Graph Contrastive Learning and Its Applications in Recommendation Systems

A thesis submitted in partial fulfilment of the requirements
for the degree of

Doctor of Philosophy
in
Computer Science and Technology

by

Haoran Yang

to

School of Computer Science
Faculty of Engineering and Information Technology
University of Technology Sydney
NSW - 2007, Australia

September 2024

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, *Haoran Yang*, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the *School of Computer Science* of the *Faculty of Engineering and Information Technology* at the University of Technology Sydney. This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of the requirements for a degree at any other academic institution except as fully acknowledged within the text. This thesis is the result of a Collaborative Doctoral Research Degree program with The Hong Kong Polytechnic University. This research is supported by the Australian Government Research Training Program.

SIGNATURE: Production Note:
Signature removed prior to publication.

Haoran Yang

DATE: 30th September, 2024

PLACE: Sydney, Australia

ABSTRACT

Graph Contrastive Learning (GCL) has emerged as a powerful tool for unsupervised graph representation learning, attracting significant attention across various applications. Its success depends on obtaining high-quality contrasting samples through graph augmentations. However, current augmentation strategies face several limitations, such as introducing noise that degrades downstream model performance, lacking flexibility for different datasets with various characteristics, and being unable to process non-embedding node features like text. These limitations hinder the full potential of GCL in practical applications. Moreover, implementing GCL in diverse application scenarios, particularly recommendation systems, is crucial for realizing its practical value. Recommendation System (RS) domains are especially suitable for GCL to perform because it can generate contrasting samples that provide self-supervised training signals, addressing the lack of related information in real-world applications caused by various factors like privacy concerns . Despite its potential, the use of GCL in recommendation systems remains underexplored, and there is a need to explore its full potential in this domain. To address the problems above, this research proposes advanced graph augmentation strategies, incorporating counterfactual mechanisms and the capabilities of Large Language Model (LLM), to overcome the limitations of existing methods. By integrating counterfactual mechanisms, the proposed strategies aim to mitigate the noise introduced during graph augmentations and achieve flexibility when facing different graph data, thereby improving the performance on downstream graph learning tasks. Additionally, the utilization of LLM capabilities enables the processing of non-embedding node features like text, enhancing the flexibility of the augmentation strategies for graph data with multimodality like text features. Furthermore, this study introduces the concept of hyper meta-path to construct contrasting samples (*i.e.,* hyper meta-graphs) for GCL for multi-behavior recommendations, providing insights into creating effective contrasting samples in this specific context, which is a pioneering research work in the domain of GCL in RS and inspires the future works in the literature. This study also investigates specific training paradigms, finding that GCL pre-training and prompt-tuning can better utilize GCL's capabilities in recommendations. By exploring these training paradigms, the research aims to provide practical guidance on how to effectively leverage GCL in recommendation systems. In summary, the findings of this study contribute to the advancement of graph augmentation strategies for GCL and demonstrate the applicability of GCL in enhancing RS.

ACKNOWLEDGMENTS

I am deeply grateful to my supervisors and other researchers who have significantly supported me throughout my PhD journey. Prof. Guandong Xu and Prof. Qing Li (The Hong Kong Polytechnic University) provided ample funding to support my studies and living expenses, allowing me to focus entirely on my research. Their patient and meticulous guidance has been invaluable. Dr. Hongxu Chen (Commonwealth Bank of Australia), my co-supervisor during the initial stage of my PhD, served as a guiding beacon for me. Dr. Xiangyu Zhao (City University of Hong Kong) offered detailed and professional guidance on my research, which has been instrumental in my development as a mature researcher.

I am profoundly grateful to my families, especially my parents, for their unwavering and selfless support. I could not have come this far without their encouragement and assistance. No words can fully express my heartfelt gratitude for them.

I am cordially thankful to my colleagues and friends, particularly the members of the DSMI group, for their invaluable support. They have helped me overcome challenges not only in my studies and research but also in my daily life and mental well-being.

Finally, I would like to extend my sincere appreciation to everyone who has been a part of my PhD journey. Their encouragement, guidance, and support have been the pillars that sustained me through the highs and lows of this PhD endeavor. This achievement is as much theirs as it is mine, and I am forever indebted to each of them for their contributions to my academic career and personal growth. I appreciate them all for being an integral part of this remarkable journey.

List of Publications

Related to The Thesis

1. Yang, H., Chen, H., Li, L., Yu, P. S., and Xu, G.

   Hyper meta-path contrastive learning for multi-behavior recommendation.

   In *IEEE International Conference on Data Mining, ICDM 2021, Auckland, New Zealand, December 7-10, 2021* (2021), IEEE, pp. 787–796

2. Yang, H., Chen, H., Zhang, S., Sun, X., Li, Q., Zhao, X., and Xu, G.

   Generating counterfactual hard negative samples for graph contrastive learning.

   In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023* (2023), ACM, pp. 621–629

3. Yang, H., Zhao, X., Li, Y., Chen, H., and Xu, G.

   An empirical study towards prompt-tuning for graph contrastive pre-training in recommendations.

   In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023* (2023)

4. Yang, H., Zhao, X., Huang, S., Li, Q., and Xu, G.

   Latex-gcl: Large language models (llms)-based data augmentation for text-attributed graph contrastive learning, 2024

Others

1. Yang, H., Chen, H., Pan, S., Li, L., Yu, P. S., and Xu, G.

   Dual space graph contrastive learning.

In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022* (2022), ACM, pp. 1238–1247

2. YANG, H., ZHAO, X., LI, M., CHEN, H., AND XU, G.

   Mitigating the performance sacrifice in dp-satisfied federated settings through graph contrastive learning.

   Inf. Sci. 648 (2023), 119552

3. YANG, H., WANG, Y., ZHAO, X., CHEN, H., YIN, H., LI, Q., AND XU, G.

   Multi-level graph knowledge contrastive learning.

   IEEE Transactions on Knowledge and Data Engineering (2024), 1–14

TABLE OF CONTENTS

ABBREVIATIONS

CL Contrastive Learning

CV Computer Vision

GCL Graph Contrastive Learning

GNN Graph Neural Network

KDE Kernel Density Estimation

LLM Large Language Model

MF Matrix Factorization

ML Machine Learning

MLP Multi-Layer Perceptron

NLP Natural Language Processing

RS Recommendation System

SOTA State-of-the-art

SSL Self-Supervised Learning

TAG Text-attributed Graph

TCL Topological Contrastive Learning

GLOSSARIES

CGC The counterfactual-based novel GCL method proposed in Generating Counterfactual Hard Negative Samples for Graph Contrastive Learning.

CPTPP A novel GCL paradigm tailored for GCL application in recommendation systems, which consists of two stages, including GCL pre-training and soft-prompting.

HMG-CR A tailored GCL framework for conducting multi-behavior recommendations, which utilizes a novel concept, namely hyper meta-path, to construct contrasting samples.

KL Divergence Kullback-Leibler Divergence.

LATEX-GCL A novel GCL framework that equips with the large language models (LLMs)-based data augmentation strategy to conduct text-attributed GCL.

RQ1.1 How to conduct flexible and efficient graph augmentations that tackle the limitations in current strategies?

RQ1.2 How to effectively augment non-embedding features, such as text, on the node in a graph to produce contrasting samples?

RQ2.1 How to produce contrasting samples tailored for the specific recommendation scenario by leveraging graph augmentation strategies?

RQ2.2 What training strategy should be adopted for GCL in recommendations? Is current the end-to-end one with joint training gooed enough?

LIST OF FIGURES

LIST OF TABLES

TABLE **Page**

CHAPTER 1

INTRODUCTION

This section presents the introduction to the thesis, which includes three parts: background, research scopes, and contributions. Specifically:

- Sec. 1.1 introduces the research background of this research by demonstrating the intuitions of Contrastive Learning (CL)
- Sec. 1.2 presents CL in graph learning domain and introduces the research scopes regarding Graph Contrastive Learning (GCL), including graph augmentation strategies in GCL and GCL's applications in Recommendation System (RS).
- Sec. 1.4 briefly summarizes the research works conducted and the related contributions to show how this research advances the related research areas.

This introductory chapter lays the foundation for the thesis by providing a comprehensive overview of the research background, scope, and contributions. The subsequent sections will delve into the detailed contents.

1.1 Research Background

Many Machine Learning (ML) methods [23] depend heavily on a large amount of manual labels or rewards as the primary learning signals during the training process. Although these ML methods have achieved significant success over the past decade, their over-reliance on supervised signals presents several limitations:

- First, the data itself contains rich information that can provide more semantics than the limited information available from manual labels [31]. Relying solely on supervised learning necessitates an enormous number of labels, which can be both time-consuming and resource-intensive to obtain. This over-reliance may also result in knowledge-specific solutions, thereby limiting the models' generalization ability [81]. In other words, models trained exclusively on labeled data may perform well on specific tasks but struggle to generalize to new, unseen scenarios.

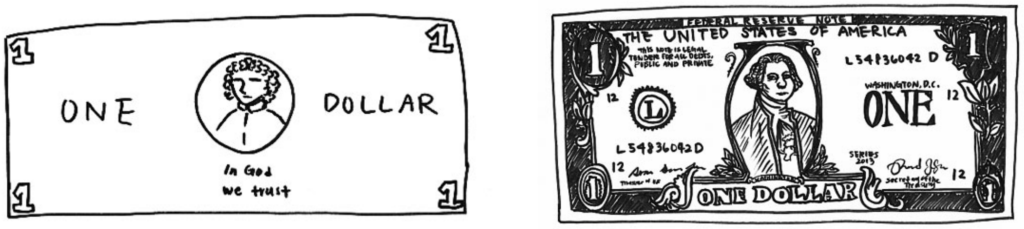
- Second, in some scenarios, such as reinforcement learning [110], the cost of acquiring rewards or labels is prohibitively high [80, 145]. For instance, in real-world applications like autonomous driving or robotic control, obtaining accurate and reliable labels or rewards can be extremely challenging and expensive. This makes it impractical to rely solely on supervision signals for solving all problems. The high cost and difficulty of obtaining these signals can hinder the development and deployment of effective ML models in such domains.

Given these limitations, there is a growing interest in exploring alternative learning paradigms that do not rely as heavily on supervised signals. Techniques such as unsupervised learning, semi-supervised learning, and self-supervised learning [31] are gaining traction as they can leverage the inherent structure and information within the data itself. These approaches aim to reduce the dependency on manual labels and rewards, thereby addressing some of the key challenges faced by traditional ML methods.

While conventional ML methods have achieved remarkable success, their dependence on large amounts of supervised signals presents significant limitations. By exploring and adopting alternative learning paradigms, researchers can develop more robust and generalizable models that are better suited to a wide range of real-world applications. This shift in focus has the potential to drive significant advancements in the field of machine learning. As an alternative solution, Self-Supervised Learning (SSL) [31] has been proposed and has garnered significant attention from researchers due to its impressive performance on various representation tasks. SSL leverages the structural information inherent in the

data to formulate training procedures without the need for manual labels. It is important to note that self-supervised learning (SSL) encompasses two main approaches: contrastive learning and generative learning. Among the different approaches within SSL, Contrastive Learning (CL) has recently gained considerable interest because of its simplicity, effectiveness, robustness, and generalizability. To gain a deeper understanding of contrastive learning, it is essential to distinguish it from generative learning.

Generative learning methods, such as Generative Adversarial Networks (GANs) [29] and Variational Autoencoders (VAEs) [52], generate new data instances from input data, ensuring that the generated instances are similar to the input instances in high-level semantics. In contrast, contrastive learning methods focus on identifying common features within data of the same class and distinguishing differences between data of different categories. Unlike generative learning, which operates at the instance level, contrastive learning distinguishes data in an abstract latent space, leading to more superficial structures and stronger generalization abilities. The primary objective of CL is to train to encode data from the same class similarly while maximizing the differences among data from different classes.



(a) Drawing of a dollar bill from memory.

(b) Drawing made with a dollar bill present.

Figure 1.1: An illustration to help understand contrastive learning borrowed from [2].

An empirical experiment from [2] illustrates the intuitions behind contrastive learning (CL). In this experiment, two subjects are asked to draw a dollar bill: one recalling from memory and the other with a dollar bill present. Predictably, the drawing made with the dollar bill present is much more detailed than the one made from memory, which are shown in Figure 1.1. Despite having seen a dollar bill countless times, people do not retain a full representation of it, they only remember enough features to distinguish it from other objects. Similarly, can researchers develop representation learning algorithms that focus not on pixel-level details but on high-level features sufficient to differentiate between objects? The answer is yes, and this is the core idea of CL. By contrasting samples, CL captures the high-level features necessary to distinguish the target from other objects.

1.2 Research Aims

The concept of CL can be applied to a variety of scenarios, including Natural Language Processing (NLP) and Computer Vision (CV), as well as to graph learning domains. Graph-structured data is a crucial form of real-world data that effectively represents complex relationships among entities. This type of data is essential for numerous applications, such as social network analysis and graph-based recommendation systems, where understanding the intricate connections between entities is paramount. Graph learning techniques, particularly Graph Neural Network (GNN), have shown significant promise in handling graph-structured data. These techniques can be further enhanced by incorporating CL to leverage its advantages. CL can help improve the performance of graph learning tasks by enabling models to learn more robust and discriminative representations of the data. This synergy between CL and graph learning opens up new possibilities for advancing the field.

Given the potential benefits, this research focuses on the application of CL in graph learning, a domain referred to as GCL. The aim is to advance progress in both CL and graph learning fields by exploring how these two areas can complement each other. By integrating CL into graph learning, we can develop more effective models that better capture the complexities of graph-structured data. In conclusion, the integration of CL with graph learning techniques such as GNNs holds great promise for enhancing the performance of various graph-related tasks. This research aims to contribute to the advancement of both fields by investigating the potential of GCL. Through this exploration, this research hopes to unlock new insights and capabilities that can be applied to a wide range of real-world applications.

1.3 Research Motivations

There are multiple aspects from which to study GCL. One of the most critical components of GCL is the graph augmentation strategy, which is essential for generating contrasting pairs and ensuring the success of GCL. The GCL process embeds critical semantic information from the input graph into the latent space by performing graph encoding and contrastive learning between appropriate contrasting samples. Current literature identifies three major types of graph augmentation strategies, including random augmentation, rule-based augmentation, and adaptive augmentation. However, these existing strategies have various limitations that prevent GCL from achieving optimal performance. Therefore, a specific focus of this research is to design advanced graph augmentation strategies to address these limitations and improve the performance of GCL.

Another perspective for advancing the development of GCL is to explore its potential in addressing real-world problems. CL has already demonstrated its practical value in NLP [116, 36] and CV [18] by solving a series of real-world issues. Similarly, GCL has been applied to various real-world scenarios, such as recommendation systems and social network analysis. GCL can introduce auxiliary self-supervised training signals, which is particularly beneficial for applications like recommendation systems where user profiles may be protected for privacy reasons and lack sufficient supervised training signals. However, at the beginning of this research, GCL in recommendation systems was underexplored, and existing methods did not adequately address the limitations of the training paradigm in these scenarios. Therefore, another specific focus of this research is to investigate the application of GCL in recommendation systems and to improve the current training paradigms.

In summary, this research is motivated by advancing the field of GCL via enhancing graph augmentation strategies and exploring its potential in recommendation systems. By proposing more advanced graph augmentation strategies to address the limitations in current literature, this study seeks to improve the performance and flexibility of GCL methods. Additionally, by investigating the application of GCL in recommendation systems, this research aims to introduce a pioneering work of applying GCL to graph-based recommendation tasks and investigate effective training paradigms that can improve current methods. Ultimately, this research aspires to contribute significantly to both the advancements in GCL methodology and its applications in recommendation systems.

1.4 Contributions

This thesis makes several key contributions to improving GCL and its applications in recommendation systems. It introduces innovative strategies to enhance graph data processing, making it more adaptable and effective. By using advanced techniques, the research reduces noise and increases flexibility in handling diverse data types, including text. It also pioneers new methods for creating contrasting samples, offering fresh insights for multi-behavior recommendations. Additionally, the study explores training methods that optimize the use of GCL, providing practical guidance for its application in recommendation systems. Overall, the research advances the field and inspires future work.

Focusing on the two research scopes outlined in the previous section, this research presents four studies, the detailed contributions of each work are listed as follows.

Chap. 3 introduces a novel method that generates high-quality contrasting samples for GCL using a counterfactual mechanism. This learning-based GCL method, known as CGC,

can adaptively process different datasets with various characteristics, addressing limitations in the current literature and proposing a flexible GCL approach. Chap. 4 addresses the limitation that current GCL methods cannot directly augment non-embedding features like text. It presents a novel GCL framework called LATEX-GCL, which leverages Large Language Model (LLM) to augment text features on the graph, producing high-quality contrasting samples. Three novel and tailored prompts for text feature augmentation are introduced and examined through extensive experiments.

To demonstrate the practical value of GCL methods, it is essential to apply them to real-world scenarios. One of the most suitable application areas for GCL methods, as previously discussed, is recommendation systems. Therefore, the remaining two sections focus on the applications of GCL in recommendation systems. Chap. 5 conducts pioneering work by applying GCL to recommendation tasks, introducing the concept of hyper meta-paths to construct hyper meta-graphs for contrastive learning. The proposed HMG-CR method outlines the implementation pipeline of GCL in RS, inspiring future research in this domain. Chap. 6 examines GCL in RS at a higher level, investigating the training paradigm of GCL for recommendation. An empirical study demonstrates the disadvantages of the current end-to-end training paradigm. Consequently, a novel framework for GCL in RS, called CPTPP, is proposed. This framework utilizes prompt learning to introduce an effective 'pre-training and prompt-tuning' paradigm.

In short, this research proposes advanced graph augmentation strategies, incorporating counterfactual mechanisms and the capabilities of LLM, to overcome the limitations of existing GCL methods. Moreover, a pioneering research work is proposed and a novel training paradigm of GCL for recommendations is examined, inspiring the future works in the domain of GCL in RS.

1.5 Chapter Summary

In summary, this chapter serves as the introduction to this thesis, providing an overview of the research background, scope, and contributions.

Specifically, Section 1.1 highlights the limitations of SSL and contrasts it with generative learning methods, thereby introducing the background and intuitions of contrastive learning (CL). Section 1.2 outlines the primary research scope, focusing on GCL. It identifies two specific sub-scopes: graph augmentation strategies in GCL and the application of GCL in RS. Finally, Section 1.4 summarizes the contributions of this research by introducing the four research works in Chapter 7, which focus on the two sub-scopes.

The subsequent content of this thesis is organized as follows:

- Chap. 2: Literature review on augmentation techniques and GCL's applications in RS.
- Chap. 3: Generating Counterfactual Hard Negative Samples for GCL (RQ1.1).
- Chap. 4: LLM-Based Data Augmentation for Text-Attributed GCL (RQ1.2).
- Chap. 5: Hyper Meta-Graph Construction for GCL in RS (RQ2.1).
- Chap. 6: A 'Pre-training and Prompt-tuning' Paradigm for GCL in RS (RQ2.2).
- Chap. 7: Thesis conclusion and future work discussion.

CHAPTER 2

LITERATURE REVIEW

This chapter provides a comprehensive literature review on GCL methods and their applications in RS, focusing on four main aspects, specifically:

- Sec. 2.1 illustrates the overview of GCL procedures by introducing graph augmentation, graph element encoding, training objective, and downstream tasks.
- Sec. 2.2 introduces current literature about graph augmentation strategies by giving a taxonomy. Subsequently, a thorough limitation analysis regarding these graph augmentation strategies is provided.
- Sec. 2.3 briefly introduces GCL's applications in the research domain of RS. It mainly analyzes the limitations in current training paradigm used to utilized GCL in RS.
- Sec. 2.4 summarizes the limitations of graph augmentation strategies and GCL's applications in RS. thereby proposing four research questions.
- Sec. 2.5 introduces some supplementary content about some other related works, including LLMs for graph learning, multi-behaviour recommendation, and prompt-tuning, which can help understand the research works within this thesis.

