feature space. Collectively, these approaches mitigate overfitting and improve the capability of the recommender systems to make accurate recommendations of unseen data, ultimately leading to better performance in real-world applications where user preferences are continuously evolving and highly diverse.

2.3.1 Data Augmentation

The commonly used data augmentation methods [86, 87] in recommender systems can be divided into three categories, including graph-based, sequence-based, and feature-based.

2.3.1.1 Graph-based Augmentation

Many existing studies construct graphs to capture relationships between users and items, as well as exclusively among users or items, resulting in user-item, user-user, and item-item graphs respectively. Based on these graph structures, there are primarily three types of augmentation methods: edge/node dropout, graph diffusion, and subgraph sampling.

Edge perturbation. This strategy involves generating augmented graphs by perturbing the connectivities within a graph, typically through randomly adding or dropping a certain ratio of edges. Various studies [88–92] propose to randomly remove a proportion of edges. In this scenario, only a selected subset of connections influences node representations, thereby enhancing their robustness by eliminating redundant or noisy interactions. In addition, Yang et al. [93] propose an enhancement to the original graph by adding edges based on a learned user-item similarity matrix. This addition aims to strengthen the graph structure by integrating possible meaningful connections that reflect user-item relationships more accurately. Furthermore, some researchers [94, 95] suggest adaptively calculating edges to maintain important connections while potentially altering less significant ones. This approach aims to refine the graph structure by prioritizing the edges that are most relevant, ensuring that the augmented graph remains representative of significant relationships and patterns.

Node dropout. With a predefined probability, nodes can be selectively removed from the graph. Dropping a proportion of nodes [88, 92, 96] aids in identifying influential nodes, thereby refining the overall graph structure and its representational efficacy [87].

Subgraph sampling. Subgraph sampling is a technique that involves extracting a subset of nodes and edges from a larger graph for analysis. This method focuses on capturing the local connectivity within the graph. Subgraphs are typically generated using various methods, including random walks, uniform sampling, ego-network sampling, and knowledge-based sampling. Random walks [88, 92, 97] involve traversing the graph randomly and selecting nodes from these paths to create subgraphs. Uniform sampling [88, 90, 98] involves randomly choosing a set of nodes and their corresponding edges to form subgraphs, with Wu et al. [88] exemplifying this by randomly dropping a portion of nodes or edges. Ego-network sampling [99] focuses on sampling a specific number of hops of neighbors for each node in the graph, thus capturing the local neighborhood structure around each node. On the other hand, Knowledge-based sampling [92, 100] incorporates domain knowledge in the sampling process of subgraphs. For instance, Yu et al. [100] design triangular motifs based on underlying semantics, which specify high-order relations like “having a mutual friend”, thereby infusing more contextual relevance into the subgraph structure.

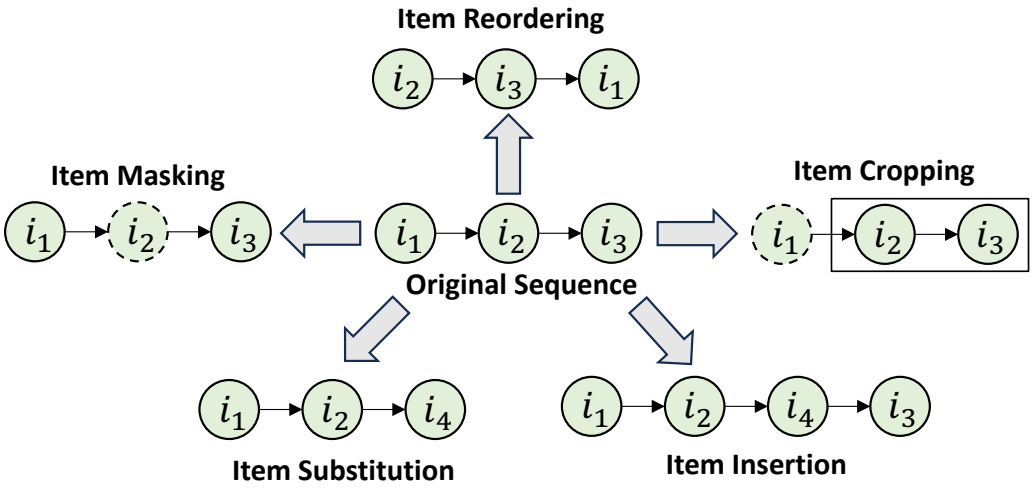


FIGURE 2.4: Sequence-based augmentation [86, 87].

Global structure-enhanced. This strategy, including graph diffusion and SVD-based augmentations, emphasizes the creation of graph augmentations by harnessing global information or patterns present in the graph. Some works [101, 102] utilize graph diffusion techniques to integrate global information and refine the original graph through the creation of new edges. For instance, Zhang et al. [102] develop a sparsified diffusion graph derived from the user-item interaction graph, employing the Personalized PageRank algorithm for its construction. Additionally, another approach [103] involves applying a Singular Value Decomposition (SVD)-based method for graph augmentation to effectively capture global collaborative signals. Specifically, this process begins with performing SVD on the adjacency matrix, followed by truncating the list of singular values to retain only the top largest values.

2.3.1.2 Sequence-based Augmentation

Taking into account the temporal factor of user interactions with items, a user's behavior can be represented as a sequence of items. Common sequence-based augmentation methods, which modify the original sequence, generally encompass five types: item masking, item cropping, item reordering, item substitution, and item insertion.

Item masking. The item masking strategy [51, 53, 104–106] involves randomly masking a portion of items in a sequence and substituting the masked items with

a special token, [mask], similar to the masking technique used in BERT [171]. This approach is viable because a user's intention tends to remain stable over a period [87]. Additionally, sequences may contain noisy items that do not accurately reflect the user's true preferences. By masking some items, the primary intent information extracted from the partial sequence is still preserved.

Item cropping. Inspired by the random crop technique commonly used in computer vision to augment image data, the item crop augmentation method in recommender systems involves randomly selecting a continuous sub-sequence from a user's historical sequence [51, 53, 104, 105]. This approach offers a localized view of the user's interaction history. The effectiveness of the item crop method lies in its ability to enhance the user representation model by learning to generalize user preferences without needing comprehensive user information [53]. This technique aids in increasing the model's robustness and adaptability by focusing on partial, yet significant, user interaction patterns.

Item reordering. Many existing methods [38, 107, 108] assume a strict sequential dependence between adjacent items in a user's historical sequences. However, in real-world scenarios, the order of user interactions is often flexible, influenced by various unobserved external factors. Consequently, different item orders may actually reflect identical user intents [36, 172]. Recognizing this, some research [53, 104] proposes the item reordering strategy for data augmentation. This technique involves randomly shuffling a continuous subsequence within the user's interaction history, which helps the model reduce its reliance on the exact order of interaction sequences, thereby enhancing its robustness, especially when encountering new interactions.

Item substitution. Random item cropping and masking can potentially exacerbate the data sparsity issue, particularly in short sequences. To address this, some studies suggest item substitution in short sequences with either random items [108] or highly correlated items [109]. When random items are injected into sequences, these modified sequences are treated as negative samples. A classifier is then employed to predict the likelihood of user interaction with these altered sequences. This approach helps in enhancing the robustness of the model by training it to distinguish between genuine and artificially altered user behaviors. In contrast, the injection of highly correlated items into sequences introduces less disruption to the original sequential information. The selection of these correlated items is based on

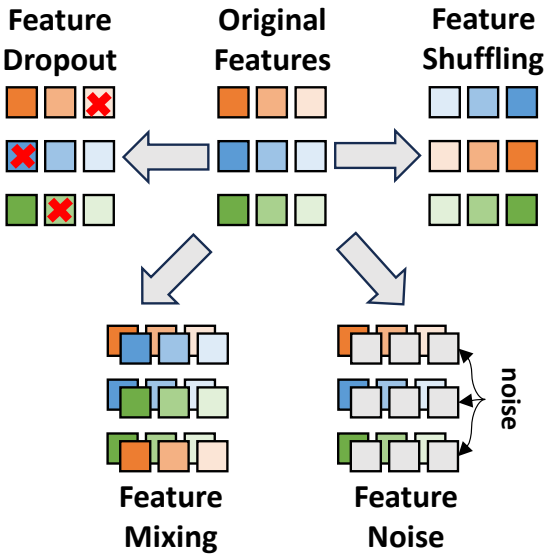


FIGURE 2.5: Feature-based augmentation [86, 87].

a correlation score, which is typically derived from item co-occurrence frequencies or the similarity of their respective representations. This method maintains the integrity of the original sequence to a greater extent, ensuring that the augmented data still closely reflects realistic user behaviors and preferences.

Item insertion. In practice, the interactions in the dataset usually represent only a fraction of users' complete behaviors, with transactions from other sources frequently missing, as noted by Liu et al. [110]. This limitation is particularly pronounced in short sequences, where the recorded interactions may not adequately capture the full spectrum of user dynamics and item correlations. As a result, training sequences derived from such limited data may fail to comprehensively represent user behaviors and preferences. To address this issue, Liu et al. [109] propose the insertion of correlated items into short sequences as a method to enhance and complete these sequences.

2.3.1.3 Feature-based Augmentation

Feature-based augmentation focuses on performing augmentations on feature vectors, which can be categorical features derived from categorical attributes or learned continuous feature embeddings.

Feature dropout. Feature dropout [111, 112] involves randomly masking or dropping a portion of features. Specifically, Wang et al. [111] suggest masking a proportion of initial features, wherein any masked feature's representation is replaced with [mask], represented as a zero vector.

Feature shuffling. Feature shuffling [93, 100, 113] involves randomly interchanging rows or columns within the feature matrix, where each row represents the feature of a graph node. As a result, the model learns representations based on changing contextual information.

Feature mixing. This augmentation strategy involves mixing various modality features from the same user or item [114], or combining features from different users or items [91, 115], for feature interpolation. For example, Zhou et al. [91] introduce embedding perturbation by leveraging historical embeddings from prior training iterations.

Feature noise. This strategy [116–118] involves generating varied views by introducing different types of noise to the original embeddings. This approach stands apart from feature-based augmentations, which typically perturb only the input embeddings or the final representations. Instead, it introduces noise at different layers of the encoder. A notable example is presented by Yu et al. [116]. They focus on regulating the uniformity of the representation distribution by adding directed random noises to the representations.

2.3.2 Contrastive Learning

The key idea of contrastive learning is to build multiple views of the input through proper transformations and maximize the agreement between different views of the same instance while minimizing the agreement between different instances in order to learn a more generalized and robust representation. The previous section outlines flexible data augmentation techniques that are instrumental in creating these varied views. Following this, the subsequent section delves into various contrastive methods, exploring how instances are contrasted and the techniques used to measure their mutual information.

Based on the type of instances used for constructing contrastive loss, contrastive methods can be categorized into local-local contrast, global-global contrast, local-global contrast, and local-context contrast. For graph structures, local refers to nodes and global refers to graphs. For sequence structures, local refers to items and global refers to sequences. The context scale is between the local and global scales and represents the subgraph or subsequence [86, 87].

2.3.2.1 Local-Local Contrast

Local-local contrast is a technique employed to distinguish between local representations. In graph-based methods, a notable example is the SGL [88], which utilizes stochastic graph augmentations such as node dropout, edge dropout, and random walk on user-item bipartite graphs. SGL begins by generating two augmented graphs using identical augmentation operators. Subsequently, it employs a shared LightGCN encoder to derive node embeddings from these augmented graphs. The model performs node-level contrast by optimizing the InfoNCE loss, utilizing in-batch negative sampling. In its final stage, SGL concurrently optimizes the InfoNCE loss and the BPR loss to enhance recommendation accuracy. Also, DCL [122] adopts stochastic edge dropout and specifically perturbs the L-hop ego-network of a node. This process yields two augmented subgraphs. DCL then focuses on maximizing the consistency between node representations learned from these two subgraphs.

Regarding sequence-based methods, particularly in sequential recommendation contexts, COTREC [113] constructs dual graph views: the item view (item-item graph) and the session view (session-session graph). It aims to align the representation of a session's final item with those of predicted positive samples, while simultaneously minimizing the agreement with negative sample representations.

2.3.2.2 Global-Global Contrast

The global-global contrast technique focuses on differentiating global representations and is frequently employed in sequential recommender systems. An example is CL4SRec [53], which employs three distinct random augmentation operators: item masking, item cropping, and item reordering, to modify input sequences.

These augmented sequences are then processed through a Transformer-based encoder [173]. The encoder facilitates the learning of representations from these modified sequences, which are subsequently employed for global-level contrast. Similarly, DuoRec [174] generates different sequence representations by applying message dropout. In this method, the same sequence is encoded twice using different dropout masks in a Transformer-based encoder. Subsequently, DuoRec conducts sequence self-discrimination, contrasting the representations derived from these differently masked sequences.

2.3.2.3 Local-Global Contrast

The local-global contrastive approach involves contrasting local and global representations. This method aims to infuse high-level global information into local structure representations, particularly in graph-based scenarios. For instance, EGLN [93] introduces a method to achieve local-global consistency. It contrasts between the edge representation and the global graph representations. Specifically, the edge representation is defined as the concatenation of the representations of its connected nodes. Meanwhile, the global graph representation is determined as the average of all edge representations. For HGCL [175], user and item node-type specific homogeneous graphs are constructed. Within each homogeneous graph, the method focuses on maximizing the mutual information between local segments of the graph and the global representation of the entire graph. Moreover, HGCL introduces a cross-type contrast, which measures both local and global information across different types of homogeneous graphs.

2.3.2.4 Local-Context Contrast

Local-context contrast methods are employed in both graph and sequence-based scenarios to create a contrast between the contextual and global representations. In these methods, the context is typically formed by sampling ego-networks or through clustering techniques.

In graph-based methods, MHCN [100] categorizes three types of triangular social relationships, which are then modeled using a multi-channel hypergraph encoder. Within each channel, MHCN optimizes mutual information among three levels

of representation: the individual user representation, the user's ego hypergraph representation, and the global hypergraph representation. This hierarchical approach ensures a comprehensive understanding of user relationships at various levels. Meanwhile, NCL [176] introduces a prototype-contrastive objective, where the positive sample for each item or user is the prototype of its cluster, and the negative sample is from other clusters' prototypes. This objective is optimized using the Expectation-Maximization algorithm. NCL also employs cross-layer contrasting, using representations from even-numbered GNN layers as positive samples for each user or item.

In sequence-based methods, ICL [177] is developed for sequential recommendation. It learns user intent distributions from behavior sequences via clustering and incorporates these intents into the sequential recommendation model using a contrastive learning loss to contrast sequence views with their respective intents.

Chapter 3

Preliminaries

In this section, we present the preliminaries of data augmentation and contrastive learning within the context of recommender systems, while introducing the relevant notations used throughout the discussion.

3.1 Data Augmentation

Data augmentation in recommender systems involves expanding the training dataset by creating modified versions of existing datasets. This process helps the model learn more robust and generalizable representations. In this thesis, we explore three primary data augmentation techniques tailored to different data structures and recommendation tasks. We denote the augmentation operator as $\mathcal{T}$.

3.1.1 Graph-based Augmentation

Consider a user-item interaction graph $\mathcal{G} = \{\mathcal{U}, \mathcal{V}, \mathcal{E}\}$ with $\mathcal{U}$ and $\mathcal{V}$ representing the sets of users and items, respectively, and $\mathcal{E}$ denoting the edges indicating interactions between them. The adjacency matrix of this graph is represented by $\mathbf{A}$. We apply various graph augmentation techniques, such as edge perturbation, node dropout, subgraph sampling, and enhancements based on global structure. These augmentations can be formalized as:

$$\widetilde{\mathcal{G}}, \widetilde{\mathbf{A}} = \mathcal{T}_{Graph}(\mathcal{G}) = (\widetilde{\mathcal{U}}, \widetilde{\mathcal{V}}, \widetilde{\mathcal{E}}), \tag{3.1}$$

where $(\widetilde{\mathcal{U}}, \widetilde{\mathcal{V}}, \widetilde{\mathcal{E}})$ represent the new set of users, items, and edges in the augmented graph.

3.1.2 Sequence-based Augmentation

Given a sequence of item interactions $\mathcal{S} = \{i_1, i_2, \ldots, i_n\}$ ordered chronologically by interaction timestamps, sequence-based augmentation methods such as item masking, cropping, reordering, substitution, and insertion can be applied. These methods redefine the original sequence into a new sequence:

$$\widetilde{\mathcal{S}} = \mathcal{T}_{Seq}(\mathcal{S}) = \{i_{r_1}, i_{r_2}, \cdots\}, \tag{3.2}$$

where each i_r represents an item in the newly generated sequence.

3.1.3 Feature-based Augmentation

Feature-based augmentation focuses on applying augmentations to feature vectors, which can consist of categorical features derived from categorical attributes or learned continuous feature embeddings. Denoting feature embeddings by $\mathbf{X}$, common methods include feature dropout, shuffling, mixing, and the addition of noise. This transformation can be represented as:

$$\mathbf{Z} = \mathcal{T}_{Feature}(\mathbf{X}) = \Phi(\mathbf{X}), \tag{3.3}$$

where Φ denotes the function to transform the original feature embeddings. For example, Φ can represent the whitening transformation (details can be found in Chapter 5) or a masking matrix to exclude certain elements in $\mathbf{X}$.

3.2 Contrastive Learning

After elucidating the various data augmentation techniques that enhance the training dataset, we utilize the augmented datasets to formulate the contrastive learning task. Contrastive learning is intended to train models that adeptly differentiate

between similar and dissimilar instances, thereby capturing deeper nuances in the data and learning a more generalizable embedding space.

Considering the representations of the same instance (which may vary across different scales such as local, global, or contextual) learned from two augmented views, $\mathbf{e}_i$ and $\mathbf{e}_j$, the objective of the contrastive loss is to maximize the mutual information between these representations. Directly maximizing mutual information presents challenges, and a practical approach involves maximizing its lower bound. In this thesis, we employ the widely used lower bound InfoNCE [178], which is formulated as follows:

$$\mathcal{L}_{\text{NCE}} = -\mathbb{E}\left[\log \frac{e^{f(\mathbf{e}_i,\mathbf{e}_j)}}{\sum_{n\in\mathcal{N}_i^-\cup\{j\}} e^{f(\mathbf{e}_i,\mathbf{e}_n)}}\right] \tag{3.4}$$

where $\mathcal{N}_i^-$ is the negative sample set of i. The negative set is often sampled within a batch. $f(\mathbf{e}_i,\mathbf{e}_j) = \frac{\mathbf{e}_i^\top \mathbf{e}_j}{\|\mathbf{e}_i\|\|\mathbf{e}_j\|}/\tau$ where τ is the temperature.

The subsequent chapters will further detail the development of effective data augmentation techniques and contrastive learning strategies for recommender systems, aimed at enhancing model generalization and alleviating data sparsity.

Chapter 4

Diffusion-based Graph Contrastive Learning for Recommendation[1]

As we have explored existing recommendation systems and established foundational concepts in Chapters 2 and 3, we now move forward to address one of the primary challenges in recommendation systems: handling large and sparse interaction graphs for data augmentation. In Chapter 4, we analyze the original user-item interaction graph and propose a data augmentation method that minimizes the introduction of noise or bias, ensuring the integrity of the underlying data structure.

4.1 Overview

The recent development of deep learning motivates the emergence of various neural network-based recommendation models [120, 121]. One representative group of studies leverages graph neural networks (GNNs). With the advantage of exploiting the high-order connectivity of users and items through iterative message propagation, GNNs-based models have shown prominent performance in recommendation tasks. However, these methods still suffer from insufficient supervision signals from observed sparse user-item interactions, unbalanced distributions of the

[1]The work in this chapter has been published in DASFAA 2022 [102].

user (or item) node degree, and unavoidable noises (*e.g.*, accidentally clicking) from users' implicit feedback for most datasets [88].

Self-supervised learning (SSL) has recently become a promising technique to address these limitations for a more robust and generalized representation learning of GNNs [179]. Researchers have begun to actively explore SSL in GNNs-based recommendation models [88, 99, 122, 123]. Most of them concentrate on contrastive learning, which maximizes the agreement between representations of augmented views and the original graph. The key part of contrastive learning is how to design the graph augmentation strategy. The most commonly endorsed approach is randomly removing edges, either on the whole graph [88] or on an h-hop enclosed subgraph [99, 122]. However, this method cannot differentiate the importance of edges. Ideally, we want to obtain an augmented graph by keeping informative edges while removing irrelevant or noisy edges.

To build a more effective augmentation view of the user-item interaction graph for contrastive learning, we propose to replace the random dropout with graph diffusion, which reconciles the spatial message passing by smoothing out the neighborhood over the graph [124]. The diffusion process defines a weighted graph exchanged from the original unweighted graph. The weights measure the relative importance of edges based on the graph structure. Hence, we can utilize these importance scores to design different sparsification methods so as to preserve a more effective neighborhood for each node in the diffusion graph. Meanwhile, most GNNs incorporate high-order information by increasing the number of convolutional layers. The iterative expansion not only includes more nodes for learning better representations but also introduces more noisy edges, which can deteriorate the recommendation performance. Graph diffusion does not have this constraint as it can extend connections in the graph from one-hop to multi-hops. We retrieve the information of a larger neighborhood by one layer of aggregation rather than stacking multiple layers of GNNs. Hence, the problem of having noises in real graphs can be further mitigated.

In this paper, we propose a simple yet effective Graph Diffusion Contrastive Learning (GDCL) framework for item recommendation with users' implicit feedback. Existing graph diffusion algorithms focus on homogeneous graphs, which include a single node type. In GDCL, we first devise the diffusion algorithm to consider different types of nodes in a heterogeneous graph (*i.e.*, user-item interaction graph).

The derived diffusion graph consists of multiple types of relations between nodes. Specifically, besides the user-item relation, user-user, and item-item relations are also introduced. These heterogeneous relations [180] are not fully captured if we simply apply graph convolutional network (GCN) based encoders to treat them uniformly as previous works [31, 88]. Hence, we extend GCN to model the heterogeneity of the diffusion graph by maintaining a dedicated representation for every type of relation, and then fuse them using a mean aggregator. To train the overall model end-to-end, we leverage a multi-task training paradigm to jointly optimize the recommendation task and self-supervised learning task. For the recommendation task, previous SSL-based recommendation models [88, 99, 122, 123] only rely on the user-item graph for user preference prediction. Differing from these works, we utilize representations learned from the auxiliary view (*i.e.*, diffusion graph) together with the user-item graph to improve the representation learning for users and items. For the self-supervised task, we contrast node representations encoded from two views by a symmetric mutual information maximization objective function. Experimental results on four publicly available datasets demonstrate that the proposed GDCL model consistently outperforms state-of-the-art recommendation methods.

4.2 The Proposed Recommendation Model

This work focuses on top-K item recommendations based on users' implicit feedback. Let $\mathcal{U} = \{u_1, u_2, \cdots, u_m\}$ and $\mathcal{V} = \{v_1, v_2, \cdots, v_n\}$ be the set of users and items, where m and n denote the number of users and items, respectively. Observed users' interactions with items can be described by a bipartite graph $\mathcal{G} = \{\mathcal{U}, \mathcal{V}, \mathcal{E}\}$, where $\mathcal{E}$ denotes the set of edges that represents all interactions between users and items. If a user u has interacted with an item v, we build an edge e_{uv} between the corresponding user node and the item node. For each node t in $\mathcal{G}$, we denote the set of its first-hop neighbors in $\mathcal{G}$ by $\mathcal{N}_t$. We denote the adjacency matrix of $\mathcal{G}$ by $\mathbf{A} \in \mathbb{R}^{(m+n)\times(m+n)}$. In this work, we consider the user-item interaction graph as an undirected graph. Then, we set $A_{ij} = A_{ji} = 1$, if there exists an edge connecting two nodes t_i and t_j in $\mathcal{G}$; Otherwise, we set $A_{ij} = A_{ji} = 0$. The degree matrix $\mathbf{D} \in \mathbb{R}^{(m+n)\times(m+n)}$ of $\mathcal{G}$ is a diagonal matrix, where the diagonal element $D_{ii} = \sum_{j=1}^{m+n} A_{ij}$. Given the interaction graph $\mathcal{G}$ between users and items,

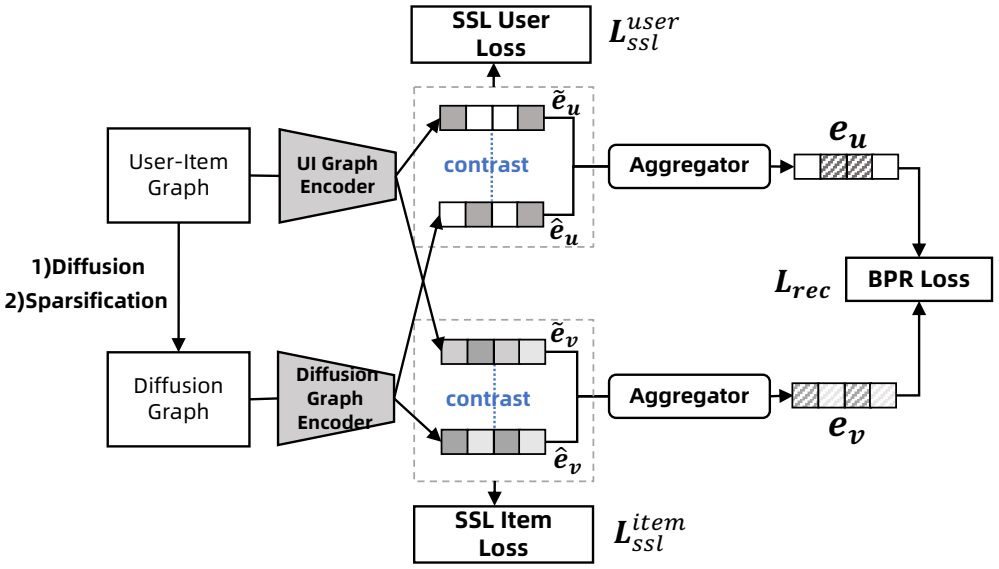


FIGURE 4.1: The overall framework of the proposed GDCL recommendation model.

our objective is to predict the probability that a user u would like to interact with a candidate item v, and then recommend K top-ranked candidate items to u. Figure 4.1 shows the overall framework of the proposed GDCL recommendation model. It consists of three main components: 1) diffusion-based graph augmentation, 2) graph encoders, and 3) self-supervised contrastive learning. Next, we introduce details of each component.

4.2.1 Diffusion-based Graph Augmentation

4.2.1.1 Graph Diffusion Approximation

For a homogeneous graph, its diffusion matrix can be formulated as [124, 181],

$$\mathbf{\Pi} = \sum_{k=0}^{\infty} \theta_k \mathbf{T}^k, \tag{4.1}$$

where $\mathbf{T}$ is the generalized transition matrix that can be defined by the symmetrically normalized adjacency matrix as $\mathbf{T} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{1/2}$, and θ_k is weighting coefficients of $\mathbf{T}^k$. In Eqn. (4.1), $\mathbf{\Pi}$ has closed-form solutions when considering two

special cases, *i.e.*, Personalized PageRank (PPR) [182] and heat kernel [183]. However, these two solutions involve matrix inverse and matrix exponential operations, which are computationally infeasible for processing large-scale graphs.

Inspired by [184], we resort to an efficient algorithm to approximate the diffusion matrix $\mathbf{\Pi}$. Specifically, we choose the following adaption of personalized PageRank [185] to instantiate $\mathbf{\Pi}$,

$$\mathbf{\Pi_{PPR}} = \alpha(\mathbf{I_n} - (1-\alpha)\mathbf{D}^{-1}\mathbf{A})^{-1}, \tag{4.2}$$

where $\mathbf{I_n}$ is the identity matrix, and α is the teleport probability in a random walk. A larger teleport probability means higher chances to return to the root node, and hence we can preserve more locality information. A smaller teleport probability allows us to reach out to a larger neighborhood. We can tune α to adjust the size of the neighborhood for different datasets. In [184], the push-flow algorithm [186] is used to obtain a sparse approximation for each row of $\mathbf{\Pi_{PPR}}$. Note that every row of this approximated matrix can be pre-computed in parallel using a distributed batch data processing pipeline. However, this algorithm is designed for homogeneous graphs, where all nodes are of the same type. We adapt it for the user-item interaction graph $\mathcal{G}$ which contains two different types of nodes. Precisely, we define teleport probabilities α_u and α_v for user nodes and item nodes, respectively. For the i-th row $\boldsymbol{\pi}_i$ in the approximated graph diffusion matrix $\mathbf{\Pi}_{PPR}$, we first determine the node type and then use its corresponding teleport probability for computation. As a result, we can control the amount of information being diffused to the neighborhood for different types of nodes. It has practical benefits for a recommender system. For example, real-world datasets usually have more long-tail items than long-tail users. In such cases, decreasing the teleport probability of items can help find more possibly relevant users and items, and hence learn better representations. Details of the diffusion matrix approximation algorithm are summarized in Algorithm 1.

4.2.1.2 Diffusion Matrix Sparsification

The diffusion matrix $\mathbf{\Pi_{PPR}}$ is a dense matrix, where each element reflects the relevance between two nodes based on the graph structure. As suggested by [186], weights of personalized PageRank vectors are usually concentrated in a small subset

Algorithm 1 Approximate graph diffusion for the user-item interaction graph

Inputs: Graph $\mathcal{G}$, teleport probability α_u and α_v, target node $z \in \mathcal{U} \cup \mathcal{V}$, max.residual ϵ, node degree vector $\boldsymbol{d}$
Initialize the estimate-vector $\boldsymbol{\pi} = \mathbf{0}$ and the residual-vector $\mathbf{r}$. $\mathbf{r}$ is a zero vector with only position z equal to α_u if z is a user node or α_v if z is an item node.
while $\exists\ t\ s.t.\ (\boldsymbol{r_t} > \alpha_u \epsilon \boldsymbol{d_t}$ if $t \in \mathcal{U})$ or $(\boldsymbol{r_t} > \alpha_v \epsilon \boldsymbol{d_t}$ if $t \in \mathcal{V})$ **do**
    if $t \in \mathcal{U}$ **then**
        $\alpha = \alpha_u$
    else if $t \in \mathcal{V}$ **then**
        $\alpha = \alpha_v$
    end if
    # approximate personalized PageRank [184]
    $\boldsymbol{\pi_t}$ += $\boldsymbol{r_t}$
    $m = (1 - \alpha) \cdot r_t / d_t$
    for $u \in \mathcal{N}_\sqcup$ **do**
        r_u += m
    end for
    $\boldsymbol{r_t} = 0$
end while
Return $\boldsymbol{\pi}$

of nodes. Thus, we can truncate small weights but still obtain a good approximation. In this work, we propose the following three methods to sparsify the diffusion matrix,

- **Topk**: For each row $\boldsymbol{\pi}_i$ in $\boldsymbol{\Pi}_{PPR}$, we retain k entries with highest weights from user nodes and item nodes respectively, and set other entries to zero. Namely, we will keep $2k$ "neighbors" for each node in the diffusion graph.
- **Topk-rand**: We firstly select k user nodes and k item nodes with highest weights in each row $\boldsymbol{\pi}_i$, following the Topk method. Then, we randomly drop selected nodes with a dropout ratio ρ (an adjustable hyper-parameter).
- **Topk-prob**: This method is similar to the Topk-rand method. The only difference is that, in this method, the probability for dropping a selected node is proportional to its weight in the weight vector $\boldsymbol{\pi}_i$.

As the Topk sparsification method is deterministic, we train the GDCL model with the fixed sparsified diffusion matrix. The other two sparsification methods Topk-rand and Topk-prob are stochastic. When using the Topk-rand and Topk-prob

methods to train the proposed model, we perform diffusion matrix sparsification at each training epoch. We denote the sparsified diffusion graph by $\widetilde{\mathcal{G}}$.

4.2.2 Graph Encoders

The original user-item interaction graph $\mathcal{G}$ and the diffusion graph $\widetilde{\mathcal{G}}$ are treated as two congruent views for contrastive learning. Two different graph encoders are designed to capture the information in $\mathcal{G}$ and $\widetilde{\mathcal{G}}$. To begin with, we randomly initialize embeddings of a user u and an item v by $\mathbf{e}_u^{(0)}$ and $\mathbf{e}_v^{(0)}$ respectively, which are shared by both graph encoders.

4.2.2.1 User-Item Interaction Graph Encoder

In this work, we use LightGCN [31] to encode the user-item interaction graph $\mathcal{G}$. As shown in [31], LightGCN only keeps the neighborhood aggregation of GCN when propagating node embeddings. At the ℓ-th layer, the graph convolution operation on a user node u is defined as,

$$\widetilde{\mathbf{e}}_u^{(\ell)} = \sum_{v \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_v|}} \widetilde{\mathbf{e}}_v^{(\ell-1)}, \tag{4.3}$$

where $\widetilde{\mathbf{e}}_u^{(\ell)}$ denotes the representation of the user u at the ℓ-th layer, and $|\cdot|$ denotes the cardinality of a set. Then, we sum multiple representations from different layers to obtain the final user embedding derived from the user-item interaction graph as follows,

$$\widetilde{\mathbf{e}}_u = \mathbf{e}_u^{(0)} + \widetilde{\mathbf{e}}_u^{(1)} + \cdots + \widetilde{\mathbf{e}}_u^{(L)}, \tag{4.4}$$

where L denotes the number of GCN layers. Similarly, we can obtain the final representation $\widetilde{\mathbf{e}}_v$ of an item v based on the interaction graph $\mathcal{G}$.