This chapter presents a thorough literature review regarding GCL, thereby introducing the research questions of this thesis.

2.1 Overview of Graph Contrastive Learning Procedures

To implement GCL, four crucial steps are required:

- Graph Augmentation: Augmenting the graph data to produce contrasting samples.
- Graph Element Encoding: Encoding the contrasting samples to obtain graph embeddings for loss calculation and training.
- Training Objective Definition: Define the training objective according to the experimental settings, graph embedding, and the mechanism of the contrastive learning.
- Addressing Downstream Tasks: Taking the graph embeddings produced by the previous steps as the input to the downstream models to conduct the specific task.

2.1.1 Graph Augmentation

The success of GCL hinges on the effective generation of contrasting samples [132, 149, 131]. By conducting CL between appropriately generated contrasting samples and the input target, GCL embeds critical semantic information of the input graph into the latent space. This embedding process is essential for capturing the underlying structure and relationships within the graph data. Graph augmentation is a key step in this process, aiming to generate diverse and meaningful variations of the original data without altering the semantic labels. The goal of graph augmentation is to create new data instances that maintain the essential characteristics of the input data while introducing variations that can be used for contrastive learning. These augmented instances are crucial for the GCL process, as they provide the necessary contrasting samples for learning robust representations. The augmented data instances are then paired together to form positive pairs in the GCL process. Positive pairs consist of the original graph and its augmented version, which are expected to be similar in the latent space. By maximizing the similarity between these positive pairs, GCL ensures that the learned representations capture the important semantic information of the input graph. This process helps the model to distinguish between different graph structures and to generalize better to new, unseen data.

The generation of contrasting samples through graph augmentation is fundamental to the success of GCL. By creating diverse and meaningful variations of the original data, GCL can effectively embed critical semantic information into the latent space. This process not only enhances the model's ability to capture the underlying structure of the graph data but also improves its generalization capabilities. Numerous studies [78, 104, 132, 149] have

demonstrated that graph augmentations are crucial in GCL. With appropriate graph augmentations, GCL methods outperform many sophisticated GNN models on various unsupervised learning tasks. This highlights the importance of selecting and applying effective augmentation techniques to enhance the performance of GCL models.

In the practice of GCL, several key insights have been reported in the literature. One significant observation is that combining different graph augmentations tends to yield better results [132]. This is because each graph augmentation method incorporates its unique underlying priors, which can introduce diverse latent semantics into the model. By leveraging multiple augmentation techniques, the model can capture a richer and more comprehensive set of features from the graph data. The effectiveness of combining different augmentations aligns with intuitive expectations. In real-world scenarios, data often contains multiple layers of information and relationships. By applying diverse augmentation techniques, GCL models can better mimic the complexity of real-world data, leading to improved performance on downstream tasks.

2.1.2 Graph Element Encoding

The graph encoder is an essential component of the entire GCL process. It is responsible for processing both the original and augmented graph elements to generate graph embeddings for CL. The effectiveness of the graph encoder determines whether the models can learn representative graph embeddings from contrasting samples.

GNN models are among the most widely used types of graph encoders. Examples include GCN [53] and GAT [103]. Technically, any GNN model can be employed in GCL, as GCL is not particularly sensitive to the choice of GNN models [78]. This flexibility allows users to select from a variety of GNN architectures based on specific needs and preferences.

Interestingly, the success of GCL does not necessarily depend on using highly sophisticated GNN models. Since properly constructed contrasting pairs carry rich semantic information, such as geometric and topological structures, even simple models like GCN can achieve satisfactory performance. This highlights the importance of the contrasting pairs themselves in capturing the essential features of the graph data. In practical applications of GCL, geometric or topology-based GNNs, such as GIN [118] and TAGCN [22], are often used as graph encoders. These models are favored for their simplicity and effectiveness. They can leverage structural information from proposed hyper meta-paths and the advantages of structure-level contrastive learning [78], making them well-suited for GCL scenarios.

2.1.3 Training Objective

As introduced in the previous background section, the core idea of GCL is to make positive pairs as similar as possible while pushing negative pairs far apart. For any data point x, GCL methods aim to train graph encoders that ensure the data point x is similar to its positive contrasting sample and distinct from its negative contrasting sample, which can be formulated as follows:

$$score\big(f(x), f(x^+)\big) >> score\big(f(x), f(x^-)\big), \tag{2.1}$$

where x^+ represents the positive sample, and x^- denotes the negative sample. The score function is used to calculate the similarity between the samples, which will be determined according to the specific scenarios.

In this context, the data point x can be considered an anchor point, *e.g.,* the input graph in GCL. To achieve the training objective, it is needed to build a classifier that accurately distinguishes between positive and negative samples. This requires the similarity function to assign high values to positive pairs and low values to negative pairs. The InfoNCE loss function [102] is widely used to accomplish this goal:

$$\mathscr{L}_{InfoNCE} = -\log \frac{\exp\big(f(x, x^+)\big)}{\sum_{x_i \in X} \exp\big(f(x, x_i)\big)}, \tag{2.2}$$

where $X = \{x_1, x_2, \cdots, x_N\}$ contains all the contrasting samples. It can be observed that the formula above is similar to cross-entropy for N-way softmax classification tasks.

It is worth noting that InfoNCE has a strong connection to mutual information:

(2.3)

$$\begin{aligned}
\mathbb{E}_X\left[-\log\frac{\exp(f(x^+,x))}{\sum_{x_i\in X} f(x_i,x)}\right] &= -\mathbb{E}_{(x^+,x)} f(x^+,x) + \mathbb{E}_{(x^+,x)}\left[\log \sum_{x_i\in X} f(x_i,x)\right] \\
&= -\mathbb{E}_{(x^+,x)} f(x^+,x) + \mathbb{E}_{(x^+,x)}\left[\log\left(\exp\big(f(x^+,x)\big) + \sum_{x_i\in X_{neg}} \exp\big(f(x_i,x)\big)\right)\right] \\
&\geq -\mathbb{E}_{(x^+,x)} f(x^+,x) + \mathbb{E}_X\left[\log \sum_{x_i\in X_{neg}} \exp\big(f(x_i,x)\big)\right] \\
&= -\mathbb{E}_{(x^+,x)} f(x^+,x) + \mathbb{E}\left[\log\frac{1}{N-1} \sum_{x_i\in X_{neg}} \exp\big(f(x_i,x)\big) + \log\frac{1}{N-1}\right],
\end{aligned}$$

which is equivalent to MINE [4] estimator. Therefore, maximizing a lower bound on this estimator while minimizing InfoNCE ensures the maximization of mutual information between the anchor data and the positive sample.

2.1.4 Downstream Tasks

In the final step of GCL procedures, the trained graph embeddings are utilized to conduct a sort of graph learning tasks. These downstream tasks can be roughly broken down into two categories, which are listed below:

- Node-level tasks: This kind of tasks focus on the properties of nodes and relations among nodes, such as node classification [53], node clustering [101], and link prediction [138] in pure graph learning context. Such tasks can be converted to real-world tasks in different scenarios. For example, node classification can be converted to anomaly detection in financial transactions [3], node clustering can be converted to community detection in social network analysis [44], and link prediction can be converted to recommendation tasks in graph-based recommendation systems [40].

- Graph-level tasks: This kind of tasks focus on analyzing the properties and structures of a whole graph, such as graph classification [132, 149] and graph generation [148]. In real-world application scenarios, graph classification can be converted to molecular property prediction in bioinformatics [14, 9] and graph generation can be used to conduct drug discovery in pharmaceutical industry [7].

The pre-trained graph embeddings can be not only directly used to conduct downstream tasks in an unsupervised manner but also incorporated with downstream models and supervised labels to enhance downstream performance in a semi-supervised manner. The specific usage manner depends on users' needs and preferences.

2.2 Graph Augmentations in Graph Contrastive Learning

This section provides a concise review of the research progress on GCL with a focus on graph augmentation strategies. The first part introduces various types of graph augmentation strategies, while the second part offers a detailed analysis of their limitations.

2.2.1 Three Types of Graph Augmentation Strategies

There are various taxonomies to categorize current GCL methods, and one of the most important one is designed according to the graph augmentation strategies [50]. This thesis roughly break down GCL methods into three categories, which are shown in Figure 2.1, including random augmentation, rule-based augmentation, and adaptive augmentation.

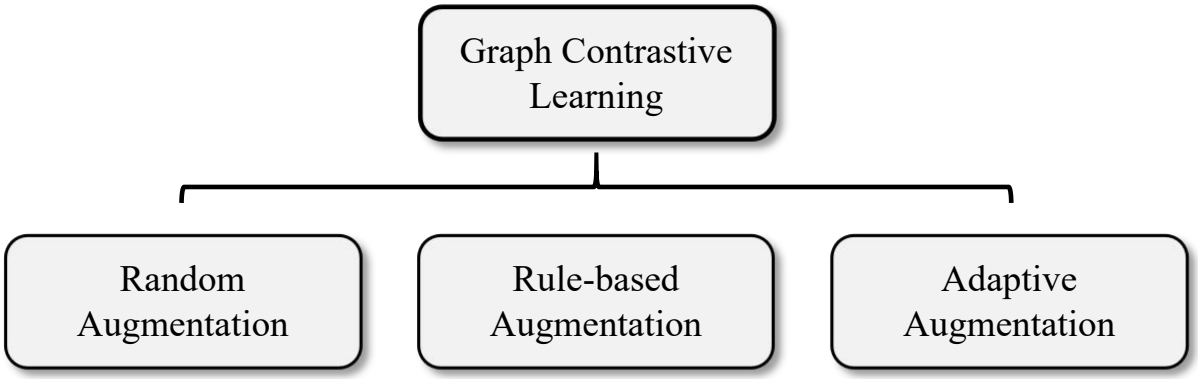


Figure 2.1: A taxonomy of GCL methods with regard to graph augmentation strategy.

2.2.1.1 Random Augmentation

Random augmentation perturb the original graph by adopting structure perturbation, feature masking, and subgraph sampling [132] to produce contrasting samples.

Structure perturbation perturbs a portion of nodes and edges in the original graph by dropping node, deleting existed edges, and adding new edges [132]. Such a process can be briefly formulated as follows:

$$\hat{\mathbf{A}} = \mathbf{P} \cdot \mathbf{A}, \tag{2.4}$$

where $\mathbf{A}$ is the adjacency matrix of the original graph, $\mathbf{P}$ is the randomly generated perturbation matrix, and $\hat{\mathbf{A}}$ denotes the adjacency matrix of the perturbed graph.

Feature masking aims to mask or alter the values in feature embedding matrix of the original graph, which can be briefly formulated as follows:

$$\hat{\mathbf{X}} = \mathbf{M} \cdot \mathbf{X}, \tag{2.5}$$

where $\mathbf{X}$ is the feature embedding matrix of the original graph, $\mathbf{M}$ denotes the randomly generated masking matrix, and $\hat{\mathbf{X}}$ is the feature embedding matrix of the perturbed graph.

Both formulations regarding the structure perturbation and feature masking above are abstract descriptions. The practical implementation could be much more complicated due to various sophisticated mechanisms and designs involved.

Subgraph sampling differs from the previously discussed graph perturbation operations, as it does not introduce any changes to the original graph. Instead, it involves extracting subgraphs from the original graph, ensuring that these subgraphs retain a portion of the original graph's semantics. By contrasting these subgraphs, one can potentially capture the local properties of the original graph. The subgraph sampling process can be suc-

cinctly described as follows:

$$\mathbf{A}_{\mathscr{S}} = \mathbf{A}[\mathscr{S}, \mathscr{S}],\ \mathbf{X}_{\mathscr{S}} = \mathbf{X}[\mathscr{S}, :], \tag{2.6}$$

where $\mathscr{S}$ denotes the node set of the sampled subgraph.

In summary, random augmentation represents one of the most straightforward yet effective graph augmentation strategies in the early stages of GCL research, significantly advancing progress in related domains. DGI [104] is a pioneering work in GCL that employs corruption operations to alter graph features, thereby generating contrasting samples. GCC [78] is among the first to leverage subgraph sampling across different datasets to obtain contrasting samples. GraphCL [132] systematically reviews all the aforementioned random augmentation operations and conducts extensive experiments, providing valuable insights into their practical application.

2.2.1.2 Rule-based Augmentation

A representative solution for rule-based augmentation is graph diffusion [35, 74, 100]. This method has gained significant attention due to its ability to enhance the structural and semantic properties of graphs [54]. Graph diffusion operates by iteratively constructing connections between previously unlinked nodes in the original graph. This process is crucial as it allows the augmentation method to utilize the global semantics and high-order relational information inherent in the graph. By establishing these new connections, graph diffusion effectively captures the broader context and relationships that may not be immediately apparent in the original graph structure. Moreover, graph diffusion is versatile and can be applied to various types of graphs, making it a robust tool for graph augmentation. Its ability to leverage global semantics and high-order relational information makes it an invaluable technique in the field of GCL by generating high-quality contrasting samples.

The diffusion process can be formulated as follows:

$$\mathbf{A}_{diff} = \sum_{k=0}^{\infty} \Theta_k \mathbf{T}^k \in \mathbb{R}^{n \times n}, \tag{2.7}$$

where $\mathbf{A}_{diff}$ denotes the adjacency matrix of the diffused graph, Θ_k is a coefficient determined by specific diffusion settings, $\mathbf{T}$ represents the transformation matrix derived from $\mathbf{A}$, and n is the number of nodes in the graph. There are multiple options for Θ_k and $\mathbf{T}$. For example, if Heat-Kernel diffusion [100] is adopted, then $\mathbf{T} = \mathbf{A}\mathbf{D}^{-1}$ and $\Theta_k = \frac{e^{-t}t^k}{k!}$, where D is the diagonal degree matrix of the original graph and t represents the diffusion time.

In summary, rule-based augmentation like graph diffusion offers a well-defined augmentation rules to process the graph to acquire global semantics and high-order relations

while avoiding random noise introduced by random augmentation strategies. MVGCL [35] is one of the most impactful research work adopting such augmentation strategy, revealing limitations of random augmentations and investigating different diffusion rules like Heat-Kernel [100] and PageRank [74] for graph augmentation.

2.2.1.3 Adaptive Augmentation

Adaptive augmentation [149] is a technique that initially identifies which elements within a graph are deemed non-essential based on specific graph metrics, such as centrality [6]. Once these less important elements are identified, they are subsequently perturbed to generate augmented views of the graph.

In contrast to this element-focused strategy, there exists another category of adaptive strategies that emphasize the augmentation process itself rather than the individual graph elements [131]. These strategies are grounded in the concept of automatic learning, wherein random augmentations are adaptively selected from a predefined strategy pool. This approach aims to optimize the augmentation process by dynamically choosing the most suitable augmentations, thereby enhancing the overall effectiveness of the graph processing. By focusing on not only the graph elements but also the augmentation techniques, these adaptive strategies offer a more flexible and potentially more powerful means of improving the performance of GCL.

In summary, adaptive augmentation strategies determines the augmentation process according to the specific context and characteristics of the input graph, which can fully utilizing the potential of conventional graph augmentation techniques [132]. In literature, GCA [149] is one of the first adaptive GCL method, which is built on GraphCL [132]. GCA adaptively selects non-essential graph elements to conduct perturbation by calculating graph metrics like centrality [6] to improve the augmentation quality. GraphCL-Auto [131] is also developed based on GraphCL, which employs automatic learning strategy to select proper augmentation techniques to process the input graph.

2.2.2 Limitations of Current Graph Augmentation Strategies

Based on the preceding discussion, current literature identifies three categories of graph augmentation strategies, including random augmentation [132, 78], rule-based augmentation [35], and adaptive augmentation [149, 131, 94]. However, each of these strategies has inherent limitations that hinder GCL from reaching optimal performance.

2.2.2.1 Limitations of Random Augmentation

One significant limitation of random augmentation strategies [132, 78] is their tendency to introduce noise into the original graph. This noise is generated during the process of creating perturbed contrasting samples. While the intention behind this perturbation is to enhance the robustness and generalizability of the model, the introduction of random noise can have unintended consequences. The primary issue with this random noise is its detrimental impact on the performance of downstream graph models [35]. When noise is added indiscriminately, it can obscure the underlying structure and relationships within the graph. This obfuscation makes it more challenging for the model to learn meaningful patterns and features, ultimately leading to suboptimal performance.

While random augmentation strategies offer a straightforward method for generating contrasting samples, their propensity to introduce noise poses a significant limitation. To address these challenges, it is essential to explore alternative augmentation strategies that minimize the introduction of random noise.

2.2.2.2 Limitations of Rule-based Augmentation

Rule-based augmentation strategies [54, 35] have been proposed to mitigate the impact of noise introduced by random augmentation strategies. These rule-based methods aim to provide a more structured and controlled approach to graph augmentation, thereby reducing the randomness and potential degradation in model performance. However, despite their advantages, rule-based strategies come with their own set of limitations.

One major drawback of rule-based augmentation strategies is their rigidity caused by no learnable parameters involved during graph augmentation [144]. These strategies are often designed with specific rules that may not be easily adaptable to different application scenarios. However, the diversity of graph data in real-world applications presents a challenge for rule-based strategies. Designing specific rules for each dataset is not only time-consuming but also impractical. For instance, a rule that works well for social network graphs may not be suitable for biological network graphs. This lack of flexibility makes it challenging to apply rule-based strategies across diverse datasets and domains.

Additionally, many rule-based strategies, such as matrix diffusion: Heat-Kernel diffusion [100] and PageRank diffusion [74], require significant computational resources. Matrix diffusion involves complex calculations that can be computationally intensive, making it difficult to incorporate these strategies into an end-to-end training process. This computational burden reduces the overall efficiency of GCL methods, as the time and resources

required for augmentation can become prohibitive.

2.2.2.3 Limitations of Adaptive Augmentation

Adaptive augmentation strategies offer a promising approach to improving graph augmentation by adaptively adjusting to the characteristics of the graph to be processed. These strategies can be broadly categorized into two types: graph element-based [149, 94] and strategy-based [131] adaptive augmentations.

Graph element-based adaptive augmentation [149, 94] focuses on identifying and removing less important elements in the graph based on certain graph metrics, such as node centrality [6] and edge importance [57]. By dropping less significant elements, the method aims to produce augmented views that retain the essential structure and information of the original graph. This approach leverages the inherent properties of the graph to guide the augmentation process, potentially leading to more meaningful and effective augmentations. However, the reliance on specific graph metrics can be problematic in real-world applications. Some graph metrics may not accurately reflect the semantics of the graph in complex real-world scenarios [43], leading to suboptimal augmentations. This lack of flexibility can limit the applicability of these strategies across diverse domains.

On the other hand, strategy-based adaptive augmentation follows the concept of automatic learning to adaptively select random augmentations from a predefined strategy pool [131]. By dynamically choosing the most suitable strategies for processing the original graph, it aims to enhance the overall augmentation process. This approach can be particularly useful in scenarios where the optimal augmentation strategy is not known a priori and needs to be determined through experimentation.

Although these adaptive strategies have shown improved performance, they still face one significant challenge that is the difficulty of incorporating both graph metric calculations and automatic augmentation selection into an end-to-end training paradigm. The computational complexity and resource requirements of these processes can hinder their seamless integration into the training pipeline.

2.2.2.4 A Common Limitation of Different Graph Augmentation Strategies

According to the literature review and summary regarding current GCL literature, a common limitation among existing graph augmentation strategies is their inability to process non-embedding features, such as text [119]. These strategies are primarily designed to handle graph structures and the embedding values of node features. This narrow focus restricts their applicability to a broader range of graph data types.

Graph data often come in various forms, including multi-modal data that incorporate different types of information. For instance, text-attributed graphs [119] are a common type of multi-modal graph where nodes are associated with textual information. In such cases, the textual content can provide valuable context and additional features that are crucial for accurate graph representation and analysis. However, current graph augmentation strategies fall short in effectively integrating and processing this textual information.

Though current graph augmentation strategies have made significant strides in processing graph structures and embedding values, their inability to handle non-embedding features like text remains a critical limitation. By developing more versatile and comprehensive augmentation techniques, researchers can unlock the full potential of GCL methods and improve their performance in real-world, multi-modal graph applications.

2.3 Applications of Graph Contrastive Learning in Recommendation Systems

To demonstrate the practical value of GCL methods, it is essential to apply them to real-world scenarios. One of the most suitable application areas for GCL methods, as previously discussed, is recommendation systems. This section provides a brief review of the applications of GCL in RS and highlights the limitations in the current research progress.

2.3.1 Implementations of Graph Contrastive Learning

Although there was limited research on the application of GCL in RS during the initial stages of GCL research, it has now become a trending topic. This surge in interest is largely due to the success of graph learning techniques in RS [40, 24].

SGL [113] proposes to utilize node-dropping and edge-dropping in random augmentation strategy [132] to generate multiple contrasting views of the user-item interaction graph. By contrasting those views, high-quality user and item embeddings can be acquired to facilitate the recommendation tasks. However, there are research works challenges the necessity of graph augmentation process in GCL for recommendation. SimGCL [136] proposes a graph augmentation-free strategy that applies random noise to directly augment user and item embeddings to acquire contrasting samples, which shows superior and robust performance compared to graph augmentation-based method [113, 136]. Based on SimGCL, a extremely simple version, XSimGCL [133], is proposed, which further simplifies SimGCL by integrating the embedding process and the augmentation process.

There is a substantial body of literature on augmentation techniques for GCL in RS, but a critical research perspective on GCL's training paradigm has been largely neglected. This oversight is evident not only in the methods previously introduced, such as SGL, SimGCL, and xSimGCL, but also in other promising approaches like QRec [134] and KGCL [129]. To advance research in GCL for RS, it is insufficient to focus solely on designing tailored graph augmentation protocols; the training paradigm must also be thoroughly investigated.

2.3.2 Limitations of Current Implementations

Recommendation systems represent one of the most suitable application scenarios for GCL methods. A successful implementation in this domain can significantly validate the practical value of GCL. The key to applying GCL to recommendation systems lies in converting user-item interactions and other auxiliary information into a graph structure. Additionally, constructing contrasting samples based on these interaction graphs is critical for the effective application of GCL methods. However, though sufficient research works exist in current literature [113, 136, 133, 134, 129], at the inception of this research area, studies on the application of GCL in recommendation systems regarding contrasting sample design were sparse. This gap underscored the necessity of pioneering research works to demonstrate how GCL methods could be effectively applied to recommendation scenarios. Such works would not only fill the existing research void but also provide valuable insights into the practical implementation of GCL in real-world applications.

Another crucial aspect that needs to be addressed is the training paradigm for GCL in recommendation scenarios. Current methods typically combine GCL objectives with downstream recommendation tasks and conduct joint training [113, 134, 129]. However, research works within this thesis has revealed that joint training requires meticulous hyperparameter tuning to balance the weights of the GCL objectives and the recommendation objectives within the overall training framework. Without careful adjustment, the performance can be suboptimal. Furthermore, it is important to note that GCL itself is fundamentally an unsupervised training paradigm, primarily aimed at pre-training. Such a gap between the characteristic of GCL and the joint training paradigm in current methods [113, 134, 129] suggests that there may be alternative training paradigms that could be more effective for recommendation systems. Inspired by these findings, it is critical to investigate and develop better training paradigms for GCL in recommendation scenarios. This exploration could lead to more robust and efficient models, ultimately enhancing the performance and applicability of GCL in RS.

2.4 Research Questions

According to the previous literature review and limitation analysis, four research questions belonging to two research scopes are summarized to guide the research within this thesis to advance the research progress in GCL domains.

For graph augmentation strategies in GCL, two primary challenges must be addressed, including enhancing flexibility and efficiency and expanding their applicability to handle non-embedding features such as text. Specifically:

- RQ1.1: How to conduct flexible and efficient graph augmentations that tackle the limitations in current strategies?

- RQ1.2: How to effectively augment non-embedding features, such as text, on the node in a graph to produce contrasting samples?

For GCL's applications in RS, two primary research questions arise, including tailored contrasting sample design and GCL's training paradigm for recommendations. Specifically:

- RQ2.1: How to produce contrasting samples tailored for the specific recommendation scenario by leveraging graph augmentation strategies?

- RQ2.2: What training strategy should be adopted for GCL in recommendations? Is current the end-to-end one with joint training good enough?

The four research questions fall into two primary areas, including graph augmentation strategies in GCL and the application of GCL in RS. Addressing these questions will not only enhance the methodology of GCL but also increase its practical value in real-world applications, thereby advancing research in GCL-related domains.

2.5 Other Related Works

Since certain methods and concepts beyond the scope of GCL are introduced in the following research work chapter as part of the proposed method or framework, this section briefly outlines the relevant research background and progress in the literature to facilitate a better understanding of the subsequent content.

2.5.1 Large Language Models for Graph Learning in LATEX-GCL

LLMs have garnered significant attention for their prowess in natural language processing tasks. However, their application in graph learning is an emerging field of research that holds great promise [17, 48]. The intersection of LLMs and graph data presents a promising avenue for enhancing various scientific disciplines, including cheminformatics [51], material informatics [66], bioinformatics [10], CV [83], and quantum computing [45]. By incorporating textual information with graph data, referred to as Text-attributed Graph (TAG), researchers can accelerate scientific discovery and analysis. This is particularly beneficial in domains where graphs are paired with critical text properties. For instance, in cheminformatics, the combination of molecular graphs with textual descriptions of chemical properties can lead to more accurate predictions and insights.

A comprehensive survey on the application of LLMs to graphs [48] categorizes the scenarios into three main types: pure graphs, text-rich graphs, and text-paired graphs. This categorization highlights the diverse contexts in which LLMs can be leveraged. Pure graphs consist solely of nodes and edges, while text-rich graphs include additional textual information associated with nodes or edges. Text-paired graphs, on the other hand, integrate extensive textual data with graph structures, offering a richer context for analysis. Several techniques have been proposed to explore the mutual enhancement between LLMs and graphs. One approach is to treat LLMs as task predictors, as demonstrated by Graphormer [128]. Another method involves using LLMs as feature encoders for Graph Neural Networks (GNNs), as seen in the TAPE framework [39]. Additionally, aligning LLMs with GNNs, as proposed in GLEM [143], offers a synergistic way to leverage the strengths of both models.

Despite the promising advancements, several challenges remain in this evolving field. Issues such as graph linearization, model optimization inefficiencies, and the need for generalizability and robustness of LLMs on graphs underscore the importance of further research [48]. Addressing these challenges will be crucial for realizing the full potential of LLMs in graph learning and their application across various scientific disciplines.

2.5.2 Multi-behavior Recommendation in HMG-CR

Multi-behavior recommendation systems leverage multiple user-item interactions to enhance the accuracy and relevance of recommendations for target behaviors. This approach recognizes that users often exhibit a variety of behaviors, such as viewing, purchasing, and rating items, which can provide richer information for recommendation algorithms. By integrating these diverse interactions, multi-behavior recommendation systems aim to offer

more personalized and effective suggestions.

Various approaches exist to utilize users' multi-behavior information effectively. One common method involves the use of multi-behavior interaction graphs. In these graphs, different types of user behaviors are represented as edges between nodes, with each edge type corresponding to a specific behavior. Studies such as [103, 86, 19] assign weights to these edges before performing aggregation, allowing the recommendation system to differentiate between the significance of various behaviors. Graph-based recommendation methods have shown strong performance in leveraging multi-behavior data. Techniques such as GAT [103] and RGCN [86] exploit the structural advantages of GNNs to capture complex relationships within the data. These methods have been particularly effective in recommendation tasks, as they can model the intricate dependencies between different user behaviors and items. In addition to graph-based methods, multi-task learning techniques have also been employed to enhance multi-behavior recommendation systems. For instance, studies like [26, 15] use multi-task learning to extract more supervision signals from multi-behavior data. This approach assumes that one behavior is strongly related to preceding behaviors and that embeddings of different user behaviors are adjacent in the embedding space. By learning from multiple tasks simultaneously, these models can capture more nuanced patterns in user behavior.

However, despite the advancements in both graph-based and multi-task learning methods, challenges remain in fully capturing the complex relationships among various types of user behaviors. The aggregation methods used in studies like [103, 86] and the assumptions made in [26, 15] may not be sufficient to model the intricate dependencies and interactions between different behaviors. Further research is needed to develop more sophisticated models that can better understand and utilize the rich information contained in multi-behavior data. In conclusion, while multi-behavior recommendation systems hold great promise for improving the accuracy and relevance of recommendations, there is still much work to be done. By continuing to explore and refine the methods for integrating and analyzing multi-behavior data, researchers can unlock new possibilities for personalized and effective recommendation systems.

2.5.3 Prompt-Tuning in CPTPP

Prompt-tuning is a novel and trending paradigm for pre-trained models in natural language processing (NLP). The core idea behind prompt-tuning is to re-formulate downstream tasks in a way that narrows the significant gap between these tasks and the pre-training objective [12, 92]. This approach aims to make the transition from pre-training

to fine-tuning more seamless and effective. There are two primary methods to achieve prompt-tuning [28]. The first method involves manually designing or searching for appropriate discrete prompts, often referred to as hard prompts [27, 47, 92]. While this method can be effective, it is also trivial and resource-intensive due to the vast search space and the need for expert knowledge in some application scenarios [115]. The complexity and resource demands of this approach can make it impractical for many applications.

To address the limitations of hard prompts, another line of methods focuses on generating continuous vector embeddings, known as soft prompts [30, 77]. These soft prompts are designed to be more flexible and less resource-intensive, as they do not require extensive manual effort or expert knowledge. By using continuous embeddings, soft prompts can adapt more easily to various tasks and datasets, making them a more scalable solution.

The application of prompt-tuning in recommendation systems has also been explored. For instance, the P5 framework [28] redefines recommendation tasks as NLP tasks and employs hard prompts to perform recommendations. This approach leverages the strengths of prompt-tuning to enhance the performance of recommendation systems. On the other hand, the PPR framework [115] adopts a soft-prompt and prefix strategy [61] to automatically generate personalized prompts for users in recommendation systems. These methods aim to provide more tailored and effective recommendations by utilizing soft prompts.

Despite these advancements, the integration of graph learning and its applications remains outside the current scope of prompt-tuning research. Additionally, most existing prompt learning methods require side information to produce high-quality prompts, which limits their applicability. The reliance on side information can constrain the use of prompt-tuning to scenarios where such information is readily available, thereby reducing its versatility. In conclusion, while prompt-tuning offers a promising approach for enhancing pre-trained models in NLP and recommendation systems, there are still challenges to be addressed. The need for side information and the current exclusion of graph learning applications highlight areas for future research. By continuing to explore and refine prompt-tuning methods, researchers can expand its applicability and effectiveness across a broader range of tasks and domains.

2.6 Future Directions

Despite the promising progress in current GCL-related research, several challenges remain that hinder the full realization of GCL methods' potential: I) Understanding the rationale and intuition behind graph learning methods is crucial for improving these methods and

applying them to real-world problems. However, most research on interpretability in graph learning focuses on supervised learning scenarios [16, 59, 19], leaving interpretability in GCL largely underexplored. There is an urgent need to develop GCL methods with high interpretability, which would enable their application in critical industries such as finance [3] and healthcare [7]. II) While GCL's applications in recommendation systems emerge nowadays, there are numerous other real-world scenarios to explore, such as finance [3], healthcare [7], and AI for science [14, 10]. The key to implementing GCL in these applications lies in designing augmentation strategies that effectively integrate domain-specific knowledge. Each application scenario inherently possesses specific priors that are crucial for models to understand the context. Relying solely on general graph augmentation strategies [132, 123, 35, 94] may prevent GCL methods from accurately capturing domain-specific semantics, leading to suboptimal performance. Therefore, it is essential to thoroughly investigate methods for integrating domain knowledge into GCL.

Both RQ1.1 and RQ1.2 focus on graph augmentation through the use of counterfactual mechanisms and LLMs, respectively. These methods will enhance the interpretability of GCL by providing insights into the augmentation process. This thesis, through RQ1.1 and RQ1.2, will contribute to advancing interpretability research in GCL. Additionally, RQ2.1 and RQ2.2 will explore the design of graph augmentation strategies and training paradigms within recommendation scenarios, offering valuable examples that can inspire research in other application domains.

2.7 Chapter Summary

This chapter provides a comprehensive literature review on GCL. It begins by detailing the procedures of GCL, offering an overview of common GCL methods. Next, it introduces graph augmentation strategies in GCL and explores GCL's applications in RS, highlighting current research progress. Following this, a detailed analysis of the limitations within these two research areas is presented. Finally, based on the literature review and limitation analysis, four research questions are formulated to guide the research presented in this thesis.

In the following chapter, the discussion will center on four research questions within two primary areas: graph augmentation strategies and the application of GCL in RS. It will present four research studies, each addressing one of these questions in detail.

CHAPTER 3

LEARNING-BASED GENERATION OF CONTRASTING SAMPLES

GCL has emerged as a powerful unsupervised graph representation learning tool. The key to the success of GCL is to acquire high-quality positive and negative samples as contrasting pairs to learn the underlying structural semantics of the input graph. Recent works usually sample negative samples from the same training batch with the positive samples or from an external irrelevant graph. However, a significant limitation lies in such strategies: the unavoidable problem of sampling false negative samples. In this paper, we propose a novel method to utilize **C**ounterfactual mechanism to generate artificial hard negative samples for **G**raph **C**ontrastive learning, namely **CGC**. Moreover, current graph augmentations have several limitations, as introduced in the literature review chapter, which also prevent GCL methods from achieving optimal performance. To address these issues,the counterfactual mechanism is utilized to produce hard negative samples, ensuring that the generated samples are similar but have labels that differ from the positive sample, which are helpful for the CL process. The proposed method achieves satisfying results on four datasets. It outperforms some traditional unsupervised graph learning methods and some SOTA GCL methods. Some supplementary experiments are also conducted to illustrate the details of the proposed method, including the performance of CGC with different hard negative samples and evaluations of different similarity measurements.

The content in this chapter focuses on RQ1.1 by proposing a novel GCL method to conduct flexible graph augmentation to generate high-quality contrasting samples, which is a critical step in the whole GCL procedure.

3.1 Brief Introduction to CGC

GCL [104, 78, 132, 149, 35, 122, 94] has emerged as a powerful learning paradigm for unsupervised graph representation learning. Inspired by the widely adopted CL framework in Computer Vision (CV) [116, 36] and Natural Language Processing (NLP) [18], GCL leverages the advanced representation learning capabilities of Graph Neural Network (GNN) [53, 118, 49, 24] and tries to distill high-quality representative graph embeddings of an input graph via comparing the differences and similarities among augmented graphs (*i.e.,* positive and negative samples) derived from the original input.

The key to a successful GCL method is to derive high-quality contrasting samples from the original input graph. To date, various kinds of methods to generate positive samples are proposed, for example, graph augmentations-based approaches [132, 149] and multi-view sample generation [35, 122], which have been becoming dominant and achieved satisfying performance. Despite this progress, especially in manipulating positive pairs, far less attention has been given to obtaining negative samples [85]. Compared to positive samples in CL, negative sampling is more challenging and non-trivial[85]. Existing methods of negative sample acquisition mainly follow traditional sampling techniques, which may encounter the deficiency caused by unnoticeable false negative samples [117]. For instance, GraphCL [132] samples other graphs as the negative samples from the same training batch where the target graph comes from. Such an approach does not guarantee that the sampled negative graphs are true. GCC [78] samples negative graphs from an external graph based on the assumption that common graph structural patterns are universal and transferable across different networks. However, this assumption neither has any theoretical guarantee nor has been validated by empirical study[78]. To alleviate the impact caused by false-negative samples, debiasing treatment has been introduced to current graph contrastive learning methods [117, 141]. The idea of these debiased graph contrastive learning methods is to estimate the probability of whether a negative sample is false. Based on this, some negative samples with low confidence will be discarded or treated with lower weights in the CL phase. Nevertheless, a typical major limitation of both GCL and the debiasing-based variants is still evident - most of these sampling strategies are stochastic and random. In other words, current methods do not guarantee the quality of the sampled negative pairs.

To address the previously mentioned problems, the high-quality negative sample is regarded as hard negative sample and the corresponding formal definition is given. According to [85], a hard negative sample is a data instance whose label differs from that of the target data, and its embedding is similar to that of the target data. Considering the limita-

tions of sampling-based strategies discussed previously, we argue that a strictly constrained generative process must be imposed to guarantee the quality of the negative samples (*i.e.*, generating hard negative samples). Inspired by counterfactual reasoning [60], a fundamental reasoning pattern of human beings, which helps people to reason out what minor behaviour changes may result in considerable differences in the final event outcome. We intuitively came up with the idea that the hard negative sample generation should apply minor changes to the target graph and finally can obtain a perturbed graph whose label is strictly different from the original graph.

To this end, two types of hard negative samples via perturbations to the original input graph, including proximity perturbed graphs and feature-masked graphs, are proposed. It is worth noting that these two types of generation processes will be adaptively conducted and constrained by sophisticating similarity-aware loss functions. However, this process is still challenging and non-trivial. We believe there are two significant challenges. First, in graph perturbation and feature masking, how to measure a generated sample is *hard*? To solve the problem, two indication matrices are designed to demonstrate the changes made to the graph structure and feature space. Then, different matrix norms are applied to indication matrices to reflect how much perturbation has been made. The calculated matrix norms will be minimized such that the perturbation applied to the original graph to generate negative samples is as minor as possible. In this case, the generated samples would be similar to the original graph in proximity and feature space. By adopting matrix norms, the perturbation can be quantified, ensuring the generated samples are *hard* ones.

After formulating a constraint that forces the generated samples to be hard to distinguish from the target in proximity and feature space, the second challenge is how to make sure the generated hard samples have different labels from the target. That is to say, how can we ensure the generated hard samples are *true negative*? The target graph and the generated samples will be first fed into a graph classifier. The classifier will then output the probability distributions of the classes to which the target graph and the generated samples belong. Following the counterfactual mechanism, an objective measuring the differences between the classifier's outputs for the target graph and that for the generated samples is applied and minimized. Specifically, the similarities between the predicted probability distributions are minimized by monitoring the KL Divergence. With the two objectives described above, high-quality hard negative samples with constraints can be generated.

In summary, a counterfactual-inspired generative method for GCL to obtain hard negative samples is proposed in this section. It explicitly introduces constraints to ensure the generated negative samples are true and hard, eliminating the random factors in current

negative sample acquiring methods in GCL methods. Furthermore, once the generation procedure is finished, we do not need further steps (*e.g.*, debiasing) to process the acquired samples. The contributions of our work are summarized as follows:

- A novel adaptively graph perturbation method, CGC, to produce high-quality hard negative samples for the GCL process is proposed in this section.
- Tthe counterfactual mechanism is introduced into the GCL domain, leveraging its advantages to make the generated negative samples be *hard* and *true*. Due to the successful application of the counterfactual mechanism, the potential feasibility of conducting counterfactual reasoning to explain GCL models is high in future works.
- Eextensive experiments are conducted to demonstrate the proposed method's effectiveness and properties, which achieved state-of-the-art performances compared to several classic graph embedding methods and some novel GCL methods.

3.2 Preliminaries about The Counterfactual Mechanism

Counterfactual reasoning is a basic way of reasoning that helps people understand their behaviours and the world's rules [60]. The definition of counterfactual reasoning is given by [25] stating that counterfactual is a probabilistic answer to a 'what would have happened if' question. Many illustrative examples are provided in [107] to help understand the ideas behind counterfactual. For instance, as shown in Figure 3.1. Someone wants to apply for a loan, but the application is rejected after a risk assessment from the financial institution. Many factors are related to the final decision, such as the applicant's age, income, and number of credit cards. The minimum change the applicant needs to make to get the loan is earning an extra $1,000 per month or cancelling two credit cards.