4.2.2.2 Diffusion Graph Encoder

The diffusion graph $\widetilde{\mathcal{G}}$ built in Section 4.2.1 is derived from the user-item interaction graph, which includes two types of nodes. Thus, three types of relations are established in $\widetilde{\mathcal{G}}$, including user-item relation, user-user relation, and item-item relation. To effectively capture the heterogeneous structure of $\widetilde{\mathcal{G}}$, we propose a

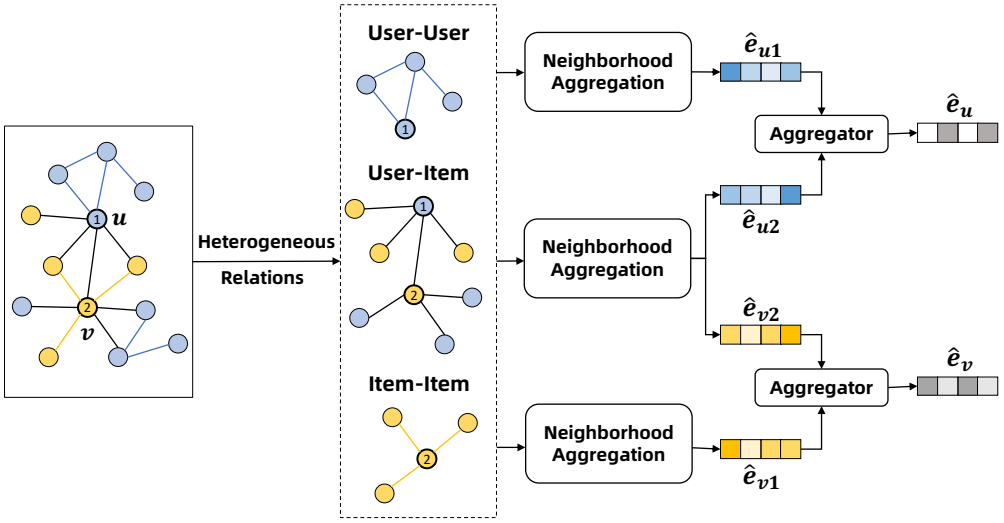


FIGURE 4.2: The illustration of diffusion graph encoder. Blue and yellow colors denote user and item nodes. Edges in black, blue, and yellow denote user-item, user-user, and item-item relations.

graph diffusion encoder to model these three relations separately and maintain a dedicated representation for every type of relation. As shown in Figure 4.2, each node's adjacent nodes can either be users or items in $\widetilde{\mathcal{G}}$. We segregate them into two groups based on the node type and perform feature aggregation within each group. Hence, two representations are generated for every user (or item). One is derived from the user-item relation and the other one is derived from the user-user (or item-item) relation.

Traditional message-passing neural networks, *e.g.*, GCN, aggregate their first-hop neighbors at each layer. Higher-order neighbors are only accessible through layer-to-layer propagation. The graph diffusion process breaks this constraint by creating connections to multi-hop nodes, and hence aggregation can be performed on a larger neighborhood without stacking multiple GNN layers [124]. Specifically, the derivation of user embeddings is as follows,

$$\widehat{\mathbf{e}}_{u1} = \sum_{u' \in \mathcal{N}_{\sqcap}{}^{(1)}} \pi_u(u') \mathbf{e}_{u'}^{(0)}, \quad \widehat{\mathbf{e}}_{u2} = \sum_{v \in \mathcal{N}_{\sqcap}{}^{(2)}} \pi_u(v) \mathbf{e}_{v}^{(0)}, \tag{4.5}$$

where $\widehat{\mathbf{e}}_{u1}$ and $\widehat{\mathbf{e}}_{u2}$ are embeddings of u obtained from the user-user diffusion graph and user-item diffusion graph, respectively. $\mathcal{N}_u^{(1)}$ and $\mathcal{N}_u^{(2)}$ denote sets of first-hop nodes of the user u on the user-user diffusion graph and user-item diffusion graph, respectively. $\boldsymbol{\pi}_u$ represents the diffusion vector of u, and $\pi_u(v)$ is the weight of the node v in the diffusion vector $\boldsymbol{\pi}_u$. Then, we combine these two embeddings by the

MEAN operation,

$$\widehat{\mathbf{e}}_u = \text{MEAN}\big(\widehat{\mathbf{e}}_{u1}, \widehat{\mathbf{e}}_{u2}\big), \tag{4.6}$$

which takes the average of user embeddings from the user-user diffusion graph and the user-item diffusion graph to be the final user representation $\widehat{\mathbf{e}}_u$ learned from the diffusion graph. Similarly, we can obtain the representation $\widehat{\mathbf{e}}_v$ of an item v learned from the diffusion graph.

4.2.3 Self-supervised Contrastive Leaning

After obtaining user and item representations from two graph views, we maximize the consistency between positive pairs and the inconsistency between negative pairs via contrastive learning. A positive pair is formed by representations of the same node from different views, whereas representations of different nodes form negative pairs. The GNNs model is enforced to distinguish positive pairs from negative pairs.

Following [187], we use the InfoNCE_Sym mutual information estimator to define contrastive learning losses in this work. Specifically, negative pairs are categorized into inter-view negative pairs and intra-view negative pairs. An inter-view negative pair consists of two nodes, which are generated from different views. If both nodes of a negative pair are from the same view, they are called an intra-view negative pair. Let $\mathcal{B}$ denote the current mini-batch of observed user-item interaction pairs, $\mathcal{U}_\mathcal{B}$ and $\mathcal{V}_\mathcal{B}$ denote the list of users and items in $\mathcal{B}$, respectively. For a user $u \in \mathcal{U}_\mathcal{B}$, its contrastive loss is defined as,

$$\ell(\widetilde{\mathbf{e}}_u, \widehat{\mathbf{e}}_u) = \log\Big(\frac{e^{cos(\widetilde{\mathbf{e}}_u, \widehat{\mathbf{e}}_u)/\tau}}{e^{cos(\widetilde{\mathbf{e}}_u, \widehat{\mathbf{e}}_u)/\tau} + \sum\limits_{u' \neq u} e^{cos(\widetilde{\mathbf{e}}_u, \widehat{\mathbf{e}}_{u'})/\tau} + \sum\limits_{u' \neq u} e^{cos(\widetilde{\mathbf{e}}_u, \widetilde{\mathbf{e}}_{u'})/\tau}}\Big), \tag{4.7}$$

where $cos(\cdot)$ is the cosine similarity function and τ is the temperature hyper-parameter. Three terms in the denominator are the score of the positive pair, the total scores of all inter-view negative pairs, and the total scores of all intra-view negative pairs. Note that we only consider negative instances from the current mini-batch. As two graph views are interchangeable, we can define the contrastive

learning loss from the user perspective in a symmetric way as follows,

$$L_{ssl}^{user} = -\frac{1}{2|\mathcal{U}_{\mathcal{B}}|} \sum_{u \in \mathcal{U}_{\mathcal{B}}} \left[\ell(\widetilde{\mathbf{e}}_u, \widehat{\mathbf{e}}_u) + \ell(\widehat{\mathbf{e}}_u, \widetilde{\mathbf{e}}_u) \right]. \tag{4.8}$$

In this case, intra-view negative pairs derived from two graph views will be taken into account.

Similarly, we can define the contrastive learning loss from the item perspective as,

$$L_{ssl}^{item} = -\frac{1}{2|\mathcal{V}_{\mathcal{B}}|} \sum_{v \in \mathcal{V}_{\mathcal{B}}} \left[\ell(\widetilde{\mathbf{e}}_v, \widehat{\mathbf{e}}_v) + \ell(\widehat{\mathbf{e}}_v, \widetilde{\mathbf{e}}_v) \right]. \tag{4.9}$$

4.2.4 Multi-Task Training

To learn model parameters, we leverage a multi-task training paradigm to jointly optimize the recommendation task and self-supervised task. For the recommendation task, we first aggregate representations from the user-item graph encoder and the diffusion graph encoder by element-wise summation,

$$\mathbf{e}_u = \widetilde{\mathbf{e}}_u + \widehat{\mathbf{e}}_u, \quad \mathbf{e}_v = \widetilde{\mathbf{e}}_v + \widehat{\mathbf{e}}_v, \tag{4.10}$$

where $\mathbf{e}_u$ and $\mathbf{e}_v$ are final representations of user u and item v. Then, the prediction of u's preference on the item v can be defined as,

$$\hat{y}_{uv} = \mathbf{e}_u^\top \mathbf{e}_v. \tag{4.11}$$

The Bayesian Pairwise Ranking (BPR) loss [10] is employed as the loss function for the recommendation task. Specifically, for each user-item pair $(u, v) \in \mathcal{B}$, we randomly sample an item w that has no interaction with u to form a triplet (u, v, w). Then, the loss function is defined as follows,

$$L_{rec} = \sum_{(u,v,w) \in \mathcal{D}_{\mathcal{B}}} -\log \sigma \left(\hat{y}_{uv} - \hat{y}_{uw} \right), \tag{4.12}$$

where $\mathcal{D}_{\mathcal{B}}$ denotes the set of triplets for all observed user-item pairs in $\mathcal{B}$. The total loss for learning the proposed recommendation model is as follows,

$$L = L_{rec} + \lambda_1 (L_{ssl}^{user} + L_{ssl}^{item}) + \lambda_2 \|\mathbf{\Theta}\|_2^2, \tag{4.13}$$

where λ_1 balances the primary recommendation loss and auxiliary self-supervised loss, λ_2 controls the L2 regularization, and $\mathbf{\Theta}$ denotes model parameters.

4.2.5 Complexity Analysis

In this section, we delve into the analysis of the time complexity associated with GDCL training. GDCL comprises three principal components: 1) diffusion-based graph augmentation, 2) graph encoders, and 3) self-supervised contrastive learning. The diffusion graph can be precomputed and is thus excluded from our time complexity analysis. The graph encoders utilize neighborhood aggregation and introduce no new trainable parameters. The primary source of complexity arises from the third component, the self-supervised contrastive learning objective, coupled with the BPR loss employed for the recommendation task. The BPR loss exhibits a complexity of $\mathcal{O}(2\mathcal{E}d)$ for each epoch. To evaluate the self-supervised loss, our analysis considers only the inner product, as specified in Equation 4.7. During the calculation of InfoNCE loss on the user side, all other user nodes within a batch are treated as negative samples. The complexities for the numerator and denominator are $\mathcal{O}(\mathcal{E}d)$ and $\mathcal{O}(2B\mathcal{E}d)$, respectively. Given that the contrastive loss is symmetric, the total complexity for the user side per epoch is $\mathcal{O}(\mathcal{E}d(1+3B))$. Since this loss is computed for both the user and item sides, the overall complexity doubles, though the factor of 2 is generally omitted for simplification. Consequently, the cumulative time complexity, incorporating both BPR and contrastive loss, is $\mathcal{O}(\mathcal{E}d(3+3B))$.

4.3 Experiments

4.3.1 Experimental Settings

4.3.1.1 Experimental Datasets

To examine the capability of the proposed model, we conduct experiments on three datasets: Amazon review [188], MovieLens-1M[2], and Yelp2018[3]. For the Amazon

[2]https://grouplens.org/datasets/movielens/1m/
[3]https://www.yelp.com/dataset

TABLE 4.1: Statistics of the experimental datasets.

Dataset	# Users	# Items	# Interactions	Sparsity
Amazon-Games	45,950	16,171	363,590	99.95%
Amazon-Arts	42,137	20,942	317,109	99.96%
MovieLens-1M	5,400	3,662	904,616	95.43%
Yelp2018	34,518	22,918	380,632	99.95%

review, we choose the subsets "Video Games" and "Arts" for evaluation. On each dataset, we only keep users and items that have at least 5 interactions. The statistics of datasets are shown in Table 4.1.

4.3.1.2 Baseline Methods

We compare the GDCL model with the following baselines:

- **BPR** [10]: This is a classical collaborative filtering method based on matrix factorization;
- **LightGCN** [31]: This is the state-of-the-art GNNs-based recommendation model. It simplifies the design of GCNs for collaborative filtering by discarding feature transformation and non-linear activation;
- **BUIR** [189]: This work proposes a recommendation framework that does not require negative sampling. It utilizes two distinct encoder networks to learn from each other;
- **BiGI** [99]: This work generates the graph-level representation and contrasts it with sampled edges' representations via a global-local infoMax objective. It incorporates global properties of a bipartite graph into the representation learning of graph nodes;
- **SGL** [88]: This work exploits self-supervised learning on the user-item graph. They devise three types of data augmentation operations on graph structure from different aspects to construct auxiliary contrastive tasks.

4.3.1.3 Evaluation Protocols

For each dataset, we sort observed user-item interactions in chronological order based on interaction timestamps. Then, the first 80% of interactions are chosen for training. The next 10% and the last 10% of interactions are used for validation and testing. The performance of a recommendation model is measured by three widely used ranking-based metrics: Recall@K, NDCG@K, and Hit Ratio@K (denoted by R@K, N@K, and HR@K), where K is set to 5, 10, and 20. For each metric, we compute the performance of each user in the testing data and report the average performance of all users.

4.3.1.4 Implementation Details

We implement GDCL by PyTorch [190]. Model parameters are initialized by the Xavier method [191] and learned by the Adam optimizer [192]. We continue to train the model until the performance is not increased for 50 consecutive epochs. All baselines are trained from scratch for fair comparison. The embedding dimension is fixed to 64, the batch size is set to 2048, and the learning rate is set to 0.001. The number of GCN layers is chosen from {1, 2, 3}. Empirically, we set teleport probability $\alpha_u = \alpha_v = 0.2$ and max.residual $\epsilon = 0.001$ for all datasets except the MovieLens-1M dataset. On MovieLens-1M dataset, α_u and α_v are set to 0.1 and ϵ is set to 0.0001. We tune the temperature τ in contrastive learning in {0.1, 0.2}. The SSL regularization λ_1 is chosen from {0.1, 0.01, 0.001, 0.0001}, and the weight decay λ_2 is chosen from {1e-2, 1e-3, 1e-4, 1e-5}. The dropout ratio ρ in diffusion matrix sparsification is set to 0.1. For each method, we use grid-search to choose optimal hyper-parameters based on its performance on validation data.

4.3.2 Performance Comparison

Table 4.2 summarizes the performance of different models. We have the following observations:

Firstly, on MovieLens-1M, graph-based baseline models are inferior to BPR for most evaluation metrics. Compared with other datasets, MovieLens-1M has a higher density. The training data has enough supervision signals from interactions;

thus simple BPR is adequate. On such datasets, GDCL can still achieve the best performance. It can be attributed to the sparsifed diffusion graph which helps to learn informative representations.

Secondly, BiGI performs worst on most datasets, which implies that contrasting the local sampled subgraph representations with global graph representations does not benefit much on the recommendation performance.

Thirdly, in most cases, the non-negative sampling method BUIR achieves better results than LightGCN. However, it cannot compete with models (*e.g.*, SGL and GDCL) that use a joint learning framework with self-supervised contrastive loss.

Fourthly, GDCL and SGL outperform purely graph-based models by leveraging self-supervised learning that contrasts node-level representations from different graph views. This result demonstrates the effectiveness of incorporating self-supervised learning into the recommendation task.

Lastly, the proposed GDCL model consistently achieves the best performance on all datasets. Compared with the best baseline method SGL, GDCL contrasts representations learned from the user-item graph with those learned from the diffusion graph. Thus, more useful graph structure information can be preserved, and negative impacts of noisy edges can be reduced. Moreover, GDCL employs representations learned from both graph encoders to enhance user and item representations for better recommendation performance.

4.3.3 Ablation Study

To study the impact of each component in GDCL, we consider the following variants of GDCL:

- **GDCL$_{\mathbf{Adj}}$**: we replace the diffusion matrix with the adjacency matrix to evaluate the effectiveness of graph diffusion;
- **GDCL$_{\mathbf{GCN}}$**: we replace the diffusion graph encoder with one layer of LightGCN to evaluate the effectiveness of modeling heterogeneous relations in graph diffusion. It performs neighborhood aggregation without differentiating various relations between nodes;

TABLE 4.2: Overall performance comparison. Best results are in **boldface** and the second best is underlined. "%Improv" refers to the relative improvement of GDCL over the best baseline. * indicates the improvements are statistically significant with $p < 0.05$.

Dataset	Model	R@5	N@5	HR@5	R@10	N@10	HR@10	R@20	N@20	HR@20
Amazon Games	BPR	0.0111	0.0096	0.0256	0.0208	0.0127	0.0440	0.0336	0.0165	0.0672
	LightGCN	0.0157	0.0134	0.0335	0.0269	0.0173	0.0545	0.0429	0.0220	0.0823
	BUIR	0.0148	0.0125	0.0320	0.0265	0.0166	0.0538	0.0456	0.0224	0.0888
	BiGI	0.0134	0.0106	0.027	0.0228	0.0139	0.0462	0.0383	0.0185	0.0728
	SGL	0.0157	0.0130	0.0331	0.0269	0.0170	0.0552	0.0451	0.0225	0.0860
	GDCL	**0.0174***	**0.0137***	**0.0367***	**0.0307***	**0.0184***	**0.0604**	**0.0469***	**0.0234***	**0.0907***
	%Improv	10.82%	2.23%	9.55%	14.12%	6.35%	9.42%	2.85%	4.00%	2.13%
Amazon Arts	BPR	0.0111	0.0092	0.0237	0.0175	0.0115	0.0363	0.0292	0.0151	0.0575
	LightGCN	0.0139	0.0114	0.0288	0.0234	0.0149	0.0463	0.0374	0.0193	0.0733
	BUIR	0.0129	0.0113	0.0268	0.0233	0.0150	0.046	0.0375	0.0195	0.0726
	BiGI	0.0098	0.0083	0.0204	0.0177	0.0113	0.0360	0.0287	0.0148	0.0570
	SGL	0.0141	0.0116	0.0281	0.0241	0.0153	0.0473	0.0381	0.0197	0.0730
	GDCL	**0.0144***	**0.0117***	**0.0291***	**0.0251***	**0.0157***	**0.0499***	**0.0391***	**0.0201***	**0.0765***
	%Improv	2.12%	0.86%	1.04%	4.14%	2.61%	5.49%	2.62%	2.03%	4.36%
MovieLens 1M	BPR	0.0299	**0.2900**	0.6012	0.0512	0.2691	0.6974	0.0859	0.2505	0.7837
	LightGCN	0.0294	0.2820	0.6012	0.0480	0.2621	0.6815	0.0836	0.2442	0.7619
	BUIR	0.0237	0.2673	0.5665	0.0387	0.2483	0.6587	0.0696	0.2322	0.7391
	BiGI	0.0274	0.2532	0.5774	0.0473	0.2441	0.6885	0.0797	0.2310	0.7837
	SGL	0.0286	0.2814	0.5972	0.0512	0.2628	0.6984	0.0848	0.2482	0.7728
	GDCL	**0.0320***	0.2877	**0.6121***	**0.0533***	**0.2696**	**0.7202***	**0.0872***	**0.2540**	**0.7917***
	%Improv	7.02%	-	1.81%	4.10%	0.18%	3.12%	1.51%	1.39%	1.02%
Yelp2018	BPR	0.0161	0.0136	0.0393	0.0290	0.0182	0.0689	0.0499	0.0248	0.1121
	LightGCN	0.0193	0.0159	0.0465	0.0328	0.0208	0.0771	0.0543	0.0275	0.1202
	BUIR	0.0196	0.0167	0.0473	0.0338	0.0217	0.0783	0.0572	0.0291	0.1271
	BiGI	0.0143	0.0113	0.0335	0.0237	0.0147	0.0556	0.0415	0.0204	0.0960
	SGL	0.0191	0.0164	0.0465	0.0346	0.0220	0.0801	**0.0608**	0.0301	0.1323
	GDCL	**0.0213***	**0.0184***	**0.0510***	**0.0356***	**0.0235***	**0.0840***	0.0606	**0.0313***	**0.1340**
	%Improv	8.67%	10.17%	7.82%	2.89%	6.81%	4.86%	-	3.98%	1.28%

- **$\text{GDCL}_{\text{w/o Diff}}$**: we predict the user preferences only using the adjacency matrix to evaluate the effectiveness of incorporating graph diffusion embeddings during inference.

The performance achieved by different GDCL variants is summarized in Table 4.3. We find that the combination of all components consistently improves the model performance on all datasets. Three GDCL variants perform differently on different datasets. Graph diffusion benefits more on Amazon-Arts and MovieLens-1M, as GDCL_{GCN} and $\text{GDCL}_{\text{w/o Diff}}$ perform better than GDCL_{Adj}. On Amazon-Games, simply adopting graph diffusion for contrastive learning (GDCL_{GCN}) is worse than using the adjacency matrix (GDCL_{Adj}). With the designed diffusion graph encoder and its generated embeddings for recommendation prediction, the performance is improved significantly.

TABLE 4.3: Performance achieved by different variants of GDCL.

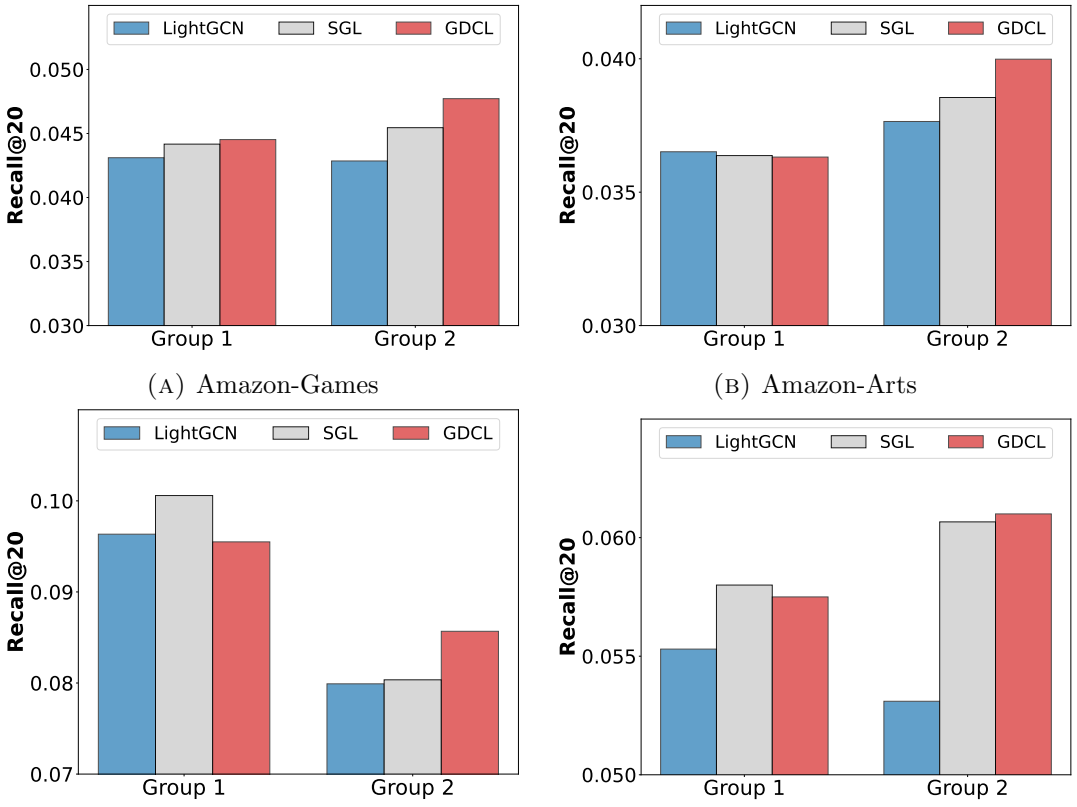
Method	Amazon-Games		Amazon-Arts		MovieLens-1M	
	R@10	N@10	R@10	N@10	R@10	N@10
$GDCL_{Adj}$	0.0293	**0.0188**	0.0237	0.0151	0.0514	0.2669
$GDCL_{GCN}$	0.0284	0.0176	0.0242	0.0154	0.0513	0.2653
$GDCL_{w/o\ Diff}$	0.0285	0.0181	0.0245	0.0154	0.0519	**0.2708**
GDCL	**0.0307**	0.0184	**0.0251**	**0.0157**	**0.0533**	0.2696

TABLE 4.4: Impacts of different sparsification methods.

Method	Amazon-Games		Amazon-Arts		MovieLens-1M	
	R@10	N@10	R@10	N@10	R@10	N@10
Topk	0.0293	0.0181	0.0242	0.0153	0.0480	0.2652
Topk-rand	**0.0307**	**0.0184**	0.0241	0.0151	**0.0533**	**0.2696**
Topk-prob	0.0294	0.0178	**0.0251**	**0.0157**	0.0485	0.2656

We also conduct experiments to investigate the impacts of different matrix sparsification methods. From Table 4.4, Topk-rand and Topk-prob settings outperform Topk. We conjecture that introducing randomness during training enhances the generalization capability of the model. Topk-rand works better on Amazon-Games and MovieLens-1M, while Topk-prob is better on Amazon-Arts.

To study the model performance on different users' popularity groups, we split testing users into two groups. Group 1 has 25% of users who have the least number of interactions, while Group 2 has the remaining 75% of users with more interactions. We compare three models, including LightGCN, SGL, and GDCL. As shown in Figure 4.3, GDCL achieves more significant improvement for users with more interaction data. We speculate that utilizing graph diffusion with sparsification can help alleviate the problem of noisy edges. For the MovieLens 1M dataset, the performance for users who engage more frequently is inferior to that of users who engage less frequently. Given the high density of this dataset, it is probable that users with many interactions have utilized the service for an extended duration. Over such periods, their preferences may shift or evolve, and the model may struggle to accurately capture these changing user preferences.



(A) Amazon-Games (B) Amazon-Arts

(C) MovieLens 1M (D) Yelp2018

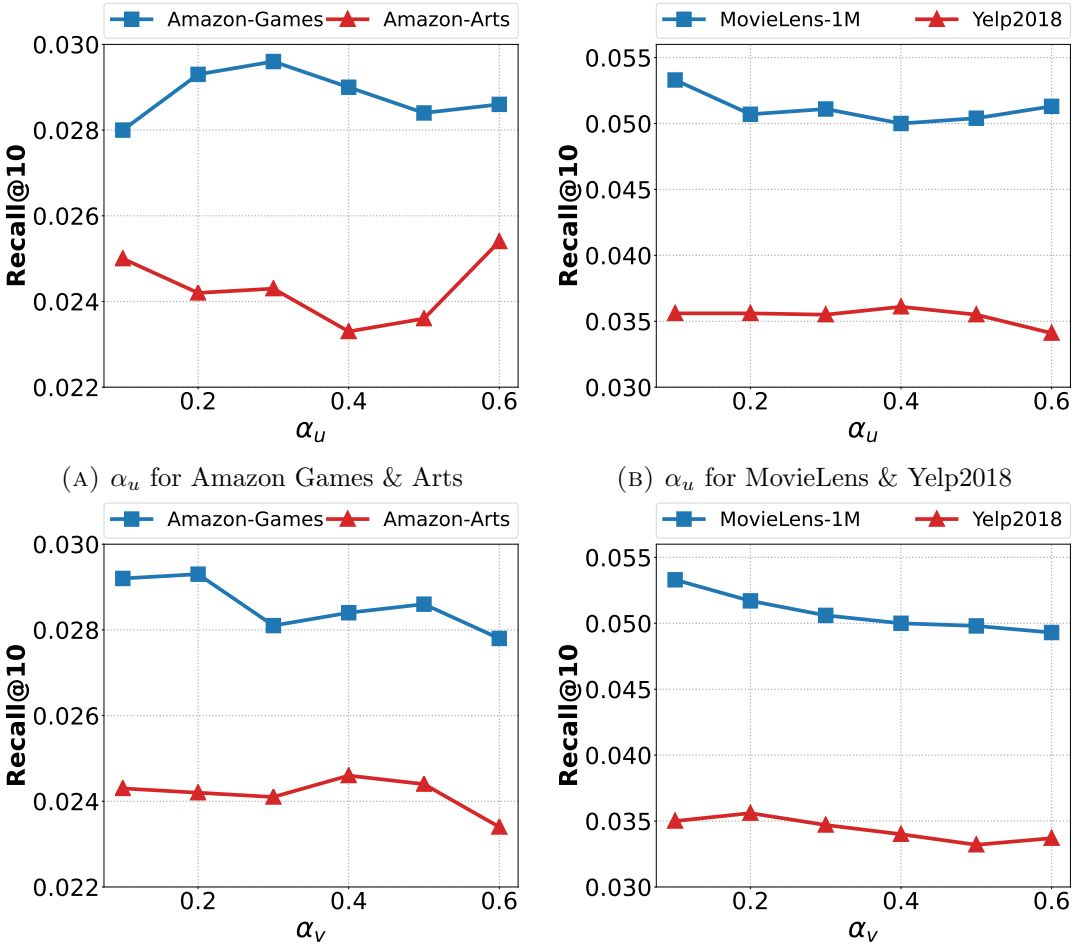
FIGURE 4.3: Performance comparison over different user groups.

TABLE 4.5: Performance of SGL and GDCL with different numbers of GCN layers

Method	Amazon-Games		Amazon-Arts		Yelp2018	
	R@10	N@10	R@10	N@10	R@10	N@10
SGL (L=1)	0.0262	0.0170	0.0228	0.0146	0.0327	0.0207
GDCL (L=1)	**0.0285**	**0.0180**	**0.0232**	**0.0148**	**0.0343**	**0.0224**
SGL (L=2)	0.0269	0.0170	**0.0242**	0.0150	0.0338	0.0216
GDCL (L=2)	**0.0290**	**0.0180**	0.0241	**0.0153**	**0.0356**	**0.0235**
SGL (L=3)	0.0252	0.0162	0.0241	0.0153	**0.0346**	0.0220
GDCL (L=3)	**0.0307**	**0.0184**	**0.0251**	**0.0157**	0.0345	**0.0225**

4.3.4 Hyper-parameter Study

To study the impacts of the GCN depth of the user-item interaction graph encoder, we vary the number of layers L in $\{1, 2, 3\}$. As shown in Table 4.5, GDCL outperforms SGL on a majority of metrics. Furthermore, GDCL with one GCN layer achieves comparable performance with SGL with three GCN layers.



(A) α_u for Amazon Games & Arts

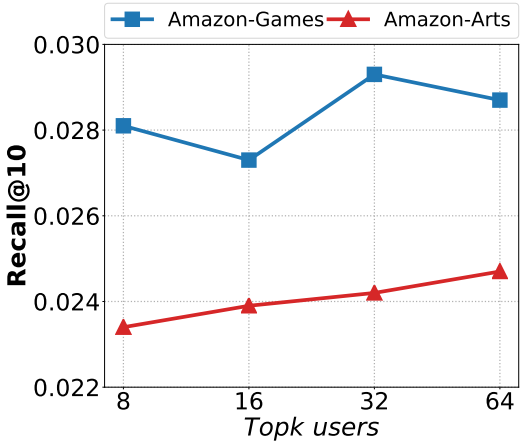
(B) α_u for MovieLens & Yelp2018

(C) α_v for Amazon Games & Arts

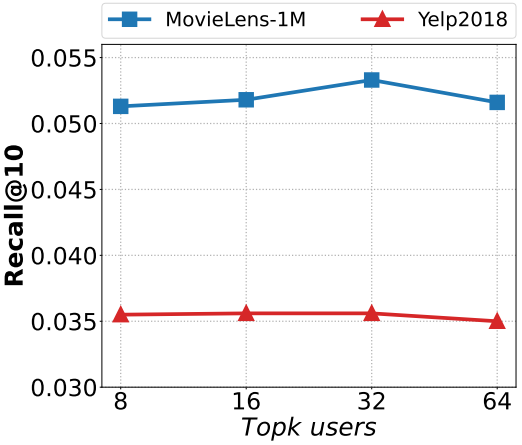
(D) α_v for MovieLens & Yelp2018

FIGURE 4.4: Performance trends of GDCL w.r.t α_u and α_v.

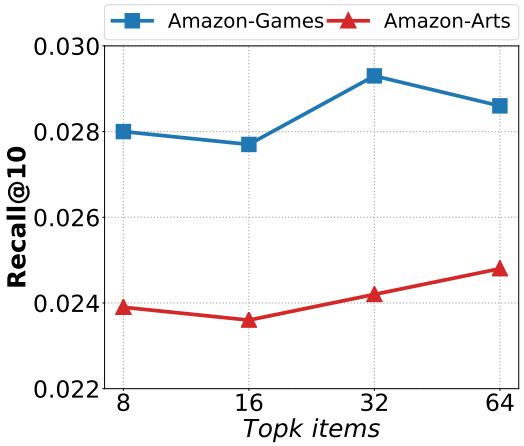
The teleport probability α is introduced to decide how much nearest neighborhood information is kept. If α is closer to 1, it is more likely to return to the starting node thus more information on the first hop nodes is retained. Otherwise, if α is closer to 0, more weights will be diffused to multi-hop nodes. We define teleport probabilities α_u and α_v for user and item nodes. Figure 4.4 shows the model performance versus different settings of α_u and α_v in $\{0.1, 0.2, ..., 0.6\}$. The plots reveal that recommendation performance is sensitive to α. For Amazon-Games and MovieLens 1M, taking smaller $\alpha = 0.1/0.2$ for both users and items can achieve higher accuracy. The performance drops with larger α. For Amazon-Arts, the performance is the best with $\alpha_u = 0.6$. As for α_v, the performance remains relatively steady when it is smaller than 0.6. When α_v is increased to 0.6, the performance declines. For Yelp2018, optimal performance is achieved at $\alpha_u = 0.4$ and $\alpha_v = 0.2$. Performance diminishes with an increase in α.



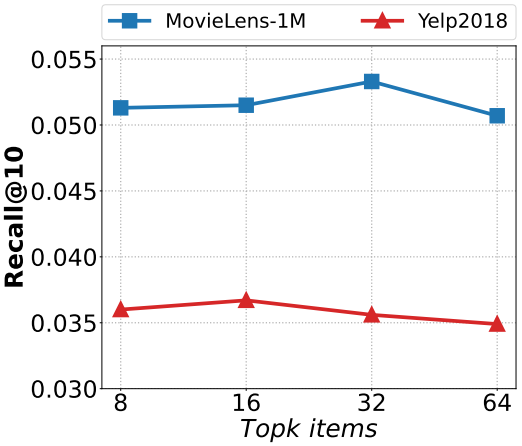
(A) *Topk* users for Amazon Games & Arts



(B) *Topk* users for MovieLens & Yelp2018



(C) *Topk* items for Amazon Games & Arts



(D) *Topk* items for MovieLens & Yelp2018

FIGURE 4.5: Performance trends of GDCL w.r.t topk users/items.

We also study how the number of selected top user and item nodes in diffusion graph sparsification affects the model performance. We fix *Topk* of items to 32 and vary *Topk* of users in {8, 16, 32, 64}, and vice versa. The results are shown in Figure 4.5. For Amazon-Games, the most effective neighborhood size is 32. The performance will decline when the size is too large or too small. For Amazon-Arts, including a larger number of users and items will gain better performance.

4.4 Summary

In this chapter, we introduce a self-supervised auxiliary task for learning user and item embeddings by contrasting two representations generated from graph structural views, including the user-item interaction graph and the sparsified diffusion graph. We propose a simple and effective model architecture to learn node embeddings from the diffusion graph by identifying its various heterogeneous relations. Extensive experiments on four datasets demonstrate that our model has achieved competitive performance compared with state-of-the-art baselines.

Chapter 5

Dual-view Whitening on Pre-trained Text Embeddings for Sequential Recommendation[1]

In Chapter 4, we propose a graph diffusion-based data augmentation method based on the interactions between users and items to minimize the introduction of bias or noises. In Chapter 5, we incorporate the side information by designing a systematic data augmentation method based on item features, thus leveraging additional contextual information to refine recommendations.

5.1 Overview

The sequential recommendation is a subfield of recommendation systems that aims to provide personalized item recommendations to users over time. It considers the order in which items are consumed by users to predict the next item the user is likely to interact with [49–51, 53, 56, 57, 195]. Recently, there has been an upsurge of interest in developing sequential recommendation methods that integrate textual information about items, such as product attributes [51, 54, 56], descriptions [57, 59, 80], and reviews [79, 98], with ID embeddings to generate more accurate and relevant recommendations. These recommendation frameworks

[1]The works in this chapter have been published in AAAI 2024 [193] and ICDE 2024 [194].

usually align text features with ID embeddings, highlighting the significance of ID embeddings in recommendations. However, there is a conspicuous absence of research exploring sequential recommendation based solely on text features. In this paper, we refer to recommendation methods that only use item text features as *text-based recommendation models*. We argue that studying text-based recommendation models, which do not necessitate ID embeddings, offers three primary advantages. Firstly, these models can greatly improve the performance in cold-start scenarios. E-commerce platforms introduce thousands of new products daily, and conventional sequential models typically integrate random ID embeddings with pre-trained text embeddings to provide recommendations for these new products. The integration of random initialized ID embeddings for these new items may inevitably have a detrimental effect on the performance of recommender systems. However, such integration may result in unwanted noise in recommendations. Secondly, text-based recommendation models can be more efficient than those that require ID embeddings. Using only text embeddings simplifies demands for both tensor storage and computational resources, as there is no requirement to maintain a large and frequently updated ID embedding matrix. Lastly, text embeddings are transferrable across platforms, whereas ID embeddings are not since user IDs and item IDs are typically not shared in practice.

However, effectively implementing sequential recommendations with only pre-trained text embeddings is non-trivial. Most existing sequential recommendation models [57, 58, 125] directly utilize text embeddings extracted by pre-trained language models (*e.g.*, BERT [196]). To identify potential issues with these pre-trained text embeddings, we first examine their cosine similarity on three recommendation datasets. Our analysis reveals that the pre-trained text embeddings exhibit a notably high average cosine similarity of approximately 0.8, indicating that their embedding spaces are highly anisotropic. We then conduct a quantitative analysis to assess the impact of embedding anisotropy on recommendation performance by comparing the performance of an ID-based method and a text-based method adapted from a widely used framework SASRec [49]. Our results show that the text-based method often yields sub-optimal results compared to the ID-based method. Although the text-based method learns from additional content information, text embeddings appear to be less expressive than standard item ID embeddings and are insufficient to achieve optimal recommendation performance on their own.

To resolve the problem of anisotropy in pre-trained text embeddings, we propose to employ a pre-processing step known as whitening transformation [197], which transforms the pre-trained text embedding distribution into a smooth and isotropic Gaussian distribution and removes the correlation among axes. We name the sequential recommendation model with whitening transformation as SASRec^W. Since the primary learning objective for recommendation is to optimize the alignment and uniformity between item representations and sequence representations [174, 198], the improved uniformity of sequence representations resulting from whitening transformation leads to enhanced recommendation performance. Notably, SASRec^W significantly improves the performance of the sequential recommendation models while using only text features, outperforming the models using ID embeddings, text embeddings, or both embeddings without whitening. SASRec^W leverages fully whitened representations, where all dimensions are decorrelated and embeddings are uniformly projected into a spherical distribution. Although whitening is effective in recommendation, excessive whitening may have a negative impact on the manifold of items that share similar textual semantics. Therefore, we can also relax the whitening criteria where partial dimensions are decorrleated and the obtained representations tend to preserve more original text semantics at the expense of embedding uniformity [199]. Although the retention of text semantics may appear advantageous for the recommendation task, our experimental results suggest that full whitening leads to the best performance compared to different degrees of relaxed whitening.

To reap the benefits of full whitening while preserving partial semantics in original text features, we propose an ensemble framework WhitenRec+, which combines both fully whitened representations and relaxed whitened representations together to enhance item representation learning for the sequential recommendation. Specifically, fully whitened representations are produced by whitening the pre-trained text embeddings with the most stringent whitening to decorrelate across all dimensions. Relaxed whitened representations are produced with less stringent whitening to decorrelate dimensions within each group of dimensions, *i.e.*, correlation among groups is kept. The fully whitened item representations and relaxed whitened item representations are subsequently combined by passing them through a shared projection head and summing their outputs. The obtained representations are then processed by the Transformer for sequential recommendation.

To further exploit the advantages of both fully and relaxed whitened representations, we have refined the model architecture of WhitenRec+ and introduced a novel Dual-view Whitening method on pre-trained text embeddings for Sequential Recommendation, named DWSRec. Initially, we employ a dual-view item encoder with a shared projection head, deriving both fully and relaxed whitened representations from pre-trained text features. Fully whitened representations are obtained by ensuring decorrelation across all dimensions. Conversely, relaxed whitened representations aim for decorrelation only within specific dimensional groups. Then, we utilize a decoupled attention-based dual-view transformer to encode sequences composed of fully and relaxed whitening representations. Namely, we generate two sets of key-query pairs in the attention layer. For learning the sequence embedding based on full whitening, it serves as the value and the initial key-query pair, while the relaxed whitening acts as the second key-query pair. Likewise, when learning the sequence embedding with relaxed whitening, it is used as the value and the primary key-query pair, with the full whitening serving as the second pair. Different extents of whitened representations are employed collectively and interchangeably to enhance the attention calculation of the transformer. Lastly, we leverage a dual-view fusion module to adaptively merge these view-specific sequence embeddings as well as item embeddings using two separate weighted attention layers for recommendation.

In summary, our contributions are the following:

- We streamline the existing sequential recommendation framework by studying models that only utilize item text features without the need for ID embeddings. Our empirical analysis reveals that anisotropy in pre-trained text embeddings restricts the performance of text-based sequential recommendation models. To resolve this issue, we employ whitening transformation to transform pre-trained text embedding distribution into an isotropic form, which can significantly improve the performance of text-based sequential recommendation models.

- Our empirical analysis of the whitening process reveals that it may hurt the manifold of items exhibiting similar textual semantics. To this end, we propose WhitenRec+ and DWSRec, which leverage different degrees of whitening transformations to reap the benefits of full whitening while preserving some of the inherent semantics in the original text features. We conduct a

thorough analysis and discussion of the merits of DWSRec in terms of representation uniformity and alignment, matrix conditioning, and information reconstruction.

- Extensive experiments are conducted on three benchmark datasets to evaluate the performance of the proposed models for the sequential recommendation. Notably, WhitenRec+ and DWSRec outperform state-of-the-art models across all metrics for all three datasets.

5.2 Related Work of Whitening

The whitening, or decorrelation, is a data transformation process with the theoretical guarantee of avoiding collapse by decorrelating each feature dimension [197]. One of the earliest approaches to whitening is Principal Component Analysis (PCA). [200] first introduces PCA for data analysis and dimensionality reduction, and it has been adapted for use in deep learning [201]. Compared with PCA, Zero-phase Component Analysis (ZCA) [202] whitening introduces an additional rotation back to the original coordinate system. Cholesky Decomposition (CD) [203] whitening proposed by [204] decomposes the covariance matrix into a lower triangular matrix and its conjugate transpose. Recently, UniSRec [57] adopts a parametric whitening (PW) method which incorporates a linear layer in the whitening transformation for better generalizability.

In the field of deep learning, prior research efforts [205–207] explore the application of whitening techniques to the activation of intermediate layers in neural networks. Batch Normalization (BN) [205] is the first to perform normalization per mini-batch, thereby enabling back-propagation and reducing the internal covariate shift during training. Decorrelated Batch Normalization (DBN) [206] builds upon BN by incorporating ZCA whitening over mini-batch data to further remove correlation among dimensions. Lately, another research direction [199, 208, 209] has emerged, focusing on employing whitening for self-supervised learning, which seeks to avoid the collapse of augmented representations into a single point. Different from these studies, our work leverages different degrees of decorrelation strength during the whitening process of pre-trained text embeddings to enhance the representation learning for the sequential recommendation.

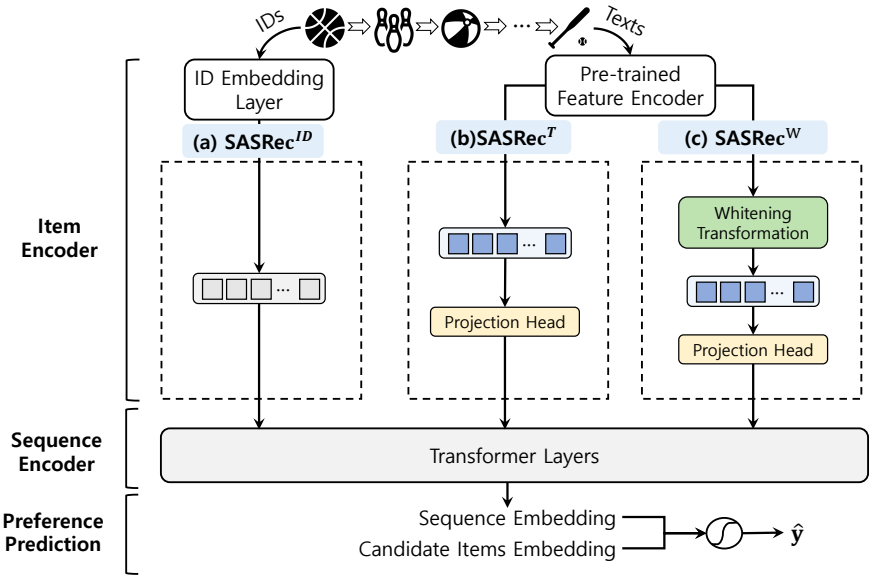


FIGURE 5.1: The illustration of presenting three variations of sequential recommendation methods, which are SASRec^{ID}, SASRec^{T}, and SASRec^{W}. Each method is composed of three components, including an item encoder, a sequence encoder, and the preference prediction layer.

5.3 Preliminaries

5.3.1 Task Formulation

In sequential recommendation, we denote a set of users as $\mathcal{U}$ with size $|\mathcal{U}|$ and a set of items as $\mathcal{I}$ with size $|\mathcal{I}|$. Each user in $\mathcal{U}$ is associated with a sequence of items $\mathcal{S} = \{i_1, \cdots, i_{|\mathcal{S}|}\}$ where $i_t \in \mathcal{I}$ denotes the item that the user has interacted with at the t-th timestamp. $|\mathcal{S}|$ is the sequence length. The objective is to consider the past item sequences of a user and predict the next item ($i_{|\mathcal{S}|+1}$) that the user is likely to adopt.

5.3.2 SASRec^{ID} & SASRec^{T} & SASRec^{W}

SASRec [49], by utilizing the self-attention mechanism, serves as a foundational model for state-of-the-art sequential recommendation methods [57, 174]. Hence, we examine three variants of SASRec [49] to illustrate the anisotropy issue inherent in pre-trained text embeddings: SASRec^{ID}, SASRec^{T}, and SASRec^{W}. Their primary distinction is in item embedding encoding before the transformer layers as presented

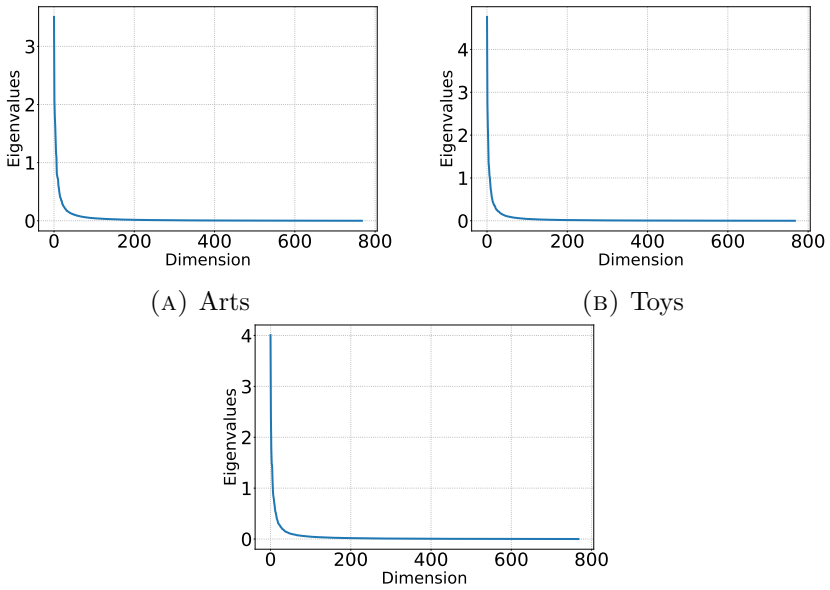
in Figure. 5.1. **SASRec**^{ID} is the base SASRec. It employs a randomly initialized ID embedding matrix, denoted as $\mathbf{E} \in \mathbb{R}^{d \times |\mathcal{I}|}$, to depict items. d indicates the embedding size. **SASRec**^{T} presumes that items possess textual information. The item embeddings are initialized using pre-trained text features $\mathbf{X} \in \mathbb{R}^{d_t \times |\mathcal{I}|}$, where d_t is the feature dimension. $\mathbf{X}$ is not updated throughout the training process. $\mathbf{X}$ is then passed to an MLP projector with two hidden layers and ReLU activations to reduce dimensionality. **SASRec**^{W} mirrors SASRec^{T} but pre-processes $\mathbf{X}$ with the whitening transformation on all items.

5.4 Methods and Main Results

In this section, we study the impact of anisotropic text embedding spaces on the performance of the text-based sequential recommendation model. We show that the anisotropy problem restricts the model's performance. To address this problem, we propose to apply the whitening transformation to eliminate the strong correlation between axes and make text embeddings more isotropic. By doing so, the recommendation performance can be significantly improved. Additionally, we introduce a simple yet effective extension that combines relaxed whitened representations with fully whitened representations, enhancing the item representation learning for the sequential recommendation.

5.4.1 Anisotropic Embedding Space Induces Poor Recommendation Performance

Recent research in the field of NLP has revealed that BERT sentence embeddings tend to degenerate into an anisotropic shape, which is referred to as the *representation degeneration problem* [126, 127, 210]. The embeddings are pushed into a similar direction that is negatively correlated with most hidden states, thus clustering in a narrow cone region of the embedding space. This phenomenon can result in high semantic similarities among embeddings and limit the effectiveness of sentence embeddings. Moreover, it has been demonstrated that this representation degeneration problem can adversely impact the performance of downstream language modeling tasks as well [126, 127]. Since BERT embeddings are commonly utilized by text-based recommendation models to extract text information



(A) Arts (B) Toys

(C) Tools

FIGURE 5.2: Normalized singular values of item text embeddings for Arts, Toys, and Tools datasets.

of items, we conduct a preliminary investigation to determine whether the representation degeneration problem also affects the performance of text-based sequential recommendation models.

We study three datasets from Amazon [188], including Arts, Toys, and Tools. Following [57, 59, 81], we first concatenate titles, categories, and brands of items as their text descriptions. Next, for each item, a special learnable symbol [`CLS`] is prepended to the beginning of its text descriptions, after which the concatenated text sequence is processed by the BERT [171]. We use the output of [`CLS`] as the text embedding of the item, which is a 768-dimensional vector.

To show that pre-trained text embeddings in these three datasets also suffer from representation degeneration, we plot their singular values in Figure. 5.2 and observe a rapid decrease in small values. This suggests an anisotropic nature in which one dimension is dominant while the effectiveness of other dimensions is limited. Additionally, for each item pair (*i.e.*, different items) in a dataset, we calculate the cosine similarity based on their pre-trained text embeddings. The average cosine similarities of all item pairs for Arts, Toys, and Tools datasets are 0.85, 0.84, and 0.85 respectively. Indeed, item representations are presented with high cosine similarities, which indicates that their semantic similarities are high and their embedding distributions are highly anisotropic. Therefore, it is difficult

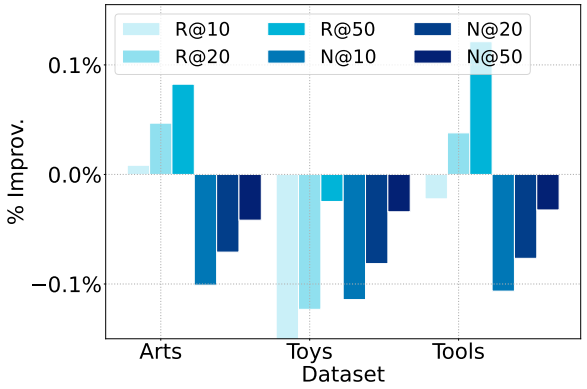


FIGURE 5.3: Performance improvement of SASRec^T compared with SASRec^{ID}.

to distinguish between items that use semantically different texts but are close to each other in the embedding space. The above analysis demonstrates that the pre-trained text embeddings in the recommendation domain also manifest the representation degeneration problem.

To demystify how the degeneration of representation in pre-trained text embeddings affects recommendation performance, we conduct a quantitative analysis of the independent impact of text embeddings on recommendation performance. In particular, we implement a specific instantiation of the general framework (Figure. 5.1), which we refer to as SASRec^T (Figure. 5.1b). It is worth noting that SASRec^T does not utilize ID embeddings. $f_{\theta_1}(\cdot)$ is a projector MLP with two hidden layers for feature transformation, where ReLU activation is appended to both hidden layers of the projector. We compare the recommendation performance of SASRec^T with SASRec^{ID}, which does not incorporate text information. The results are visualized in Figure. 5.3, where the relative performance improvement of SASRec^T over SASRec^{ID} is calculated. The figure reveals that the use of text embeddings improves the Recall but worsens the NDCG for Arts and Tools datasets in most cases. However, for the Toys dataset, all metrics exhibit deterioration. Despite incorporating more informative item contents as opposed to randomly initialized ID embeddings in SASRec^{ID}, the effectiveness of SASRec^T is inferior to that of SASRec^{ID}. We suspect that anisotropic item embedding spaces may be the underlying cause of performance limitations in text-based sequential recommendation models.

5.4.2 Whitening Transformation to Resolve Anisotropy Problem

The anisotropy of pre-trained text embeddings is a widely recognized form of feature degeneration in representation learning. As such, prior works [199, 207] have demonstrated that the application of a whitening transformation [197] to project the elements of pre-trained text embeddings onto a spherical distribution can mitigate the anisotropy problem and reduce similarity among distinct instances. The whitened representation removes the correlation among axes and ensures the item set is scattered in a spherical distribution to avoid the feature collapse with theoretical guarantee [199].

To perform the whitening transformation, given pre-trained text embeddings of all items $\mathbf{X} \in \mathbb{R}^{d_t \times |\mathcal{I}|}$, the whitened output $\mathbf{Z}$ is derived as

$$\mathbf{Z} = \Phi(\mathbf{X} - \mu \cdot \mathbf{1}^\top), \tag{5.1}$$

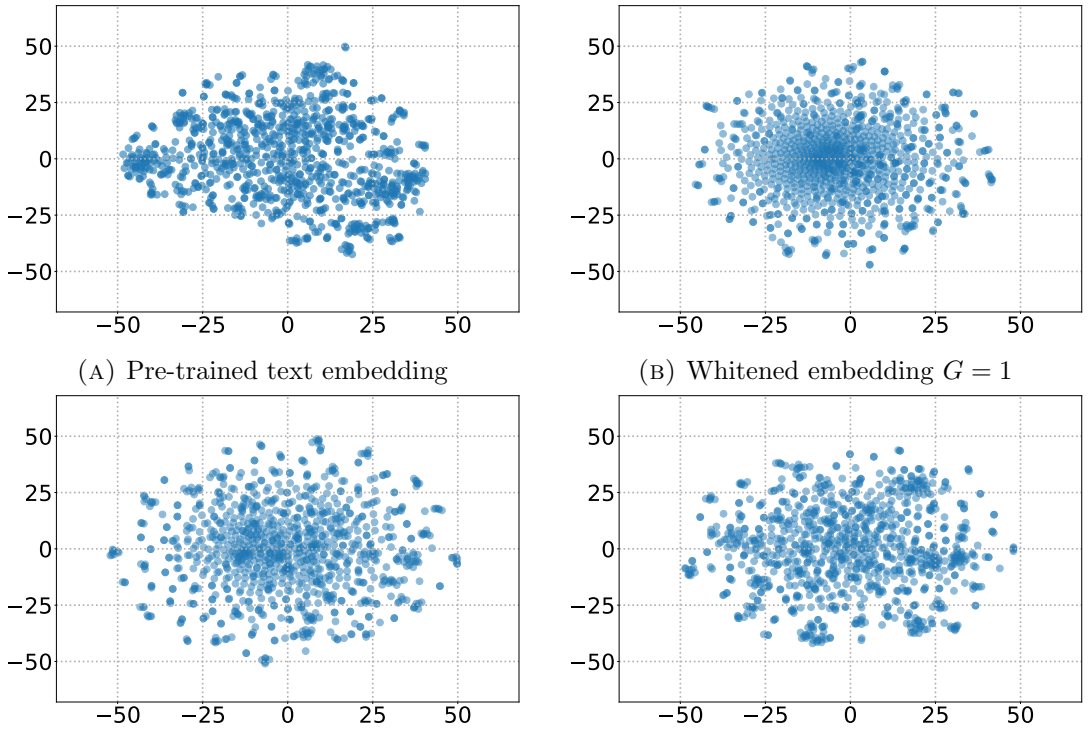
where $\Phi : \mathbb{R}^{d_t \times |\mathcal{I}|} \to \mathbb{R}^{d_t \times |\mathcal{I}|}$ denotes the function for whitening transformation, $\mu = \frac{1}{|\mathcal{I}|}\mathbf{X} \cdot \mathbf{1}$ is the mean of $\mathbf{X}$, $\mathbf{1}$ is the column vector of all ones. There are many possible ways to perform whitening, including BN [205], PCA [197, 206], CD [211], and ZCA [202] whitening. Different whitening methods differ in the choice of Φ. By default, we choose ZCA whitening, which yields the best performance for most experimental datasets. We also compare different whitening operations and report the details of experimental results in Section 5.5.6. For ZCA whitening, Φ is defined as follows,

$$\Phi = \mathbf{D}\Lambda^{-\frac{1}{2}}\mathbf{D}^\top, \tag{5.2}$$

where $\Lambda = \text{diag}(\sigma_1, \cdots, \sigma_{d_t})$ and $\mathbf{D} = [\mathbf{d}_1, \cdots, \mathbf{d}_{d_t}]$ are the eigenvalues and associated eigenvectors of $\mathbf{\Sigma} = \mathbf{D}\Lambda\mathbf{D}^\top$. $\mathbf{\Sigma} = \frac{1}{|\mathcal{I}|}(\mathbf{X} - \mu \cdot \mathbf{1}^\top)(\mathbf{X} - \mu \cdot \mathbf{1}^\top)^\top + \epsilon\mathbf{I}$ is the covariance matrix of the centered input $\mathbf{X}$. Φ ensures the transformed output $\mathbf{Z}$ has the property of $\mathbf{Z}\mathbf{Z}^\top = \mathbf{I}_{d_t}$ to make $\mathbf{X}$ fully whitened. Given that the ZCA assumes a full rank covariance matrix, conducting ZCA on all items in which the cardinality of $\mathcal{I}$ significantly exceeds the dimensionality of d_t (*i.e.*, $|\mathcal{I}| \gg d_t$) ensures that Σ is full rank. We visualize the t-SNE of item text embeddings before and after ZCA whitening in Figure. 5.4a and Figure. 5.4b, respectively. We observe that the distribution of item text embeddings that have undergone whitening exhibits spherical symmetry around the origin and is uniformly spread in all directions.

TABLE 5.1: Performance comparison of SASRec^{ID}, SASRec^{T}, SASRec^{T+ID}, and SASRec^{W} in terms of R@50 and N@50.

Model	Arts		Toys		Tools	
	R@50	N@50	R@50	N@50	R@50	N@50
SASRec^{ID}	0.1967	<u>0.0887</u>	0.1581	0.0558	0.0941	0.0463
SASRec^{T}	<u>0.2129</u>	0.0850	0.1542	0.0539	<u>0.1055</u>	0.0448
SASRec^{T+ID}	0.2009	0.0879	<u>0.1664</u>	<u>0.0610</u>	0.0954	<u>0.0490</u>
SASRec^{W}	**0.2348**	**0.0939**	**0.1798**	**0.0639**	**0.1196**	**0.0519**
%Improv	10.3%	5.9%	8.1%	4.8%	13.4%	5.9%



(A) Pre-trained text embedding

(B) Whitened embedding $G = 1$

(C) Whitened embedding $G = 4$

(D) Whitened embedding $G = 32$

FIGURE 5.4: t-SNE plots of item text embeddings under different settings for Arts.

We incorporate the ZCA whitening of Eqn. (5.1) into SASRec^{T} and refer to the resulting model as SASRec^{W}, which is illustrated in Figure. 5.1c. As indicated in Table 5.1, the corresponding representation yields a significant improvement of 10.3%, 8.1%, 13.4% in Recall@50 compared to SASRec^{ID}, SASRec^{T}, or SASRec^{T+ID} on Arts, Toys, and Tools respectively. It is evident that the application of the whitening transformation in SASRec^{W}, without introducing any additional trainable parameters, results in a significant improvement in performance.

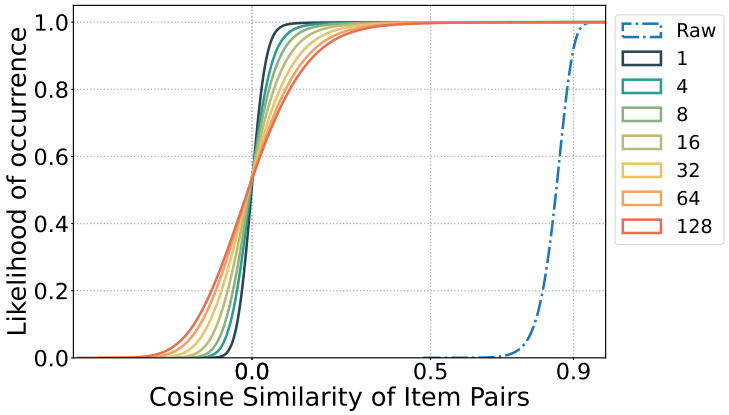


FIGURE 5.5: CDF plot of item pairs in terms of the cosine similarities for Arts dataset. "Raw" denotes the original text features without whitening.

5.4.3 Relaxed Whitening for Retaining Text Semantics

Although ZCA whitening can effectively decorrelate pre-trained text embeddings, our observations indicate that the fully whitened representation (Figure. 5.4b) may have an adverse impact on the manifold of items sharing similar textual semantics, in comparison to the original text representation (Figure. 5.4a).

Inspired by [206, 207], we adapt "group whitening" to standardize covariance matrices within dimensional groups to relax the extent of whitening and retain more original textual semantics. Specifically, the relaxed whitening with the number of groups G takes as its input a matrix $\mathbf{X} \in \mathbb{R}^{d_t \times |\mathcal{I}|}$ and its output is a matrix $\mathbf{Y} \in \mathbb{R}^{d_t \times |\mathcal{I}|}$ computed as:

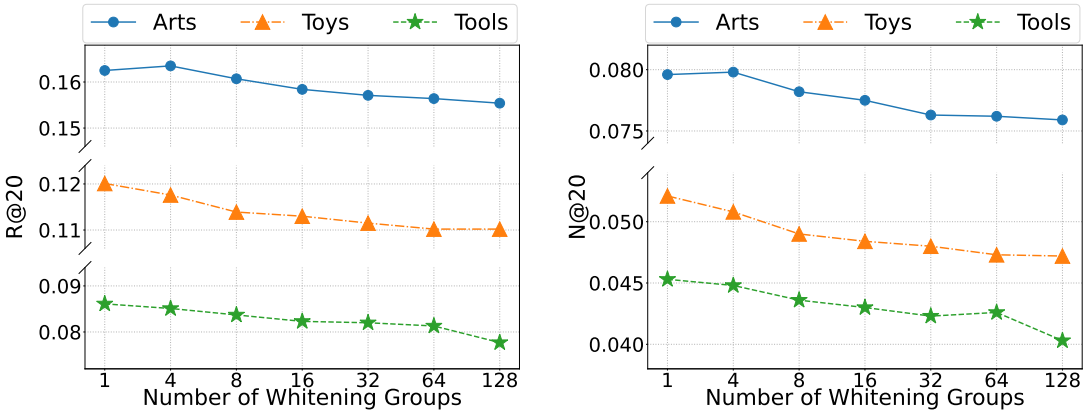
$$\mathbf{Y}^{[h]} = \text{ZCA}(\mathbf{X}^{[h]}), \tag{5.3}$$

$$\mathbf{X}^{[h]} = \left(\left(\mathbf{X}_{(h-1)\cdot \frac{d_t}{G}+1} \right)^\top, \cdots, \left(\mathbf{X}_{h\cdot \frac{d_t}{G}} \right)^\top \right)^\top \in \mathbb{R}^{\frac{d_t}{G} \times |\mathcal{I}|}, \tag{5.4}$$

$$\mathbf{Y}^{[h]} = \left(\left(\mathbf{Y}_{(h-1)\cdot \frac{d_t}{G}+1} \right)^\top, \cdots, \left(\mathbf{Y}_{h\cdot \frac{d_t}{G}} \right)^\top \right)^\top \in \mathbb{R}^{\frac{d_t}{G} \times |\mathcal{I}|}, \tag{5.5}$$

where $\text{ZCA}(\mathbf{X}) = \mathbf{D}\Lambda^{-\frac{1}{2}}\mathbf{D}^\top(\mathbf{X} - \mu \cdot \mathbf{1}^\top)$ follows Eqn. (5.1) and (5.2). In other words, $\mathbf{Y}$ is derived by dividing all feature dimensions d_t into G groups and applying ZCA whitening to each group independently.

We visualize the Cumulative Distribution Function (CDF) plot of item pairs concerning different extents of whitening on text embeddings of Arts, *i.e.*, different



(A) Recall@20 (B) NDCG@20

FIGURE 5.6: Performance of different groups G for Whitening for SASRec^W.

G, in Figure. 5.5. The legend specifies G involved in the whitening transformation, with a decreasing extent of whitening resulting from an increase in G. Other datasets showing similar distributions are omitted for space constraints. Namely, a smaller value of G corresponds to a higher degree of decorrelation, indicating a stronger suppression of redundant information. From Figure.5.5, weaker whitening leads to a less concentrated CDF line within a broader range, indicating increasingly similar item representations and more preserving of textual semantics.

Despite the apparent advantage of retaining text semantics for recommendation tasks, our findings suggest that the exclusive use of relaxed whitened item representations for recommendation may result in cluttered item embedding distributions, as demonstrated in Figure. 5.4b, Figure. 5.4c and Figure. 5.4d. In these figures, we perform whitening with values of G equal to 1, 4, and 32, respectively. It is apparent that as G increases, the distribution becomes increasingly non-uniform. To investigate the impact of G on the recommendation performance, we conduct experiments on SASRec^W by varying G in the range of $\{1, 4, 8, 16, 32, 64, 128\}$ and report the results in Figure. 5.6. The results indicate that optimal performance is achieved when G is set to a smaller value, suggesting that further decorrelation enhances the representation learning of sequential recommendation.

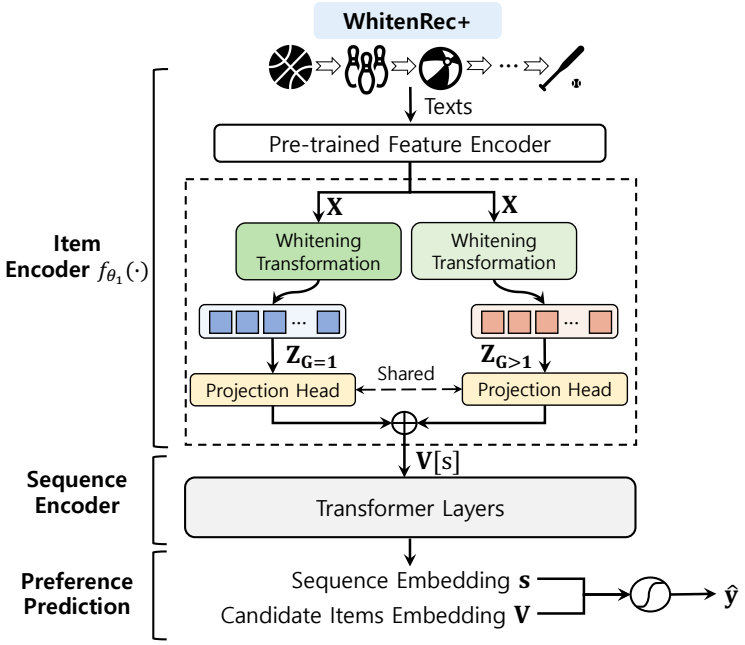


FIGURE 5.7: The illustration of WhitenRec+.

5.4.4 WhitenRec+: Ensemble of Relaxed Whitening for Further Gains

The preceding sections present evidence that applying whitening techniques for dimension decorrelation mitigates the feature degeneration issue, consequently enhancing the sequential recommendation performance. Nonetheless, a relaxation of the whitening criteria aiming to retain more of the original text semantics results in sub-optimal performance.