Algorithm 1: Heuristic counterfactual generation algorithm

```
sample a random instance as the initial x'
optimise L(x, x', y', λ) with x'
while |f̂(x') − y'| > ε do
    increase λ by step-size α
    optimise L(x, x', y', λ) with new x'
end while
return x'
```

Counterfactual is a kind of thinking mechanism to discover the facts that contradict existing facts and could potentially alter the outcomes of a decision-making process. There

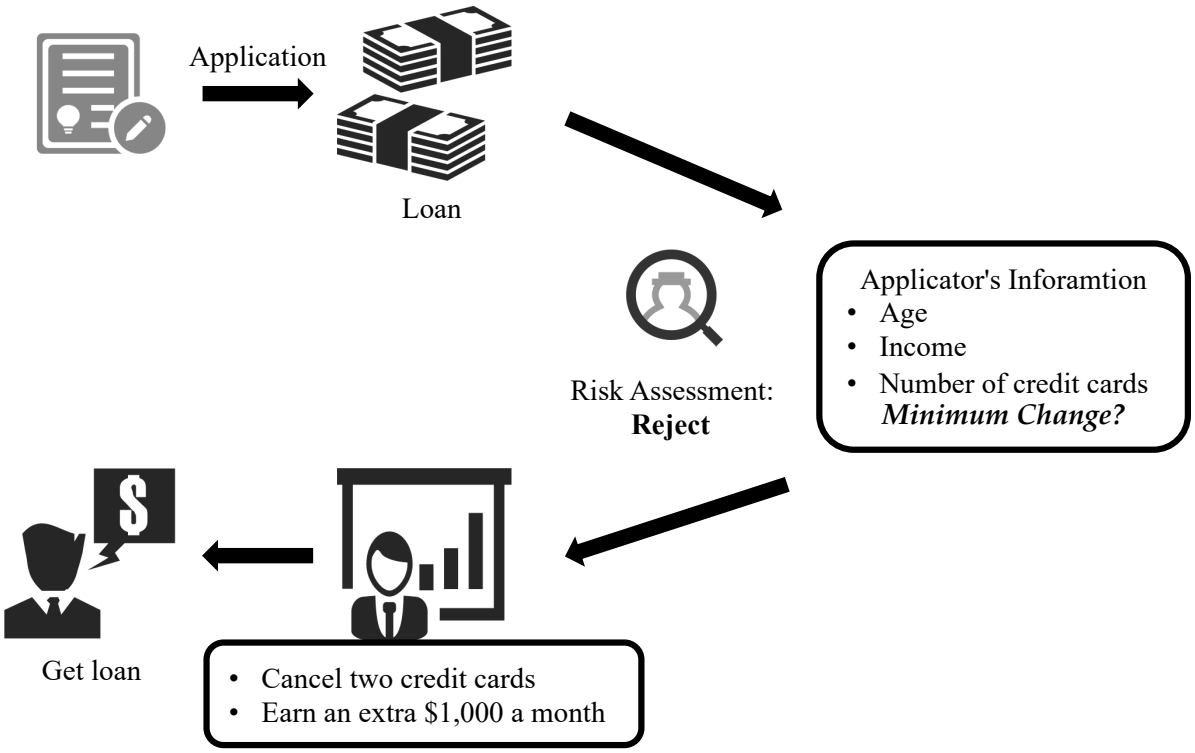


Figure 3.1: An illustrative example about a counterfactual explanation.

are some restrictions on counterfactuals. First, many factors could potentially affect the final results. However, counterfactuals must apply as small as possible changes to achieve such a goal. Second, counterfactuals must be feasible and reasonable. In Figure 3.1, the financial institution would release the loan without hesitation if the applicator earns an extra one million dollars per month. Nevertheless, the applicator cannot have such a high salary quickly. So, earning an extra one million dollars per month is not a counterfactual.

A classical counterfactual method is heuristic counterfactual generation [106], which is shown in Algorithm 1, where:

$$L(x, x', y', \lambda) = \lambda \overbrace{(\hat{f}(x') - y')^2}^{\text{distance in predictions}} + \underbrace{d(x, x')}_{\text{distance in instances}}, \tag{3.1}$$

and x denotes the target instance, x' is counterfactual, y' represents the desired outcome, λ is the term used to balance two distances, and ϵ denotes tolerance for the distance. This equation is the objective function of the heuristic counterfactual generation algorithm. It maximizes the distances in predictions and minimizes the distance between the original instance x and the counterfactual x'.

3.3 CGC Method Design

This section will give a detailed illustration of the proposed method and the training procedure. The overview of the proposed method is illustrated in Figure 3.2. The counterfactual hard negative sample generation is first conducted to acquire a proximity-perturbed and feature-masked sample. Then, the target and the two generated hard negative samples will be fed into the graph contrastive learning module to learn graph embeddings.

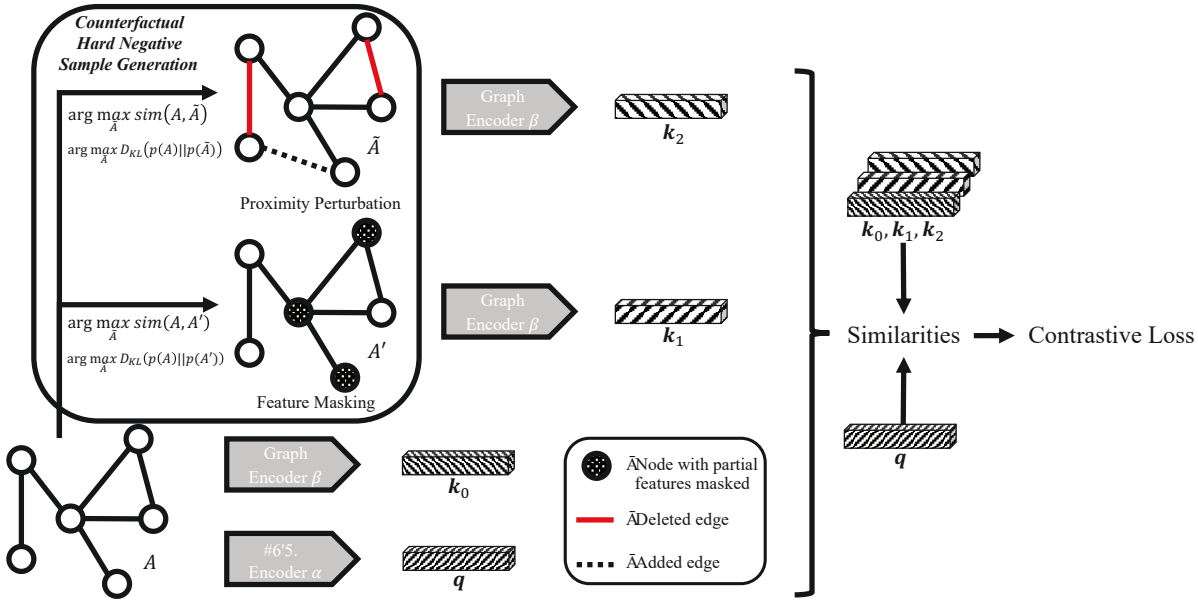


Figure 3.2: The overview of CGC.

3.3.1 Problem Definition

Given a graph $\mathscr{A} = \{\mathcal{V}, \mathcal{E}, \mathbf{X}\}$, where $\mathcal{V}$ denotes all the nodes, $\mathcal{E}$ represents all the edges, and $\mathbf{X}$ is the feature matrix consisting of the features of all nodes. If there are N nodes and the dimension of the feature is h, then, $\mathbf{X} \in \mathbb{R}^{N \times h}$. The proposed method aims to derive some negative graphs from the input graph based on counterfactual mechanisms. For simplicity's sake, in this paper, the scenario where two kinds of hard negative graphs are generated is taken into consideration, the proximity perturbed graph $\mathscr{A}' = \{\mathcal{V}, \mathcal{E}', \mathbf{X}\}$ and the feature masked graph $\tilde{\mathscr{A}} = \{\mathcal{V}, \mathcal{E}, \tilde{\mathbf{X}}\}$, such that:

$$\underset{\mathcal{E}', \tilde{\mathcal{X}}}{\arg\max}\, sim(\mathscr{A}, \mathscr{A}') + sim(\mathscr{A}, \tilde{\mathscr{A}}), \tag{3.2}$$

$$\underset{\mathcal{E}', \tilde{\mathcal{X}}}{\arg\max}\, D_{KL}(p(\mathscr{A}) || p(\mathscr{A}')) + D_{KL}(p(\mathscr{A}) || p(\tilde{\mathscr{A}})), \tag{3.3}$$

where $sim(*)$ denotes the metric to measure the similarity between two items (e.g., graph adjacency matrices, feature matrices), $D_{KL}(*)$ is the KL-Divergence [56] function, which is used to measure the similarity between two probability distributions, and $p(*)$ denotes predictor outputting the probabilities of classes to which the graph belongs. The intuition behind the two formulas is to derive hard negative graphs with different labels while forcing the derived graphs to be as similar to the original graph as possible. In other words, we want to achieve dramatic change at the semantics level with minor perturbations at the graph's essential elements (*e.g.,* edges, node features).

The problem above is formulated as an optimization problem to maximize the similarities between the generated negative samples and the target in proximity and feature and force them to have different labels.

3.3.2 Counterfactual Adaptive Perturbation

This section discusses two adaptive perturbation matrices: the proximity perturbation matrix and the feature masking matrix.

3.3.2.1 Proximity Perturbation

This aims to change the graph structure to generate a contrasting sample. This can help the model learn the critical structural information in the original graph [132, 149]. First, let us focus on how to conduct adaptive perturbation. To achieve the goal of adaptive perturbation, a trainable matrix $\mathbf{M}_a \in \mathbb{R}^{N\times N}$ is required, such that:

$$\mathbf{A}_a = \mathbf{M}_a \times \mathbf{A}, \tag{3.4}$$

where $\mathbf{A} \in \mathbb{R}^{N\times N}$ is the adjacency matrix of $\mathscr{A}$, and $\mathbf{A}_a$ denotes the adjacency matrix of the proximity perturbed graph $\mathscr{A}'$. Note that we adopt matrix multiplication here instead of taking the Hadamard product since it cannot add an edge to the adjacency matrix.

Moreover, values of the entries of $\mathbf{A}_a$ are in $\mathbb{R}^{N\times N}$, which conflict with the definition domain of the adjacency matrix, $\{0,1\}^{N\times N}$. An extra step is required such that $f : \mathbf{A}_a \in \mathbb{R}^{N\times N} \rightarrow \mathbf{A}_a^{'} \in \{0,1\}^{N\times N}$. Tthe following formula is used to conduct the mapping process:

$$\mathbf{A}_a^{'} = \mathbb{I}(sigmoid(\mathbf{A}_a) \geq \omega), \tag{3.5}$$

where $\mathbb{I}(*)$ is the indicator function, and ω is a threshold determining whether to set the entry as 1 or 0. Finally, we have the adjacency matrix for the negative sample $\mathscr{A}'$. Consequently, a perturbed set $\mathscr{E}'$ of edges is obtained.

Though in this procedure, there is no modification made to nodes, we can adaptively discard some nodes by deleting all the edges of the nodes, which will be isolated from the generated graph after perturbation. In summary, we can utilise the proximity perturbation matrix and procedure mentioned above to simulate all the proximity perturbation methods in [132], including edge dropping or adding and node dropping.

3.3.2.2 Feature Masking

This tries to mask the original feature matrix to help the contrastive learning model obtain the critical information in the features [132, 149], which would determine the decisive factors in the feature domain. The whole procedure of feature masking is similar to the proximity perturbation, but there are some minor differences. First, same as the previous procedure, we need to initialize a trainable matrix $\mathbf{M}_b \in \mathbb{R}^{N\times h}$, which serves as a mask matrix here.It is required to make sure the definition domain of it is $\{0,1\}^{N\times h}$. To meet such a requirement, the following process is needed:

$$\mathbf{M}_b^{'} = \mathbb{I}(sigmoid(\mathbf{M}_b) \geq \gamma), \tag{3.6}$$

where γ is a threshold determining whether to mask some feature entries.

To fulfil the feature masking procedure, the Hadamard element-wise product between $\mathbf{M}_b^{'}$ and $\mathbf{X}$ is needed instead of matrix multiplication in the previous perturbation:

$$\tilde{\mathscr{X}} = \mathbf{M}_b^{'} \circ \mathbf{X}. \tag{3.7}$$

Once the feature masking procedure is finished, the values of the masked feature entries will be replaced with 0.

After the proximity perturbation and the feature masking, two hard negative samples are acquired, which are $\mathscr{A}' = \{\mathscr{V}, \mathscr{E}', \mathbf{X}\}$ and $\tilde{\mathscr{A}} = \{\mathscr{V}, \mathscr{E}, \tilde{\mathbf{X}}\}$, respectively.

3.3.2.3 Perturbation Measurement

Once two perturbed graphs are obtained, the next step is how to ensure the generated graphs are hard negatives. As mentioned previously, the counterfactual mechanism is utilized to solve this problem because this method naturally meets the requirements of hard negative sample generation. Both of them aim to output something different at the semantic level but similar at the structural level.

First, we discuss maximising the similarity between the original and the perturbed graphs. This objective function tries to ensure the perturbation we made is as minor as possible. We

utilise the Frobenius norm of the difference between $\mathbf{A}$ and $\mathbf{A}'_a$. The smaller the Frobenius norm is, the more similar they are. The Frobenius norm of the masking matrix can measure the similarity between features. A relatively greater norm indicates that a small portion of feature entries are masked. Therefore, the similarity between $\mathbf{X}$ and $\tilde{\mathbf{X}}$ would be high. The objective is formulated as follows:

$$\mathscr{L}_s = ||\mathbf{A} - \mathbf{A}_a^{'}||_F - ||\mathbf{M}'_b||_F. \tag{3.8}$$

Next, it is required to ensure the generated graphs are different from the original graph at the semantic level. Here, we consider the classification problem, where we minimize the similarity between probability distributions of classes between the original and the perturbed graphs. Therefore, this part of objective can be formulated as:

$$\mathscr{L}_c = -D_{KL}(p(\mathscr{A}), p(\mathscr{A}')) - D_{KL}(p(\mathscr{A}), p(\tilde{\mathscr{A}})). \tag{3.9}$$

Finally, the overall objective for counterfactual pre-training for hard negative sample generation is the combination of the previous objectives shown as below:

$$\mathscr{L}_{pre} = \mathscr{L}_s + \mathscr{L}_c. \tag{3.10}$$

3.3.3 Contrastive Learning Procedure

The counterfactual mechanism is adopted to generate hard negative samples. After that, GCL between the original graph and the perturbed graphs will be conducted. In this work, a simple and widely-used GCL schema is adopted to conduct it, which is *dictionary look-up* method [78], shown in Figure 3.2.

Given an original input graph $\mathscr{A}$, two negative graphs $\mathscr{A}'$ and $\tilde{\mathscr{A}}$, and two graph encoders, $g_p(\cdot)$ and $g_n(\cdot)$, we will have a sort of graph embeddings: $\mathbf{q} = g_p(\mathscr{A})$, $\mathbf{k}_+ = \mathbf{k}_0 = g_n(\mathscr{A})$, $\mathbf{k}_1 = g_n(\mathscr{A}')$, and $\mathbf{k}_2 = g_n(\tilde{\mathscr{A}})$. Specifically, the target graph will be encoded by both graph encoders, and $g_n(\cdot)$ will only be used to encode the generated hard negative samples. *Dictionary look-up* method here tries to look up a single key (denoted by $\mathbf{k}_+$) that $\mathbf{q}$ matches in $\mathbb{K}$. Let $\mathbf{q}$ denotes the query key and $\mathbb{K} = \{\mathbf{k}_0, \mathbf{k}_1, \mathbf{k}_2\}$ be the dictionary.

InfoNCE in [102] is adopted to formulate CL procedure. Therefore, the training objective for the GCL phase can be formulated as:

$$\mathscr{L}_{contra} = -\log \frac{\exp(sim(\mathbf{q}, \mathbf{k}_+)/\tau)}{\sum_{t=0}^{|\mathscr{K}|-1} \exp(sim(\mathbf{q}, \mathbf{k}_t)/\tau)}, \tag{3.11}$$

where τ is the temperature hyperparameter.

After finishing the two training phases, including the counterfactual mechanism-based hard negative sample generation and the GCL process, the trained embeddings of all nodes and graphs can be obtained. The graph embeddings will be fed into a downstream prediction model to conduct graph classification and evaluate the trained embeddings' quality.

3.4 Experiments on CGC

Comparison experiments are conducted to show the superiority of the proposed method. Supplementary experimental results are given to analyze the properties of the proposed method. This section discloses sufficient experimental settings and datasets for readers to reproduce the experiments.

3.4.1 Experiment Setup

Detailed experiment setup is listed in this section, including datasets, baselines, and experimental settings to facilitate reproducibility.

Table 3.1: Statistics of four graph datasets for CGC experiments.

Dataset	Num. of Graphs	Avg. Num. of Nodes	Avg. Num. of Edges	Node Attr. Dim.	Num. of Classes
PROTEINS_full	1,113	39.06	72.82	29	2
FRANKENSTEIN	4,337	16.90	17.88	780	2
Synthie	400	95.00	172.93	15	4
ENZYMES	600	32.63	62.14	18	6

3.4.1.1 Datasets

To fully demonstrate the performances of the proposed method compared to baselines, we choose several public and widely-used datasets from TUDataset [70]. All the datasets are available on the webpage[1]. Recall that the feature masking operation necessary for our proposed method is a hard negative sample generation procedure. Hence, the graph datasets we use must contain high-quality node features. We select four datasets, which are PROTEINS_full [10, 87], FRANKENSTEIN [73], Synthie [71], and ENZYMES [10, 87]. The detailed statistics of four datasets are shown in Table 3.1.

[1] https://ls11-www.cs.tu-dortmund.de/staff/morris/graphkerneldatasets

3.4.1.2 Baselines

To verify the effectiveness and superiority of the proposed framework, we compare it with several unsupervised learning methods in three categories: graph kernels, graph embedding methods, and GCL methods. For graph kernel methods, we choose four different kernels, including **RandomWalk Kernel** [105], **ShortestPath Kernel** [8], **Graphlet Kernel** [89], and **Weisfeiler-Lehman Kernel** [88]. For graph embedding methods, we select two methods, including **sub2vec** [1] and **graph2vec** [72]. Since the proposed method in this paper belongs to GCL, it is important to compare our method to current state-of-the-art GCL methods. Four impactful methods are selected in the literatur, including **InfoGraph** [94], **MVGCL** [35], **GraphCL** [132], and **GCA** [149].

3.4.1.3 Settings

For reproducibility, the detailed settings of the proposed method are introduced in this section. For PROTEINS_full, FRANKENSTEIN, and Synthie, GCN [53] with three layers is adopted as the graph encoder. The learning rates for hard negative sample generation and CL are 0.0001. The training epochs for the two training stages are 80 and 30, respectively. For dataset ENZYMES, a 2-layer GIN [118] is adopted as the graph encoder. The learning rates for hard negative sample generation and CL are 0.001. The training epochs for both training stages are 100. The batch sizes for all the experiments are set to 256, while 128 is also feasible if GPU memory is limited for large graphs such as Synthie. The threshold ω and γ mentioned previously are both 0.3. As to the temperature hyperparameter for CL, it is set to 1 for all the experiments. The proposed method is evaluated via graph classification under the linear evaluation protocol. Specifically, we closely follow the evaluation protocol in InfoGraph and report the mean 10-fold cross-validation F1-Micro and F1-macro scores with standard deviation output by a linear SVM. The SVM is implemented via scikit-learn[2].