To maximize the advantages of complete whitening while retaining some semantic content from the original text features, we propose an ensemble method WhitenRec+, which leverages both fully whitened representations and relaxed whitened representations to further improve the learning of item representations. The framework is depicted in Figure. 5.7. Specifically, we apply both the most stringent and relaxed whitening on item text features $\mathbf{X}$. The resultant embeddings are denoted as $\mathbf{Z}_{G=1}$ and $\mathbf{Z}_{G>1}$, respectively. Then, we map both $\mathbf{Z}_{G=1}$ and $\mathbf{Z}_{G>1}$ into a latent representation space using the item encoder $f_{\theta_1}(\cdot)$, *i.e.*, a shared projection head consisting of two MLP layers. The outputs from $f_{\theta_1(\cdot)}$ are combined using

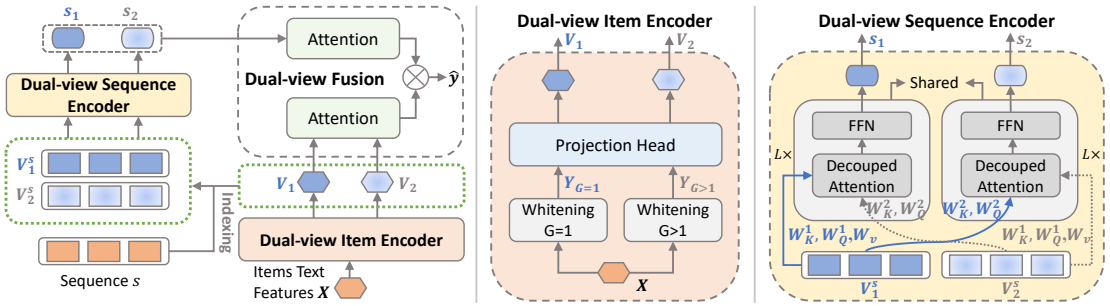


FIGURE 5.8: The illustration of DWSRec: 1) Dual-view Item Encoder extracts item embeddings using both full and relaxed whitening; 2) Dual-view Sequence Encoder optimizes attention calculations by leveraging different degrees of whitening; 3) Dual-view Fusion to merge view-specific embeddings via attention layers.

element-wise summation:

$$\mathbf{V} = f_{\theta_1}\left(\mathbf{Z}_{G=1}\right) + f_{\theta_1}\left(\mathbf{Z}_{G>1}\right). \tag{5.6}$$

Subsequently, $\mathbf{V}$ is used as the input to the sequence encoder for the generation of recommendations.

Intuitively, the recommendation model can learn representations that are both discriminative and robust to variations in the input data by leveraging both fully whitened representations and relaxed whitened representations.

5.4.5 DWSRec: Dual-view Whitening for Sequential Recommendation

To build upon the advantages of both fully and relaxed whitened representations, we have refined the model architecture of WhitenRec+ and introduced DWSRec. DWSRec employs varying degrees of whitened representations, utilizing them as multi-faceted views for updating the Transformer alternatively. This approach significantly enhances the modeling of users and items, leveraging the nuanced perspectives provided by the different levels of whitened representations. The illustration of DWSRec is shown in Figure. 5.8.

5.4.5.1 Dual-view Item Encoder

We apply both the full and relaxed whitening on item text features $\mathbf{X}$. The resultant embeddings are denoted as $\mathbf{Y}_{G=1}$ and $\mathbf{Y}_{G>1}$, respectively. Then, we map both $\mathbf{Y}_{G=1}$ and $\mathbf{Y}_{G>1}$ into a latent space using a shared projection head consisting of two MLP layers. The outputs from the projection are denoted as $\mathbf{V}_1 \in \mathbb{R}^{d\times|\mathcal{I}|}$ and $\mathbf{V}_2 \in \mathbb{R}^{d\times|\mathcal{I}|}$ for $\mathbf{Y}_{G=1}$ and $\mathbf{Y}_{G>1}$ respectively.

5.4.5.2 Dual-view Sequence Encoder

The Transformer model [196] has become a preeminent method for sequence encoding. However, its conventional design primarily focuses on the self-correlation within a single type of sequence. Inspired by [54], we adapt the Transformer with a decoupled attention mechanism to encode sequences from diverse whitening views. This refines the attention computation for a given view by leveraging an alternative view, thus enabling adaptive gradient adjustments to capture the nuances of different whitenings.

As shown in Figure. 5.8, the dual-view sequence encoder comprises two transformer modules with shared parameters. A transformer module contains multiple stacked blocks, with each block consisting of a decoupled attention layer and a feed-forward layer (FFN). These blocks utilize two input types generated from the dual-view item encoder: full and relaxed whitening. Given a user sequence s, the embeddings for all items in s retrieved from $\mathbf{V}_1$ and $\mathbf{V}_2$ are denoted as $\mathbf{V}_1^s \in \mathbb{R}^{d\times|\mathcal{S}|}$ and $\mathbf{V}_2^s \in \mathbb{R}^{d\times|\mathcal{S}|}$ respectively. For each decoupled attention layer, two sets of key-query projection matrices and one value projection matrix are generated for each of h heads. They are denoted as $\mathbf{W}_K^{1,i}$, $\mathbf{W}_Q^{1,i}$, $\mathbf{W}_K^{2,i}$, $\mathbf{W}_Q^{2,i}$, $\mathbf{W}_V^i \in \mathbb{R}^{d_h\times d}$, $i \in [h]$, and $d_h = d/h$. We first learn the sequence representation using $\mathbf{V}_1^s$ and correlate it with $\mathbf{V}_2^s$. We use $\mathbf{V}_1^s$ as the input to the value and first key-query pair projections, whereas $\mathbf{V}_2^s$ is used for the second key-query pair projections in attention computation. Specifically, the attention matrices of a head i for $\mathbf{V}_1^s$ given $\mathbf{V}_2^s$ are formulated as follows:

$$att_1^i = (\mathbf{W}_Q^{1,i}\mathbf{V}_1^s)(\mathbf{W}_K^{1,i}\mathbf{V}_1^s)^\top, att_2^i = (\mathbf{W}_Q^{2,i}\mathbf{V}_2^s)(\mathbf{W}_K^{2,i}\mathbf{V}_2^s)^\top.$$

These matrices are fused with a function $\mathcal{F}$ using addition to produce outputs for each head:

$$head^i = \sigma(\frac{\mathcal{F}(att_1^i, att_2^i)}{\sqrt{d}})(\mathbf{W}_V^i \mathbf{V}_1^s). \tag{5.7}$$

The concatenated outputs of all attention heads serve as the input to the FFN. After L decoupled transformer layers, following [51], the embedding of the sequence's last item represents the sequence and is denoted as $\mathbf{s}_1 \in \mathbb{R}^d$.

Next, we derive the sequence representation using $\mathbf{V}_2^s$ and correlate it with $\mathbf{V}_1^s$. Here, $\mathbf{V}_2^s$ serves as input to the value and first key-query pair projections, whereas $\mathbf{V}_1^s$ is employed for the second key-query pair projections. The corresponding attention matrices for $\mathbf{V}_2^s$ in relation to $\mathbf{V}_1^s$ are detailed below:

$$\widehat{att}_1^{\,i} = (\mathbf{W}_Q^{1,i}\mathbf{V}_2^s)(\mathbf{W}_K^{1,i}\mathbf{V}_2^s)^\top, \widehat{att}_2^{\,i} = (\mathbf{W}_Q^{2,i}\mathbf{V}_1^s)(\mathbf{W}_K^{2,i}\mathbf{V}_1^s)^\top,$$
$$\widehat{head}^{\,i} = \sigma(\frac{\mathcal{F}(\widehat{att}_1^{\,i}, \widehat{att}_2^{\,i})}{\sqrt{d}})(\mathbf{W}_V^i \mathbf{V}_2^s). \tag{5.8}$$

The output sequence embedding is denoted as $\mathbf{s}_2 \in \mathbb{R}^d$.

Our proposed dual-view sequence encoder leverages diverse views of whitened embeddings to interchangeably update the attention heads within the transformer model. This approach can be perceived as generating augmented data instances, thereby enriching the training process and enhancing overall performance.

5.4.5.3 Dual-view Fusion

Given two views of sequence embeddings, $\mathbf{s}_1$ and $\mathbf{s}_2$, and two views of item embeddings $\mathbf{V}_1$ and $\mathbf{V}_2$, we aim to adaptively merge these view-specific embeddings using learnable attentive weights. The resulting aggregated sequence representation, $\mathbf{s}$, is computed as follows:

$$\mathbf{s} = \sum_{\mathbf{e}_i \in \{\mathbf{s}_1, \mathbf{s}_2\}} f(\mathbf{e}_i)\mathbf{e}_i, \quad f(\mathbf{e}_i) = \frac{\exp(\mathbf{a}^\top \cdot \mathbf{W}_a \mathbf{e}_i)}{\sum_i \exp(\mathbf{a}^\top \cdot \mathbf{W}_a \mathbf{e}_i)},$$

where $\mathbf{a} \in \mathbb{R}^d$ and $\mathbf{W}_a \in \mathbb{R}^{d \times d}$ are trainable attention parameters. Similarly, the aggregated item representation $\mathbf{V}$ is computed as:

$$\mathbf{V} = \sum_{\mathbf{E}_i \in \{\mathbf{V}_1, \mathbf{V}_2\}} f(\mathbf{E}_i)\mathbf{E}_i, \quad f(\mathbf{E}_i) = \frac{\exp(\mathbf{b}^\top \cdot \mathbf{W}_b \mathbf{E}_i)}{\sum_i \exp(\mathbf{b}^\top \cdot \mathbf{W}_b \mathbf{E}_i)},$$

where $\mathbf{b} \in \mathbb{R}^d$ and $\mathbf{W}_b \in \mathbb{R}^{d \times d}$ are trainable attention parameters. With $\mathbf{V} \in \mathbb{R}^{d \times |\mathcal{I}|}$ and $\mathbf{s} \in \mathbb{R}^d$, the model is optimized with the cross-entropy loss:

$$\mathcal{L} = -\log(\hat{\mathbf{y}})\,\text{onehot}(\mathbf{y}), \quad \hat{\mathbf{y}} = \text{softmax}(\mathbf{V}^\top \mathbf{s}), \tag{5.9}$$

where $\mathbf{y}$ is the ground-truth next item given a user sequence.

5.4.6 Discussion and Analysis

We further investigate the merits of SASRec^W and DWSRec through both empirical and theoretical analyses. Our examination focuses on representation uniformity and alignment, conditioning, and information reconstruction.

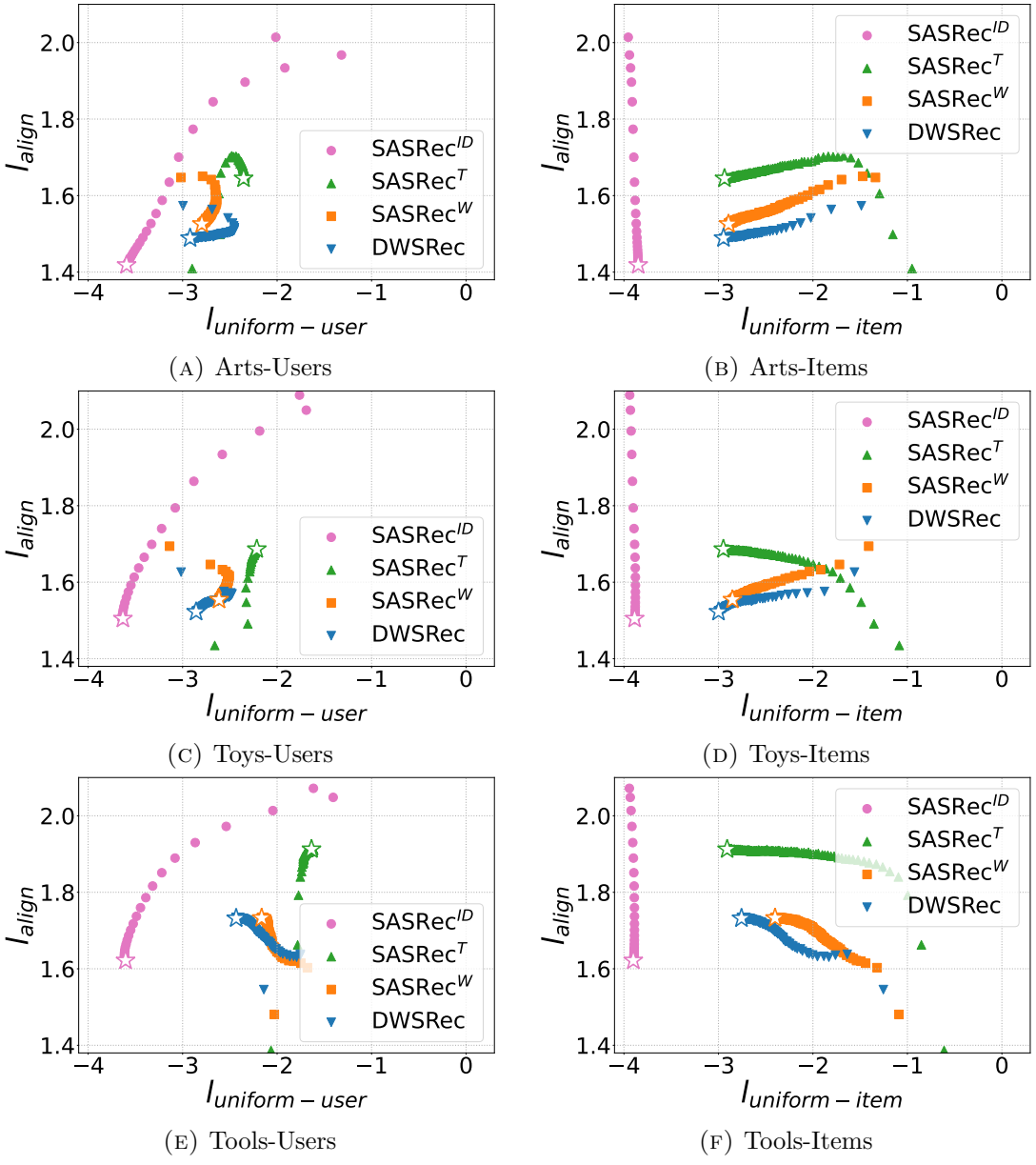
5.4.6.1 Uniformity and Alignment

We analyze the user embedding $\mathbf{s}$ (*i.e.*, generated by the sequence encoder) and the item embedding $\mathbf{v}$ retrieved from $\mathbf{V}$ (*i.e.*, generated by the item encoder) with respect to their uniformity and alignment [198]. The uniformity and alignment in the context of recommendation are formulated as follows:

$$\begin{aligned} l_{align} &= \mathop{\mathbb{E}}_{(u,i)\sim p_{pos}} ||f(\mathbf{s}_u) - f(\mathbf{v}_i)||^2, \\ l_{uniform-user} &= log \mathop{\mathbb{E}}_{(u,u')\sim p_{user}} e^{-2||f(\mathbf{s}_u) - f(\mathbf{s}_{u'})||^2}, \\ l_{uniform-item} &= log \mathop{\mathbb{E}}_{(i,i')\sim p_{item}} e^{-2||f(\mathbf{v}_i) - f(\mathbf{v}_{i'})||^2}, \end{aligned} \tag{5.10}$$

where $f(\cdot)$ indicates l_2 normalized representations. p_{pos}, p_{user}, and p_{item} are the distribution of positive user-item pairs, users, and items, respectively.

We present visualizations of the learned user and item representations from four models with respect to uniformity and alignment on the Arts, Toys, and Tools



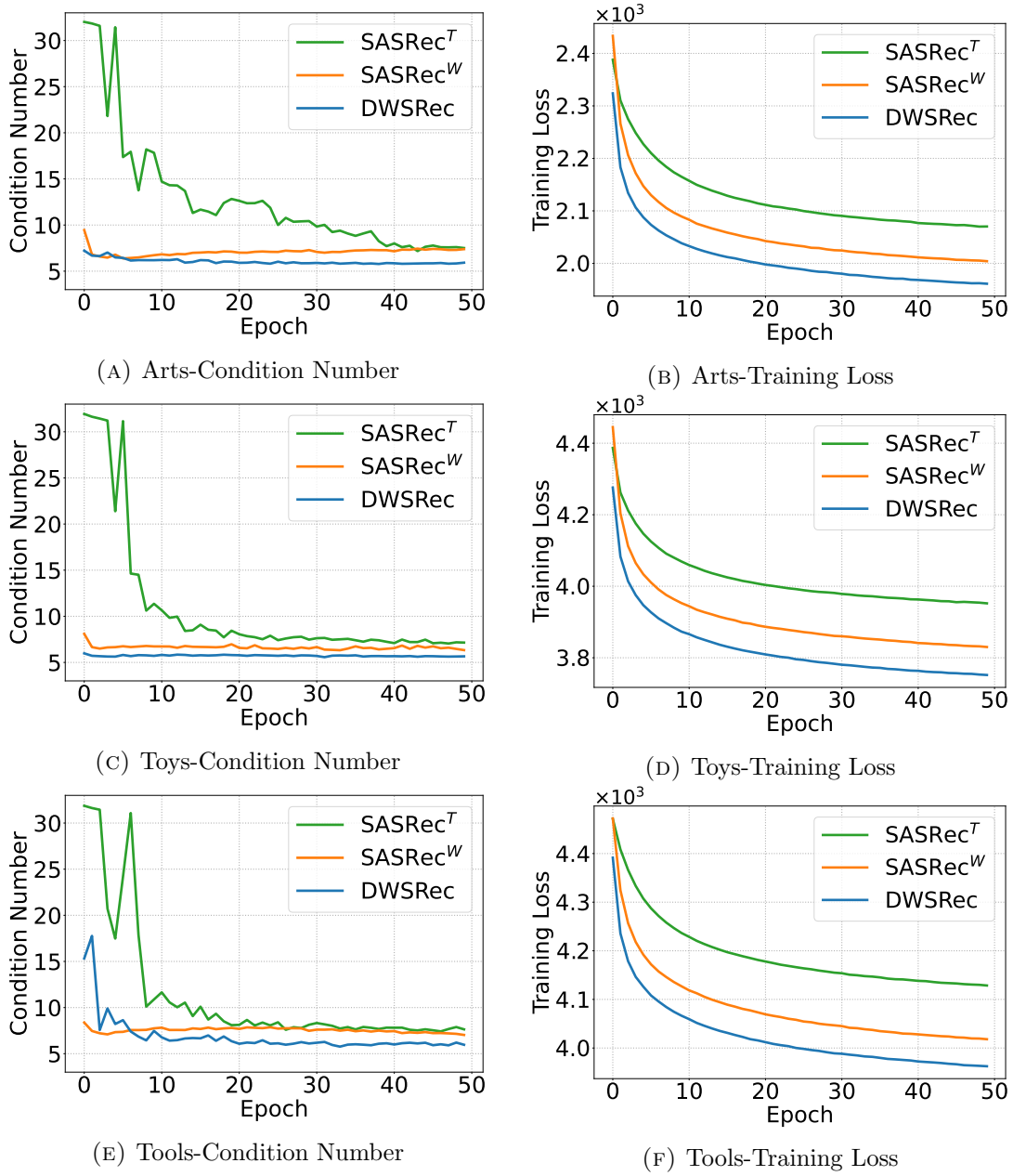
(A) Arts-Users (B) Arts-Items

(C) Toys-Users (D) Toys-Items

(E) Tools-Users (F) Tools-Items

FIGURE 5.9: $l_{align} - l_{uniform}$ plots for representations of users and items during training. We visualize these two metrics in each epoch, and the stars indicate the last converged epoch. For l_{align} and $l_{uniform}$, lower numbers are better.

datasets in Figure.5.9. For comparative analysis, we include one ID-based method (*i.e.*, SASRec^{ID}) and three text-based methods(*i.e.*, SASRec^{T}, SASRec^{W}, and DWSRec). Both SASRec^{W} and DWSRec achieve better alignment and user uniformity compared to SASRec^{T}, leading to enhanced performance. DWSRec further improves all metrics compared with SASRec^{W} and achieves the best performance. It is important to note that despite SASRec^{ID} exhibiting the highest level of uniformity, its performance is poor. This suggests that the positive correlation between



(A) Arts-Condition Number

(B) Arts-Training Loss

(C) Toys-Condition Number

(D) Toys-Training Loss

(E) Tools-Condition Number

(F) Tools-Training Loss

FIGURE 5.10: Conditioning analysis on Arts, Toys, and Tools dataset for SASRec^T, SASRec^W, and DWSRec. We plot the condition number (log-scale) calculated for the item embedding matrix after projection and the training loss with respect to each epoch.

uniformity and performance is limited in scope. Excessive pursuit of uniformity may overlook the proximity of semantically similar items and ultimately impair recommendation performance.

5.4.6.2 Conditioning Analysis

We highlight the benefits of SASRec^W and DWSRec in achieving improved conditioning of item embedding matrix $\mathbf{V}$. Given the covariance matrix $\mathbf{A}$ of $\mathbf{V}$, we measure its conditioning by the condition number [212]:

$$\kappa(\mathbf{A}) = \lambda_{max}(\mathbf{A})\lambda_{min}^{-1}(\mathbf{A}), \tag{5.11}$$

where $\lambda(\cdot)$ denotes the eigenvalue of the matrix. Well-conditioned matrices have a low condition number, while ill-conditioned matrices have a high condition number. For neural networks, ill-conditioned covariance matrices cause detrimental effects on training stability and optimization. Figure. 5.10(A), Figure. 5.10(C), and Figure. 5.10(E) show the evolution of the condition number throughout training epochs for Arts, Toys, and Tools respectively. Figure. 5.10(B), Figure. 5.10(D), and Figure. 5.10(F) show the training loss over training epochs for the same datasets. The results demonstrate that both SASRec^W and DWSRec converge more rapidly and achieve better conditioning compared to SASRec^T. This outcome highlights the effectiveness of the whitening transformation in simplifying the optimization problem. Also, DWSRec achieves the best conditioning and highest convergence rate.

5.4.6.3 More Preserved Information in WhitenRec+ and DWSRec

We also conduct a mathematical proof to demonstrate that WhitenRec+ and DWSRec is capable of preserving more information than SASRec^W. Given a pre-trained text embedding matrix $\mathbf{X} \in \mathbb{R}^{d \times |\mathcal{I}|}$, where $n = |\mathcal{I}|$ is the number of items and d is the dimension size. We can derive the following proposition:

Proposition 5.1. *WhitenRec+ and DWSRec preserve at least* $(1 - \frac{1}{G})d^2$ *more information in its whitened representations compared to SASRec*^{W}*.*

Proof. Given $\mathbf{X}$, we define the Gram matrix of $\mathbf{X}$ as $\mathbf{K} = \mathbf{X}^\top\mathbf{X} \in \mathbb{R}^{n \times n}$. Based on [213], the prediction for recommendation models depends on training inputs only through $\mathbf{K}$[2]. Denote the text features whitened by SASRec^W as $\mathbf{Z}$. Since

[2]pre-trained text embeddings are typically generated from a linear projector in conjunction with BERT.

SASRec^W performs full data whitening on $\mathbf{X}$, we have $\mathbf{Z}\mathbf{Z}^\top = \mathbf{I}_d$. Thus, the Gram matrix of $\mathbf{Z}$ is:

$$\mathbf{K}_\mathbf{Z} = \mathbf{Z}^\top\mathbf{Z} = \mathbf{Z}^\top\mathbf{Z}\mathbf{Z}^+\mathbf{Z} = \mathbf{Z}^+\mathbf{Z}, \tag{5.12}$$

where $^+$ is the Moore–Penrose inverse and $\mathbf{Z} = \mathbf{Z}\mathbf{Z}^+\mathbf{Z}$ holds true.

To determine the amount of decorrelated information preserved in $\mathbf{Z}$, we perform a transformation $\mathbf{Q}$ on $\mathbf{Z}$, resulting in $\widehat{\mathbf{Z}} = \mathbf{Q}\mathbf{Z} = [\mathbf{I}\cdots]$. Here, $\mathbf{Q} \in \mathbb{R}^{d\times d}$ is the inverse of the submatrix formed by the first d columns of $\widehat{\mathbf{Z}}$. As the first d columns are deterministic, hence, $\widehat{\mathbf{Z}}$ can preserve $(n-d)d$ real values. To reconstruct $\mathbf{K}_\mathbf{Z}$,

$$\mathbf{K}_\mathbf{Z} = \mathbf{Z}^+\mathbf{Z} = \mathbf{Z}^+\mathbf{Q}^{-1}\mathbf{Q}\mathbf{Z} = (\mathbf{Q}\mathbf{Z})^+\mathbf{Q}\mathbf{Z} = \widehat{\mathbf{Z}}^+\widehat{\mathbf{Z}}. \tag{5.13}$$

The above equation indicates that both the whitened text feature matrix $\mathbf{Z}$ and $\widehat{\mathbf{Z}}$ contain an equivalent amount of information for reconstructing $\mathbf{K}_\mathbf{Z}$.

Analogously, we can infer a submatrix (one group) whitened by WhitenRec+ and DWSRec preserves $(n-\frac{d}{G})\frac{d}{G}$ real values. With G groups of submatrix, thus, our methods have $(n-\frac{d}{G})d$ values to reconstruct $\mathbf{K}$. Because the addition function used in our methods is injective, WhitenRec+ and DWSRec preserve at least $(1-\frac{1}{G})d^2$ more information for training sequential models. □

5.4.7 Complexity Analysis

Note that various degrees of the whitening transformation can be pre-computed; hence, SASRec^W and WhitenRec+ exhibit identical complexity levels. The time complexity of WhitenRec+ primarily arises from the projection head with MLPs and the attention-based transformer layers. Each contributes a time complexity of $\mathcal{O}(|\mathcal{S}|(d_t d + d^2))$ and $\mathcal{O}(|\mathcal{S}|^2 d + |\mathcal{S}|d^2)$, respectively. Consequently, the aggregate time complexity is $\mathcal{O}(|\mathcal{S}|d_t d + |\mathcal{S}|d^2 + |\mathcal{S}|^2 d)$.

We also analyze the time complexity of our proposed DWSRec, which primarily arises from the projection head with MLPs, decoupled attention-based transformers, and attention layers for fusion. Each contributes a time complexity of $\mathcal{O}(|\mathcal{S}|(d_t d + d^2))$, $\mathcal{O}(|\mathcal{S}|^2 d + |\mathcal{S}|d^2)$, and $\mathcal{O}(d^2 + |\mathcal{S}|d^2)$. Consequently, the total time complexity is $\mathcal{O}(|\mathcal{S}|d_t d + |\mathcal{S}|d^2 + |\mathcal{S}|^2 d)$, which shows no order of magnitude

TABLE 5.2: Statistics of the experimental datasets. "Avg. n" and "Avg. i" denote the average length of interaction sequences and the average actions of items.

Datasets	#Users	#Items	#Inter.	Avg. n	Avg. i	Sparisity
Arts	45,486	21,019	395,150	7.69	16.63	99.96%
Toys	85,694	40,483	618,738	7.22	15.28	99.98%
Tools	90,599	36,244	623,248	6.88	17.20	99.98%

difference with SASRec^W. In the experimental section, we show that our methods have significantly reduced the number of parameters in practice.

5.5 Experiments

5.5.1 Experimental Settings

5.5.1.1 Datasets

To evaluate the performance of the proposed method, we conduct experiments on three representative categories of widely-used Amazon review dataset [188]: *Arts, Crafts and Sewing*, *Toys and Games*, and *Tools and Instruments*. We abbreviate them as Arts, Toys, and Tools. The statistics of the pre-processed experimental datasets are summarized in Table 5.2.

5.5.1.2 Baseline Methods

To evaluate the effectiveness of our proposed method, we compare it with several state-of-the-art recommendation models. These baselines fall into three groups: general recommendation models with text features (*i.e.*, GRCN, BM3), sequential recommendation models (*i.e.*, SASRec, HGN, CL4SRec), and sequential recommendation models with text features (*i.e.*, SASRec^T, SASRec^{T+ID}, S^3-Rec, FDSA, UniSRec, UniSRec^T, UniSRec^{T+ID}).

- **GRCN** [80] is a graph-based multimodal recommendation model that refines the user-item interaction graph by identifying false-positive feedback and pruning noisy edges. Only item text representations are exploited;

- **BM3** [59] utilizes contrastive learning losses for multimodal recommendation. Only item text representations are exploited;
- **SASRec**^{ID} [49] is a directional self-attention method for next item prediction. Item representations are randomly initialized ID embeddings;
- **HGN** [195] represents a sequence of items at the group level using a hierarchical gating network, and explicitly captures the item-item relationship via the item-item product;
- **CL4SRec** [53] designs three data augmentation approaches to construct contrastive tasks and extract self-supervised signals to improve the sequential recommendation performance;
- **SASRec**^{T} and **SASRec**$^{T+ID}$ are two extensions of SASRec [49]. **SASRec**^{T} transforms item text representations with MLPs as the input to the self-attention blocks. **SASRec**$^{T+ID}$ combines item ID embeddings with transformed text representations of items as the input to the self-attention blocks;
- **FDSA** [56] models the transition patterns between items as well as features by separate self-attention blocks;
- **S^3-Rec** [51] devises supervised learning objectives to learn the correlations between items and features;
- **DIF-SR** [54] decouples side information and item ID representation from the calculation of attention. The model is optimized with an auxiliary attribute prediction loss to enhance the interaction of side information and item representation;
- **UniSRec** [57] leverages item text representations with an MoE-based adaptor and employs contrastive learning tasks to learn transferable sequence representations. For a fair comparison, we remove its pre-training stage and fine-tune the model with the inductive setting and the transductive setting, which are denoted as **UniSRec**^{T} and **UniSRec**$^{T+ID}$ respectively. The inductive setting takes into account only item text representations, whereas the transductive setting takes into account both item text and ID representations.

5.5.1.3 Evaluations

We conduct experiments in both warm-start and cold-start settings.

Warm-start settings. Following [50, 51], we keep the five-core datasets and discard users and items with fewer than five interactions. We apply the *leave-one-out* strategy to evaluate the performance of recommendation models. Specifically, for each user, the last item of her interaction sequence is used for testing, the second last item is used for validation, and the remaining items are used for model training.

Cold-start settings. Following [214], a subset of items (15% of all items) is randomly selected, and all user-item interactions related to this subset are removed. We preserve sequences containing the aforementioned "cold" items as target items in the validation and testing sets. Since these items are not encountered by the model during training, we can assess the model's capability to generalize to previously unseen items.

Each method is evaluated on the entire item set without sampling to avoid inconsistent results [215]. The recommendation performance is evaluated by two widely used metrics, *i.e.*, Recall@K and Normalized Discounted Cumulative Gain@K (respectively denoted by R@K and N@K). In the experiments, K is empirically set to 20 and 50.

5.5.1.4 Implementation Details

The proposed method and all baselines are implemented by Pytorch [216] and an open-source recommendation framework RecBole [217]. The Adam optimizer [192] is used to learn model parameters. For a fair comparison with baselines and model variants, we set the maximum sequence length, embedding size, and batch size to 50, 300, and 1024, respectively. We also consistently set the number of self-attention blocks, attention heads, and MLP layers in the projection head at 2. Other hyper-parameters of baseline methods are chosen as per their original papers, with optimal settings derived from the model performance on validation data. For our proposed methods, we tune the learning rate in $\{1e^{-5}, 5e^{-5}, 1e^{-4}, 5e^{-4}, 1e^{-3}\}$ and weight decay in $\{0, 1e^{-3}, 1e^{-4}, 1e^{-6}\}$. The group number G is empirically set to 4. The number of decoupled attention-based Transformer layers L is set to 2.

Besides, we adopt an early stopping strategy, *i.e.*, we apply a premature stopping if N@20 on the validation data does not increase for 10 epochs to avoid over-fitting.

5.5.2 Performance Comparison

5.5.2.1 Overall performance

Table 5.3 shows the performance comparison results for warm-start settings, from which we can observe:

- General recommendation methods utilizing text features perform worse than sequential methods, highlighting the effectiveness of sequence encoders in capturing sequential data patterns.
- Sequential methods utilizing text features yield better performance overall, suggesting that text features provide rich semantic information about items, and can enhance recommendation accuracy.
- Our methods SASRec^{W}, WhitenRec+, and DWSRec significantly outperform both general recommendation methods with text features and sequential recommendation methods, demonstrating the effectiveness of the whitening for text features extracted from pre-trained encoders.
- The performances of SASRec^{W}, WhitenRec+, and DWSRec are comparable or superior to that of SASRec^{T+ID} or UniSRec^{T+ID}.This finding suggests that the proposed whitening transformation approach can achieve improved results without depending on ID embeddings, while also reducing the number of learnable parameters.
- DWSRec and WhitenRec+ can further improve recommendation performance compared with SASRec^{W}. This indicates leveraging both fully whitened and relaxed whitened text representations can enhance the item representation learning, and therefore improve the sequential recommendation performance.
- DWSRec outperforms all methods, showing that using both fully and relaxed whitened representations with the dual-view encoders enhances user and item representation learning.

TABLE 5.3: Performance of different methods on the warm-start setting. The best results are in **boldface**, and the best results for baselines are underlined. * denotes SASRec^W, WhitenRec+, or DWSRec surpasses the best baseline using a paired t-test ($p < 0.01$).