3.4.2 Comparison Experiment

The comparison experiment results for all baselines and our proposed method on all four datasets are shown in Table 3.2. Generally, the proposed method outperforms the best baselines on all the datasets except PROTEINS_full. Though our method has a mean F1-macro score lower than that of GraphCL, the gap between its F1-macro score and ours is insignificant as the standard deviation exists. We note that the proposed method significantly improves the dataset Synthie and ENZYMES. According to Table 3.1, both of these datasets

[2]https://scikit-learn.org/

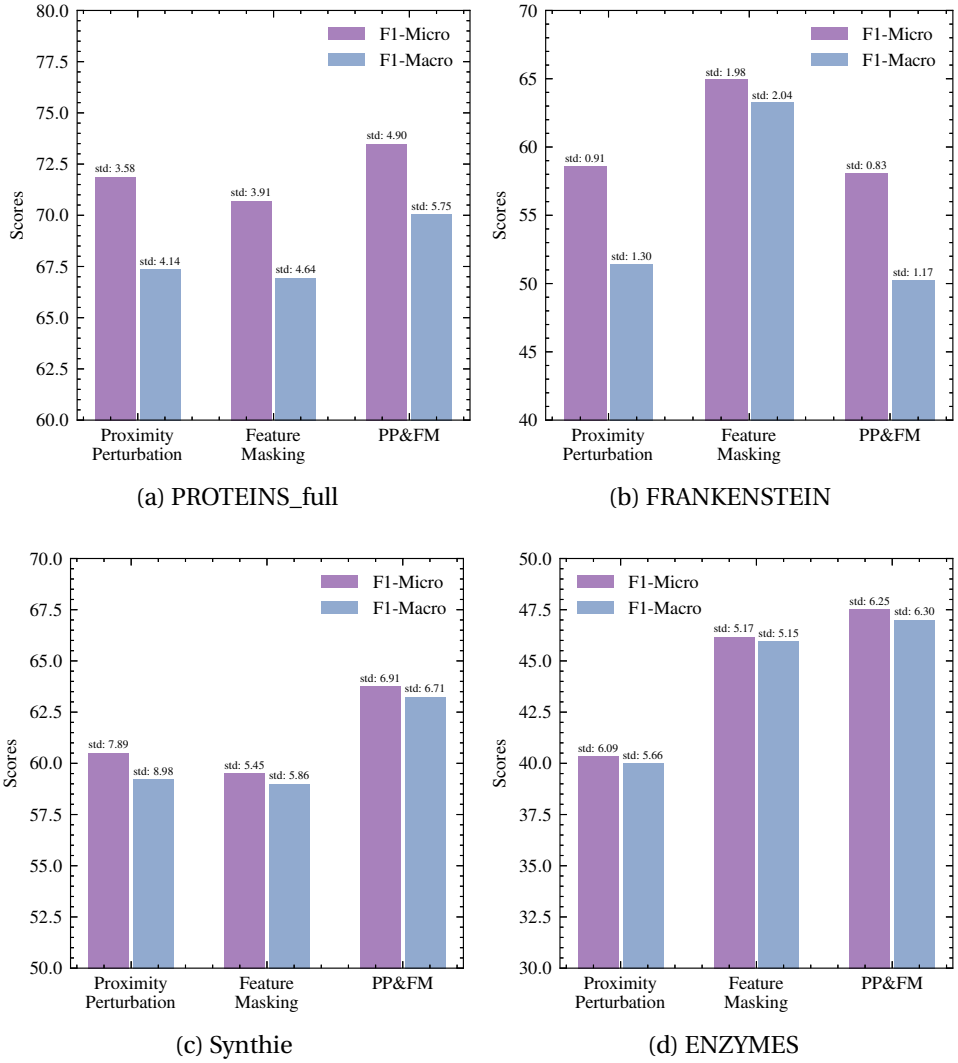
Table 3.2: Comparison experiment results of CGC.

Method \ Dataset	PROTEINS_full		FRANKENSTEIN		Synthie		ENZYMES	
	F1-Micro	F1-Macro	F1-Micro	F1-Macro	F1-Micro	F1-Macro	F1-Micro	F1-Macro
RandomWalk	-	-	57.97(std 2.15)	57.45(std 1.92)	18.50(std 4.06)	16.86(std 3.58)	-	-
ShortestPath	70.88(std 4.91)	69.88(std 4.99)	62.39(std 1.95)	59.81(std 2.02)	50.75(std 9.69)	47.32(std 9.86)	27.83(std 6.37)	27.18(std 5.93)
GL	69.89(std 3.25)	68.57(std 3.43)	61.26(std 2.85)	53.94(std 2.46)	52.50(std 10.49)	50.24(std 10.37)	31.67(std 7.03)	30.02(std 7.13)
WL	72.32(std 3.11)	71.36(std 3.41)	-	-	-	-	37.83(std 4.95)	36.42(std 5.78)
sub2vec	70.17(std 2.06)	66.26(std 0.44)	54.97(std 1.80)	46.83(std 4.00)	29.75(std 4.67)	22.07(std 3.75)	19.67(std 3.64)	13.34(std 4.33)
graph2vec	68.65(std 3.45)	64.16(std 5.00)	61.70(std 3.04)	59.68(std 0.22)	54.25(std 0.62)	35.17(std 0.26)	25.67(std 4.84)	22.41(std 5.04)
InfoGraph	71.61(std 4.67)	70.48(std 5.06)	63.57(std 2.12)	62.95(std 2.20)	54.5(std 8.05)	54.17(std 7.87)	38.33(std 7.03)	37.07(std 6.89)
MVGCL	72.06(std 3.29)	69.53(std 3.61)	61.89(std 1.40)	59.65(std 1.50)	62.00(std 9.07)	61.59(std 9.52)	40.50(std 7.85)	38.7(std 9.12)
GraphCL	73.05(std 3.29)	**71.04(std 3.35)**	62.62(std 2.49)	61.89(std 2.57)	57.50(std 9.08)	55.87(std 8.87)	33.67(std 4.58)	33.46(std 4.96)
GCA	71.71(std 4.40)	69.59(std 4.44)	63.20(std 1.70)	62.17(std 1.57)	52.25(std 5.18)	43.27(std 9.85)	34.00(std 5.01)	33.62(std 5.01)
CGC	**73.48(std 4.90)**	70.03(std 5.75)	**64.93(std 1.98)**	**63.25(std 2.04)**	**63.75(std 6.91)**	**63.23(std 6.71)**	**47.50(std 6.25)**	**46.99(std 6.30)**

have multiple classes, which are 4 and 6, respectively. It indicates that the proposed method is superior in multiclass graph classification tasks. Recall one of the training objectives of hard negative samples generation, Equation (3.9), which minimizes the similarity between the probability distributions of which class the original graph and hard negative samples are. If there were a multiclass classification task, the Equation (3.9) would minimize the similarity between two vectors (the vector refers to the probability distribution in our context) with more dimensions. Comparing two vectors with higher dimensionality could help the model to learn more information. So, it is reasonable that the proposed method has advantages on dataset Synthie and ENZYMES.

Graph kernel methods also achieve better performance than novel neural network methods. Nevertheless, some of them spend time on computing. Compared to our proposed method, they cannot be accelerated by GPUs, which is unaffordable under some real-world scenarios. Sub2Vec and Graph2Vec are two impactful graph embedding methods, which both leverage the idea of Word2Vec [68]. According to the experiment results, all of them cannot compete with the GCL methods, which is a novel and effective unsupervised graph learning paradigm with significant superiority. Note that four GCL methods are selected as baselines. All of them are impactful methods in the GCL domain. InfoGraph is one of the first methods to introduce the idea of contrastive learning into the graph representation learning area. It achieved promising performances on several graph learning tasks. As shown in Table 3.2, it has satisfying results on dataset PROTEINS_full and FRANKENSTEIN. However, it may not be compatible with multiclass graph classification tasks, as the proposed CGC significantly outperforms it. Conversely, MVGCL performs better on Synthie and ENZYMES than on the other two datasets with only two classes. GCA is an updated version of GraphCL, and they share the same framework, but the improvement achieved by GCA is not significant. It has minor improvement on the dataset FRANKENSTEIN and ENZYMES. On dataset Synthie, it even has much worse performances. GraphCL and GCA

try to conduct proper perturbations on the target to have positive or negative samples to form contrasting pairs. Specifically, GraphCL follows a random setting to perturb the graph, which cannot ensure the quality of the generated samples. GCA tries to adaptively locate the essential elements in the graph and perturb such identified elements according to their centrality. However, elements with high centrality are not always the critical factor determining the labels or semantics of the graph. Compared to our counterfactual hard negative samples generation method, these two methods have limitations in contrasting pairs generation. It is worth noting that GraphCL and GCA are incompatible with multiclass graph classification tasks. This is because the implementations of GraphCL and GCA both take InfoGraph as the backbone. It is reasonable for them to have such a phenomenon.



(a) PROTEINS_full (b) FRANKENSTEIN

(c) Synthie (d) ENZYMES

Figure 3.3: Graph classification results of CGC with different hard negative samples.

3.4.3 Ablation Study

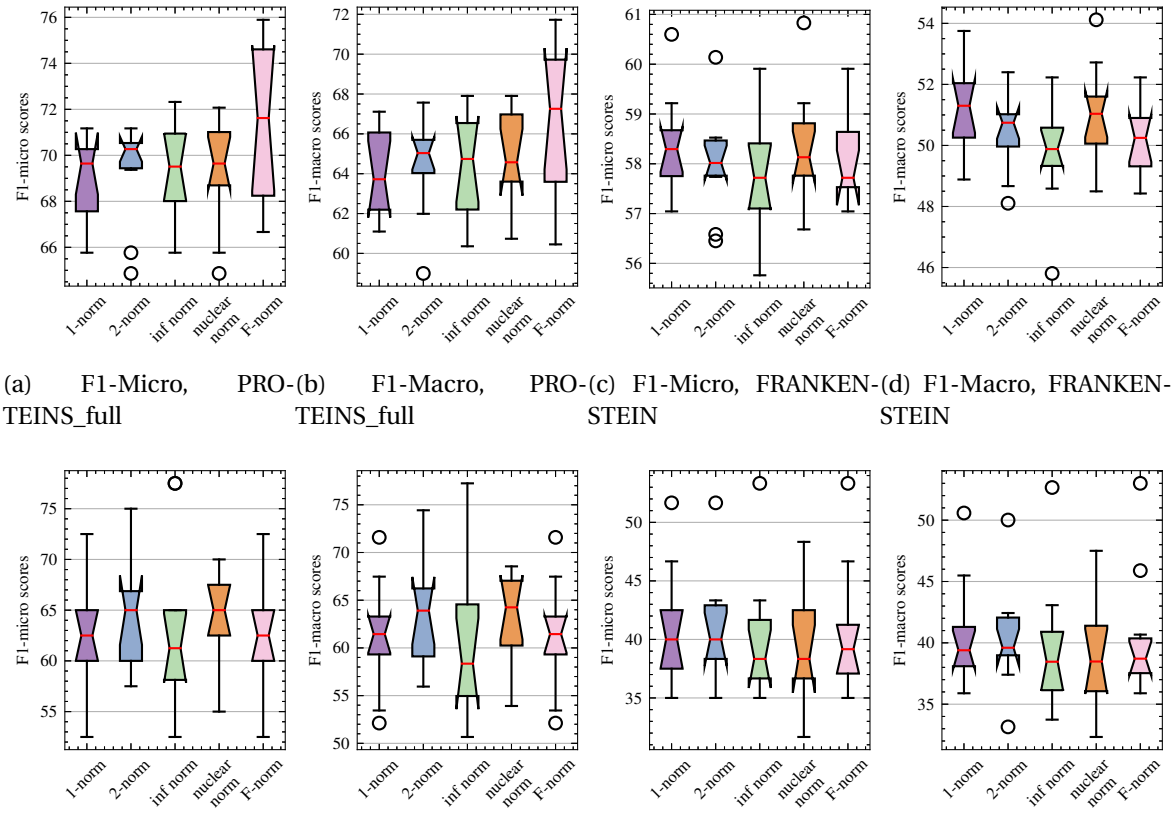
The impact on the GCL with different types of generated hard negative samples. Recall that two different types of hard negative samples are proposed. This section aims to find out how much improvement can be brought by different hard negative samples. Several experiments are conducted on the proposed method. We proceed with the GCL procedure under three scenarios, which are 1) only to use the proximity perturbed graphs as a negative sample, 2) only to use the feature-masked graphs as negative samples, and 3) utilizing both types of graphs as negative samples. The results of the experiments are illustrated in Figure 3.3. Utilizing both types of negative samples can achieve the best performances on all the datasets except FRANKENSTEIN. To achieve better results, utilizing two hard negative samples can help the model capture the key semantics in proximity and feature space simultaneously. Moreover, we can form more contrasting pairs with more negative samples. Hence, the model can receive sufficient self-supervised signals to update parameters and consequently perform better.

On dataset FRANKENSTEIN, the experiment results are not as expected. The model trained only with the feature-masked graphs achieved the best performance. There is a significant gap between the performance of the model trained only with the proximity perturbed graphs and the model trained only with the feature-masked graphs. Such a gap makes the collaboration of two types of negative samples unsatisfying, resulting in the worse performance of the model trained with both types of generated negative samples. Though the gap between the mean F1 scores of the model trained only with the proximity perturbed graphs and the model trained only with the feature-masked graphs on dataset ENZYMES is also significant, we note there is a larger standard deviation in the experimental results on the dataset ENZYMES. In this case, such a phenomenon indicates that the differences between the performances of the model trained only with the proximity perturbed graphs and the model trained only with the feature-masked graphs on dataset ENZYMES are not as significant as that on dataset FRANKENSTEIN. According to Table 3.1, graphs in dataset FRANKENSTEIN have much fewer nodes and edges than the other three datasets, but they have significantly larger node feature dimensionality. Masking features can bring more advantages to the model on dataset FRANKENSTEIN since the feature matrices are more complicated than the adjacency matrices. Such imbalance results in a considerable gap between the performances of the model trained only with the proximity perturbed graphs and the model trained only with the feature-masked graphs on dataset FRANKENSTEIN. We claim that perturbation to the aspects containing more informative semantics would bring more advantages to GCL. Similar phenomena appears in the rest datasets. For

example, graphs in dataset PROTEINS_full and Synthie have complicated adjacency with simple features. On these two datasets, the model trained only with the feature-masked graphs outperforms the model trained only with the proximity perturbed graphs.

Table 3.3: Analysis of five types of matrix norms in CGC.

Matrix Norm	Definition	Complexity
1-norm	$\|\|M\|\|_1 = \max_{1\le j\le n}\sum_{i=1}^{m}\|m_{ij}\|$	$\mathcal{O}(mn)$
2-norm	$\|\|M\|\|_2 = \sqrt{\lambda_{max}(M^*M)}$	$\mathcal{O}(m^3)$
inf norm	$\|\|M\|\|_\infty = \max_{1\le i\le m}\sum_{j=1}^{n}\|m_{ij}\|$	$\mathcal{O}(mn)$
nuclear norm	$\|\|M\|\|_* = tr(\sqrt{M^TM})$	$\mathcal{O}(mn^2)$
F-norm	$\|\|M\|\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}\|m_{ij}\|^2}$	$\mathcal{O}(mn)$



(a) F1-Micro, PROTEINS_full (b) F1-Macro, PROTEINS_full (c) F1-Micro, FRANKENSTEIN (d) F1-Macro, FRANKENSTEIN

(e) F1-Micro, Synthie (f) F1-Macro, Synthie (g) F1-Micro, ENZYMES (h) F1-Macro, ENZYMES

Figure 3.4: The performances of CGC with different matrix norms.

How to measure the similarity in hard negative samples generation procedure? Ensur-

ing the generated negative samples have similar forms to the original input in proximity and feature space is the key to making the negative samples be *hard*. A proper similarity measurement is important to achieve such a goal. In the methodology section, we introduced that we measure the similarity between the original input and the generated negative samples via calculating the norms of difference matrices $||A - A'_a||$ and M'_b. However, there are many different matrix norms. In this section, we examine the performances of the model trained with the negative samples in which different matrix norms were applied. We consider five different matrix norms shown in Table 3.3, and the experimental results are illustrated in Figure 3.4.

3.5 Summary of CGC

In this chapter, a novel GCL method, named CGC, is proposed to generate hard negative samples to improve GCL performance, which aims to address RQ1.1. Compared to current GCL methods and some classical graph kernel and graph embedding methods, it achieved the SOTA performances in most cases. The effectiveness of the model trained with different types of generated hard negative samples is also extensively studied. It is worth noting that perturbation made on the more complicated part of the graph data (*e.g.,* node features or proximity) would bring more advantages to the following CL procedure. Furthermore, this section explores how to choose similarity measurement for hard negative sample generation from a perspective of matrix norm. There will be more methods to conduct such a task, and it would be interesting future work to improve the proposed CGC method.

The next chapter goes beyond merely improving the current methodology for constructing contrasting samples. It addresses a common limitation among existing methods: the inability to directly augment non-embedding features in graphs, as summarized in RQ1.2. To expand the potential application scenarios of GCL techniques, it is essential to enable GCL to process non-embedding features for constructing contrasting samples.

CHAPTER 4

LARGE LANGUAGE MODELS-BASED DATA AUGMENTATION

GCL is a potent paradigm for self-supervised graph learning that has attracted attention across various application scenarios. However, GCL for learning on Text-attributed Graph (TAG) has yet to be explored. Because conventional augmentation techniques like feature embedding masking cannot directly process textual attributes on TAGs. A naive strategy for applying GCL to TAGs is to encode the textual attributes into feature embeddings via a language model and then feed the embeddings into the following GCL module for processing. Such a strategy faces three key challenges: I) failure to avoid information loss, II) semantic loss during the text encoding phase, and III) implicit augmentation constraints that lead to uncontrollable and incomprehensible results. In this paper, we propose a novel GCL framework named LATEX-GCL to utilize LLMs to produce textual augmentations and LLMs' powerful NLP abilities to address the three limitations aforementioned to pave the way for applying GCL to TAG tasks. Extensive experiments on four high-quality TAG datasets illustrate the superiority of the proposed LATEX-GCL method.

This chapter presents an LLM-based GCL framework to address RQ1.2. The proposed novel framework can not only conduct flexible graph augmentations with tailored but also enable GCL methods to process non-embedding features like text in TAGs.

4.1 Brief Introduction to LATEX-GCL

In numerous real-world scenarios, graph data is often enriched with textual attributes, for instance, user-item interaction graphs in recommendation systems that include tex-

tual user profiles and product descriptions [38, 67]. This type of graph data is referred to as TAGs [17]. More than recommendation systems, the application scenarios of TAGs also include bioinformatics [9], CV [84], and quantum computing [45]. The development of effective methodologies for processing and analyzing TAGs is crucial for advancing applications that rely on such data. With the advent of graph learning techniques, a variety of paradigms have been introduced. Notably, GCL [104, 35, 122] has gained prominence as a powerful self-supervised technique for graph representation learning, capitalizing on the benefits of self-supervision in cases of lacking sufficient labels. Current GCL approaches typically employ perturbations to manipulate graph structures and feature embeddings, thereby generating contrasting samples for GCL [132, 131, 149, 123]. Despite the diversity of these strategies, they fall short in directly augmenting the textual attributes inherent in TAGs. Consequently, there is a pressing need to devise a framework that synergizes GCL with TAGs, potentially enhancing the performance of graph learning tasks within TAG application scenarios by harnessing the strengths of GCL techniques.

Despite the advancements in GCL, the literature reveals a gap in the development of GCL methodologies specifically tailored for TAG settings [17, 48, 119]. An initial attempt to address this, referred to as Topological Contrastive Learning (TCL) for TAGs, is outlined in [119]. This approach begins by encoding textual attributes into feature embeddings for each node. Subsequently, it employs conventional GCL augmentations such as feature masking and proximity perturbation [132] to process the graph, followed by the execution of the remaining GCL steps in sequence. While this rudimentary approach enables the adaptation of GCL to TAG settings, it is not without significant drawbacks that could potentially compromise its effectiveness. There are three limitations lie ahead: I) **Information Loss.** Existing research [35] has identified information loss as a significant issue during the augmentation phase of conventional GCL methods, attributable to randomness and noise inherent in these processes. Adhering to the aforementioned rudimentary pipeline and employing standard random-based augmentation techniques, such as feature masking, inevitably leads to this loss of information. To enhance the performance of graph models within the GCL framework, it is imperative to implement strategies that mitigate such information loss. II) **Incapable Language Models.** The encoding of textual attributes in TAGs presents challenges when using both shallow text embedding methods, such as bag-of-words [34] and skip-gram [69], and advanced deep language models like BERT [20], DeBERTa [37], and GPT-2 [79]. Shallow embedding methods are constrained by their limited capacity to capture nuanced semantic features, whereas deep language models, despite their sophistication, fall short in complex reasoning tasks [17]. The reliance on these in-

adequate language models for the text encoding phase leads to an inevitable semantic degradation contained in the original textual attributes. III) **Implicit Constraint on Augmentations.** Conventional GCL methods [104, 132], as well as those having sophisticated adaptive augmentation strategies [149, 123] employed, share a fundamental challenge: the absence of explicit constraints on the augmentation process. This deficiency hinders users from monitoring and comprehending the effects of augmentation techniques, leading to augmented outcomes that are both uncontrollable and incomprehensible.

To overcome the aforementioned limitations, we introduce a novel approach named LATEX-GCL that employs an LLM to generate auxiliary texts, which act as augmented textual attributes for GCL applied to TAGs. This method circumvents the information loss associated with conventional feature augmentation techniques (*e.g.,* random feature masking). Thanks to the general knowledge contained in the LLMs [75], our strategy effectively enriches the semantics of the original text via the LLM-based augmentation, compensating for potential semantic deficits incurred during the text encoding phase. Furthermore, the utilization of LLMs involves natural language inputs, carefully crafted prompts to steer the augmentation process, and outputs that are inherently understandable for human beings. This process ensures that the augmentation constraints and results are explicit and comprehensible, enhancing the transparency and control over the augmentation. Nevertheless, employing LLMs for textual attribute augmentation in GCL is challenging, as there is a dearth of precedents in the literature to guide such an application. In this section, a suite of prompts for textual attribute augmentation using LLMs are proposed by drawing inspiration from the foundational principles of conventional graph augmentations as cataloged in GraphCL [132], including *shorten*, *rewriting*, and *expansion*, to facilitate the LLM-based textual attribute augmentation process.

In short, to address the limitations in current methods and better adapt GCL techniques to TAG settings, I) a novel GCL framework that can leverage the advantages of LLMs to conduct textual attribute augmentation is proposed, II) three types of LLM-based textual attribute augmentations are seminally summarized and the related prompt designs are listed, and III) comprehensive experiments are conducted to illustrate the performance and verify the effectiveness of the proposed LATEX-GCL method.

4.2 Preliminaries and Notations about LATEX-GCL

Before giving detailed descriptions of the proposed method, some necessary notations and formulations related to TAGs, LLMs, the text encoder, and the graph encoder are listed here.

Text-Attributed Graphs. Technically, a TAG can be defined as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \{t_n\}_{n \in \mathcal{V}}\}$, where $\mathcal{V}$ is the set of all nodes, $\mathcal{E}$ is the set of all existing links between the nodes in $\mathcal{V}$, and t_n is a sequence of text attributes associated with the n-th node. To facilitate the presentation of a graph, an adjacency matrix $\mathbf{A} \in \{0,1\}^{N \times N}$, where N is the number of nodes, is adopted to demonstrate nodes and links.

Large Language Model as Augmentor. In the proposed method, an LLM is applied as an augmentor to augment the original text attributes in the given TAG guided by the properly designed prompt. In this section, the $LLM(\cdot)$ is used to denote this augmentor. Given the original text attribute t_n and the prompt p, we can have the prompted text attribute $\hat{t}_n$. The augmentor $LLM(\cdot)$ finally takes the prompted text attribute $\hat{t}_n$ to output o_n.

Text Attribute Encoder. To facilitate the utilization of the original and the augmented text attributes, a text encoder, such as BERT [20] and DeBERTa [37], is required to obtain feature embeddings. Specifically, $LM(\cdot)$ is used to denote the text encoder, which takes the original text attribute t_n or the augmented text attribute o_n as the input to produce feature embedding $\mathbf{h}_n$. Then, the feature embeddings of all the nodes are concatenated to construct the overall feature matrix $\mathbf{H}$.

Graph Encoder. A GNN model, such as GCN [53], is implemented to serve as the graph encoder to capture the graph structure information. The graph encoder takes the adjacency matrix and the feature matrix as the inputs to update the feature matrix iteratively, where $g(\cdot,\cdot)$ denotes the graph encoder. A K-layer graph encoder $g(\cdot,\cdot)$ will output $\mathbf{H}^{(K)}$ at the last layer as the final feature embedding matrix.

4.3 LATEX-GCL Framework

This section illustrates the details of the LATEX-GCL method, starting with the preliminaries, followed by the descriptions for each module, including I) LLM-based text feature augmentation, II) text attribute encoding, III) graph encoding, and IV)GCL module, which is demonstrated in Figure 4.1 below.

4.3.1 Large Language Model-Based Text Feature Augmentation

An LLM is adopted in our proposed method LATEX-GCL as an augmentor to conduct augmentations on the original textual attributes in the input TAG. Adopting the LLM aims to effectively address the three limitations in the aforementioned rudimentary TCL strategy [119] in the introduction section, including information loss, incapable language models,

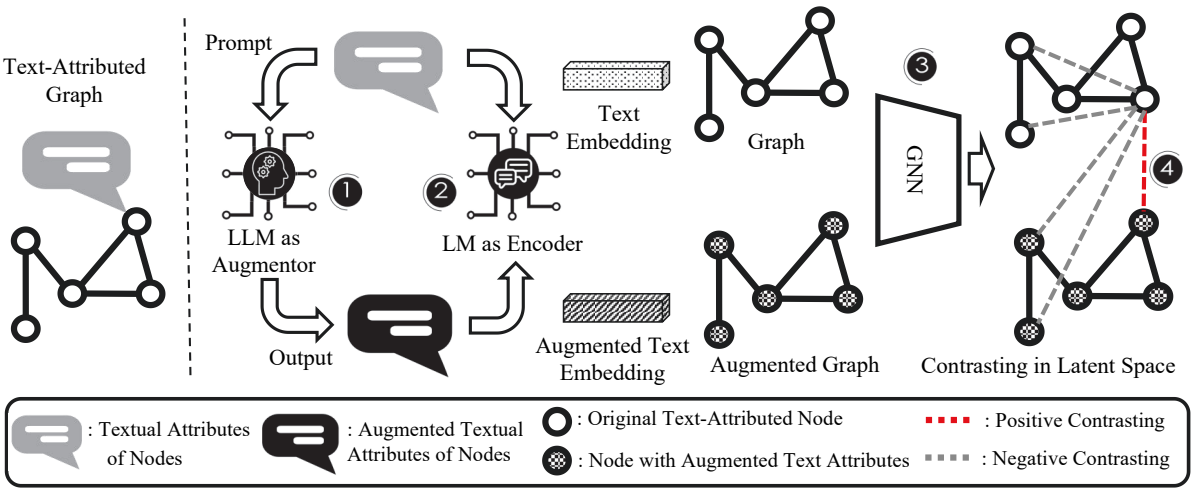


Figure 4.1: The overview of LATEX-GCL.

and implicit constraints on the augmentation process. However, the adoption of the LLM is non-trivial. A dearth of precedents in the current literature guides how to prompt the LLM to acquire proper augmented texts for the following GCL procedures. In this section, we innovatively propose and summarize a suite of prompts in order to employ the LLM to conduct textual attribute augmentations tailored for GCL on TAGs.

Table 4.1: Augmentation strategies in LATEX-GCL.

Augmentation	Prompt Design	Underlying Prior
Shorten	**Request:** The following content is the description of {XXX}. *Please simplify and summarize the provided content in one short sentence.* **Content:** {.....}	The shorten augmentation can help filter out the redundant contents and maintain the key information.
Rewriting	**Request:** The following content is the description of {XXX}. *Please rewrite the provided content to improve the spelling, grammar, clarity, concision, logical coherence, and overall readability.* **Content:** {.....}	The rewriting augmentation can help identify the invariant semantics contained in the original texts.
Expansion	**Request:** The following content is the description of {XXX}. *Please expand the provided content to give more related and necessary information.* **Content:** {.....}	The expansion augmentation can help introduce auxiliary information to enrich the original text features.

The paradigm of the LLM is known as 'pre-train, prompt, and output' [17], which is different from the existing language models. An LLM is normally trained on large-scale text corpora and possesses massive general knowledge [17, 75]. A properly designed prompt is required to help the LLM output the desired content from the massive knowledge. The prompt has various forms, such as several words or a sentence, and can include additional information to guide and constrain the output of the LLM [39].

Formally, let t_n be the original text attributes of a node and p denote the prompt to be placed in front of t_n, the prompted textual attributes after tokenization can be formalized as $\hat{t}_n = (p_1, p_2, \cdots, p_a, t_{n,1}, t_{n,2}, \cdots, t_{n,b})$. The LLM-based augmentor $LLM(\cdot)$ is trained to as-

sign a probability to each possible output $o_n = (o_{n,1}, o_{n,2}, \cdots, o_{n,c})$ that consists of c tokens, where the most satisfactory output is expected to have the largest probability value. The probability of the output o given t_n can be formalized as:

$$p(o_n|\hat{t}_n) = \prod_{i=1}^{b} p(o_{n,i}|o_{n,<i}, \hat{t}_n). \tag{4.1}$$

To guide the LLM-based augmentor $LLM(\cdot)$ to adapt to the scenario of text-attributed GCL, three general text augmentations are proposed, which are listed in Table 4.1. The related discussions about the intuitive priors behind these augmentations are shown below:

Shorten. Given an original text attribute t_n, the *shorten* augmentation applies a prompt p^s to produce $\hat{t}_n^s$ to guide $LLM(\cdot)$ output o_n^s. Such an augmentation aims to simplify the original text attribute. The underlying prior enforced by it is that simplified content can help filter out redundant information and maintain the key points in the original text.

Rewriting. Given an original text attribute t_n, the *rewriting* augmentation applies a prompt p^r to produce $\hat{t}_n^r$ to guide $LLM(\cdot)$ output o_n^r. Such an augmentation aims to rewrite the original text attribute so that the invariant semantics contained in the original text attribute can be identified. Moreover, the readability of the text attributes can also be improved to help produce high-quality feature embeddings.

Expansion. Given an original text attribute t_n, the *expansion* augmentation applies a prompt p^e to produce $\hat{t}_n^e$ to guide $LLM(\cdot)$ output o_n^e. Such an augmentation aims to expand the original text attribute to introduce more related and necessary information to leverage the advantages of the knowledge base, which is trained on a large volume of the corpus.

Without loss of generality, the *shorten* augmentation is adopted as the example, denoted by superscript s, to describe the workflow of the proposed method LATEX-GCL in the methodology section and omit the two other augmentations. Formally, we prompt the original text attribute t_n of the n-th node in the TAG $\mathcal{G}$ to obtain the prompted input $\hat{t}_n^s$ for the augmentor $LLM(\cdot)$ to have:

$$o_n^s = LLM(\hat{t}_n^s). \tag{4.2}$$

The operations above repeat on each node in the original TAG $\mathcal{G}$ to have augmented text attributes $\{o_n^s | n \in \mathcal{V}\}$. Finally, we can have the augmented TAG $\mathcal{G}^s = (\mathcal{V}, \mathcal{E}, \{o_n^s\}_{n\in\mathcal{V}})$.

4.3.2 Text Attribute Encoding

The proposed method LATEX-GCL applies an LLM to directly augment the original text attributes to produce augmented text attributes instead of adopting the feature masking

augmentation, which is one of the conventional graph augmentations [132]. Though, as introduced in the introduction section, the adopted strategy can reduce information loss and leverage the advantages of the LLM's superior semantic comprehension capability, the augmented text attributes are in the form of natural language that cannot be processed by the following graph encoding module (*i.e.,* the GNN model). Therefore, we adopt a text encoder in the proposed method to encode the original and the augmented text attributes to acquire feature embeddings to facilitate the following procedures.

A relatively small language model, such as BERT [20] and DeBERTa [37], is adopted to serve as the text encoder because they are more powerful than those conventional text embedding methods [34, 69] and more efficient than the LLMs. Following the LLM-based augmentation phase, the text encoder $LM(\cdot)$ takes the original and the augmented text attributes to produce the original and augmented feature embeddings, shown as follows:

$$\mathbf{h}_n = LM(t_n) \in \mathbb{R}^{d\times 1}, \ \mathbf{h}_n^s = LM(o_n^s) \in \mathbb{R}^{d\times 1}, \tag{4.3}$$

where d is the size of feature embeddings. Then, the feature matrix of the original TAG $\mathcal{G}$ and the augmented TAG $\mathcal{G}^s$ can be acquired as follows:

$$\mathbf{H} = [h_1; h_2; \cdots; h_N]^T \in \mathbb{R}^{N\times d}, \ \mathbf{H}^s = [h_1^s; h_2^s; \cdots; h_N^s] \in \mathbb{R}^{N\times d}. \tag{4.4}$$

The feature matrices obtained above can cooperate with the adjacency matrix $\mathbf{A}$ of the input TAG to facilitate the following graph encoding procedures.

To enhance performance, it is a common practice to train the text encoder in conjunction with subsequent modules, yet this approach demands substantial computational resources. In practical applications, an adaptor module, typically a straightforward neural network component such as a linear layer, is employed to refine the text encoder's output, thereby boosting performance without incurring the costs associated with fine-tuning. Nevertheless, optimizing the adaptor module often necessitates ample supervised training data from specific downstream tasks. The efficacy of the adaptor module within the context of GCL in this section remains an open question. This issue will be investigated in the following experiment section, where the impact of the adaptor module on the performance of LATEX-GCL is examined.

4.3.3 Graph Encoding

TAGs contain a rich repository of information. In addition to the previously mentioned textual attribute information, graph structure is also essential for the graph learning tasks on

TAGs. Encoding only the text features is insufficient for acquiring comprehensive graph representation, necessitating the adoption of GNN models (*e.g.,* GCN [53]) to learn the structural information in the graph.

Given the feature matrices $\mathbf{H}$ and $\mathbf{H}^s$ obtained in the previous text encoding module, the adjacency matrix $\mathbf{A}$, and a K-layer graph encoder $g(\cdot,\cdot)$, we can have the updated feature matrices that possess graph structure information as follows:

$$\mathbf{H}^{(K)} = g(\mathbf{A},\mathbf{H}) \in \mathbb{R}^{N\times d},\ \mathbf{H}^{s(K)} = g(\mathbf{A},\mathbf{H}^s) \in \mathbb{R}^{N\times d}. \tag{4.5}$$

Each layer of the graph encoder functions as a message passing and aggregation process, collecting neighbor information and updating the node feature iteratively.

4.3.4 Graph Contrastive Learning

Typically, TAGs possess extensive text attributes to describe the nodes. However, in real-world scenarios, label sparsity is a common and unavoidable issue, making it infeasible to manually label each node in the TAG due to the prohibitive costs involved. To broaden the applications of TAGs, it is vital to investigate how to employ self-supervised learning paradigms to obtain high-quality graph embeddings from TAGs without label information. GCL has demonstrated the powerful capability to conduct self-supervised graph learning, to this end, being a viable option for the self-supervised learning paradigm on TAGs. This section utilizes a GCL module to process the LLM-augmented graphs, finalizing the workflow of the proposed LATEX-GCL method.

A rough GCL setting is revealed in the fourth part of Figure 4.1. During the training, the node embeddings are usually processed in a mini-batch manner. We use $\mathcal{V}_b$ to denote the set of nodes in a training batch. Formally, suppose that the i-th node $i \in \mathcal{V}_b$ is the target. The original feature embedding of the target and the augmented feature embedding can be obtained as follows:

$$\mathbf{h}_i^{(K)} = \mathbf{H}_{i,:}^{(K)\,T} \in \mathbb{R}^{d\times 1},\ \mathbf{h}_i^{s(K)} = \mathbf{H}_{i,:}^{s\ (K)\,T} \in \mathbb{R}^{d\times 1} \tag{4.6}$$

The two feature embeddings mentioned above originate from the same target node, thus they are expected to exhibit a high degree of similarity. Therefore, we treat such a pair of embeddings as positive contrasting samples. Then, a subset $\mathcal{V}_M \subseteq \mathcal{V}_b \setminus i$ of nodes, where $|\mathcal{V}_b| = M$, is randomly sampled from the mini-batch to collaborate with the original feature embedding of the target, generating $2M$ negative contrasting samples. The negative contrasting sample's original feature embedding and its LLM-augmented embedding are

denoted by $\{\mathbf{h}_j^{(K)} | j \in \mathcal{V}_M\}$ and $\{\mathbf{h}_j^{s(K)} | j \in \mathcal{V}_M\}$. A similarity function $sim(\cdot,\cdot)$ is adopted to measure the distance between two feature embeddings. Then, InfoNCE [102] is adopted as the loss function for the GCL training:

$$\mathscr{L} = -\log \frac{e^{sim(\mathbf{h}_i^{(K)}, \mathbf{h}_i^{s(K)})/\tau}}{e^{sim(\mathbf{h}_i^{(K)}, \mathbf{h}_i^{s(K)})/\tau} + \sum_{j \in \mathcal{V}_M}^{M} (e^{sim(h_i^{(K)}, \mathbf{h}_j^{(K)})} + e^{sim(\mathbf{h}_i^{(K)}, \mathbf{h}_j^{s(K)})})}, \tag{4.7}$$

where τ denotes the temperature hyperparameter. After the GCL training, the feature matrix $\mathbf{H}^{(K)}$ is updated, and we can obtain the final feature matrix $\mathbf{H}_{final}$ for inference and evaluation of the downstream graph classification task.

4.4 Experiments on LATEX-GCL

To demonstrate the effectiveness and the performance of the proposed LATEX-GCL method, extensive experiments are conducted and the results with insightful analysis are shown in this section. The related experimental settings are also provided in this section.

Table 4.2: Statistics of Amazon Datasets in LATEX-GCL Experiments

Dataset	#Node	#Edge	#Class	Raw Text Content
Books-Children	76,875	1,554,578	24	Book Introduction
Books-History	41,551	358,574	12	Book Introduction
Ele-Computers	87,229	721,081	10	Consumer Review
Ele-Photo	48,362	500,928	12	Consumer Review

4.4.1 Experimental Settings

4.4.1.1 Datasets

Considering the research scope of this section, experiments on the graph datasets with promising text attributes are required. Multiple text-attributed graphs are collected by [119] from which four datasets, including Books-Children, Books-History, Ele-Computers, and Ele-Photo, are selected as the experiment datasets. These datasets are extracted from the Amazon dataset [38, 67], which have raw text descriptions for each node and are large-scale compared to previous text-attributed graph datasets [119]. The statistics and the content of the raw text of each dataset are listed in Table 4.2.

4.4.1.2 Baselines

Besides the datasets, five impactful GCL methods are selected as baselines for the comparison study. These baselines can be roughly broken down into three categories: I) **GraphCL** [132] is the most classical GCL method that involves several conventional random-based augmentations, II) **GCA** [149] and **GraphCL-Auto** [131] are both the adaptive augmentation-based GCL methods, where GCA conducts automatic selection from the conventional augmentation techniques and GRACE performs trainable augmentations based on the input graph data, and III) both **BGRL** [99] and **GBT** [5] method follow a novel GCL paradigm that utilizes different training objectives instead of InfoNCE [102] based on DGI [104] to eliminate the requirement of negative contrasting samples to achieve storage efficient.