Dataset	Model	R@20	R@50	N@20	N@50
Arts	GRCN	0.0851	0.1296	0.0411	0.0499
	BM3	0.1233	0.1782	0.0642	0.075
	SASRec^{ID}	0.1410	0.1967	0.0776	0.0887
	HGN	0.1293	0.1880	0.0693	0.0810
	CL4SRec	0.1388	0.1967	0.0653	0.0768
	SASRec^{T}	0.1476	0.2129	0.0721	0.0850
	SASRec^{T+ID}	0.1435	0.2009	0.0766	0.0879
	FDSA	0.1284	0.1788	0.0785	0.0888
	S^3-Rec	0.1411	0.2007	0.0762	0.0880
	DIF-SR	0.1510	0.2126	0.0701	0.0823
	UniSRec^{T}	0.1500	0.2165	0.0738	0.0869
	UniSRec^{T+ID}	0.1611	0.2322	0.0774	0.0915
	SASRec^{W}	0.1625	0.2348	0.0796	0.0939*
	WhitenRec+	0.1688*	0.2403*	0.0810*	0.0952*
	DWSRec	**0.1710***	**0.2419***	**0.0822***	**0.0962***
Toys	GRCN	0.0651	0.0981	0.0304	0.0369
	BM3	0.0965	0.1383	0.0478	0.0560
	SASRec^{ID}	0.1121	0.1581	0.0467	0.0558
	HGN	0.0983	0.1466	0.0435	0.0530
	CL4SRec	0.1094	0.1609	0.0426	0.0528
	SASRec^{T}	0.0983	0.1542	0.0429	0.0539
	SASRec^{T+ID}	0.1163	0.1664	0.0511	0.0610
	FDSA	0.0895	0.1242	0.0475	0.0543
	S^3-Rec	0.1068	0.1533	0.0488	0.0581
	DIF-SR	0.1176	0.1663	0.0487	0.0584
	UniSRec^{T}	0.1042	0.1607	0.0451	0.0563
	UniSRec^{T+ID}	0.1257	0.1801	0.0513	0.0621
	SASRec^{W}	0.1201	0.1798	0.0521	0.0639*
	WhitenRec+	0.1257	0.1874*	0.0537*	0.0659*
	DWSRec	**0.1307***	**0.1931***	**0.0560***	**0.0683***
Tools	GRCN	0.0452	0.0682	0.0234	0.0280
	BM3	0.0530	0.0714	0.0299	0.0335
	SASRec^{ID}	0.0712	0.0941	0.0418	0.0463
	HGN	0.0647	0.0902	0.0375	0.0425
	CL4SRec	0.0781	0.1027	0.0385	0.0433
	SASRec^{T}	0.0739	0.1055	0.0386	0.0448
	SASRec^{T+ID}	0.0728	0.0954	0.0445	0.0490
	FDSA	0.0633	0.0812	0.0432	0.0468
	S^3-Rec	0.0707	0.0943	0.0424	0.0470
	DIF-SR	0.0732	0.0955	0.0414	0.0458
	UniSRec^{T}	0.0772	0.1091	0.0407	0.0470
	UniSRec^{T+ID}	0.0828	0.1116	0.0420	0.0477
	SASRec^{W}	0.0861*	0.1196*	0.0453	0.0519*
	WhitenRec+	0.0888*	0.1236*	0.0462*	0.0531*
	DWSRec	**0.0918***	**0.1254***	**0.0479***	**0.0546***

5.5.2.2 Performance in cold-start settings

The cold-start problem persists in recommendation systems. Item text features provide rich content information that can alleviate the cold-start problem. We conduct cold-start experiments by comparing them with representative baselines, including SASRec^T and UniSRec^T. Table 5.4 shows the results of the performance comparison, from which we can observe:

- UniSRec^T performs better than SASRec^T, indicating the effectiveness of utilizing the Mixture-of-Experts adaptor with parametric whitening to transform text embeddings for the recommendation task.
- Full whitening SASRec$^W{}_{G=1}$ is either surpassed by or yields similar performance to UniSRec^T in the Arts and Toys datasets. In contrast, relaxed whitening SASRec$^W{}_{G>1}$ outperforms SASRec$^W{}_{G=1}$ and baselines. It suggests that the utilization of relaxed whitened representations facilitates improved generalization for unseen data, ultimately leading to greater performance enhancement.
- Our proposed methods, WhitenRec+ and DWSRec, demonstrate the best performance across the majority of baselines for all three datasets. Leveraging both full and relaxed whitening transformation on text features is proved to be effective under the cold-start setting.

5.5.3 Ablation Study

To assess DWSRec designs, we examine four variants outlined in Table 5.5:

- w/o $\mathbf{s}_1$: excludes the left segment of the dual-view sequence encoder, using only $\mathbf{s}_2$ for prediction.
- w/o $\mathbf{s}_2$: omits the right segment, relying solely on $\mathbf{s}_1$ for prediction.
- w/o D-Trm: uses conventional Transformer to derive sequence embeddings via both full and relaxed whitening separately.
- w/o Attn: replaces attention fusion layers with element-wise summation.

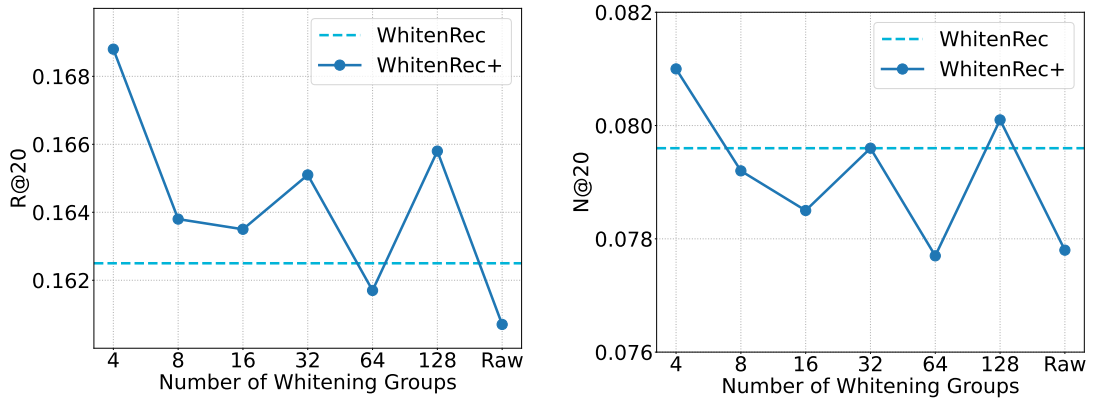
TABLE 5.4: Performance comparison of different methods on the cold-start setting. The best results are in **boldface**, and the second best results are underlined.

Dataset	Model	R@20	R@50	N@20	N@50
Arts	SASRec^T	0.0300	0.0504	0.0130	0.0170
	UniSRec^T	0.0617	0.0796	0.0281	0.0316
	SASRec$^W_{G=1}$	0.0554	0.0655	0.0271	0.0290
	SASRec$^W_{G>1}$	0.0656	0.0820	0.0297	0.0329
	WhitenRec+	0.0693	0.0834	**0.0315**	**0.0343**
	DWSRec	**0.0706**	**0.0864**	0.0310	0.0341
Toys	SASRec^T	0.0239	0.0397	0.0100	0.0131
	UniSRec^T	0.0519	0.0751	0.0222	0.0268
	SASRec$^W_{G=1}$	0.0530	0.0660	0.0238	0.0264
	SASRec$^W_{G>1}$	0.0624	0.0851	0.0265	0.0309
	WhitenRec+	0.0626	**0.0873**	0.0266	0.0316
	DWSRec	**0.0647**	0.0845	**0.0280**	**0.0319**
Tools	SASRec^T	0.0153	0.0261	0.0057	0.0079
	UniSRec^T	0.0298	0.0381	0.0158	0.0175
	SASRec$^W_{G=1}$	0.0431	0.0487	0.0234	0.0245
	SASRec$^W_{G>1}$	0.0501	0.0588	0.0252	0.0280
	WhitenRec+	**0.0537**	**0.0642**	**0.0268**	**0.0288**
	DWSRec	0.0514	0.0594	0.0260	0.0276

TABLE 5.5: Ablation study on DWSRec components. w/o stands for without.

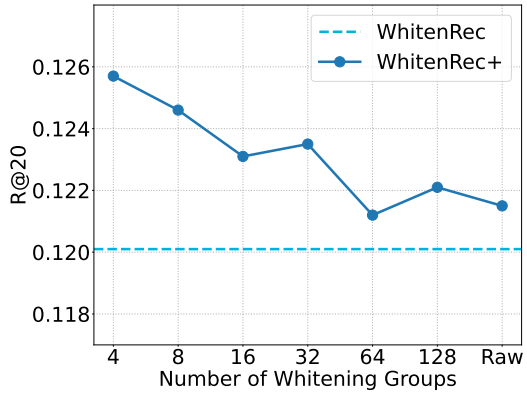
Model	Arts		Toys		Tools	
	R@20	N@20	R@20	N@20	R@20	N@20
DWSRec	**0.1710**	**0.0822**	**0.1307**	**0.0560**	**0.0918**	**0.0479**
1) w/o $\mathbf{s}_1$	0.1650	0.0802	0.1217	0.0525	0.0866	0.0454
2) w/o $\mathbf{s}_2$	0.1652	0.0808	0.1241	0.0538	0.0869	0.0456
3) w/o D-Trm	0.1647	0.0801	0.1222	0.0527	0.0884	0.0463
4) w/o Attn	0.1702	0.0818	0.1253	0.0533	0.0908	0.0468

From Table 5.5, it is observed that removing either $\mathbf{s}_1$ or $\mathbf{s}_2$ diminishes performance, emphasizing the importance of using both embeddings to update the transformer and enhance its generalization. Additionally, removing the decoupled attention leads to further decline, illustrating the effectiveness of the interaction facilitated by the decoupled attention mechanism in harnessing fully and relaxed whitened embeddings. Also, the attention fusion module enhances performance by adaptively combining embeddings from various views.

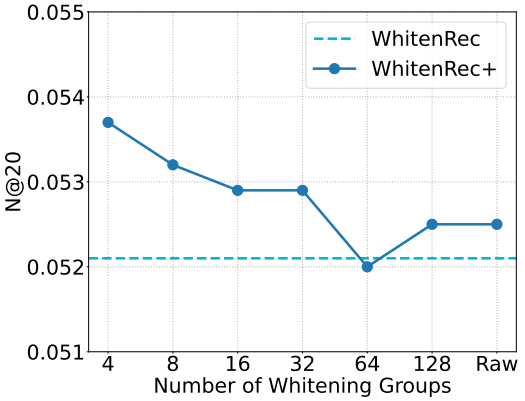


(A) Arts-R@20

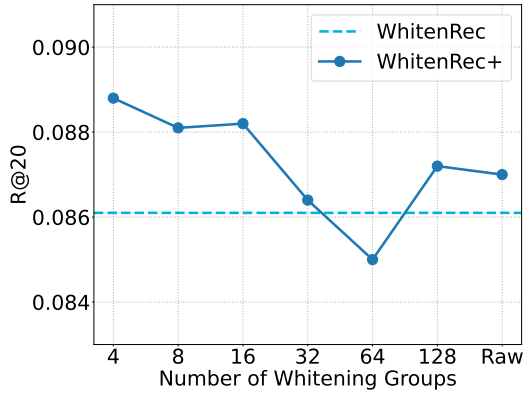
(B) Arts-N@20



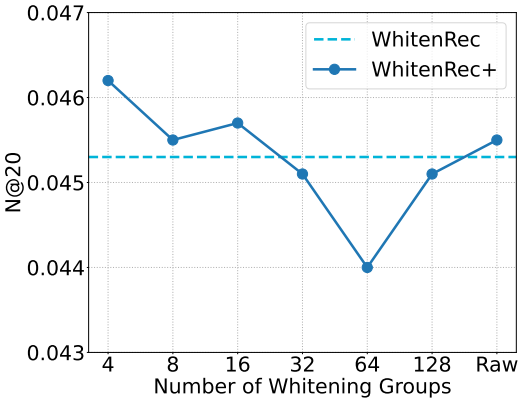
(C) Toys-R@20



(D) Toys-N@20



(E) Tools-R@20

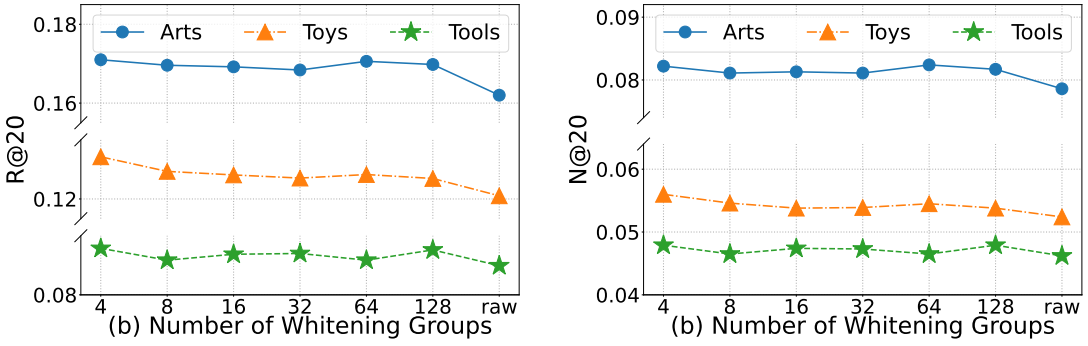


(F) Tools-N@20

FIGURE 5.11: Performance by Whitening Groups for WhitenRec+ for Arts, Toys, and Tools datasets.

5.5.4 Effect of Group Size

To examine the impact of different levels of decorrelation strength of whitening transformation on WhitenRec+ and DWSRec, we experiment with two whitening transformations by fixing one of them with a group number G of 1, representing



(A) R@20 (B) NDCG@20

FIGURE 5.12: Performance by Whitening Groups for DWSRec for Arts, Toys, and Tools datasets.

fully whitened representations. We then vary the G of relaxed whitened representations in {4, 8, 16, 32, 64, 128, Raw}. "Raw" denotes text features without whitening transformation.

For WhitenRec+, the results are shown in Figure. 5.11. We also include SASRec^W's accuracy in the plots for comparison. Figure. 5.11 demonstrates that as the number of groups increases, the performance decreases. Furthermore, we observe that a larger G leads to worse performance than SASRec^W. This indicates that overly relaxed whitened representations are not beneficial and could impede the model's performance. As such, selecting a relatively smaller value with a stronger decorrelation strength is recommended when choosing G for relaxed whitened representations.

For DWSRec, Figure. 5.12 reveals that the optimal G differs by dataset: the Arts favors a larger G, the Toys favors a smaller one, while the Tools shows no distinct preference. "raw" in general performs the worst across all datasets.

5.5.5 Effect of Projection Head

We conduct a preliminary experiment to examine the influence of the projection head on WhitenRec+ and DWSRec's performance. We vary the number of hidden layers in the projection head, choosing from the set {1,2,3}. These configurations are represented as MLP-1, MLP-2, and MLP-3, respectively. We also evaluate the model using a linear projection without a non-linear activation function, termed "Linear". Furthermore, the performance is assessed using the Mixture-of-Experts

TABLE 5.6: Performance comparison of projection head for WhitenRec+. The best results are in **boldface**, and the second best results are underlined.

Dataset	Model	R@20	R@50	N@20	N@50
Arts	Linear	0.1476	0.2090	0.0724	0.0846
	MLP-1	0.1627	0.2330	0.0782	0.0921
	MLP-2	0.1688	**0.2403**	**0.0810**	0.0952
	MLP-3	0.1655	0.2399	0.0808	**0.0955**
	MoE	**0.1690**	0.2366	0.0784	0.0918
Toys	Linear	0.1029	0.1569	0.0448	0.0555
	MLP-1	0.1168	0.1757	0.0494	0.0610
	MLP-2	0.1257	0.1874	0.0537	0.0659
	MLP-3	**0.1261**	**0.1883**	**0.0547**	**0.0670**
	MoE	0.0896	0.1220	0.0446	0.0510
Tools	Linear	0.0751	0.1054	0.0396	0.0456
	MLP-1	0.0836	0.1165	0.0427	0.0492
	MLP-2	0.0888	0.1236	0.0462	0.0531
	MLP-3	**0.0894**	**0.1248**	**0.0469**	**0.0540**
	MoE	0.0852	0.1172	0.0438	0.0501

TABLE 5.7: Performance comparison of projection head for DWSRec. The best results are in **boldface**, and the second best results are underlined.

Dataset	Model	R@20	R@50	N@20	N@50
Arts	Linear	0.1504	0.2130	0.0748	0.0872
	MLP-1	0.1651	0.2358	0.0790	0.0930
	MLP-2	**0.1710**	**0.2419**	0.0822	0.0962
	MLP-3	0.1684	0.2405	**0.0828**	**0.0971**
	MoE	0.1672	0.2361	0.0776	0.0912
Toys	Linear	0.1081	0.1628	0.0453	0.0562
	MLP-1	0.1228	0.1827	0.0521	0.0640
	MLP-2	**0.1307**	**0.1931**	0.0560	0.0683
	MLP-3	0.1291	0.1913	**0.0564**	**0.0687**
	MoE	0.1262	0.1848	0.0520	0.0636
Tools	Linear	0.0777	0.1083	0.0400	0.0460
	MLP-1	0.0888	0.1229	0.0455	0.0522
	MLP-2	0.0918	0.1254	0.0479	0.0546
	MLP-3	**0.0920**	**0.1287**	**0.0484**	**0.0557**
	MoE	0.0896	0.1205	0.0444	0.0505

TABLE 5.8: Performance comparison of whitening methods for WhitenRec+. The best results are in **boldface**, and the second best results are underlined.

Dataset	Method	R@20	R@50	N@20	N@50
Arts	PW	0.1243	0.1801	0.0599	0.0709
	PCA	0.1283	0.1852	0.0633	0.0746
	BN	0.1628	0.2339	0.0789	0.0930
	CD	0.1664	0.2377	0.0798	0.0939
	ZCA	**0.1688**	**0.2403**	**0.0810**	**0.0952**
Toys	PW	0.0843	0.1321	0.0363	0.0457
	PCA	0.0748	0.1194	0.0333	0.0421
	BN	0.1150	0.1739	0.0494	0.0610
	CD	0.1230	0.1847	0.0528	0.0650
	ZCA	**0.1257**	**0.1874**	**0.0537**	**0.0659**
Tools	PW	0.0626	0.0904	0.0322	0.0377
	PCA	0.0625	0.0877	0.0334	0.0384
	BN	0.0799	0.1130	0.0418	0.0483
	CD	**0.0891**	**0.1244**	**0.0465**	**0.0535**
	ZCA	0.0888	0.1236	0.0462	0.0531

adaptor (*i.e.*, MoE) [173]. The results are presented in Table 5.6 and Table 5.7 for WhitenRec+ and DSWRec respectively. It is observed that augmenting the number of layers—and subsequently the model's complexity—enhances performance. The "Linear" configuration was the least effective, emphasizing the necessity of a non-linear activation function for refining pre-trained text embeddings in downstream recommendation applications. Notably, MoE is surpassed by the performance achieved with multiple stacked MLP layers.

5.5.6 Whitening Transformations

Here, we perform experiments to investigate the impact of utilizing different whitening transformations, including both non-parametric and parametric methods. The non-parametric methods examined are PCA, BN, CD, and ZCA, while the parametric method evaluated is PW [57], which employs a linear layer for whitening transformation.

The results of using different whitening methods for WhitenRec+ and DWSRec are presented in Table 5.8 and Table 5.9 respectively. Our results reveal that the parametric method PW performs interior in comparison to the non-parametric methods

TABLE 5.9: Performance comparison of whitening methods for DWSRec. The best results are in **boldface**, and the second best results are underlined.

Dataset	Method	R@20	R@50	N@20	N@50
Arts	PW	0.1532	0.2160	0.0724	0.0848
	PCA	0.1256	0.1821	0.0610	0.0722
	BN	0.1616	0.2308	0.0778	0.0914
	CD	0.1696	0.2415	**0.0823**	**0.0965**
	SVD	**0.1710**	**0.2419**	0.0822	0.0962
Toys	PW	0.1135	0.1692	0.0475	0.0585
	PCA	0.0779	0.1225	0.0340	0.0428
	BN	0.1207	0.1806	0.0519	0.0638
	CD	0.1275	0.1895	0.0549	0.0672
	SVD	**0.1307**	**0.1931**	**0.0560**	**0.0683**
Tools	PW	0.0812	0.1126	0.0414	0.0476
	PCA	0.0624	0.0884	0.0330	0.0381
	BN	0.0866	0.1201	0.0450	0.0516
	CD	0.0891	0.1235	0.0462	0.0530
	SVD	**0.0919**	**0.1260**	**0.0479**	**0.0546**

except PCA. This can be attributed to the fact that a linear layer cannot ensure the transformed output is in fact whitened. Of all the non-parametric whitening methods, PCA exhibits the worst performance due to the issue of stochastic axis swapping, which can impede training progress as noted in [206]. On the other hand, CD and ZCA outperform BN by producing more informative representations through further decorrelation between axes. CD and ZCA show comparable performances across all three datasets.

5.5.7 Efficiency Analysis

In this section, we compare our models, SASRec^W, WhitenRec+, and DWSRec, alongside the leading baselines, UniSRec^T and UniSRec^{T+ID}, focusing on parameter size, training time per epoch (in seconds), and inference time per epoch (in seconds). From the Table 5.10, our analysis yields several insights:

- UniSRec^{T+ID} carries the highest parameter size among the four methods, leading to a training/inference time that is approximately 10% longer than that of UniSRec^T. This increase is due to the integration of item ID embeddings, indicating a potential compromise in model efficiency.

Model	UniSRec^{T}	UniSRec^{T+ID}	SASRec^{W}	WhitenRec+	DWSRec
#Params	2.9M	13.8M	1.4M	1.4M	2.2M
Training Time (s/Epoch)	90	102	63	64	122
Inference Time (s/Epoch)	3.3	3.6	2.8	2.8	7.2

TABLE 5.10: Efficiency Comparison.

- SASRec^{W} and WhitenRec+, leveraging pre-computable whitening, achieve enhanced performance with the least complexity.
- DWSRec has a parameter size similar to UniSRec^{T} but incurs a slightly longer training/inference time in comparison to the baselines due to the implementation of a dual-view-based transformer. Yet, this can be optimized through parallel processing. Using solely text embeddings, SASRec^{W}, WhitenRec+, and DWSRec run with fewer parameters, which reduces over-fitting risks and benefits the cold-start scenarios.

5.6 Summary

In this chapter, we present the frameworks SASRec^{W}, WhitenRec+, and DWSRec to effectively exploit text features of items in sequential recommendation. We contend that relying on text embeddings from pre-trained language models is sub-optimal because such embeddings exist in an anisotropic semantic space, which limits the differentiation among item representations. To address this issue, we propose the SASRec^{W} method, which transforms the anisotropic text embedding distribution into an isotropic distribution through whitening. When an excessive whitening transformation is applied, text embeddings can deviate from their original semantics. Relying solely on the relaxed whitening results in a clustered embedding distribution and sub-optimal performance. To benefit from both ends, we introduce WhitenRec+ and DWSRec, which leverage both fully whitened and relaxed whitened item representations to balance differentiation and similarity. Our experimental results on three public benchmark datasets demonstrate that our proposed methods outperform existing state-of-the-art models for sequential recommendation on both warm and cold settings.

Chapter 6

Multimodal Contrastive Learning for Sequential Recommendation[1]

In Chapter 5, we explore the integration of textual information associated with items by developing a systematic data augmentation strategy that capitalizes on the textual features of items. In Chapter 6, we advance our methodology by incorporating multimodal features alongside the sequential order of user interactions into our data augmentation strategy. This approach offers a robust solution that effectively captures the complexity and richness of real-world user behaviors.

6.1 Overview

Sequential recommendation systems aim to capture users' dynamic preferences based on their historical behaviors, with the objective of predicting the next item of interest [36]. The primary supervision signal for learning the parameters of these models typically derives from users' sequential interactions with items. However, due to the sparsity of user behavior data, sequential recommendation methods relying solely on such data are prone to the problem of data sparsity [56, 58, 125, 218], resulting in suboptimal performance.

In practice, a wealth of multimodal content (*i.e.*, images and text descriptions) associated with items is available, and this has been employed to mitigate the data

[1]The work in this chapter has been published in ACM Transactions on Recommender Systems 2024 [114].

sparsity issue in constructing traditional recommendation systems. For example, studies [66, 68] utilize item multimodal content as a regularization factor and incorporate it into collaborative filtering frameworks. More recent research [55, 80, 81] deploys graph neural networks to discover hidden links between different modalities, thereby deepening the understanding of users' preferences. Multimodal data is proven to be effective in enhancing recommendation performance.

However, although considerable advancements have been made by these methods, they are not designed for sequential recommendation and fall short in modeling the dynamic nature of users' preferences for multimodal information over time. Currently, only a handful of studies exploit multimodal data for sequential recommendation. For example, MV-RNN [219] merges multimodal features at its input and employs a recurrent structure to dynamically track users' interests. MML [218] implements a modality-specific meta-learner to identify the sequential pattern from different modalities and adaptively merges their predictions using a learnable fusion layer.

Despite these efforts, current methods fail to effectively explore and capture correlations among behavior sequences of users and items across different modalities. Specifically, 1) existing works usually perform modality fusion (*i.e.*, concatenation, addition, or attention) at the item level, neglecting the correlation among sequence representations in different modalities and the correlation between sequence representations with items in different modality spaces; 2) existing works rely on a supervised learning framework that utilizes item prediction loss (*i.e.*, cross-entropy loss) to learn the entire model. This approach tends to overemphasize final performance while insufficiently capturing the association or fusion between multimodal data and sequence data.

In this work, we explore multimodal pre-training [128, 220–223] in the context of sequential recommendation with the aim of enhancing multimodality fusion and utilization of multimodal information. The core idea of multimodal pre-training involves leveraging self-supervised signals to aggregate and align visual and textual information. Pre-training enables efficient exploitation of unlabeled data space, captures intrinsic data correlations, and ultimately improves the performance of downstream tasks. Although prevalent in computer vision, multimodal pre-training is relatively under-explored in sequential recommendation. Our objective is to integrate the advantages of multimodal pre-training into sequential recommendation

representation learning. This is a non-trivial work as pre-training for sequential recommendation fundamentally differs from that for computer vision tasks. Multimodal pre-training for sequential recommendation predominantly focuses on modeling users' evolving preferences for multimodal information over time, rather than aligning images and text.

To enhance multimodality fusion and more effectively harness multimodal information in sequential recommendation, we propose a pre-training approach, Multimodal Pre-training for Sequential Recommendation (MP4SR). MP4SR utilizes contrastive losses to capture the correlation among behavior sequences of users and items across different modalities. It comprises three key components: multimodal feature extraction, a backbone network Multimodal Mixup Sequence Encoder (M^2SE), and pre-training tasks. Initially, we tokenize each item image into multiple text keywords using a language-image pre-trained model [128], and then apply the Sentence-BERT model [129] to extract initial text and image features of items. This step eliminates discrepancies between the textual and visual modalities while preserving meaningful information from images and discarding redundant information. Next, M^2SE integrates different user modality sequences via a complementary sequence mixup strategy, subsequently processed by a Transformer to obtain mix-modality sequence representations. Lastly, we employ contrastive learning to identify modality interactions at both sequence-to-sequence and sequence-to-item levels. Specifically, we use a *modality-specific next item prediction loss* to capture the correlation between a mix-modality sequence and the subsequent item within each modality space, and a *cross-modality contrastive learning loss* to calibrate the discrepancies of mix-modality sequence representations across different modality spaces.

We conduct extensive experiments on three real-world datasets to evaluate the effectiveness of MP4SR. Our experimental results show that MP4SR outperforms state-of-the-art approaches for sequential recommendation under both normal and cold-start settings. We also show that restricting the feasible starting points in parameter space to those minimizing the self-supervised pre-training criterion yields similar effects to a good regularizer on the parameters, thereby enhancing the recommendation performance.

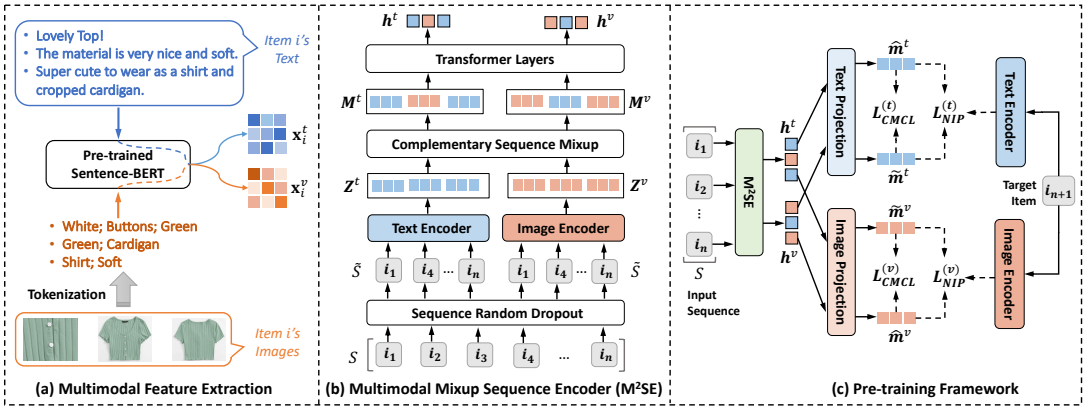


FIGURE 6.1: a) The multimodal feature extraction module used to obtain initial multimodal features of items; b) The structure of the proposed multimodal mixup sequence encoder that fuses items' multimodal content with users' behavior sequence; c) The workflow of the proposed pre-training framework, where $\mathcal{S}$ is the input sequence and i_{n+1} is the target item.

6.2 Methodology

In this section, we introduce the main components of MP4SR, including multimodal feature extraction, backbone network M^{2}SE, pre-training objectives, and fine-tuning objectives.

6.2.1 Notations

Let $\mathcal{I}$ denote the set of items, and $\mathcal{S} = \{i_1, i_2, \cdots, i_n\}$ denote a user behavior sequence, where n items are sorted in chronological order based on the interaction timestamp. In this work, we consider the text and image content of items to build the model. For each item i, it is associated with a chunk of text descriptions that is split into sentences as $\mathcal{T}_i = \{t_1^i, t_2^i, \cdots, t_{|\mathcal{T}_i|}^i\}$, and a set of images $\mathcal{V}_i = \{v_1^i, v_2^i, \cdots, v_{|\mathcal{V}_i|}^i\}$, where $|\mathcal{T}_i|$ and $|\mathcal{V}_i|$ denote the number of sentences and images, respectively.

6.2.2 Multimodal Feature Extraction

Figure 6.1(a) shows the workflow using pre-trained models to obtain the initial text and image features of items to eliminate the modality gap between the text and image embeddings.

6.2.2.1 Text Feature Extraction

For each sentence in $\mathcal{T}_i$, we feed it into the pre-trained Sentence-BERT [129] to obtain its latent representation. The initial text feature $\mathbf{x}_i^t$ of item i is obtained by stacking representations of all the sentences in $\mathcal{T}_i$ as follows,

$$\mathbf{x}_i^t = stack\big[\text{BERT}(t_1^i), \text{BERT}(t_2^i), \cdots, \text{BERT}(t_{|\mathcal{T}_i|}^i)\big], \tag{6.1}$$

where $\mathbf{x}_i^t \in \mathbb{R}^{|\mathcal{T}_i| \times d}$, $stack[,]$ denotes stacking multiple vectors into a matrix, and d is the embedding dimension.

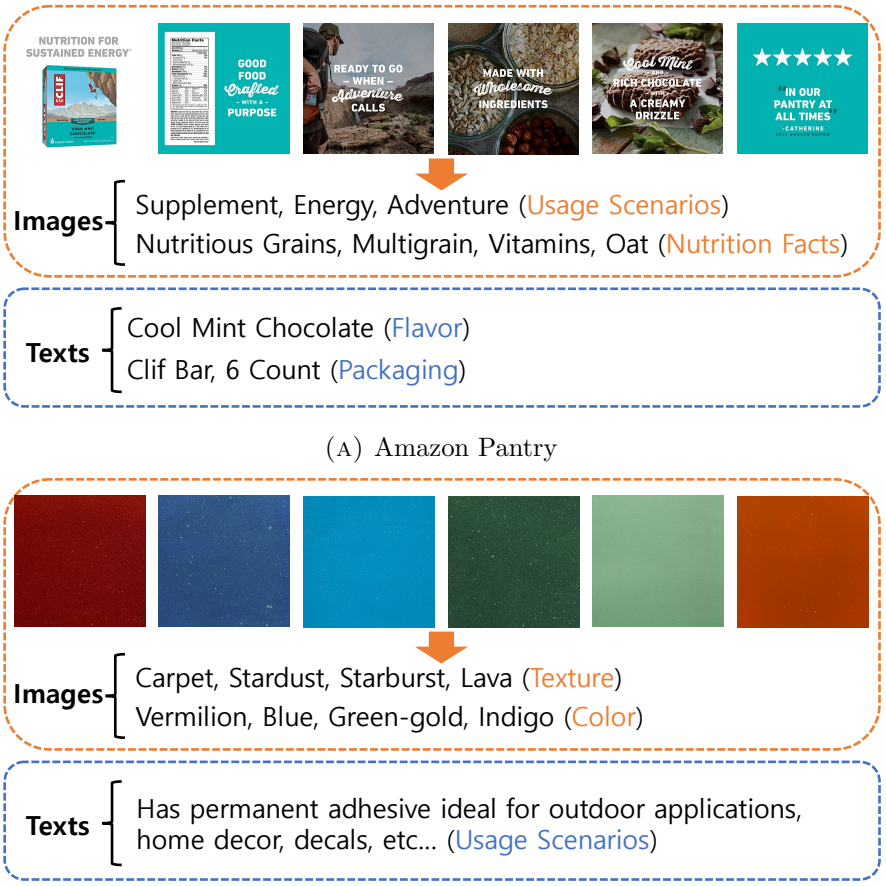
6.2.2.2 Image Feature Extraction

Inspired by [224], we use a pre-trained language-image model, *i.e.*, CLIP [128], to describe each image by text tokens in order to eliminate the modality gap between text and image representations and remove irrelevant information from images. To capture the key visual information of an image, N most relevant text tokens are retained based on their similarities to the image. Then, we obtain the initial feature $\mathbf{v}_\ell^i$ for an item image $v_\ell^i \in \mathcal{V}_i$, by concatenating these text tokens as a sentence and feeding it into the same pre-trained Sentence-BERT model. The initial image feature $\mathbf{x}_i^v$ of item i can be obtained by stacking the features of all images in $\mathcal{V}_i$ as follows,

$$\begin{aligned} f(w) &= \text{sim}(\text{CLIP}(v_\ell^i), \text{CLIP}(w)) \quad \forall w \in \mathcal{D}, \\ \mathbf{v}_\ell^i &= \text{BERT}(concat(\text{TopN}(\{f(w_1), \cdots, f(w_{|\mathcal{D}|})\}, N))), \\ \mathbf{x}_i^v &= stack[\mathbf{v}_1^i, \mathbf{v}_2^i, \cdots, \mathbf{v}_{|\mathcal{V}_i|}^i], \end{aligned} \tag{6.2}$$

where $\mathbf{x}_i^v \in \mathbb{R}^{|\mathcal{V}_i| \times d}$, w is a text token in the word dictionary $\mathcal{D}$, and $|\mathcal{D}|$ denotes the size of the dictionary. $\text{sim}(\cdot)$ is to compute the cosine similarity between the embedding of an image v_ℓ^i and a word w obtained by the CLIP model. $\text{TopN}(\cdot)$ function selects N words that have the highest similarities with the image. $concat(\cdot)$ is the operation of concatenating N words into one sentence. Note that $\mathbf{x}_i^t$ and $\mathbf{x}_i^v$ are derived during the data pre-processing stage.