4.4.1.3 Implementation Details

The LLM used for dataset augmentations in our settings is *GPT-3.5-turbo*, and the specific version is default and decided by OpenAI update schedule[1]. The prompts for guiding the LLM to generate augmented text are listed in Table 4.1 in the methodology section. Moreover, we adopt a pre-trained BERT [20] model, whose version is *bert-base-uncased*, to embed the original and augmented text attributes. The pre-trained model are used according to the guidance of *Pytorch-Transformers*[2]. The pre-trained model and other related components can be publicly accessed on *Hugging Face* via this link[3]. Some important hyperparameter settings are listed here. The embedding size of the text encoder is set to 768, and the output size of the graph encoder is set to 256. The learning rate for the whole framework training is $2e^{-5}$. The training batch size and the epoch number are set to 512 and 10.

4.4.1.4 Evaluation Protocol

The proposed method is evaluated based on the node classification task, which is subject to the linear evaluation protocol. The linear evaluation is to train and test a support vector machine (SVM) on node feature embeddings trained by the method to be evaluated to verify the quality of the outputs of the proposed LATEX-GCL method, where the SVM is implemented by a third-party toolkit named *scikit-learn*[4]. Specifically, to ensure the reliability of the experiment results, we repeat the experiment five times. For each time, 20% of the nodes are selected as the training set, and 10% of the rest of the nodes are the test

[1] https://platform.openai.com/docs/models/gpt-3-5-turbo
[2] https:pytorch.orghub/huggingface_pytorch-transformers
[3] https://huggingface.co/google-bert/bert-base-uncased
[4] https://scikit-learn.org/

Table 4.3: The comparison study between the baselines and LATEX-GCL.

Dataset	Books-Children				Books-History			
Methods \ Metrics	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
GraphCL	33.87 (std 0.87)	11.63 (std 0.96)	6.92 (std 0.28)	5.94 (std 0.34)	72.42 (std 0.52)	22.83 (std 0.49)	20.64 (std 0.70)	20.86 (std 0.64)
GraphCL-Auto	37.23 (std 0.91)	20.15 (std 1.07)	9.93 (std 0.24)	10.87 (std 0.29)	72.87 (std 0.63)	27.58 (std 0.61)	22.07 (std 0.75)	22.94 (std 0.69)
GCA	OOM	OOM	OOM	OOM	7.53 (std 0.58)	4.34 (std 2.32)	4.85 (std 0.83)	6.01 (std 0.72)
BGRL	7.99 (std 0.81)	28.16 (std 2.03)	12.73 (std 0.22)	13.08 (std 0.30)	75.36 (std 0.49)	30.02 (std 2.24)	23.73 (std 0.92)	23.97 (std 0.83)
GBT	36.98 (std 0.83)	8.77 (std 1.59)	3.09 (std 0.18)	4.01 (std 0.27)	74.97 (std 0.42)	31.17 (std 3.42)	23.35 (std 0.87)	25.13 (std 0.79)
LATEX-GCL (S)	38.71 (std 0.65)	27.86 (std 2.62)	11.89 (std 0.27)	12.40 (std 0.43)	78.65 (std 0.69)	32.58 (std 4.47)	25.91 (std 0.77)	25.55 (std 0.56)
LATEX-GCL (R)	39.30 (std 0.56)	28.07 (std 1.14)	12.70 (std 0.10)	13.38 (std 0.21)	79.08 (std 0.65)	35.55 (std 7.17)	26.98 (std 0.81)	27.02 (std 0.73)
LATEX-GCL (E)	**41.72 (std 0.45)**	**31.27 (std 2.52)**	**15.50 (std 0.21)**	**16.81 (std 0.11)**	**79.22 (std 0.61)**	**37.28 (std 5.17)**	**27.31 (std 0.89)**	**27.51 (std 0.84)**
Dataset	Ele-Computers				Ele-Photo			
Methods \ Metrics	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
GraphCL	33.48 (std 0.23)	35.77 (std 5.37)	15.44 (std 2.65)	13.79 (std 0.36)	42.24 (std 0.45)	36.78 (std 8.00)	8.97 (std 0.18)	6.21 (std 0.32)
GraphCL-Auto	40.79 (std 0.63)	47.23 (std 4.23)	21.99 (std 1.96)	24.67 (std 0.58)	45.74 (std 0.27)	40.39 (std 7.59)	15.61 (std 0.21)	14.95 (std 0.19)
GCA	OOM	OOM	OOM	OOM	5.65 (std 0.37)	9.37 (std 1.87)	9.56 (std 0.73)	3.97 (std 1.17)
BGRL	44.36 (std 0.61)	9.78 (std 1.39)	28.43 (std 2.11)	2.27 (std 0.54)	53.77 (std 0.40)	68.73 (std 2.39)	28.88 (std 0.69)	32.74 (std 0.95)
GBT	5.31 (std 0.59)	49.12 (std 2.03)	9.59 (std 1.05)	31.97 (std 0.48)	54.68 (std 0.49)	67.56 (std 1.59)	29.02 (std 0.84)	32.93 (std 1.07)
LATEX-GCL (S)	48.87 (std 0.56)	52.60 (std 1.62)	29.48 (std 0.38)	31.50 (std 0.41)	56.54 (std 0.40)	**71.48 (std 1.64)**	29.14 (std 0.92)	35.10 (std 1.31)
LATEX-GCL (R)	**50.80 (std 0.51)**	52.49 (std 1.16)	**31.55 (std 0.41)**	**33.89 (std 0.50)**	**57.73 (std 0.16)**	69.64 (std 0.70)	**30.77 (std 0.68)**	**37.14 (std 0.96)**
LATEX-GCL (E)	47.24 (std 0.55)	**53.26 (std 1.81)**	27.58 (std 0.33)	28.99 (std 0.30)	56.39 (std 0.30)	70.88 (std 1.11)	28.35 (std 0.52)	33.86 (std 0.72)

set. Sufficient metrics, including Accuracy, Precision, Recall, and F1 scores with standard deviations, are used to demonstrate the results of the linear evaluation.

4.4.2 Experiment Results & Analysis

This section lists the experiment results, including the comparison study, the ablation study, and the adaptor module experiment, which are accompanied by detailed analysis.

4.4.2.1 Comparison Experiment

The results of the comparison study are listed in Table 4.3, demonstrating the performance of the proposed LATEX-GCL method and the selected baselines regarding the node classification task on the graph. The figures underlined denote the best performance achieved by baselines, the figures in boldface represent the best result among all methods, and 'OOM' indicates that the method is out of the memory when performing on the specific dataset. The suffixes of LATEX-GCL, including (S), (R), and (E), denote different augmentation prompts used for the experiment, which are *shorten*, *rewriting*, and *expansion*, respectively. According to the results, we have the following three findings:

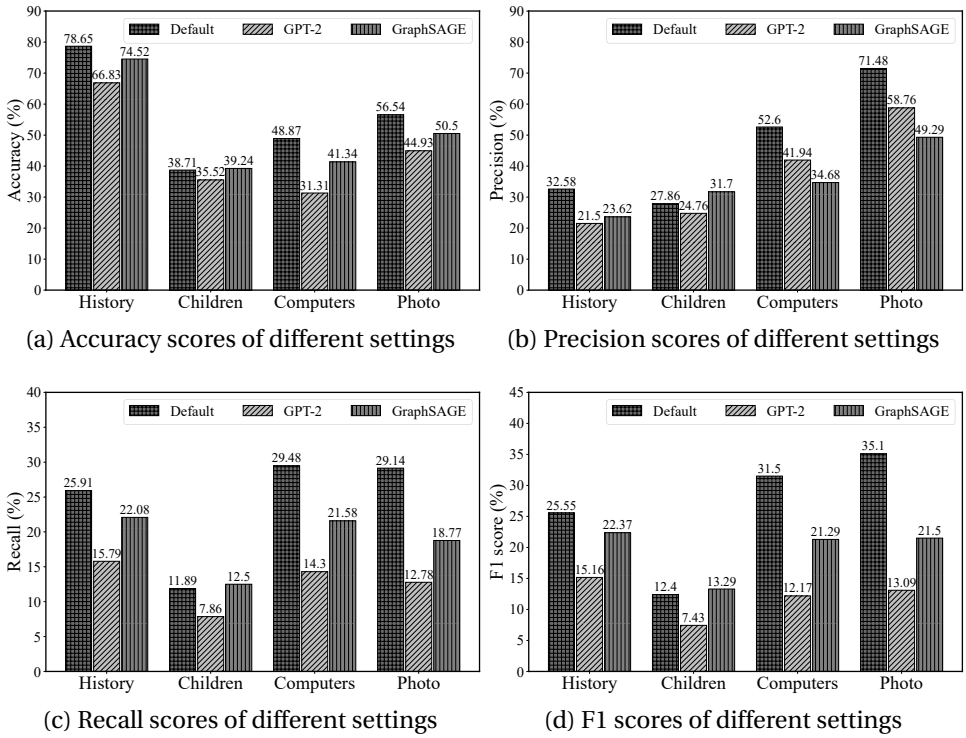
- Generally, the proposed LATEX-GCL method achieves the best performance in the comparison study among all datasets compared to the selected baselines. Such an observation verifies the effectiveness and the superiority of our proposed LATEX-GCL method. For different augmentation settings, the results reflect a clear pattern. Specifically, LATEX-GCL equipped with *expansion* augmentation performs better on

the two Amazon-Books datasets, and LATEX-GCL equipped with *rewriting* augmentation performs better on the two Amazon-Electronics datasets.

- The differences among the performance of different augmentation settings of LATEX-GCL are largely due to the difference in the raw text content of the two types of datasets. As listed in Table 4.2, the raw text content in the book datasets is the book introduction, and that of the electronic datasets is the consumer review. The book introduction usually contains the correct title of the book, which can help the LLM prompted by the *expansion* augmentation to produce informative content that is highly related to the specific book as the augmented textual attributes, which can significantly benefit the following GCL. However, the consumer reviews of the electronic datasets are normally short and neglect to list the full name of the product reviewed. Such textual attributes prevent the LLM prompted by *expansion* augmentation from producing informative content. Even worse, it may lead the LLM to introduce more noise (*i.e.,* unrelated content). Therefore, utilizing the LLM to extract key information in the consumer review would be more suitable instead of producing auxiliary information. The experiment results confirm our analysis. On dataset Books-Children and Books-History, LATEX-GCL equipped with *shorten* augmentation and *rewiriting* augmentation, which are both helpful for key information extraction from the original textual attributes as discussed previously, outperform LATEX-GCL equipped with *expansion* augmentation. Moreover, in the scenarios of lacking sufficient computational resources, the *shorten* augmentation would be a promising alternative for the *rewriting* expansion as the gap between the performance of these two augmentations is insignificant on both electronic datasets.

- GraphCL has the lowest scores across all metrics and datasets. This is because GraphCL uses classical augmentation techniques to conduct GCL, outperformed by those adaptive augmentation strategies. GraphCL-Auto adopts an automatic selection strategy to pick conventional augmentations used in GraphCL, slightly improving the performance. GRACE proposes an adaptive strategy to augment the graph according to the specific input data. However, such a strategy significantly increases the complexity. Consequently, GRACE is out of memory when performing on the two large datasets, including Books-Children and Ele-Computers. The significant improvement brought by the adaptive augmentation strategy is reflected by GCA's performance on Books-History and Ele-Photo. Specifically, GRACE achieved the best results among all the baselines on these two datasets. Both BGRL and GBT methods follow the same idea

of utilizing different training objectives instead of InfoNCE to eliminate the requirement of negative contrasting samples and achieve better performance. We can observe that both methods can perform well on large datasets. However, on the relatively small datasets where GraphCL-Auto can function, BGRL and GBT are outperformed by GraphCL-Auto due to both methods taking the same conventional augmentation techniques as adopted by GraphCL, which is less advanced compared to the adaptive augmentation strategy.



(a) Accuracy scores of different settings

(b) Precision scores of different settings

(c) Recall scores of different settings

(d) F1 scores of different settings

Figure 4.2: The LATEX-GCL's performance with different text encoders and graph encoders.

4.4.2.2 Ablation Study of Text Encoder and Graph Encoder

There are two critical components in LATEX-GCL: text encoder and graph encoder. In this ablation study, we examined different models for the two components. The text encoder adopts BERT [20], and the graph coder is GCN [53] in the default settings. Two supplementary experiments are conducted with BERT [20] being replaced by GPT-2 [79] and GCN [53] being replaced by GraphSAGE [32]. The experiment results are illustrated in Figure 4.2.

Both BERT and GPT-2 are representative language models in NLP areas. However, there are significant differences between the two models. BERT is a bidirectional model that

utilizes tasks like Masked Language Modeling to train word representations, focusing on context-based text understanding. But GPT-2 is a single-direction model trained by self-regression paradigms to predict the next word based on the previous content, which is designed for generative tasks. We can observe that LATEX-GCL equipped with GPT-2 is significantly outperformed by LATEX-GCL equipped with BERT. It indicates that the generative language model is unsuitable for acquiring text feature embeddings. This phenomenon is reasonable as the generative language models are designed for content generation, lacking powerful embedding abilities to obtain informative text representations.

The graph encoder module in LATEX-GCL incorporates the embedded text features and graph structural information to obtain the final representation embedding of the node in the TAG. In practice, the graph encoder selected for LATEX-GCL should be simple and efficient for processing large-scale graphs like GCN and GraphSAGE. Though LATEX-GCL equipped with GraphSAGE is functional, it is outperformed by LATEX-GCL equipped with GCN. GraphSAGE is designed for very large graphs and randomly drops some nodes and edges to facilitate the training, which causes information loss.

In short, to ensure the normal functionality and satisfying performance of LATEX-GCL, the text encoder should not adopt generative language models such as GPT-2, and the graph encoder should be simple and efficient enough like GCN and GraphSAGE to incorporate the language model to train on large TAGs.

4.4.2.3 Adaptor Module Experiment

As mentioned previously, adopting an adaptor module is a common practice for employing pre-trained language models for various downstream applications while avoiding fine-tuning. However, the adaptor module is usually combined with the downstream models to be trained together by supervised signals. But, in our settings, the training phase is motivated by graph contrastive learning, a self-supervised learning paradigm, instead of the supervised one. This section investigates if the adaptor module can apply to LATEX-GCL.

Without losing generality, we employ a single linear layer to decorate the outputs of the text encoder. The adaptor-processed outputs' size is a hyperparameter selected from $\{256, 512, 768\}$. Moreover, the default setting in this experiment denotes the vanilla LATEX-GCL equipped with *shorten* augmentation. The experiment results are shown in Table 4.4.

According to the results, the adaptor module is effective in improving the performance of LATEX-GCL in most scenarios. Specifically, the improvement occurs when the output size of the adaptor is relatively small (*i.e.,* smaller than the output size of the text encoder).

Table 4.4: The performance of LATEX-GCL with different adaptor settings.

Dataset	Books-Children				Books-History			
Settings \ Metrics	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Default	38.71 (std 0.65)	27.86 (std 2.62)	11.89 (std 0.27)	12.40 (std 0.43)	78.65 (std 0.69)	32.58 (std 4.47)	25.91 (std 0.77)	25.55 (std 0.56)
256	40.96 (std 0.51)	31.19 (std 2.83)	14.47 (std 0.25)	15.44 (std 2.75)	78.86 (std 0.33)	32.95 (std 3.29)	26.16 (std 0.67)	25.75 (std 0.49)
512	39.55 (std 0.67)	28.88 (std 1.85)	12.73 (std 0.23)	13.45 (std 0.21)	79.17 (std 0.43)	36.33 (std 4.91)	26.40 (std 0.35)	25.95 (std 0.22)
768	35.28 (std 0.96)	13.20 (std 2.38)	8.26 (std 0.33)	7.37 (std 0.44)	78.48 (std 0.66)	29.50 (std 4.16)	25.62 (std 0.93)	24.98 (std 0.78)
Dataset	Ele-Computers				Ele-Photo			
Settings \ Metrics	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Default	48.87 (std 0.56)	52.60 (std 1.62)	29.48 (std 0.38)	31.50 (std 0.41)	56.54 (std 0.40)	71.48 (std 1.64)	29.14 (std 0.92)	35.10 (std 1.31)
256	50.63 (std 0.66)	53.15 (std 1.34)	31.22 (std 0.58)	33.61 (std 0.79)	57.06 (std 0.51)	70.94 (std 1.13)	29.00 (std 0.89)	34.94 (std 1.14)
512	53.44 (std 1.17)	53.06 (std 1.45)	34.26 (std 0.95)	37.01 (std 1.16)	49.23 (std 0.56)	49.31 (std 6.85)	16.58 (std 0.64)	18.67 (std 0.96)
768	48.62 (std 0.30)	53.59 (std 1.37)	28.82 (std 0.29)	30.61 (std 0.40)	53.86 (std 0.33)	64.91 (std 5.10)	24.07 (std 0.70)	28.96 (std 0.98)

It can be speculated that the role of the adaptor is to condense the text feature embeddings produced by the text encoder to facilitate the following GCL training process.

4.5 Summary of LATEX-GCL

This section proposes a novel GCL framework, namely LATEX-GCL, which successfully incorporates LLMs to conduct augmentations to construct contrasting samples and addresses RQ1.2. The purpose of the proposed augmentation strategy is to leverage the advantages of LLMs to tackle the limitations of information loss, incapable language models, and implicit constraints of current GCL methods for TAGs, including alleviating information loss during the augmentation, enhancing insufficient NLP abilities of conventional language models, and imposing explicit constraints on the augmentation process. Comprehensive experiments verify the effectiveness and superiority of the proposed LATEX-GCL method. This research is expected to be a pioneering work that encourages the exploration of LLMs for GCL. The future directions are two-fold, including investigating more comprehensive augmentation prompting strategies for different scenarios and how to improve the computation efficiency of employing LLMs in real-world applications.

Both Chap. 3 and this chapter focus on the perspective of methodology of augmentation techniques in GCL. However, to demonstrate the practical value of GCL methods, it is essential to apply them to real-world scenarios. One of the most suitable application areas for GCL methods, as previously discussed, is recommendation systems. Next two chapters present novel implementations of GCL in recommendations to achieve the practical value of GCL methods in real-world applications.

CHAPTER 5

CONSTRUCTION OF CONTRASTING SAMPLES FOR RECOMMENDATIONS

User purchasing prediction with multi-behavior information remains a challenging problem for current RS. Various methods have been proposed to address it via leveraging the advantages of GNNs or multi-task learning. However, most existing works do not take the complex dependencies among different behaviors of users into consideration. They utilize simple and fixed schemes, like neighborhood information aggregation or mathematical calculation of vectors, to fuse the embeddings of different user behaviors to obtain a unified embedding to represent a user's behavioral patterns which will be used in downstream recommendation tasks. To tackle the challenge, in this section, the concept of hyper meta-path is first proposed, which can be used to construct hyper meta-graphs to explicitly illustrate the dependencies among different behaviors of a user. How to obtain a unified embedding for a user from hyper meta-paths and avoid the previously mentioned limitations simultaneously is critical. Thanks to the recent success of GCL, it can be leveraged to learn embeddings of user behavior patterns adaptively instead of assigning a fixed scheme to understand the dependencies among different behaviors. A new GCL-based framework is proposed by coupling with hyper meta-path, namely HMG-CR, which consistently and significantly outperforms all baselines in extensive comparison experiments.

This chapter presents a pioneering research, which is one of the first work in the literature to demonstrate how to implement GCL in RS and is the answer to RQ2.1. Inspired by the previous research and the understanding of different augmentation techniques, a novel

concept of hyper meta-path and the tailored contrasting sample construction method are proposed, and they would inspire the future works in this research domain.