To verify text tokens retrieved from images using the pre-trained language-image model, we select two representative items from the Amazon Pantry and Arts



(A) Amazon Pantry

(B) Amazon Arts

FIGURE 6.2: Two examples of converting images of an item into text tokens. Items are retrieved from the Amazon Pantry and Arts dataset. Text tokens are generated using CLIP [128].

dataset for analysis (Figure 6.2). The left item is an energy bar to provide carbohydrates and protein, with incomplete information in the seller's summary. By generating text tokens from images, key information, including nutritional facts and usage modes, can be obtained. The right item is a carpet with multiple colors, and the merchant's product descriptions lack information on the carpet's appearance. By using text tokens from images, we can describe the characteristics of the carpet based on its colors and designs. Compared to the image feature extracted from pre-trained image encoders (*e.g.*, ResNet [55, 81]), word tokens converted from images can discard irrelevant information to achieve better recommendation performance.

6.2.3 Multimodal Mixup Sequence Encoder

To encode user sequences with multimodal features extracted from items, we explain the backbone network, *i.e.*, M^{2}SE. The structure of M^{2}SE is shown in Figure 6.1(b). Observe that M^{2}SE includes four main components: sequence random dropout, text and image encoders, complementary sequence mixup, and transformer layers.

6.2.3.1 Sequence Random Dropout

To help the model achieve better generalization performance, M^{2}SE randomly drops a portion of items from $\mathcal{S}$ with a drop ratio ρ for a user behavior sequence $\mathcal{S}$ [53]. The obtained sub-sequence after the random dropout operation is denoted by $\widetilde{\mathcal{S}}$. ρ is fixed during the pre-training stage.

6.2.3.2 Text and Image Encoders

These two encoders are used to adapt the initial modality features of items obtained from the pre-trained language model to learn users' sequential behaviors. Both encoders share the same structure, including an attention layer and a Mixture-of-Expert (MoE) architecture [173].

In the text encoder, each item $i \in \widetilde{\mathcal{S}}$ is represented by its initial textual feature $\mathbf{x}_i^t$. The attention layer is composed of two linear transformations to fuse i's sentence-level embeddings as follows,

$$\begin{aligned} \alpha^t &= \text{softmax}\big((\mathbf{x}_i^t \mathbf{W}_1^t + \mathbf{b}_1^t)\mathbf{W}_2^t + b_2^t\big), \\ \mathbf{m}_i^t &= \sum_{j=1}^{|\mathcal{T}_i|} \alpha_j^t \mathbf{x}_i^t[j,:], \end{aligned} \tag{6.3}$$

where $\mathbf{W}_1^t \in \mathbb{R}^{d \times d_a}$, $\mathbf{W}_2^t \in \mathbb{R}^{d_a}$, $\mathbf{b}_1^t \in \mathbb{R}^{d_a}$, and $b_2^t \in \mathbb{R}$ are learnable parameters. d_a is the attention dimension size. α_j^t is the j-th element of α^t, and $\mathbf{x}_i^t[j,:]$ denotes the j-th row of feature matrix $\mathbf{x}_i^t$. Then, MoE is used to increase the model's capacity for adapting the fused modality representation $\mathbf{m}_i^t$. Each expert in MoE consists of a linear transformation, followed by a dropout layer and a normalization layer.

Let $E_k(\mathbf{m}_i^t) \in \mathbb{R}^{d_0}$ denote the output of the k-th expert network, and $\mathbf{g}^t \in \mathbb{R}^O$ is the output of the gating network as follows,

$$\begin{aligned} E_k(\mathbf{m}_i^t) &= \text{LayerNorm}(\text{Dropout}(\mathbf{m}_i^t \mathbf{W}_k^t)), \\ \mathbf{g}^t &= \text{softmax}(\mathbf{m}_i^t \mathbf{W}_3^t), \end{aligned} \tag{6.4}$$

where $\mathbf{W}_3^t \in \mathbb{R}^{d \times O}$ and $\mathbf{W}_k^t \in \mathbb{R}^{d \times d_0}$ are learnable parameters, O is the number of experts, and d_0 is the dimension of the hidden embedding. Then, the output of MoE for item i is formulated as follows,

$$\mathbf{z}_i^t = \sum_{k=1}^{O} g_k^t E_k(\mathbf{m}_i^t), \tag{6.5}$$

where $\mathbf{z}_i^t \in \mathbb{R}^{d_0}$, and g_k^t is the weight derived from k-th gating router. Here, we omit bias terms in the equation for simplicity. The outputs of MoE network for all items in $\widetilde{\mathcal{S}}$ are stacked to form the output of the text encoder, which is denoted by $\mathbf{Z}^t = stack[\mathbf{z}_1^t, \mathbf{z}_2^t, \cdots, \mathbf{z}_{|\widetilde{\mathcal{S}}|}^t]$.

Similarly, in the image encoder, each item in $\widetilde{\mathcal{S}}$ is represented by its image feature $\mathbf{x}_i^v$. The output of the image encoder is denoted by $\mathbf{Z}^v = stack[\mathbf{z}_1^v, \mathbf{z}_2^v, \cdots, \mathbf{z}_{|\widetilde{\mathcal{S}}|}^v]$, where $\mathbf{z}_i^v$ is the output of the MoE network for the i-th item in $\widetilde{\mathcal{S}}$.

6.2.3.3 Complementary Sequence Mixup

To alleviate the representation discrepancy between two different modality sequences, we propose a complementary sequence mixup method that mixes up text representations and image representations in a complementary manner. Specifically, we define a mixup ratio p between 0 to 0.5, which is randomly generated during model training. For each item in $\widetilde{\mathcal{S}}$, we swap its embedding in $\mathbf{Z}^t$ and $\mathbf{Z}^v$ with probability p and generate two mix-modality sequence embeddings $\mathbf{M}^t$ and $\mathbf{M}^v$. The definition of $p \leq 0.5$ ensures the generated mix-modality sequence embedding dominates with information from the same modality. In this case, $\mathbf{M}^t$ and $\mathbf{M}^v$ complement each other in terms of the modality choice for each item in the sequence.

6.2.3.4 Transformer Layers

The Transformer [196] structure is used to further encode $\mathbf{M}^t$ and $\mathbf{M}^v$. We first add positional encodings to $\mathbf{M}^t$ and $\mathbf{M}^v$, and then feed the summed embeddings into L Transformer layers. Note that each Transformer layer consists of a multi-head self-attention sub-layer and a point-wise feed-forward network. Let $\mathbf{H}_L^t$ and $\mathbf{H}_L^v$ denote the output of the L-th Transformer layer based on $\mathbf{M}^t$ and $\mathbf{M}^v$, respectively. Following [51], we use the last rows in $\mathbf{H}_L^t$ and $\mathbf{H}_L^v$ as two mix-modality representations of the input sequence, which are denoted by $\mathbf{h}^t$ and $\mathbf{h}^v$.

6.2.4 Pre-training Objectives

To better capture the correlation between representations across different modalities, we propose two optimization objectives, *i.e.*, modality-specific next item prediction and cross-modality contrastive learning, based on mix-modality sequence representations for pre-training the backbone model. The workflow in the pre-training phase is shown in Figure 6.1(c).

Let $\mathcal{B} = \{(\mathcal{S}_j, i_j)\}_{j=1}^{|\mathcal{B}|}$ denote a batch of pre-training data, where $\mathcal{S}_j$ denotes a user's behavior sequence and i_j is her next interaction item after $\mathcal{S}_j$. With M^{2}SE, we can obtain two mix-modality sequence representations $\mathbf{h}_j^t$ and $\mathbf{h}_j^v$ for $\mathcal{S}_j$. As $\mathbf{h}_j^t$ and $\mathbf{h}_j^v$ are obtained by mixing up modalities, we first use two linear transformations to map them into the text feature space and image feature space for calculating the pre-training losses, respectively,

$$\begin{aligned} \widehat{\mathbf{e}}_j^t &= \mathbf{h}_j^t \mathbf{W}_t + \mathbf{b}_t, & \widetilde{\mathbf{e}}_j^t &= \mathbf{h}_j^v \mathbf{W}_t + \mathbf{b}_t, \\ \widehat{\mathbf{e}}_j^v &= \mathbf{h}_j^v \mathbf{W}_v + \mathbf{b}_v, & \widetilde{\mathbf{e}}_j^v &= \mathbf{h}_j^t \mathbf{W}_v + \mathbf{b}_v, \end{aligned} \tag{6.6}$$

where $\mathbf{W}_t, \mathbf{W}_v \in \mathbb{R}^{d_0 \times d_0}$ and $\mathbf{b}_t, \mathbf{b}_v \in \mathbb{R}^{d_0}$ are learnable parameters. Motivated by the success of contrastive learning in model pre-training, we define the pre-training objective functions in a contrastive manner.

6.2.4.1 Modality-specific Next Item Prediction

Modality-specific Next Item Prediction (NIP) aims to predict the next item based on the mix-modality sequence representations. For each $(\mathcal{S}_j, i_j)$ pair, $\mathcal{S}_j$ is the

input sequence and i_j is the target item. Thus, in the text feature space, we pair $\widehat{\mathbf{e}}_j^t$ and $\widetilde{\mathbf{e}}_j^t$ with the i_j's text embedding $\mathbf{z}_j^t$ obtained by the text encoder as a positive sample, and pair $\widehat{\mathbf{e}}_j^t$ and $\widetilde{\mathbf{e}}_j^t$ with the text embeddings of other items $\{i_{j'}|j' \neq j, 1 \leq j' \leq |\mathcal{B}|\}$ from $\mathcal{B}$ as negative samples. The next item prediction loss defined in the text feature space is as follows,

$$\mathcal{L}_{\text{NIP}}^{(t)} = -\sum_{j=1}^{|\mathcal{B}|} \log \frac{f(\widehat{\mathbf{e}}_j^t, \mathbf{z}_j^t) + f(\widetilde{\mathbf{e}}_j^t, \mathbf{z}_j^t)}{\sum_{j'=1}^{|\mathcal{B}|}[f(\widehat{\mathbf{e}}_j^t, \mathbf{z}_{j'}^t) + f(\widetilde{\mathbf{e}}_j^t, \mathbf{z}_{j'}^t)]}, \tag{6.7}$$

where $f(\mathbf{s}, \mathbf{z}) = \exp(\text{sim}(\mathbf{s}, \mathbf{z})/\tau)$, and τ is a temperature hyper-parameter. Similarly, we can define the next item prediction loss $\mathcal{L}_{\text{NIP}}^{(v)}$ in the image feature space as follows,

$$\mathcal{L}_{\text{NIP}}^{(v)} = -\sum_{j=1}^{|\mathcal{B}|} \log \frac{f(\widehat{\mathbf{e}}_j^v, \mathbf{z}_j^v) + f(\widetilde{\mathbf{e}}_j^v, \mathbf{z}_j^v)}{\sum_{j'=1}^{|\mathcal{B}|}[f(\widehat{\mathbf{e}}_j^v, \mathbf{z}_{j'}^v) + f(\widetilde{\mathbf{e}}_j^v, \mathbf{z}_{j'}^v)]}. \tag{6.8}$$

6.2.4.2 Cross-Modality Contrastive Learning

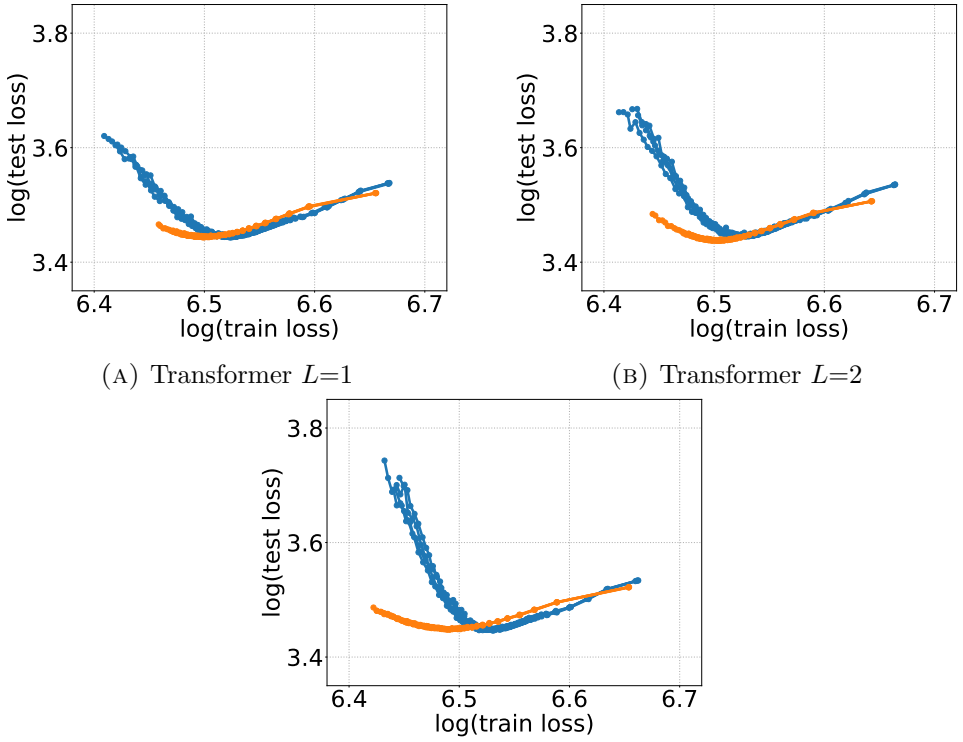
To capture the semantic relationship between different modality sequences, we develop a Cross-Modality Contrastive Loss (CMCL). Specifically, the complementary mix-modality sequence representations mapped to the same feature space, *e.g.*, ($\widehat{\mathbf{e}}_j^t$, $\widetilde{\mathbf{e}}_j^t$) and ($\widehat{\mathbf{e}}_j^v$, $\widetilde{\mathbf{e}}_j^v$), are paired as positive samples, while randomly-selected samples in the training batch are paired as negative samples. Following [187], CMCL for the text space is defined in a symmetric contrastive way as follows,

$$\ell(\widehat{\mathbf{e}}_j^t, \widetilde{\mathbf{e}}_j^t) = \log \frac{f(\widehat{\mathbf{e}}_j^t, \widetilde{\mathbf{e}}_j^t)}{\sum\limits_{j'=1}^{|\mathcal{B}|} f(\widehat{\mathbf{e}}_j^t, \widetilde{\mathbf{e}}_{j'}^t) + \sum\limits_{j'=1, j' \neq j}^{|\mathcal{B}|} f(\widehat{\mathbf{e}}_j^t, \widehat{\mathbf{e}}_{j'}^t)},$$

$$\mathcal{L}_{\text{CMCL}}^{(t)} = -\frac{1}{2}\sum_{j=1}^{|\mathcal{B}|}(\ell(\widehat{\mathbf{e}}_j^t, \widetilde{\mathbf{e}}_j^t) + \ell(\widetilde{\mathbf{e}}_j^t, \widehat{\mathbf{e}}_j^t)). \tag{6.9}$$

Similarly, CMCL in the image space is defined as follows,

$$\mathcal{L}_{\text{CMCL}}^{(v)} = -\frac{1}{2}\sum_{j=1}^{\mathcal{B}}(\ell(\widehat{\mathbf{e}}_j^v, \widetilde{\mathbf{e}}_j^v) + \ell(\widetilde{\mathbf{e}}_j^v, \widehat{\mathbf{e}}_j^v)). \tag{6.10}$$



(A) Transformer L=1 (B) Transformer L=2

(C) Transformer L=3

FIGURE 6.3: Evolution without pre-training (blue) and with pre-training (orange) on **Pantry** dataset of the log of the test loss plotted against the log of the train loss as training proceeds. Each group has 5 curves representing a different initialization. During training, the trajectories move from right (high error) to left (low error) due to the decrease in training error.

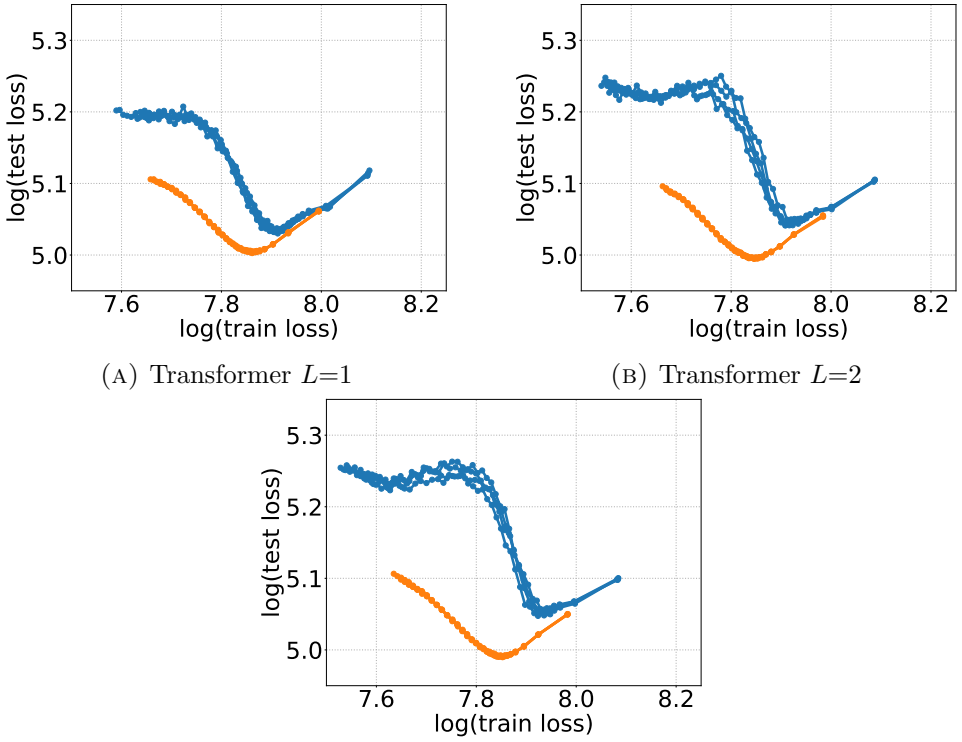
The overall loss function for model pre-training is formulated as follows,

$$\mathcal{L}_{\text{pre-train}} = \mathcal{L}_{\text{NIP}}^{(t)} + \mathcal{L}_{\text{NIP}}^{(v)} + \lambda(\mathcal{L}_{\text{CMCL}}^{(t)} + \mathcal{L}_{\text{CMCL}}^{(v)}), \tag{6.11}$$

where λ is a hyper-parameter to balance these two groups of losses.

6.2.4.3 Discussion

Inspired by [225], we plot test loss against train loss along the optimization trajectory in parameter space. Figure 6.3 and Figure 6.4 show five of these curves without pre-training (blue), originating from a random initialization point in parameter space, and five initiated from pre-trained parameters (orange), for Pantry and Office datasets respectively. The experiments are performed for MP4SR with 1, 2, and 3 Transformer layers. We observe that, at the same level of train loss,



(A) Transformer L=1 (B) Transformer L=2

(C) Transformer L=3

FIGURE 6.4: Evolution without pre-training (blue) and with pre-training (orange) on Office of the log of the test loss plotted against the log of the train loss as training proceeds. Each group has 5 curves representing a different initialization.

pre-trained models consistently exhibit a lower test cost than randomly initialized models when training proceeds to convergence. This finding suggests that pre-training tasks serve as effective regularizers for parameters used for modeling user sequences based on multimodal data. Additionally, as the number of layers increases, the impact of generalization becomes more pronounced, which is evident from the downward shift of the orange line.

6.2.5 Fine-tuning for Sequential Recommendation

We treat sequential recommendation as a supervised classification problem [49, 51] and use the cross-entropy loss incorporating all modality sequences for model fine-tuning.

Let $\widetilde{\mathcal{B}}$ denote a batch of fine-tuning data. For each $(\mathcal{S}, i) \in \widetilde{\mathcal{B}}$, i is the user's next interaction item after her interaction sequence $\mathcal{S}$. From the classification

perspective, i can be treated as the target label of the input sequence $\mathcal{S}$. In the fine-tuning stage, we disable the sequence random dropout and complementary sequence mixup operations in the pre-trained M^{2}SE network, by setting the dropout ratio ρ and sequence mixup ratio p to 0. Moreover, we also incorporate the ID embeddings $\mathbf{E}_{\mathcal{S}}$ of items in the sequence $\mathcal{S}$ with its text and image representations $\mathbf{M}^t$ and $\mathbf{M}^v$ using element-wise summation, and then feed the summed embeddings into the Transformer layers to obtain the sequence embeddings $\mathbf{h}^t$ and $\mathbf{h}^v$. Then, for the sequence $\mathcal{S}$, the predicted probability distribution $\widehat{\mathbf{y}}^{(\mathcal{S})}$ of its potential target labels (*i.e.*, items) is defined as follows,

$$\widehat{\mathbf{y}}^{(\mathcal{S})} = \text{softmax}\big(\mathbf{h}^t(\mathbf{F}^t + \mathbf{E})^\top + \mathbf{h}^v(\mathbf{F}^v + \mathbf{E})^\top\big), \tag{6.12}$$

where $\widehat{\mathbf{y}}^{(\mathcal{S})} \in \mathbb{R}^{|\mathcal{I}|}$, $\mathbf{E} \in \mathbb{R}^{|\mathcal{I}| \times d_0}$ denotes the ID embedding matrix of all items. $\mathbf{F}^t, \mathbf{F}^v \in \mathbb{R}^{|\mathcal{I}| \times d_0}$ denote the text and image modality embedding matrices of all items, which are obtained by the text encoder and image encoder in M^{2}SE. The loss function for model fine-tuning is defined as follows,

$$\mathcal{L}_{\text{finetune}} = -\sum_{(\mathcal{S},i)\in\widetilde{\mathcal{B}}} \log\big(\widehat{\mathbf{y}}^{(\mathcal{S})}(i)\big), \tag{6.13}$$

where $\widehat{\mathbf{y}}^{(\mathcal{S})}(i)$ denotes the predicted probability for the ground-truth label i of the sequence $\mathcal{S}$. By minimizing Eqn. (6.13), we fine-tune the parameters of the pre-trained M^{2}SE network as well as the ID embedding of items.

6.2.6 Complexity Analysis

The time complexity of MP4SR primarily arises from the text and image encoders, attention-based transformer layers, and contrastive learning-based pre-training objectives. Considering a batch of samples $\mathcal{B}$, each contributes a time complexity of $\mathcal{O}(|\mathcal{B}||\mathcal{S}|Odd_0)$, $\mathcal{O}(|\mathcal{B}||\mathcal{S}|^2d_0 + |\mathcal{B}||\mathcal{S}|d_0^2)$, and $\mathcal{O}(|\mathcal{B}|^2d_0)$ respectively. Consequently, the aggregate time complexity is $\mathcal{O}(|\mathcal{B}||\mathcal{S}|Odd_0 + |\mathcal{B}||\mathcal{S}|^2d_0 + |\mathcal{B}||\mathcal{S}|d_0^2)$.

TABLE 6.1: Statistics of the experimental datasets. "Avg. n" denotes the average length of interaction sequences.

Datasets	#Users	#Items	#Inter.	Avg. n
Pantry	13,614	7,670	131,311	9.65
Arts	32,216	52,557	264,465	8.21
Office	68,224	59,705	527,209	7.73

6.3 Experiments

6.3.1 Experimental Settings

6.3.1.1 Datasets

The experiments are conducted on the Amazon review dataset [188], which provides the multimodal information of items. We use three "5-core" subsets for experimental evaluation, *i.e.*, *Pantry*, *Arts*, and *Office*. Following [51, 159], we convert each rating into an implicit feedback record. On each dataset, we group interactions by users and construct the interaction sequence for each user by sorting her interactions in chronological order. The statistics of the pre-processed experimental datasets are summarized in Table 6.1.

6.3.1.2 Evaluation Settings

Following [50, 51], we apply the *leave-one-out* strategy to evaluate the performance of recommendation models in both pre-training and fine-tuning stages. Specifically, for each user, the last item of her interaction sequence is used for testing, the second last item is used for validation, and the remaining items are used for model training. The performance of a recommendation model is evaluated by two widely used metrics, *i.e.*, Recall@K and Normalized Discounted Cumulative Gain@K (respectively denoted by R@K and N@K). K is empirically set to 10 and 20. All evaluation metrics are computed on the whole candidate item set without negative sampling.

6.3.1.3 Baseline Methods

We compare the proposed model with four groups of baseline methods:

1) General Recommendation Model:

- **LightGCN** [226] is one of the representative GNNs-based recommendation methods.

2) Multimodal Recommendation Models:

- **GRCN** [80] is a graph-based multimodal recommendation model that refines the user-item interaction graph by pruning noisy edges;
- **DualGNN** [82] explicitly models the user's attention over different modalities.

3) Sequential Recommendation Models:

- **SASRec** [49] is a directional self-attention method for next item prediction;
- **SINE** [227] uses a sparse-interest module that adaptively infers a sparse set of concepts and outputs multiple embeddings for each user;
- **CL4SRec** [53] designs three data augmentation approaches to extract self-supervised signals to improve the sequential recommendation performance;

4) Sequential Recommendation Models with Side Information:

- **MV-RNN** [219] is a multimodal sequential recommendation model that combines multimodal features at its input and applies a recurrent structure to dynamically capture users' interests;
- **FDSA** [56] models the transition patterns between items as well as features by separate self-attention blocks. We feed both text and image features into the model for fair comparison;
- **S^3-Rec** [51] devises four self-supervised learning objectives based on the mutual information maximization principle;

- **DIF-SR** [54] decouples attribute information and item representation from the calculation of attention;
- **SASRec+** integrates multimodal information of items with the SASRec. The model encodes modality features using the same text/image encoder as MP4SR, sums them with item ID embeddings, and feeds the output into SASRec to generate recommendations.

6.3.1.4 Implementation Details

The proposed method is implemented by Pytorch [216] and an open-source recommendation framework, RecBole [217]. The Adam optimizer [192] is used to learn model parameters. Following [51], we set the maximum sequence length to 50. Our training phase consists of two stages: pre-training and fine-tuning. The learned parameters from the pre-training stage are used to initialize the M^2SE network in the fine-tuning stage. Both the pre-training and fine-tuning are performed on the same dataset to obtain the final recommendation results. More implementation details can be found in the supplementary material.

In the multimodal feature extraction stage of MP4SR, the pre-trained Sentence-BERT model maps every sentence of text descriptions or a group of word tokens extracted from an image into a 768-dimensional dense vector, *i.e.*, $d = 768$. For each item, we consider up to 10 sentences and 10 images.

For pre-training MP4SR, we set the learning rate to 0.001, the batch size to 1024, and the number of experts O to 8 on all datasets. In addition, we set ρ, τ, and λ to 0.2, 0.07, and 0.01, respectively. The attention dimension d_a and embedding dimension d_0 are fixed to 64. The proposed model is pre-trained for 300 epochs.

For fine-tuning MP4SR and training all baseline methods, we apply grid search to identify the best hyper-parameter settings based on the validation data for all methods. The search space is as follows: learning rate in $\{0.0001, 0.0005, 0.001\}$, batch size in $\{256, 512, 1024\}$, and weight decay in $\{0.0001, 0.0005, 0.001\}$. For a fair comparison, the hyper-parameters of Transformer layers are kept identical for MP4SR and transformer-based baselines (*i.e.*, SASRec, S^3-Rec, and DIF-SR). Specifically, the number of attention heads and the number of self-attention blocks are set to 2. The remaining hyper-parameters for baseline methods follow the

original papers. Additionally, we adopt an early stopping strategy, *i.e.*, we apply premature stopping if R@20 on the validation data does not increase for 10 epochs.

6.3.2 Performance Comparison

We summarize the overall performance comparison results in Table 6.2, from which we have the following observations.

Firstly, the multimodal recommendation methods (*i.e.*, GRCN and DualGNN) consistently outperform the general recommendation model (*i.e.*, LightGCN). This suggests that leveraging the multimodal information of items can effectively enhance recommendation performance.

Secondly, sequential recommendation models generally perform better than non-sequential recommendation models, by capturing users' sequential behavior patterns. Notably, an exception is observed in the Pantry dataset, where the non-sequential model GRCN, which utilizes multimodal data, achieves better performance than sequential baseline methods. This may be attributed to the fact that the multimodal features of items in this dataset are more informative than those in the other two datasets. Meanwhile, S^3-Rec usually surpasses other sequential recommendation baselines, highlighting the effectiveness of using self-supervised signals and side information for pre-training sequential recommendation models.

Thirdly, MP4SR and SASRec+ both outperform baseline methods by leveraging the designed multimodal data encoders and capturing the sequential behavior patterns of users to enhance recommendation accuracy.

Lastly, the proposed MP4SR model consistently outperforms all baseline methods by a significant margin. This is attributed to the utilization of two pre-training objectives to capture the correlation of multimodal data with user behaviors, thereby improving the generalization capabilities of sequential recommendation models and resulting in the best overall performance.

TABLE 6.2: The overall performance achieved by different methods. The best results are in **boldface**, and the second best results are underlined. * denotes MP4SR surpasses the best baseline using a paired t-test ($p < 0.01$).

Dataset	Model	R@10	R@20	N@10	N@20
Pantry	LightGCN	0.0460	0.0774	0.0236	0.0315
	GRCN	0.0552	0.0856	0.0289	0.0366
	DualGNN	0.0485	0.0739	0.0254	0.0318
	SASRec	0.0457	0.0722	0.0204	0.0271
	SINE	0.0534	0.0873	0.0243	0.0329
	CL4SRec	0.0487	0.0796	0.0236	0.0314
	MV-RNN	0.0276	0.0467	0.0134	0.0184
	FDSA	0.0357	0.0588	0.0194	0.0252
	S^3-Rec	0.0535	0.0845	0.0257	0.0335
	DIF-SR	0.0473	0.0736	0.0219	0.0284
	SASRec+	0.0600	0.0934	0.0298	0.0382
	MP4SR	**0.0673***	**0.1040***	**0.0321***	**0.0414***
Arts	LightGCN	0.0726	0.0967	0.044	0.0501
	GRCN	0.0741	0.0999	0.0448	0.0513
	DualGNN	0.0788	0.1033	0.0495	0.0557
	SASRec	0.0910	0.1125	0.0509	0.0563
	SINE	0.0935	0.1237	0.0491	0.0567
	CL4SRec	0.0899	0.1162	0.0484	0.0550
	MV-RNN	0.0446	0.0661	0.0232	0.0283
	FDSA	0.0772	0.0948	0.0545	0.0589
	S^3-Rec	0.0961	0.1250	0.0546	0.0619
	DIF-SR	0.0899	0.1126	0.0510	0.0567
	SASRec+	0.1099	0.1430	0.0616	0.0699
	MP4SR	**0.1184***	**0.1570***	**0.0637***	**0.0735***
Office	LightGCN	0.0518	0.0752	0.0281	0.0339
	GRCN	0.0714	0.0911	0.046	0.0509
	DualGNN	0.0661	0.0843	0.0431	0.0477
	SASRec	0.1025	0.1222	0.0617	0.0667
	SINE	0.1059	0.1305	0.0618	0.0680
	CL4SRec	0.1016	0.1256	0.0602	0.0662
	MV-RNN	0.0416	0.0641	0.0210	0.0266
	FDSA	0.0832	0.0997	0.0616	0.0657
	S^3-Rec	0.1027	0.1254	0.0641	0.0698
	DIF-SR	0.1039	0.1241	0.0620	0.0671
	SASRec+	0.1060	0.1316	0.0652	0.0716
	MP4SR	**0.1206***	**0.1480***	**0.0797***	**0.0866***

TABLE 6.3: The performance of cold-items achieved by CLCRec, MASR, $\text{MP4SR}_{\text{w/o Pre-train}}$, and MP4SR.

Dataset	Model	R@10	R@20	N@10	N@20
Pantry	MASR	0.0111	0.0119	0.0064	0.0066
	CLCRec	0.0166	0.0251	0.0081	0.0103
	$\text{MP4SR}_{\text{w/o Pre-train}}$	0.0257	0.0337	0.0119	0.0139
	MP4SR	**0.0360**	**0.0491**	**0.0176**	**0.0208**
Arts	MASR	0.0136	0.0165	0.0080	0.0090
	CLCRec	0.0178	0.0239	0.0101	0.0116
	$\text{MP4SR}_{\text{w/o Pre-train}}$	0.0314	0.0455	0.0145	0.0181
	MP4SR	**0.0403**	**0.0552**	**0.0191**	**0.0229**
Office	MASR	0.0079	0.0094	0.0050	0.0054
	CLCRec	0.0094	0.0120	0.0049	0.0056
	$\text{MP4SR}_{\text{w/o Pre-train}}$	0.0128	0.0183	0.0061	0.0074
	MP4SR	**0.0235**	**0.0312**	**0.0117**	**0.0136**

6.3.3 Cold-start Performance

To validate the effectiveness of our model for the cold-start recommendation, we include the following methods for evaluation alongside $\textbf{MP4SR}_{\text{w/o Pre-train}}$ and **MP4SR**:

- **CLCRec** [214]: this method explores the mutual dependency between item multimodal features and collaborative representations to alleviate the cold-start item problem.
- **MASR** [228]: authors construct two memory banks to store historical user sequences and a retriever-copy network to search for similar sequences to enhance the recommendation performance for cold-start items.

In our experiments, counting all items in the training set, we categorize those that appear less than 10 times as cold items, and the rest as warm items. CLCRec, MASR, and $\text{MP4SR}_{\text{w/o Pre-train}}$ are trained based on the full dataset, including both cold and warm items, and are evaluated based on user sequences that take cold items as the target item for prediction. MP4SR is first pre-trained on warm items, followed by fine-tuning using the entire dataset. Its performance is evaluated in the same manner as the other three baselines. Given that cold items lack sufficient interaction data, item ID embeddings are excluded during fine-tuning.

TABLE 6.4: The ablation study of MP4SR and its variants on Pantry and Office datasets.

Dataset	Model	R@10	R@20	N@10	N@20
Pantry	MP4SR	**0.0673**	**0.1040**	**0.0321**	**0.0414**
	$\text{MP4SR}_{\text{ResNet}}$	0.0647	0.1007	0.0317	0.0408
	$\text{MP4SR}_{\text{w/o NIP}}$	0.0501	0.0816	0.0245	0.0324
	$\text{MP4SR}_{\text{w/o CMCL}}$	0.0662	0.1030	0.0310	0.0403
	$\text{MP4SR}_{\text{w/o C-Mixup}}$	0.0649	0.1014	0.0307	0.0399
	$\text{MP4SR}_{\text{w/o Pre-train}}$	0.0595	0.0920	0.0286	0.0369
	$\text{MP4SR}_{\text{w/o Proj}}$	0.0630	0.1001	0.0300	0.0393
	$\text{MP4SR}_{\text{E2E}}$	0.0605	0.0937	0.0290	0.0373
Office	MP4SR	**0.1206**	**0.1480**	**0.0797**	**0.0866**
	$\text{MP4SR}_{\text{ResNet}}$	0.1159	0.1435	0.0749	0.0818
	$\text{MP4SR}_{\text{w/o NIP}}$	0.1062	0.1302	0.0637	0.0697
	$\text{MP4SR}_{\text{w/o CMCL}}$	0.1095	0.1349	0.0649	0.0713
	$\text{MP4SR}_{\text{w/o C-Mixup}}$	0.1192	**0.1480**	0.0768	0.0841
	$\text{MP4SR}_{\text{w/o Pre-train}}$	0.1094	0.1335	0.0665	0.0726
	$\text{MP4SR}_{\text{w/o Proj}}$	0.1177	0.1456	0.0724	0.0794
	$\text{MP4SR}_{\text{E2E}}$	0.1013	0.1243	0.0589	0.0647

Table 6.3 shows the performance achieved by CLCRec, MASR, $\text{MP4SR}_{\text{w/o Pre-train}}$, and MP4SR on cold items. We can note that both $\text{MP4SR}_{\text{w/o Pre-train}}$ and MP4SR outperform the two baseline methods, illustrating the effectiveness of using multimodal information to alleviate the cold-start item problem. Overall, MP4SR performs the best by a substantial margin across all evaluation metrics. This result suggests that cold items can benefit more from self-supervised multimodal pre-training tasks that leverage items with more interactions.

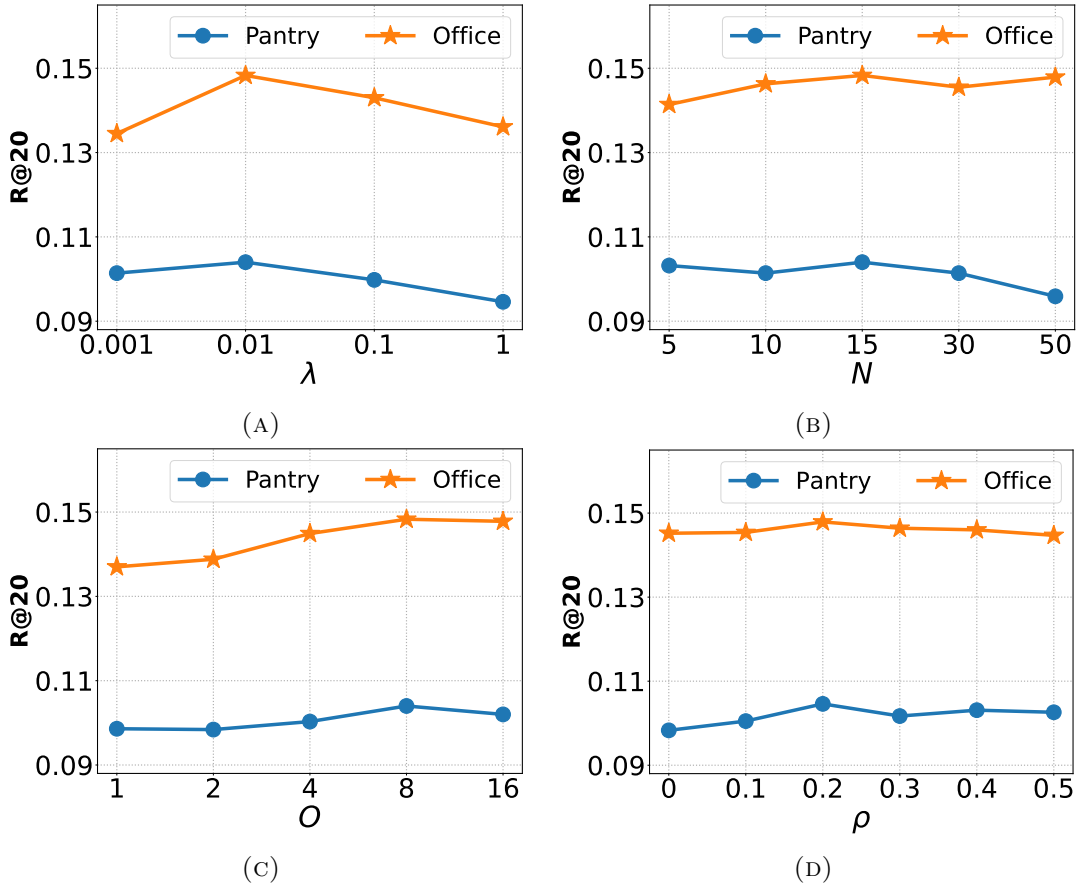
6.3.4 Ablation Study

To study the contribution of each component of MP4SR, we consider the following variants of MP4SR for evaluation:

- **$\text{MP4SR}_{\text{ResNet}}$**: we use ResNet to extract features of item images, instead of converting an item image into keywords;
- **$\text{MP4SR}_{\text{w/o NIP}}$**: we remove the modality-wise next item prediction losses in the pre-training stage;

- **MP4SR**$_{\text{w/o CMCL}}$: we remove the cross-modality contrastive losses in the pre-training stage;
- **MP4SR**$_{\text{w/o C-Mixup}}$: we remove the complementary sequence mixup module in M^{2}SE;
- **MP4SR**$_{\text{w/o Pre-train}}$: we remove the pre-training tasks and train the proposed model from scratch based on the multimodal fine-tuning setting;
- **MP4SR**$_{\text{w/o Proj}}$: we remove the two projection heads and calculate the pre-training contrastive losses on $\mathbf{h}^t$ and $\mathbf{h}^v$;
- **MP4SR**$_{\text{E2E}}$: we optimize the proposed model in an end-to-end manner by summing up the pre-training loss $\mathcal{L}_{\text{pre-train}}$ and the fine-tuning loss $\mathcal{L}_{\text{finetune}}$.

Table 6.4 presents the performance of MP4SR and its variants on Pantry and Office datasets. It shows that each proposed component of MP4SR consistently improves recommendation performance. The modality-wise next item prediction losses are particularly important for pre-training MP4SR for the sequential recommendation. Omitting them results in a significant decline in performance. This is likely due to the fact that the model's primary objective is the optimization of next item prediction throughout both pre-training and fine-tuning stages. Furthermore, cross-modality contrastive losses are more effective in improving recommendation performance on the Office dataset compared to the Pantry dataset, indicating that items in the Office dataset provide more training signals to align various modalities. Additionally, we note that the recommendation performance decreases significantly when pre-training tasks are eliminated, which further validates the effectiveness of applying pre-training for the multimodal sequential recommendation. Also, projection heads facilitate the calculation of contrastive losses by mapping each sequence representation into a common semantic space. If they are removed, the performance is negatively affected. Lastly, if the model is trained end-to-end by combining the $\mathcal{L}_{\text{pre-train}}$ and $\mathcal{L}_{\text{finetune}}$, the performance deteriorates. This is because pre-training losses aim to learn interactions across different modalities, whereas fine-tuning losses prioritize recommendation tasks using cross-entropy losses. If these are optimized together, the model struggles to converge to the optimal solution for the recommendation task.



(A) (B) (C) (D)

FIGURE 6.5: The performance trends of MP4SR with respect to different settings of λ, N, O, ρ on Pantry and Office datasets based on R@20.

6.3.5 Parameter Sensitivity Study

In this experiment, we study the impact of four hyper-parameters, including the λ to balance between modality-wise next item prediction loss and cross-modality contrastive loss, the number of tokens retrieved for each image N, the number of experts used in the MoE architecture O, and the random dropout probability of a sequence ρ. We conduct experiments on Pantry and Office and report R@20 for comparison.

6.3.5.1 Impact of λ

The performance comparison using different values of λ is shown in Figure 6.5(A) and Figure 6.6(A). We vary λ in $\{0.001, 0.01, 0.1, 1.0\}$. We can note that the best performance is achieved when λ is set to 0.01. Recommendation performance is compromised with either a large or a small λ.

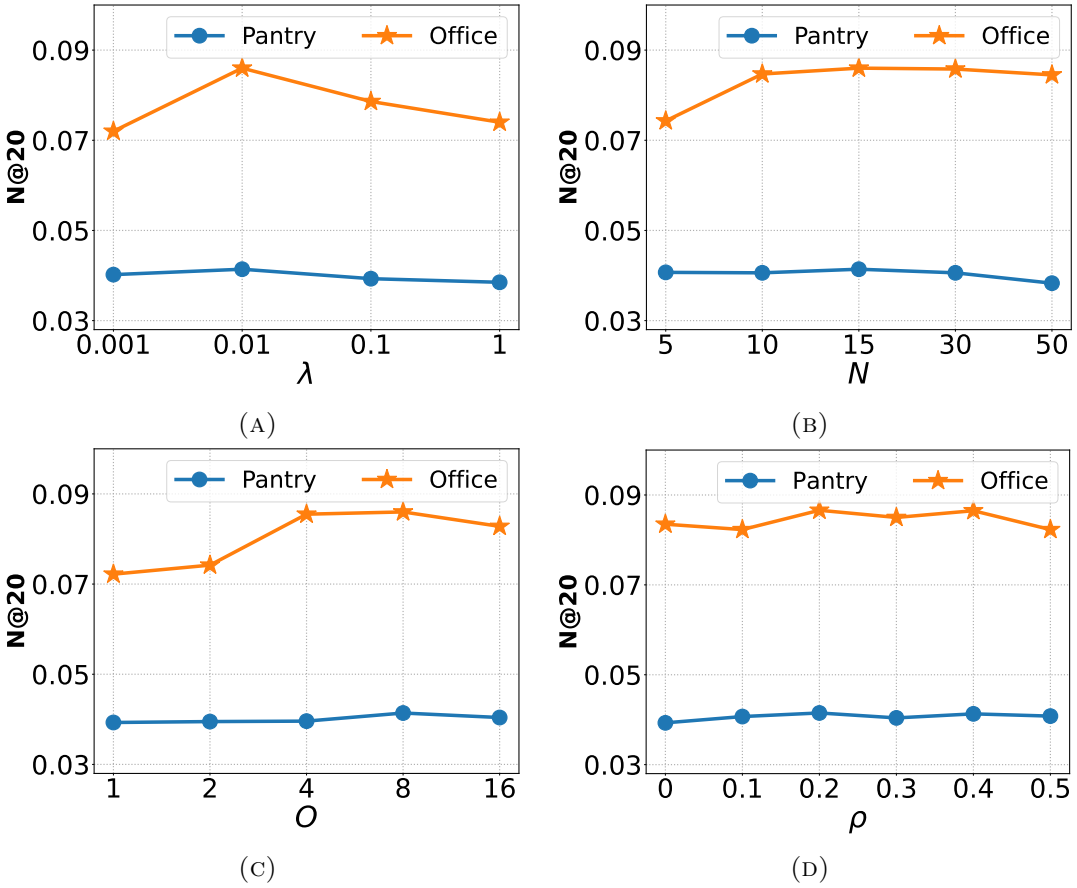


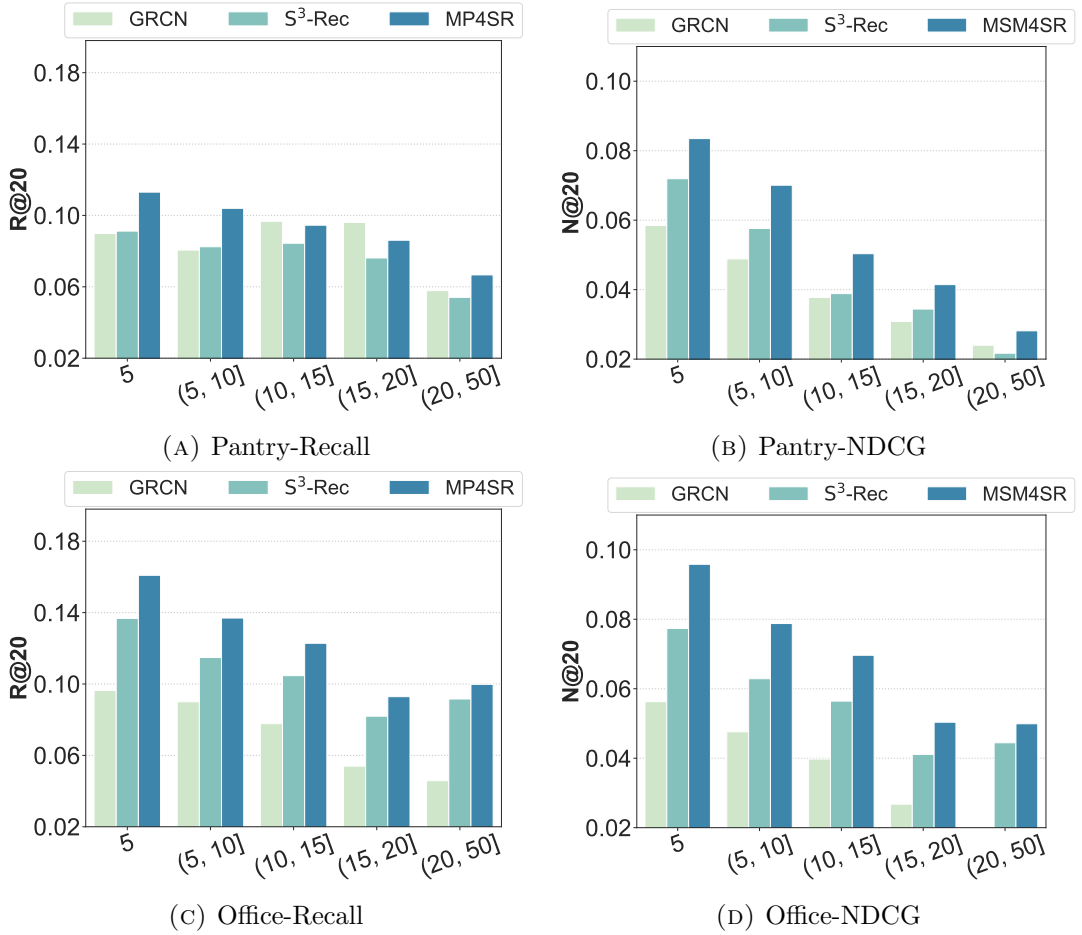
FIGURE 6.6: The performance trends of MP4SR with respect to different settings of λ, N, O, ρ on Pantry and Office datasets based on N@20.

6.3.5.2 Impact of N

The number of tokens retrieved for each image N is varied in $\{5, 10, 15, 30, 50\}$. As shown in Figure 6.5(B) and Figure 6.6(B), 15 word tokens per image appear to be the optimal setting for both datasets. Insufficient information from images is captured when fewer tokens are used, while excessive token usage usually introduces noise.

6.3.5.3 Impact of O

The number of experts used in the MoE architecture O is chosen from $\{1, 2, 4, 8, 16\}$. From Figure 6.5(C) and Figure 6.6(C), we can notice the results with respect to O are consistent on both datasets. The best performance is achieved when O is set to 8. However, the further increase of O does not help with the performance.



(A) Pantry-Recall (B) Pantry-NDCG

(C) Office-Recall (D) Office-NDCG

FIGURE 6.7: The performance on different user groups achieved by GRCN, S^3-Rec, and MP4SR on Pantry and Office datasets.

6.3.5.4 Impact of ρ

As shown in Figure 6.5(D) and Figure 6.6(D), we examine the model performance with different dropout probabilities of the user sequence ρ, which ranges from 0 to 0.5 with a step size of 0.1. We can observe that the model performance is relatively stable with the change in the random dropout probability.

6.3.6 Performance on Different User Groups

The performance comparison results outlined in Table 6.2 allow us to examine the influence of data sparsity on users. Specifically, we split all users into five groups based on the length of their interaction sequences and assess the performance of the models in each user group. Figure 6.7 presents a comparison of performance

on two datasets, from which we have the following observations. *Firstly*, for the Office dataset, the proposed MP4SR model performs better than GRCN and S^3-Rec across all user groups. When considering the Pantry dataset, both GRCN and MP4SR exceed S^3-Rec's performance when user sequences are longer, affirming the importance of multimodal features. *Secondly*, a decrease in the sequence length of user behaviors leads to greater improvements for MP4SR compared to GRCN and S^3-Rec. This demonstrates the superiority of MP4SR in handling sparse scenarios.

6.3.7 Unimodal vs. Multimodal Performance

MP4SR can be applied when only one modality is available. To study the effectiveness of MP4SR in exploiting different modality information, we consider the following two variants of MP4SR for evaluation:

- **MP4SR-V**: the text modality information is not used for model fine-tuning (*i.e.*, removing $\mathbf{h}^t(\mathbf{F}^t + \mathbf{E})^\top$ from Eqn. (12));
- **MP4SR-T**: the image modality information is not used for model fine-tuning (*i.e.*, removing $\mathbf{h}^v(\mathbf{F}^v + \mathbf{E})^\top$ from Eqn. (12)).

In MP4SR-V, MP4SR-T, and MP4SR, the model is pre-trained with both text and image modalities.

Table 6.5 presents the recommendation performance of MP4SR-V, MP4SR-T, and MP4SR on each dataset. We can note that MP4SR-T outperforms MP4SR-V on all datasets, indicating that text information of items contributes more to performance gain than item images. Leveraging both text and image modality information leads to the best recommendation performance for most datasets. This illustrates the effectiveness of exploiting items' multimodal information for sequential recommendation.

6.3.8 Cross-domain Recommendation Performance

We study the knowledge transfer capability of the pre-trained model under the MP4SR framework. Specifically, we evaluate the cross-domain recommendation

TABLE 6.5: The performance achieved by MP4SR under unimodal and multimodal-based settings on each dataset.

Dataset	Model	R@10	R@20	N@10	N@20
Pantry	MP4SR-V	0.0596	0.0928	0.0290	0.0373
	MP4SR-T	0.0649	0.1001	0.0318	0.0407
	MP4SR	**0.0673**	**0.1040**	**0.0321**	**0.0414**
Arts	MP4SR-V	0.1018	0.1345	0.0581	0.0663
	MP4SR-T	0.1144	0.1523	**0.0652**	**0.0748**
	MP4SR	**0.1184**	**0.1570**	0.0637	0.0735
Office	MP4SR-V	0.1153	0.1415	0.0764	0.0830
	MP4SR-T	0.1186	0.1464	0.0772	0.0841
	MP4SR	**0.1206**	**0.1480**	**0.0797**	**0.0866**

TABLE 6.6: The recommendation performance achieved by RecGURU, UniSRec, $\text{MP4SR}_{\text{w/o Pre-train}}$, and MP4SR under cross-domain setting on Pantry and Arts datasets.

Dataset	Model	R@10	R@20	N@10	N@20
Pantry	RecGURU	0.0308	0.0537	0.0152	0.0210
	UniSRec	0.0582	0.0932	0.0265	0.0353
	$\text{MP4SR}_{\text{w/o Pre-train}}$	0.0595	0.0920	0.0286	0.0369
	$\text{MP4SR}_{\text{Cross}}$	**0.0622**	**0.0944**	**0.0294**	**0.0375**
Arts	RecGURU	0.0890	0.1174	0.0569	0.0641
	UniSRec	0.0995	0.1300	0.0565	0.0642
	$\text{MP4SR}_{\text{w/o Pre-train}}$	0.1030	**0.1374**	0.0558	0.0644
	$\text{MP4SR}_{\text{Cross}}$	**0.1041**	0.1367	**0.0573**	**0.0657**

performance of MP4SR, using the Office dataset as the source domain, Pantry and Arts datasets as target domains. In the experiments, RecGURU, UniSRec, and two variants of MP4SR are used for comparison:

- **RecGURU** [229]: this baseline model is a cross-domain sequential recommendation framework that exploits adversarial learning to construct a generalized user representation unified across different domains.
- **UniSRec** [57]: authors utilize item texts to learn more transferable and universal representations from multiple domains for sequential recommendation.
- **$\text{MP4SR}_{\text{w/o Pre-train}}$**: we use the target domain data to train the proposed model from scratch based on the multimodal fine-tuning strategy.

- **MP4SR$_{\text{Cross}}$**: we use the source domain data to pre-train the MP4SR framework and fine-tune it based on the target domain data. For UniSRec and MP4SR variants, we perform parameter-efficient fine-tuning for target domains by fixing the parameters of the Transformer architecture and only fine-tuning the modality encoders.

The results of cross-domain recommendation performance are shown in Table 6.6. Compared with RecGURU, UniSRec, and MP4SR$_{\text{w/o Pre-train}}$, MP4SR$_{\text{Cross}}$ achieves the best performance in terms of most evaluation metrics. This indicates the proposed pre-training framework is effective in transferring knowledge from the source domain to the target domain. Additionally, the proposed contrastive learning tasks enable MP4SR to learn generalized multimodal representations for user behavior sequences to benefit sequential recommendation. It is worth noting that RecGURU performs comparably with MP4SR on the Arts dataset in terms of NDCG but performs the worst on the Pantry dataset. This can be attributed to the fact that Office and Arts have more common users (*i.e.*, 4,068) than Office and Pantry (*i.e.*, 1,525). As a result, RecGURU, which is designed to capture generalized user representations using adversarial learning, is more successful in knowledge transfer from Office to Arts but fails from Office to Pantry.

6.4 Summary

In this chapter, we propose a novel pre-training framework, called MP4SR (*i.e.*, Multimodal Pre-training for Sequential Recommendation), for boosting sequential recommendation performance. In MP4SR, item images are first represented by textual tokens to eliminate the discrepancy between text and image modalities. Then, MP4SR employs a backbone network, M^{2}SE (*i.e.*, Multimodal Mixup Sequence Encoder), to integrate items' multimodal content with the user behavior sequence. Two contrastive learning losses are designed to help M^{2}SE learn generalized multimodal sequence representations. The experiments on real datasets demonstrate that the proposed pre-training framework can help improve sequential recommendation performance in different settings by effectively regularizing the parameter space for sequential recommendation.

Chapter 7

Conclusions and Future Work

This thesis explores various recommendation approaches that utilize data augmentation and contrastive learning methods across different data structures. These approaches aim to mitigate the data sparsity problem and improve the generalization capabilities of recommender systems. In this chapter, we summarize the main contributions of the dissertation and outline promising avenues for future research in this field.

7.1 Conclusions

Recommender systems become increasingly important in improving user experiences through personalized content and suggestions. However, they encounter the challenge of data sparsity, as real-world datasets usually lack sufficient user-item interaction data, leading to compromised performance. Integrating data augmentation with contrastive learning presents a promising approach to alleviate the data sparsity problem and enhance the generalization capabilities of recommender systems. This dissertation introduces innovative data augmentation and contrastive learning methods applicable to diverse data structures, aiming to yield more effective recommender systems.

In Chapter 3, we explore methods for augmenting graph data, focusing on creating augmented data that distinguish between informative and noisy edges in the augmented graph. We leverage a novel technique that employs graph diffusion.

This method effectively smooths neighborhood interactions across the graph, converting the original unweighted graph into a weighted one. The weighting scheme, anchored in the structural significance of each edge, aids in sustaining an efficient neighborhood for each node within the diffusion graph. Specifically, We propose the graph diffusion-based contrastive learning framework for recommendation. In this framework, the diffusion graph is encoded to maintain heterogeneity. Additionally, a symmetric contrastive learning objective is used to compare local node representations of the diffusion graph with those in the user-item interaction graph. Extensive experiments on real-world datasets demonstrate that GDCL consistently outperforms state-of-the-art recommendation methods.

In Chapter 4, we focus on using varying degrees of whitening transformation on pre-trained text features as an alternative to the random-based feature data augmentation methods prevalent in existing research. Our investigation begins with an analysis of a sequential recommendation framework that utilizes item text features. We observe that anisotropy in pre-trained text embeddings can negatively impact performance. To counter this, we apply a whitening transformation that restructures the distribution of pre-trained text embeddings into an isotropic form, resulting in a notable enhancement in model performance. However, empirical studies reveal that this whitening transformation may disrupt the manifold of items sharing similar textual semantics. To overcome this challenge, we introduce two methods for sequential recommendation to exploit the benefits of fully and partially whitened embeddings. The first method WhitenRec+ combines fully and partially whitened embeddings via a simple summation to enhance the representation learning of users and items. The second method DWSRec employs different degrees of whitened embeddings to update the attention heads within the transformer model. This strategy effectively serves as a form of data augmentation, leading to a further improvement in the overall performance of the recommender system.

Chapter 5 delves into the exploration of user behavior sequence augmentations across various features. This approach stands in contrast to most existing methods, which tend to focus on augmenting a single type of feature and often fall short in exploring augmentations involving user behavior sequences across diverse features. We introduce the innovative multimodal pre-training for sequential recommendation framework. This framework leverages contrastive losses to discern

correlations between different modality sequences of users and between these sequences and items. MP4SR incorporates a sequence mixup strategy, which effectively blends different modality sequences. It further applies contrastive learning at both the sequence-to-sequence and sequence-to-item levels. This multimodal pre-training method acts as a potent regularizer, refining the parameter space to optimize the multimodal recommendation task.

In conclusion, this dissertation presents a series of innovative methods that focus on developing diverse data augmentation techniques and contrastive learning across various data structures, aimed at improving the performance of recommender systems. The effectiveness of these methods has been thoroughly validated through extensive experiments using real-world datasets.

7.2 Future Work

This research has explored the intricacies of effective data augmentations with contrastive learning for various types of data structures in recommender systems. Our work opens up multiple avenues for future exploration. Looking ahead, we aim to investigate a spectrum of methodologies, ranging from traditional data-based augmentation to innovative model-based strategies. Additionally, we plan to rigorously evaluate the effectiveness of contrastive objective functions in capturing mutual information. Furthermore, venturing into the vital area of recommendation debiasing is also a key aspect of our future research agenda. Lastly, we discuss the current challenges in negative sample selection within contrastive learning, highlighting the necessity for more effective and specifically tailored strategies in the context of recommender systems.

7.2.1 Advanced Data Augmentation

In this thesis, our focus has primarily centered on data-based augmentation methods, including graph-based, sequence-based, and feature-based augmentations. While these methods have proven to be effective, they often involve the manual selection of augmentation strategies, which can limit their generalizability. To overcome

this limitation, we are interested in exploring model-based augmentation methods [86, 87]. These strategies involve generating different views by perturbing the model itself, such as the user or item encoder. A key advantage of model-based augmentations lies in their enhanced generalizability. They modify the learned representations without being constrained by the nature of the original data. This approach offers a promising direction for future research, potentially leading to more robust and versatile recommender systems.

7.2.2 Advanced Contrastive Objective

In this thesis, we have employed the InfoNCE contrastive objective function to measure mutual information, leveraging its simplicity and effectiveness [86]. However, there are two issues that necessitate further exploration. Firstly, the mutual information measurement in InfoNCE is based on KL divergence, which introduces issues associated with KL divergence, such as asymmetrical estimation and unstable training. To address this, there is a need for a more robust measure of mutual information. Few studies have delved into this, but notable is the work by Fan et al. [230], who propose using the Wasserstein discrepancy measure, based on the 2-Wasserstein distance, for measuring mutual information. However, its application has been limited to sequential recommendation, and its suitability for other types of recommendation tasks remains to be thoroughly investigated. Secondly, the use of mutual information for measuring agreement, while common, faces challenges in accurate estimation and can lead to suboptimal representations, as noted in [231]. This indicates that exploring alternative measures for agreement could be a promising research direction. Such exploration could lead to more accurate and stable models in the field of recommender systems.

7.2.3 Recommendation Debiasing

In this thesis, we have investigated the application of graph diffusion processes for the purpose of recommendation denoising, thereby enhancing the robustness of recommendation models. Beyond the scope of denoising, our research can be extended to the critical issue of recommendation debiasing, an effort to mitigate

pervasive biases, such as popularity and selection biases, within recommender systems. For example, DCRec [232] explores the intersection of contrastive learning with debiasing techniques. This model adeptly addresses popularity bias by discerning and separating user conformity from genuine interest, subsequently refining the application of contrastive regularization. UnKD [233], presents an innovative unbiased knowledge distillation methodology. This approach can be seamlessly integrated with contrastive learning, either by embedding contrastive learning within the knowledge transfer process or by treating partition groups as unique contrastive views. The exploration of data augmentation in conjunction with contrastive learning as a strategy to mitigate these biases is a promising area of research.

7.2.4 Negative Sampling

In the context of contrastive pretext tasks, the selection of negative samples is a critical yet challenging aspect. In this thesis, we have employed the widely used uniform sampling strategy for obtaining negative samples through random sampling. However, this approach has limitations, including the risk of encountering false negatives. Moreover, the presence of easy negative samples, which provide minimal informational value, can potentially diminish the effectiveness of contrastive learning. Consequently, there is a need for more effective negative sampling strategies. Although some studies [234, 235] in the field of computer vision have begun to address this issue, their methodologies are tailored specifically for image data and are not readily adaptable to recommendation systems. Furthermore, considering that current methods often require a large number of negative samples, the development of efficient negative sample strategies is also a crucial area for further research.

List of Publications

- **Lingzi Zhang**, Yong Liu, Xin Zhou, Chunyan Miao, Guoxin Wang, and Haihong Tang. Diffusion-based graph contrastive learning for recommendation with implicit feedback. In *International Conference on Database Systems for Advanced Applications (DASFAA 2022)*, pages 232–247. Springer, 2022.

- **Lingzi Zhang**, Xin Zhou, Zhiwei Zeng, and Zhiqi Shen. Dual-view whitening on pre-trained text embeddings for sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2024)*.

- **Lingzi Zhang**, Xin Zhou, Zhiwei Zeng, and Zhiqi Shen. Are ID Embeddings Necessary? Whitening Pre-trained Text Embeddings for Effective Sequential Recommendation. In *IEEE International Conference on Data Engineering (ICDE 2024)*.

- **Lingzi Zhang***, Yinan Zhang*, Xin Zhou, and Zhiqi Shen. GreenRec: A Large-Scale Dataset for Green Food Recommendation. In *Companion Proceedings of the ACM on Web Conference 2024 (WWW 2024)*.

- **Lingzi Zhang**, Xin Zhou, Zhiwei Zeng, and Zhiqi Shen. Multimodal pretraining framework for sequential recommendation via contrastive learning. In *ACM Transactions on Recommender Systems 2024*.

- Chenyi Lei, Yong Liu, **Lingzi Zhang**, Guoxin Wang, Haihong Tang, Houqiang Li, and Chunyan Miao. Semi: A sequential multi-modal information transfer network for e-commerce micro-video recommendations. In *Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD 2021)*, pages 3161–3171, 2021.

- Hongyu Zhou, Xin Zhou, **Lingzi Zhang**, and Zhiqi Shen. Enhancing dyadic relations with homogeneous graphs for multimodal recommendation. In *European Conference on Artificial Intelligence (ECAI 2023)*. IOS Press, 2023.

- Hongyu Zhou, Xin Zhou, Zhiwei Zeng, **Lingzi Zhang**, and Zhiqi Shen. A comprehensive survey on multimodal recommender systems: Taxonomy, evaluation, and future directions. *Under Review.*

Bibliography

[1] Le Wu, Xiangnan He, Xiang Wang, Kun Zhang, and Meng Wang. A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):4425–4445, 2022.

[2] Yashar Deldjoo, Markus Schedl, Paolo Cremonesi, and Gabriella Pasi. Recommender systems leveraging multimedia content. *ACM Computing Surveys (CSUR)*, 53(5):1–38, 2020.

[3] Yang Li, Kangbo Liu, Ranjan Satapathy, Suhang Wang, and Erik Cambria. Recent developments in recommender systems: A survey. *arXiv preprint arXiv:2306.12680*, 2023.

[4] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1):1–38, 2019.

[5] Joseph A Konstan, Bradley N Miller, David Maltz, Jonathan L Herlocker, Lee R Gordon, and John Riedl. Grouplens: Applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77–87, 1997.

[6] Yue Shi, Martha Larson, and Alan Hanjalic. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys (CSUR)*, 47(1):1–45, 2014.

[7] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[8] Andriy Mnih and Russ R Salakhutdinov. Probabilistic matrix factorization. *Advances in Neural Information Processing Systems*, 20, 2007.

[9] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *IEEE International Conference on Data Mining (ICDM)*, pages 263–272. IEEE, 2008.

[10] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, page 452–461, 2009.

[11] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 549–558, 2016.

[12] Jan Van Balen and Bart Goethals. High-dimensional sparse embeddings for collaborative filtering. In *Proceedings of the Web Conference 2021*, pages 575–581, 2021.

[13] Steffen Rendle. Factorization machines. In *IEEE International Conference on Data Mining (ICDM)*, pages 995–1000. IEEE, 2010.

[14] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: a factorization-machine based neural network for ctr prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 1725–1731, 2017.

[15] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. Wide and deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pages 7–10, 2016.

[16] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the Web Conference 2015*, pages 111–112, 2015.

[17] Florian Strub, Jeremie Mary, and Preux Philippe. Collaborative filtering with stacked denoising autoencoders and sparse inputs. In *NIPS Workshop on Machine Learning for eCommerce*, 2015.

[18] Wanqi Ma, Xiancong Chen, Weike Pan, and Zhong Ming. Vae++ variational autoencoder for heterogeneous one-class collaborative filtering. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining*, pages 666–674, 2022.

[19] Xiangnan He, Xiaoyu Du, Xiang Wang, Feng Tian, Jinhui Tang, and Tat-Seng Chua. Outer product-based neural collaborative filtering. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 2227–2233, 07 2018.

[20] Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*, pages 565–573, 2018.

[21] Feng Xue, Xiangnan He, Xiang Wang, Jiandong Xu, Kai Liu, and Richang Hong. Deep item-based collaborative filtering for top-n recommendation. *ACM Transactions on Information Systems (TOIS)*, 37(3):1–25, 2019.

[22] Zhiyong Cheng, Fan Liu, Shenghan Mei, Yangyang Guo, Lei Zhu, and Liqiang Nie. Feature-level attentive icf for recommendation. *ACM Transactions on Information Systems (TOIS)*, 40(4):1–24, 2022.

[23] Shoujin Wang, Liang Hu, Yan Wang, Xiangnan He, Quan Sheng, Mehmet Orgun, Longbing Cao, Francesco Ricci, and Philip Yu. Graph learning based recommender systems: A review. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, 04 2021.

[24] Zhengshen Jiang, Hongzhi Liu, Bin Fu, Zhonghai Wu, and Tao Zhang. Recommendation in heterogeneous information networks based on generalized random walk model and bayesian personalized ranking. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*, page 288–296, 2018. ISBN 978-1-4503-5581-0.

[25] Xiaotian Han, Chuan Shi, Senzhang Wang, Philip S. Yu, and Li Song. Aspect-level deep collaborative filtering via heterogeneous information networks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, page 3393–3399, 2018.

[26] Yukuo Cen, Xu Zou, Jianwei Zhang, Hongxia Yang, Jingren Zhou, and Jie Tang. Representation learning for attributed multiplex heterogeneous network. In *Proceedings of the 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, page 1358–1368, Jul 2019.

[27] Xiao Wang, Ruijia Wang, Chuan Shi, Guojie Song, and Qingyong Li. Multi-component graph convolutional collaborative filtering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6267–6274, 2020.

[28] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *Proceedings of the Web Conference 2019*, pages 417–426, 2019.

[29] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. Graph contextualized self-attention network for session-based recommendation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, page 3940–3946, Aug 2019.

[30] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, page 950–958, Jul 2019.

[31] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng 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.

[32] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 346–353, 2019.

[33] Priyanka Gupta, Diksha Garg, Pankaj Malhotra, Lovekesh Vig, and Gautam M Shroff. Niser: Normalized item and session representations with graph neural networks. *arXiv preprint arXiv:1909.04276*, 2019.

[34] Mengqi Zhang, Shu Wu, Meng Gao, Xin Jiang, Ke Xu, and Liang Wang. Personalized graph neural networks with attention mechanism for session-aware recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[35] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. Graph contextualized self-attention network for session-based recommendation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 3940–3946, 2019.

[36] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z Sheng, and Mehmet Orgun. Sequential recommender systems: Challenges, progress and prospects. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 6332–6338, 2019.

[37] Ghim-Eng Yap, Xiao-Li Li, and Philip S Yu. Effective next-items recommendation via personalized sequential pattern mining. In *International Conference on Database Systems for Advanced Applications*, pages 48–64. Springer, 2012.

[38] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the Web Conference 2010*, pages 811–820, 2010.

[39] Ruining He, Chen Fang, Zhaowen Wang, and Julian McAuley. Vista: A visually, socially, and temporally-aware model for artistic recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 309–316, 2016.

[40] Ruining He and Julian McAuley. Fusing similarity models with markov chains for sparse sequential recommendation. In *IEEE International Conference on Data Mining (ICDM)*, pages 191–200. IEEE, 2016.

[41] Bo Peng, Zhiyun Ren, Srinivasan Parthasarathy, and Xia Ning. Ham: Hybrid associations models for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[42] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. Recurrent recommender networks. In *Proceedings of the 10th ACM International Conference on Web Search and Data Mining*, pages 495–503, 2017.

[43] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. Sequential user-based recurrent neural network recommendations. In *Proceedings of the 11th ACM Conference on Recommender Systems*, pages 152–160, 2017.

[44] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. A simple convolutional generative network for next item recommendation. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*, pages 582–590, 2019.

[45] Fajie Yuan, Xiangnan He, Haochuan Jiang, Guibing Guo, Jian Xiong, Zhezhao Xu, and Yilin Xiong. Future data helps training: Modeling future contexts for session-based recommendation. In *Proceedings of the Web Conference 2020*, pages 303–313, 2020.

[46] Ruihong Qiu, Jingjing Li, Zi Huang, and Hongzhi Yin. Rethinking the item order in session-based recommendation with graph neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 579–588, 2019.

[47] Ruihong Qiu, Hongzhi Yin, Zi Huang, and Tong Chen. Gag: Global attributed graph neural network for streaming session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 669–678, 2020.

[48] Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. Sequential recommendation with graph neural networks. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 378–387, 2021.

[49] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *IEEE International Conference on Data Mining (ICDM)*, pages 197–206. IEEE, 2018.

[50] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1441–1450, 2019.

[51] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, pages 1893–1902, 2020.

[52] Zhiwei Liu, Ziwei Fan, Yu Wang, and Philip S Yu. Augmenting sequential recommendation with pseudo-prior items via reversely pre-training transformer. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1608–1612, 2021.

[53] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. Contrastive learning for sequential recommendation. In *IEEE International Conference on Data Engineering (ICDE)*, pages 1259–1273, 2022.

[54] Yueqi Xie, Peilin Zhou, and Sunghun Kim. Decoupled side information fusion for sequential recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 1611–1621, 2022.

[55] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. Mmgcn: Multi-modal graph convolution network for personalized recommendation of micro-video. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 1437–1445, 2019.

[56] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, and Xiaofang Zhou. Feature-level deeper self-attention network for sequential recommendation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 4320–4326, 2019.

[57] Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. Towards universal sequence representation learning for recommender systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 585–593, 2022.

[58] Zheng Yuan, Fajie Yuan, Yu Song, Youhua Li, Junchen Fu, Fei Yang, Yunzhu Pan, and Yongxin Ni. Where to go next for recommender systems? id-vs. modality-based recommender models revisited. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2639–2649, 2023.

[59] Xin Zhou, Hongyu Zhou, Yong Liu, Zhiwei Zeng, Chunyan Miao, Pengwei Wang, Yuan You, and Feijun Jiang. Bootstrap latent representations for multi-modal recommendation. In *Proceedings of the ACM Web Conference 2023*, pages 845–854, 2023.

[60] Yupeng Hou, Zhankui He, Julian McAuley, and Wayne Xin Zhao. Learning vector-quantized item representation for transferable sequential recommenders. In *Proceedings of the ACM Web Conference 2023*, pages 1162–1171, 2023.

[61] Ruyu Li, Wenhao Deng, Yu Cheng, Zheng Yuan, Jiaqi Zhang, and Fajie Yuan. Exploring the upper limits of text-based collaborative filtering using large language models: Discoveries and insights. *arXiv preprint arXiv:2305.11700*, 2023.

[62] Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. Neural news recommendation with multi-head self-attention. In *Proceedings*

of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 6389–6394, 2019.

[63] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. Recommendation as language processing (rlp): A unified pretrain, personalized prompt and predict paradigm (p5). In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 299–315, 2022.

[64] Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. Text is all you need: Learning language representations for sequential recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1258—-1267, 2023.

[65] Chenyi Lei, Dong Liu, Weiping Li, Zheng-Jun Zha, and Houqiang Li. Comparative deep learning of hybrid representations for image recommendations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2545–2553, 2016.

[66] Qidi Xu, Fumin Shen, Li Liu, and Heng Tao Shen. Graphcar: Content-aware multimedia recommendation with graph autoencoder. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 981–984, 2018.

[67] Xun Yang, Xiangnan He, Xiang Wang, Yunshan Ma, Fuli Feng, Meng Wang, and Tat-Seng Chua. Interpretable fashion matching with rich attributes. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 775–784, 2019.

[68] Ruining He and Julian McAuley. Vbpr: Visual bayesian personalized ranking from implicit feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016.

[69] Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the Web Conference 2016*, pages 507–517, 2016.

[70] Xingchen Li, Xiang Wang, Xiangnan He, Long Chen, Jun Xiao, and Tat-Seng Chua. Hierarchical fashion graph network for personalized outfit recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 159–168, 2020.

[71] Joonseok Lee, Sami Abu-El-Haija, Balakrishnan Varadarajan, and Apostol Natsev. Collaborative deep metric learning for video understanding. In *Proceedings of the 24th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 481–490, 2018.

[72] Joonseok Lee and Sami Abu-El-Haija. Large-scale content-only video recommendation. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 987–995, 2017.

[73] Qiang Liu, Shu Wu, and Liang Wang. Deepstyle: Learning user preferences for visual recommendation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 841–844, 2017.

[74] Jinhui Tang, Xiaoyu Du, Xiangnan He, Fajie Yuan, Qi Tian, and Tat-Seng Chua. Adversarial training towards robust multimedia recommender system. *IEEE Transactions on Knowledge and Data Engineering*, 32(5):855–867, 2019.

[75] Shang Liu, Zhenzhong Chen, Hongyi Liu, and Xinghai Hu. User-video co-attention network for personalized micro-video recommendation. In *Proceedings of the Web Conference 2019*, pages 3020–3026, 2019.

[76] Fan Liu, Zhiyong Cheng, Changchang Sun, Yinglong Wang, Liqiang Nie, and Mohan Kankanhalli. User diverse preference modeling by multimodal attentive metric learning. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 1526–1534, 2019.

[77] Xu Chen, Hanxiong Chen, Hongteng Xu, Yongfeng Zhang, Yixin Cao, Zheng Qin, and Hongyuan Zha. Personalized fashion recommendation with visual explanations based on multimodal attention network: Towards visually explainable recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 765–774, 2019.

[78] Yong Liu, Susen Yang, Chenyi Lei, Guoxin Wang, Haihong Tang, Juyong Zhang, Aixin Sun, and Chunyan Miao. Pre-training graph transformer with multimodal side information for recommendation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 2853–2861, 2021.

[79] Nhu-Thuat Tran and Hady W Lauw. Aligning dual disentangled user representations from ratings and textual content. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1798–1806, 2022.

[80] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, and Tat-Seng Chua. Graph-refined convolutional network for multimedia recommendation with implicit feedback. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 3541–3549, 2020.

[81] Jinghao Zhang, Yanqiao Zhu, Qiang Liu, Shu Wu, Shuhui Wang, and Liang Wang. Mining latent structures for multimedia recommendation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3872–3880, 2021.

[82] Qifan Wang, Yinwei Wei, Jianhua Yin, Jianlong Wu, Xuemeng Song, and Liqiang Nie. Dualgnn: Dual graph neural network for multimedia recommendation. *IEEE Transactions on Multimedia*, 2021.

[83] Jing Yi and Zhenzhong Chen. Multi-modal variational graph auto-encoder for recommendation systems. *IEEE Transactions on Multimedia*, 24:1067–1079, 2021.

[84] Penghang Yu, Zhiyi Tan, Guanming Lu, and Bing-Kun Bao. Multi-view graph convolutional network for multimedia recommendation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 6576–6585, 2023.

[85] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):857–876, 2021.

[86] Mengyuan Jing, Yanmin Zhu, Tianzi Zang, and Ke Wang. Contrastive self-supervised learning in recommender systems: A survey. *ACM Transactions on Information Systems (TOIS)*, 42(2):1–39, 2023.

[87] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Jundong Li, and Zi Huang. Self-supervised learning for recommender systems: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[88] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. Self-supervised graph learning for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 726–735, 2021.

[89] Dongha Lee, SeongKu Kang, Hyunjun Ju, Chanyoung Park, and Hwanjo Yu. Bootstrapping user and item representations for one-class collaborative filtering. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 317–326, 2021.

[90] Junwei Zhang, Min Gao, Junliang Yu, Lei Guo, Jundong Li, and Hongzhi Yin. Double-scale self-supervised hypergraph learning for group recommendation. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, pages 2557–2567, 2021.

[91] Xin Zhou, Aixin Sun, Yong Liu, Jie Zhang, and Chunyan Miao. Selfcf: A simple framework for self-supervised collaborative filtering. *ACM Transactions on Recommender Systems*, 1(2):1–25, 2023.

[92] Hui Wang, Kun Zhou, Xin Zhao, Jingyuan Wang, and Ji-Rong Wen. Curriculum pre-training heterogeneous subgraph transformer for top-n recommendation. *ACM Transactions on Information Systems (TOIS)*, 41(1):1–28, 2023.

[93] Yonghui Yang, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. Enhanced graph learning for collaborative filtering via mutual information maximization. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 71–80, 2021.

[94] Junjie Huang, Qi Cao, Ruobing Xie, Shaoliang Zhang, Feng Xia, Huawei Shen, and Xueqi Cheng. Adversarial learning data augmentation for graph contrastive learning in recommendation. In *International Conference on Database Systems for Advanced Applications*, pages 373–388. Springer, 2023.

[95] Yangqin Jiang, Chao Huang, and Lianghao Huang. Adaptive graph contrastive learning for recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4252–4261, 2023.

[96] Bowen Hao, Hongzhi Yin, Jing Zhang, Cuiping Li, and Hong Chen. A multi-strategy-based pre-training method for cold-start recommendation. *ACM Transactions on Information Systems (TOIS)*, 41(2):1–24, 2023.

[97] Haoran Yang, Hongxu Chen, Lin Li, S Yu Philip, and Guandong Xu. Hyper meta-path contrastive learning for multi-behavior recommendation. In *IEEE International Conference on Data Mining (ICDM)*, pages 787–796. IEEE, 2021.

[98] Jie Shuai, Kun Zhang, Le Wu, Peijie Sun, Richang Hong, Meng Wang, and Yong Li. A review-aware graph contrastive learning framework for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1283–1293, 2022.

[99] Jiangxia Cao, Xixun Lin, Shu Guo, Luchen Liu, Tingwen Liu, and Bin Wang. Bipartite graph embedding via mutual information maximization. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 635–643, 2021.

[100] Junliang Yu, Hongzhi Yin, Jundong Li, Qinyong Wang, Nguyen Quoc Viet Hung, and Xiangliang Zhang. Self-supervised multi-channel hypergraph convolutional network for social recommendation. In *Proceedings of the Web Conference 2021*, pages 413–424, 2021.

[101] Xiaoling Long, Chao Huang, Yong Xu, Huance Xu, Peng Dai, Lianghao Xia, and Liefeng Bo. Social recommendation with self-supervised metagraph informax network. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, pages 1160–1169, 2021.

[102] Lingzi Zhang, Yong Liu, Xin Zhou, Chunyan Miao, Guoxin Wang, and Haihong Tang. Diffusion-based graph contrastive learning for recommendation with implicit feedback. In *International Conference on Database Systems for Advanced Applications*, pages 232–247. Springer, 2022.

[103] Xuheng Cai, Chao Huang, Lianghao Xia, and Xubin Ren. Lightgcl: Simple yet effective graph contrastive learning for recommendation. In *International Conference on Learning Representations*, 2023.

[104] Mingyue Cheng, Fajie Yuan, Qi Liu, Xin Xin, and Enhong Chen. Learning transferable user representations with sequential behaviors via contrastive pre-training. In *IEEE International Conference on Data Mining (ICDM)*, pages 51–60. IEEE, 2021.

[105] Yicong Li, Hongxu Chen, Xiangguo Sun, Zhenchao Sun, Lin Li, Lizhen Cui, Philip S Yu, and Guandong Xu. Hyperbolic hypergraphs for sequential recommendation. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, pages 988–997, 2021.

[106] Chenyang Wang, Weizhi Ma, Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. Sequential recommendation with multiple contrast signals. *ACM Transactions on Information Systems*, 41:1–27, 2023.

[107] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In *International Conference on Learning Representations*, 2016.

[108] Xu Yuan, Hongshen Chen, Yonghao Song, Xiaofang Zhao, Zhuoye Ding, Zhen He, and Bo Long. Improving sequential recommendation consistency with self-supervised imitation. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, page 3321–3327, 2021.

[109] Zhiwei Liu, Yongjun Chen, Jia Li, Philip S Yu, Julian McAuley, and Caiming Xiong. Contrastive self-supervised sequential recommendation with robust augmentation. *arXiv preprint arXiv:2108.06479*, 2021.

[110] Zhiwei Liu, Lei Zheng, Jiawei Zhang, Jiayu Han, and S Yu Philip. Jscn: Joint spectral convolutional network for cross domain recommendation. In *IEEE International Conference on Big Data*, pages 850–859. IEEE, 2019.

[111] Fangye Wang, Yingxu Wang, Dongsheng Li, Hansu Gu, Tun Lu, Peng Zhang, and Ning Gu. Cl4ctr: A contrastive learning framework for ctr prediction. In *Proceedings of the 16th ACM International Conference on Web Search and Data Mining*, pages 805–813, 2023.

[112] Tiansheng Yao, Xinyang Yi, Derek Zhiyuan Cheng, Felix Yu, Ting Chen, Aditya Menon, Lichan Hong, Ed H Chi, Steve Tjoa, Jieqi Kang, et al. Self-supervised learning for large-scale item recommendations. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, pages 4321–4330, 2021.

[113] Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Lizhen Cui, and Xiangliang Zhang. Self-supervised hypergraph convolutional networks for session-based recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4503–4511, 2021.

[114] Lingzi Zhang, Xin Zhou, Zhiwei Zeng, and Zhiqi Shen. Multimodal pretraining framework for sequential recommendation via contrastive learning. *ACM Transactions on Recommender Systems*, 2024.

[115] Tinglin Huang, Yuxiao Dong, Ming Ding, Zhen Yang, Wenzheng Feng, Xinyu Wang, and Jie Tang. Mixgcf: An improved training method for graph neural network-based recommender systems. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 665–674, 2021.

[116] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung 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*, pages 1294–1303, 2022.

[117] Junliang Yu, Xin Xia, Tong Chen, Lizhen Cui, Nguyen Quoc Viet Hung, and Hongzhi Yin. Xsimgcl: Towards extremely simple graph contrastive learning for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[118] Haibo Ye, Xinjie Li, Yuan Yao, and Hanghang Tong. Towards robust neural graph collaborative filtering via structure denoising and embedding perturbation. *ACM Transactions on Information Systems (TOIS)*, 41(3):1–28, 2023.

[119] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607, 2020.

[120] Chenyi Lei, Yong Liu, Lingzi Zhang, Guoxin Wang, Haihong Tang, Houqiang Li, and Chunyan Miao. Semi: A sequential multi-modal information transfer network for e-commerce micro-video recommendations. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3161–3171, 2021.

[121] Yinan Zhang, Boyang Li, Yong Liu, and Chunyan Miao. Initialization matters: Regularizing manifold-informed initialization for neural recommendation systems. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2263–2273, 2021.

[122] Zhuang Liu, Yunpu Ma, Yuanxin Ouyang, and Zhang Xiong. Contrastive learning for recommender system. *arXiv:2101.01317*, 2021.

[123] Hao Tang, Guoshuai Zhao, Yuxia Wu, and Xueming Qian. Multisample-based contrastive loss for top-k recommendation. *IEEE Transactions on Multimedia*, 2021.

[124] Johannes Klicpera, Stefan Weißenberger, and Stephan Günnemann. Diffusion improves graph learning. In *Advances in Neural Information Processing Systems*, 2019.

[125] Jie Wang, Fajie Yuan, Mingyue Cheng, Joemon M Jose, Chenyun Yu, Beibei Kong, Zhijin Wang, Bo Hu, and Zang Li. Transrec: Learning transferable recommendation from mixture-of-modality feedback. *arXiv preprint arXiv:2206.06190*, 2022.

[126] Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[127] Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*, 2021.

[128] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763, 2021.

[129] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, 2019.

[130] Gediminas Adomavicius and Alexander 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*, 17(6): 734–749, 2005.

[131] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. Graph neural networks in recommender systems: a survey. *ACM Computing Surveys*, 55 (5):1–37, 2022.

[132] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the Web Conference 2001*, pages 285–295, 2001.

[133] Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1):143–177, 2004.

[134] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the 9th ACM International Conference on Web Search and Data Mining*, pages 153–162, 2016.

[135] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *Proceedings of the Web Conference 2018*, pages 689–698, 2018.

[136] Ilya Shenbin, Anton Alekseev, Elena Tutubalina, Valentin Malykh, and Sergey I Nikolenko. Recvae: A new variational autoencoder for top-n recommendations with implicit feedback. In *Proceedings of the 13th ACM International Conference on Web Search and Data Mining*, pages 528–536, 2020.

[137] Lei Zheng, Vahid Noroozi, and Philip S Yu. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the 10th ACM International Conference on Web Search and Data Mining*, pages 425–434, 2017.

[138] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 335–344, 2017.

[139] Hakan Bagci and Pinar Karagoz. Context-aware friend recommendation for location based social networks using random walk. In *Proceedings of the Web Conference 2016*, pages 531–536, 2016.

[140] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and S Yu Philip. Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 31(2):357–370, 2018.

[141] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S. Yu. Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In *Proceedings of the 24th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, page 1531–1540, Jul 2018.

[142] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[143] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 974–983, 2018.

[144] Haoyu Wang, Defu Lian, and Yong Ge. Binarized collaborative filtering with distilling graph convolutional network. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, page 4802–4808, Aug 2019.

[145] Chen Lei, Wu Le, Hong Richang, Zhang Kun, and Wang Meng. Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[146] Qiaoyu Tan, Ninghao Liu, Xing Zhao, Hongxia Yang, Jingren Zhou, and Xia Hu. Learning to hash with graph neural networks for recommender systems. In *Proceedings of the Web Conference 2020*, pages 1988–1998, 2020.

[147] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

[148] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 165–174, Jul 2019.

[149] Rianne van den Berg, Thomas N. Kipf, and Max Welling. Graph convolutional matrix completion. In *Proceedings of the 24th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2017.

[150] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the Web Conference 2017*, page 173–182, 2017.

[151] Muhan Zhang and Yixin Chen. Inductive matrix completion based on graph neural networks. In *International Conference on Learning Representations*, 2020.

[152] Balázs Hidasi and Domonkos Tikk. General factorization framework for context-aware recommendations. *Data Mining and Knowledge Discovery*, 30(2):342–371, 2016.

[153] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. Learning hierarchical representation model for nextbasket recommendation. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 403–412, 2015.

[154] Shoujin Wang, Liang Hu, Longbing Cao, Xiaoshui Huang, Defu Lian, and Wei Liu. Attention-based transactional context embedding for next-item recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[155] Ruining He, Wang-Cheng Kang, and Julian McAuley. Translation-based recommendation. In *Proceedings of the 11th ACM Conference on Recommender Systems*, pages 161–169, 2017.

[156] Wen Wang, Wei Zhang, Shukai Liu, Qi Liu, Bo Zhang, Leyu Lin, and Hongyuan Zha. Beyond clicks: Modeling multi-relational item graph for session-based target behavior prediction. In *Proceedings of the Web Conference 2020*, pages 3056–3062, 2020.

[157] Cheng Hsu and Cheng-Te Li. Retagnn: Relational temporal attentive graph neural networks for holistic sequential recommendation. In *Proceedings of the Web Conference 2021*, pages 2968–2979, 2021.

[158] Huachi Zhou, Qiaoyu Tan, Xiao Huang, Kaixiong Zhou, and Xiaoling Wang. Temporal augmented graph neural networks for session-based recommendations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1798–1802, 2021.

[159] Yixin Zhang, Yong Liu, Yonghui Xu, Hao Xiong, Chenyi Lei, Wei He, Lizhen Cui, and Chunyan Miao. Enhancing sequential recommendation with graph contrastive learning. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence*, pages 2398–2405, 2022.

[160] Jianling Wang, Kaize Ding, Ziwei Zhu, and James Caverlee. Session-based recommendation with hypergraph attention networks. In *Proceedings of the 2021 SIAM International Conference on Data Mining*, pages 82–90. SIAM, 2021.

[161] Mengqi Zhang, Shu Wu, Xueli Yu, Qiang Liu, and Liang Wang. Dynamic graph neural networks for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):4741–4753, 2022.

[162] Ziwei Fan, Zhiwei Liu, Jiawei Zhang, Yun Xiong, Lei Zheng, and Philip S Yu. Continuous-time sequential recommendation with temporal graph collaborative transformer. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, pages 433–442, 2021.

[163] Lingzi Zhang, Yinan Zhang, Xin Zhou, and Zhiqi Shen. Greenrec: A large-scale dataset for green food recommendation. In *Companion Proceedings of the ACM on Web Conference 2024*, pages 625–628, 2024.

[164] Chengkai Huang, Shoujin Wang, Xianzhi Wang, and Lina Yao. Dual contrastive transformer for hierarchical preference modeling in sequential recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 99–109, 2023.

[165] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. Empowering news recommendation with pre-trained language models. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1652–1656, 2021.

[166] Hongyu Zhou, Xin Zhou, Zhiwei Zeng, Lingzi Zhang, and Zhiqi Shen. A comprehensive survey on multimodal recommender systems: Taxonomy, evaluation, and future directions. *arXiv preprint arXiv:2302.04473*, 2023.

[167] Hongyu Zhou, Xin Zhou, Lingzi Zhang, and Zhiqi Shen. Enhancing dyadic relations with homogeneous graphs for multimodal recommendation. In *European Conference on Artificial Intelligence*, volume 372, pages 3123–3130. IOS Press, 2023.

[168] Xin Zhou and Zhiqi Shen. A tale of two graphs: Freezing and denoising graph structures for multimodal recommendation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 935–943, 2023.

[169] Xin Zhou and Chunyan Miao. Disentangled graph variational auto-encoder for multimodal recommendation with interpretability. *IEEE Transactions on Multimedia*, 2024.

[170] Xin Zhou. Mmrec: Simplifying multimodal recommendation. In *Proceedings of the 5th ACM International Conference on Multimedia in Asia Workshops*, pages 1–2, 2023.

[171] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019*, pages 4171–4186, 2019.

[172] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 191–198, 2016.

[173] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

[174] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining*, pages 813–823, 2022.

[175] Desheng Cai, Shengsheng Qian, Quan Fang, Jun Hu, Wenkui Ding, and Changsheng Xu. Heterogeneous graph contrastive learning network for personalized micro-video recommendation. *IEEE Transactions on Multimedia*, 2022.

[176] Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In *Proceedings of the ACM Web Conference 2022*, pages 2320–2329, 2022.

[177] Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. Intent contrastive learning for sequential recommendation. In *Proceedings of the ACM Web Conference 2022*, pages 2172–2182, 2022.

[178] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[179] Wei Jin, Tyler Derr, Haochen Liu, Yiqi Wang, Suhang Wang, Zitao Liu, and Jiliang Tang. Self-supervised learning on graphs: Deep insights and new direction. *arXiv preprint arXiv:2006.10141*, 2020.

[180] Chuan Shi, Xiao Wang, and S Yu Philip. *Heterogeneous Graph Representation Learning and Applications*. Springer, 2022.

[181] Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning*, pages 4116–4126, 2020.

[182] Sergey Brin. The pagerank citation ranking: bringing order to the web. *Proceedings of ASIS, 1998*, 98:161–172, 1998.

[183] Imre Kondor RISI. Diffusion kernels on graphs and other discrete input spaces. In *International Conference on Machine Learning*, 2002.

[184] Aleksandar Bojchevski, Johannes Gasteiger, Bryan Perozzi, Amol Kapoor, Martin Blais, Benedek Rózemberczki, Michal Lukasik, and Stephan Günnemann. Scaling graph neural networks with approximate pagerank. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2464–2473, 2020.

[185] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *International Conference on Learning Representations*, 2019.

[186] Reid Andersen, Fan Chung, and Kevin Lang. Local graph partitioning using pagerank vectors. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 475–486. IEEE, 2006.

[187] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. In *Proceedings of the Web Conference 2021*, pages 2069–2080, 2021.

[188] Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 188–197, 2019.

[189] Dongha Lee, SeongKu Kang, Hyunjun Ju, Chanyoung Park, and Hwanjo Yu. Bootstrapping user and item representations for one-class collaborative filtering. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 317–326, 2021.

[190] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.

[191] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.

[192] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[193] Lingzi Zhang, Xin Zhou, Zhiwei Zeng, and Zhiqi Shen. Dual-view whitening on pre-trained text embeddings for sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

[194] Lingzi Zhang, Xin Zhou, Zhiwei Zeng, and Zhiqi Shen. Are id embeddings necessary? whitening pre-trained text embeddings for effective sequential recommendation. In *IEEE International Conference on Data Engineering (ICDE)*, 2024.

[195] Chen Ma, Peng Kang, and Xue Liu. Hierarchical gating networks for sequential recommendation. In *Proceedings of the 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 825–833, 2019.

[196] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[197] Agnan Kessy, Alex Lewin, and Korbinian Strimmer. Optimal whitening and decorrelation. *The American Statistician*, pages 309–314, 2018.

[198] Chenyang Wang, Yuanqing Yu, Weizhi Ma, Min Zhang, Chong Chen, Yiqun Liu, and Shaoping Ma. Towards representation alignment and uniformity in collaborative filtering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1816–1825, 2022.

[199] Xi Weng, Lei Huang, Lei Zhao, Rao Anwer, Salman H Khan, and Fahad Shahbaz Khan. An investigation into whitening loss for self-supervised learning. *Advances in Neural Information Processing Systems*, pages 29748–29760, 2022.

[200] Ian T Jolliffe. *Principal component analysis for special types of data*. Springer, 2002.

[201] Guillaume Desjardins, Karen Simonyan, Razvan Pascanu, et al. Natural neural networks. *Advances in Neural Information Processing Systems*, 2015.

[202] Anthony J Bell and Terrence J Sejnowski. The "independent components" of natural scenes are edge filters. *Vision Research*, pages 3327–3338, 1997.

[203] Dariusz Dereniowski and Marek Kubale. Cholesky factorization of matrices in parallel and ranking of graphs. In *International Conference on Parallel Processing and Applied Mathematics*, pages 985–992, 2004.

[204] Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening and coloring transform for GANs. In *International Conference on Learning Representations*, 2019.

[205] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[206] Lei Huang, Dawei Yang, Bo Lang, and Jia Deng. Decorrelated batch normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 791–800, 2018.

[207] Tianyu Hua, Wenxiao Wang, Zihui Xue, Sucheng Ren, Yue Wang, and Hang Zhao. On feature decorrelation in self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9598–9608, 2021.

[208] Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. In *International Conference on Machine Learning*, pages 3015–3024, 2021.

[209] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022.

[210] Kawin Ethayarajh. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, 2019.

[211] Dariusz Dereniowski and Marek Kubale. Cholesky factorization of matrices in parallel and ranking of graphs. In *International Conference on Parallel Processing and Applied Mathematics*, pages 985–992, 2003.

[212] Yue Song, Nicu Sebe, and Wei Wang. Improving covariance conditioning of the svd meta-layer by orthogonality. In *European Conference on Computer Vision*, 2022.

[213] Neha Wadia, Daniel Duckworth, Samuel S Schoenholz, Ethan Dyer, and Jascha Sohl-Dickstein. Whitening and second order optimization both make information in the dataset unusable during training, and can reduce or prevent generalization. In *International Conference on Machine Learning*, pages 10617–10629, 2021.

[214] Yinwei Wei, Xiang Wang, Qi Li, Liqiang Nie, Yan Li, Xuanping Li, and Tat-Seng Chua. Contrastive learning for cold-start recommendation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 5382–5390, 2021.

[215] Walid Krichene and Steffen Rendle. On sampled metrics for item recommendation. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1748–1757, 2020.

[216] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, pages 8026–8037, 2019.

[217] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, et al. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, pages 4653–4664, 2021.

[218] Xingyu Pan, Yushuo Chen, Changxin Tian, Zihan Lin, Jinpeng Wang, He Hu, and Wayne Xin Zhao. Multimodal meta-learning for cold-start sequential recommendation. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*, pages 3421–3430, 2022.

[219] Qiang Cui, Shu Wu, Qiang Liu, Wen Zhong, and Liang Wang. Mv-rnn: A multi-view recurrent neural network for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 32(2):317–331, 2018.

[220] Hangbo Bao, Wenhui Wang, Li Dong, Qiang Liu, Owais Khan Mohammed, Kriti Aggarwal, Subhojit Som, Songhao Piao, and Furu Wei. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. *Advances in Neural Information Processing Systems*, pages 32897–32912, 2022.

[221] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13041–13049, 2020.

[222] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in Neural Information Processing Systems*, 32, 2019.

[223] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023.

[224] Xudong Lin, Simran Tiwari, Shiyuan Huang, Manling Li, Mike Zheng Shou, Heng Ji, and Shih-Fu Chang. Towards fast adaptation of pretrained contrastive models for multi-channel video-language retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14846–14855, 2023.

[225] Dumitru Erhan, Aaron Courville, Yoshua Bengio, and Pascal Vincent. Why does unsupervised pre-training help deep learning? In *Proceedings of the*

13th International Conference on Artificial Intelligence and Statistics, pages 201–208, 2010.

[226] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng 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*, pages 639–648, 2020.

[227] Qiaoyu Tan, Jianwei Zhang, Jiangchao Yao, Ninghao Liu, Jingren Zhou, Hongxia Yang, and Xia Hu. Sparse-interest network for sequential recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 598–606, 2021.

[228] Yidan Hu, Yong Liu, Chunyan Miao, and Yuan Miao. Memory bank augmented long-tail sequential recommendation. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*, pages 791–801, 2022.

[229] Chenglin Li, Mingjun Zhao, Huanming Zhang, Chenyun Yu, Lei Cheng, Guoqiang Shu, Beibei Kong, and Di Niu. Recguru: Adversarial learning of generalized user representations for cross-domain recommendation. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining*, pages 571–581, 2022.

[230] Ziwei Fan, Zhiwei Liu, Hao Peng, and Philip S Yu. Mutual wasserstein discrepancy minimization for sequential recommendation. In *Proceedings of the Web Conference 2023*, 2023.

[231] Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. In *International Conference on Learning Representations*, 2020.

[232] Yuhao Yang, Chao Huang, Lianghao Xia, Chunzhen Huang, Da Luo, and Kangyi Lin. Debiased contrastive learning for sequential recommendation. In *Proceedings of the ACM Web Conference 2023*, pages 1063–1073, 2023.

[233] Gang Chen, Jiawei Chen, Fuli Feng, Sheng Zhou, and Xiangnan He. Unbiased knowledge distillation for recommendation. In *Proceedings of the 16th ACM International Conference on Web Search and Data Mining*, pages 976–984, 2023.

[234] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *Advances in neural information processing systems*, 33:8765–8775, 2020.

[235] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. *Advances in Neural Information Processing Systems*, 33:21798–21809, 2020.