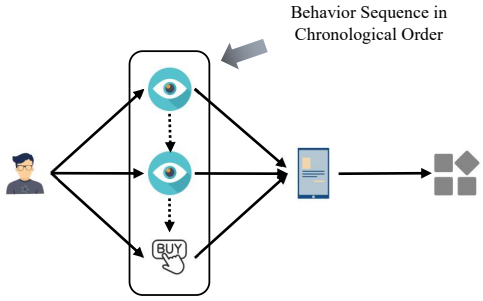
5.1 Brief Introduction to HMG-CR

Online shopping is becoming more and more essential nowadays, and it generates a large volume of user behavioral data depicting users' purchasing motivations, interests, behavioral patterns, etc. However, many traditional recommendation systems [41, 32] pay significant attention to purchasing alone, leaving other associated behavioral data unexploited. Though recent works [142, 86, 26, 15, 130] reveal the gap and try to leverage multi-behavior information to improve recommendation quality, there are still limitations. For instance, some path-based works [142, 16] leverage meta-paths [98, 11] to extract recommendation context to better characterize users' multiple behaviors. However, there exist many meta-path schemes observed in heterogeneous graphs, resulting in the difficulty of finding out the best one from multiple meta-path schemes via exhaustive search or learning a specific

ru

el

sc



(b) A hyper meta-path of the user

Figure 5.1: Differences between meta-path and hyper meta-path in HMG-CR.

To overcome the above limitation of existing path-based approaches, we propose a new concept of hyper meta-path that consolidates multiple paths in a well-organized and holistic way. Similar to hyperedge in hypergraph [96, 97, 62, 65] where an edge can connect more than two nodes, a hyper meta-path is a composition of multiple meta-paths between specified two end nodes in a heterogeneous network. As shown in Figure 5.1, let us assume that before purchasing a phone, the user had viewed the item twice. If meta-path-based approaches are adopted to model the different shopping-related behaviors, three indepen-

dent meta-path instances (Figure. 5.1. (a)) can be discovered to characterize this purchasing context. From these three meta-paths, it can learn that the user has viewed and purchased the item, but it does not explicitly reflect the exact behavioral pattern of the user. That is, it is unclear whether the user purchased the item directly or viewed the item carefully and for multiple times before purchasing.

In contrast, the proposed hyper meta-path is capable of achieving such a goal. As shown in part (b) of Figure. 5.1, a hyper meta-path between the user and the phone consolidates all three behaviors in a sequential order, which explicitly shows that the user carefully viewed the phone twice before the final purchasing action. Note that the way of consolidating related meta-paths into a hyper meta-path can be flexible and generalized to any reasonable rule depending on a particular application scenario. Besides, the concept of hyper meta-path is also useful for differentiating different behavior patterns between different users or when facing various categories of items presented to a user. For instance, normally, technical people may research different substitutable electronic products and take a longer time to compare them, while non-technical people are not keen on investigating them and would probably directly buy one based on someone's recommendation. Even for the same user, no matter their age and gender, they usually exhibit quite diverged buying patterns when facing different categories of products. For example, a user may have totally different buying patterns when purchasing large items (*e.g.,* white goods like fridges or TVs) and small fast-moving consumer goods (*e.g.,* periodically buying tissues from an online market without viewing them again and again).

Nevertheless, it is not straightforward to incorporate the modeling of hyper meta-path into existing learning frameworks. Currently, graph-based unsupervised learning approaches are mainly used for path-based recommendation. For example, GNNs-based approaches [86] are a popular means for multi-behavior recommendation via aggregating information passed from different types of edges or nodes in heterogeneous information networks [120]. Despite its popularity, these methods usually fuse the learned features of different behaviors independently, which is too naive to reflect hyper meta-path context for recommendation. Moreover, multi-task learning-based models [26, 15] are also possible ways that introduce additional supervision signals from the observed multiple behavior data to improve recommendation quality. However, extra efforts on well-elaborated tasks are tricky, and researchers have to carefully work out the effective dependencies among related tasks. For example, taking purchasing prediction as a primary task while modeling the *add-to-cart* behavior prediction as an auxiliary task might not always be right as some users may buy some items directly without putting them into the cart.

Thus, to further reveal and capture the differences between buying patterns, together with hyper meta-paths, the GCL [78, 132] paradigm is innovatively leveraged for the multi-behavior recommendation problem. The main idea of graph contrastive learning is to distinguish the differences among graphs to obtain the useful structure information of each graph, raising a recent surge of interest [78, 132]. The rationale for incorporating contrastive learning with our proposed hyper meta-paths is that a user may have multiple hyper meta-paths explicitly illustrating his/her behavioral patterns when facing different products. Since hyper meta-path explicitly describes users' behaviors towards purchasing different items, GCL becomes the best fit for comparing and extracting the key structures in the hyper meta-graph consisting of hyper meta-paths.

More specifically, we combine multiple hyper meta-paths of a user to construct several hyper meta-graphs. Each hyper meta-graph contains a different number of types of behaviors. For example, the first hyper meta-graph contains *buy*, and the second hyper meta-graph contains *buy* and *page view*. In this case, different hyper meta-graphs reflect different behavioral patterns of the user regarding different products. Then, we conduct graph contrastive learning among the constructed hyper meta-graphs to adaptively obtain the complex dependencies among different behaviors and the embeddings representing different behavioral patterns. For instance, in HMG-CR, we first build the target contrastive graph that only contains *buy* interactions between users and items as it is the target behavior for recommendation systems, and the other contrastive hyper meta-graphs are added for comparison by incrementally introducing auxiliary behaviors to the precedent hyper meta-graph. After that, we conduct graph contrastive learning between the constructed contrastive hyper meta-graphs to successively obtain progressive and comprehensive representations for each type of behavior. Finally, the recommendation will be performed based on those discovered behavior patterns and features.

The contributions of this section can be summarized into three aspects:

- Tthe concept of hyper meta-path is proposed to explicitly illustrate the logical relations among a collection of meta-paths, which tackles the limitation of meta-path that is insufficient to model the interactions among meta-paths. Hyper meta-path can be regarded as an approach to enrich graph structures.
- GCL is innovatively utilized to capture the complex behavior patterns of users adaptively, alleviating existing methods' limitations.
- A novel recommendation framework is proposed by coupling GCL with hyper meta-path, achieving superior performances in the comparison experiments.

5.2 Preliminaries about Hyper Meta-Path

This section introduces some necessary preliminaries and definitions about the meta-path and the proposed concept of hyper meta-path.

5.2.1 Meta-Path

Heterogeneous networks have been intensively studied by a lot of researchers due to their ability to utilize multi-model, multi-typed graph data. To illustrate the power of heterogeneous networks, Sun *et al.* [98] proposed the concept of meta-path, which is widely used by many existing works [21, 11] in the research area of heterogeneous networks modeling. Each meta-path captures the features among the nodes on the meta-path from a particular semantic perspective. Due to the diversity of meta-paths in a heterogeneous graph, multiple meta-paths exist for the target (*e.g.,* a node or an edge). Thus, the informative meta-paths give heterogeneous network models the chance to obtain the multi-model multi-typed features of nodes and their relations. This kind of data structure indeed shows the advantage in many real-world graph data mining applications [46, 91]. However, there are limitations existing in meta-paths mentioned in previous section, failing to capture the interaction information among multiple meta-paths.

5.2.2 Hyper Meta-Path

Though people can build extra meta-paths based on the interactions among existing meta-paths, we cannot take an exhaustive method to compute every meta-path since the computation complexity is unaffordable. Inspired by the concepts of hyperedge and hypergraph, we find a way to integrate interaction information among meta-paths into the target. According to the limitations of the conventional meta-path mentioned above and the advantages of hyperedge and hypergraph, we propose the concept of hyper meta-path to capture meta-path features and interaction information among them simultaneously.

Definition 1. ***Hyper meta-path****. A hyper meta-path is a logical composition of multiple meta-path schemas connecting two end nodes in a heterogeneous information network. Hyper meta-path has the following properties:*

- *It describes the logical relations (e.g., chronological order, spatial order, and topological order) among a sort of meta-paths with the same end nodes.*
- *Multiple hyper meta-paths, having the same start node, compose a hyper meta-graph.*

5.3 Methodology

In this section, the details of the proposed HMG-CR method are introduced. The overview

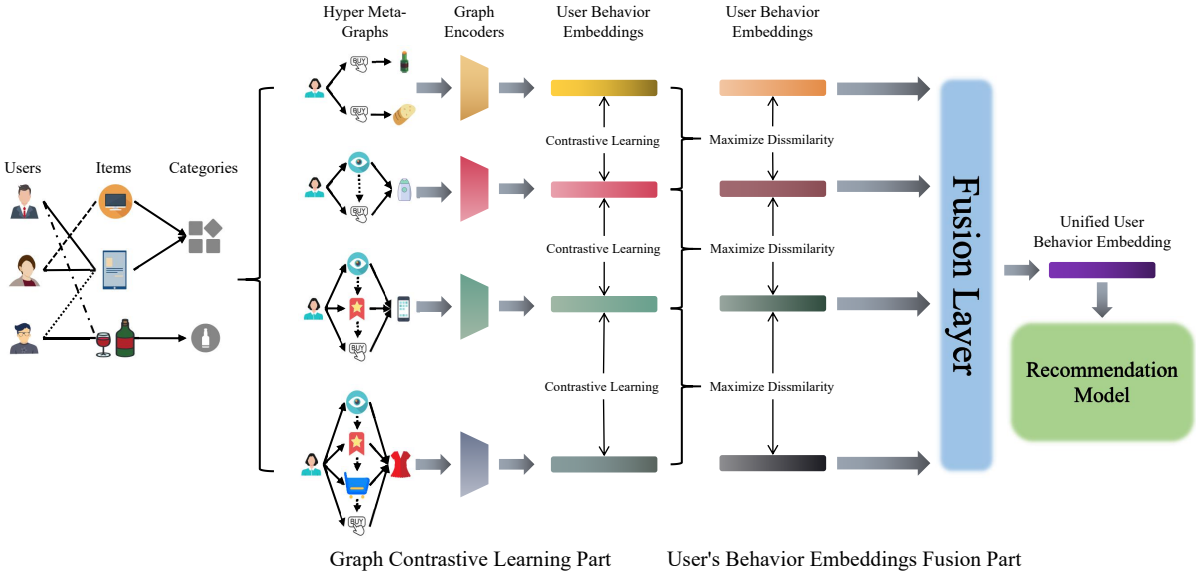


Figure 5.2: The overview of HMG-CR.

5.3.1 Hyper Meta-Graph Generation

GNN-based recommendation methods [40] have recently achieved tremendous success due to the power of GNNs. Data is critical to neural network models' performances. One way to leverage GNN-based recommendation models is to construct proper graphs for them. The most common way to construct graphs in recommendation systems is by building bipartite graphs via user-item interaction history. Since user-item interaction graphs are bipartite graphs, they lack semantic information because of their simple structure. To tackle this limitation, researchers have taken measures to further enrich semantic information carried by graphs, for example, by adding auxiliary information into the graph [76], utilizing meta-paths existing in the graph [16], and constructing more sophisticated graph structures like hypergraphs [108, 13].

To improve recommendation results, in this work, the proposed concept of hyper meta-path is utilized to construct hyper meta-graphs carrying rich semantic information. Next, details on how to construct hyper meta-graphs.

Given a set of interaction records in a recommendation system, $\{(u_j, r_k, i_q)|u_j \in \mathscr{U}, i_q \in \mathscr{I}, r_k \in \mathscr{R}\}$, where $\mathscr{U} = \{u_0, u_1, \cdots, u_n\}$ denotes the set of all users, $\mathscr{I} = \{i_0, i_1, \cdots, i_m\}$ de-

notes the set of all items, and $\mathscr{R} = \{r_0, r_1, \cdots, r_l\}$ denotes the set of all different kinds of user behaviors. According to the number of types of different user behaviors, we construct $|\mathscr{R}| = l + 1$ hyper meta-graphs for each user. For the t-th hyper meta-graph of user u_j, it is defined as $\mathscr{HG}_t^j = \{(u_j, (r_a, r_b, \cdots, r_c), i_q) | i_q \in \mathscr{I}, \forall r \in \{r_0, r_1, \cdots, r_{t-2}, r_l\}\}$, where behavior sequence $(r_a, r_b, \cdots, r_c)$ is sorted in chronological order, and each behavior r in the sequence solely bridges user u_j and item i_q. Hence, we will have a set of hyper meta-graphs $\mathscr{HG}^j = \{\mathscr{HG}_0^j, \mathscr{HG}_1^j, \cdots, \mathscr{HG}_l^j\}$ to illustrate user-item interactions of user u_j in the recommendation system. Note that the order among different behaviors in $\mathscr{R}$ is based on the distance between behaviors and behavior *buy* in the semantic space. For example, there are four types of common user behaviors: *page view*, *favorite*, *add to cart*, *buy*. Behavior *page view* is farthest from behavior *buy*, a sorted set of behaviors can be defined here, $\{r_{pv}, r_{fav}, r_{cart}, r_{buy}\}$. So, the first hyper meta-graph of a user solely contains the behavior of *buy*, the second one contains *page view* and *buy*, the third one contains all the behaviors except *add to cart*, and the last hyper meta-graph contains all of four types of behaviors.

5.3.2 Graph Encoders

Graph encoder is the essential part of the whole framework since it determines whether the framework can learn representative embeddings for users' behavior patterns from hyper meta-graphs. GNN models are widely used graph encoders, *e.g.,* GCN [53] and GAT [103]. Technically, any GNN models can be used in our framework with sufficient information (*e.g.,* edge types and node features). Note that exquisite GNN models are not required since hyper meta-graphs that carry rich semantic information (*e.g.,* geometric information, topological structures) have been built. In the practice of HMG-CR, geometric or topoloy-based GNNs, like GIN [118] and TAGCN [22], could be applied as the graph encoder because of their simplicity and effectiveness. GNN-based encoders can effectively leverage the structure information from the proposed hyper meta-path.

As mentioned in the previous section, each user in the recommendation system have $|\mathscr{R}|$ hyper meta-graphs. We assign $|\mathscr{R}|$ independent graph encoders to process these hyper meta-graphs accordingly. Note that these graph encoders are shared among different users. Given the t-th hyper meta-graph of user u_j and a graph encoder $g_t(\cdot)$, where $\cdot$ denotes a hyper meta-graph, we will have the embedding of the t-th hyper meta-graph of user u_j:

$$\mathbf{h}_t^j = g_t(\mathscr{HG}_t^j), \tag{5.1}$$

where $\mathbf{h}_t^j \in \mathbb{R}^h$ and h denotes the hidden dimension of the user behavior pattern embeddings and item embeddings, respectively.

5.3.3 Hyper Meta-Graph Contrastive Learning

For each user, we build several hyper meta-graphs. The graphs carry the user's interaction records. We can capture this information via graph encoders learning on the hyper meta-graphs separately. However, the behavior patterns of a user would be complicated. According to the example mentioned in the previous section, we have four different hyper meta-graphs for each user. The complexity of the hyper meta-graph is increasing following the number of behavior types it contains. For example, the first hyper meta-graph solely includes *buy*, and the second hyper meta-graph includes *page view* and *buy*. The second hyper meta-graph contains at least two purchasing patterns: buying the item directly, which is also contained in the first hyper meta-graph, and buying the item after viewing. Suppose we adopt graph encoders to learn on each hyper meta-graph separately. In that case, different behavior patterns in the same hyper meta-graph will be fused. This result may neglect the performances when using the learned behavior patterns for the recommendation. It is critical to extract different behavior patterns from a sequence of hyper meta-graphs whose complexities are cascadingly increasing. A potential solution is to contrast the hyper meta-graph with its previous one to obtain the differences (*e.g.,* different behavior patterns) between these two adjacent hyper meta-graphs.

Thanks to the recent success of GCL, *hyper meta-graphs discrimination* is proposed as the solution to obtain different behavior patterns by contrasting different hyper meta-graphs and InfoNCE [102] as the contrastive learning objective.

An example is given here. For the user u_j, it can give out two adjacent hyper meta-graphs of u_j, which are $\mathcal{HG}^j_{t-1}$ and $\mathcal{HG}^j_t$. $g_{t-1}(\cdot)$ and $g_t(\cdot)$ are assigned as their graph encoders, respectively. Hence, the embeddings of two hyper meta-graphs can be obtained:

$$\mathbf{h}^j_{t-1} = g_{t-1}(\mathcal{HG}^j_{t-1}), \tag{5.2}$$

$$\mathbf{h}^j_t = g_t(\mathcal{HG}^j_t). \tag{5.3}$$

In this example, $\mathbf{h}^j_{t-1}$ and $\mathbf{h}^j_t$ compose the negative pair. To satisfy the setting of InfoNCE, the positive pair must be constructed to fulfill the contrastive learning process. Following the GCL settings in GCC [78], $g_{t-1}(\cdot)$ is used to encode $\mathcal{HG}^j_t$ to obtain $\hat{\mathbf{h}}^j_t$:

$$\hat{\mathbf{h}}^j_t = g_{t-1}(\mathcal{HG}^j_t), \tag{5.4}$$

which is together with $\mathbf{h}^j_t$ to compose the positive pair. InfoNCE is adopted such that:

$$\mathcal{L}^j_{t-1,t} = -\log \frac{\exp\left(d(\mathbf{h}^j_t, \hat{\mathbf{h}}^j_t)\right)}{\exp\left(d(\mathbf{h}^j_t, \hat{\mathbf{h}}^j_t)\right) + \exp\left(d(\mathbf{h}^j_t, \mathbf{h}^j_{t-1})\right)}, \tag{5.5}$$

where $d(\cdot,\cdot)$ denotes the metrics measuring the distance between two vectors. For the recommendation system having $n+1$ users and $l+1$ different types of user behaviors, we will have a overall contrastive learning objective:

$$\mathscr{L}_{contra} = \frac{1}{n+1}\sum_{j=0}^{n}\sum_{t=1}^{l}\mathscr{L}^{j}_{t-1,t}. \tag{5.6}$$

The intuitions of adopting such a strategy are twofold:

- **Avoid generating the negative pair via graph augmentation**. Some works [132] utilize graph augmentation to generate negative pairs. However, in the recommendation scenario, graph augmentation would disturb the users' interaction records and affect behavior pattern generation, which may cause misleading results in the downstream recommendation tasks. Such a strategy is an alternative solution for us to generate the negative pair without disturbing the original semantics.
- **Bridge two contrasting hyper meta-graphs**. It is hard for us to link the embeddings generated from different graph encoders with different graphs in semantic space. However, with such a strategy, we can build an implicit connection between contrasting hyper meta-graphs in the contrastive learning process.

Finally, $|\mathscr{R}|$ user behavior embeddings for a user can be acquired after the contrastive learning process, which will be fed into fusion layer and downstream recommendation tasks.

5.3.4 Users' Multi-behavior Pattern Fusion

After obtaining $|\mathbf{R}|$ different embeddings which denote different behavior patterns of a user, we have to fuse them and obtain a unified embedding to conduct recommendations. There is a sort of widely used linear fusion methods, like *sum* and *mean*. And there is another type of fusion method, which is neural network-based methods (*e.g.,* Multi-Layer Perceptron (MLP) and Personalized Non-Linear Fusion (PNLF) [90]). Given a fusion function $f(*)$, we can have a unified behavior pattern embedding for the user:

$$\mathbf{h}^{j}_{uni} = f(\mathbf{h}^{j}_{0}, \mathbf{h}^{j}_{1}, \cdots, \mathbf{h}^{j}_{l}) \in \mathbb{R}^{h}. \tag{5.7}$$

5.3.5 Recommendation Task

There are plenty of collaborative filtering-based recommendation frameworks that leverage the explicit or implicit feedback of users [41]. To fully demonstrate the ability of the

proposed model and evaluate the quality of the user embeddings generated by HMG-CR, a simple vector product is used to make predictions instead of those complex and SOTA models to avoid the improvement brought by sophisticated recommendation models.

Let $\mathbf{h}^k$ denote the embedding of item i_k. With the unified behavior pattern embedding of user u_j, we can obtain the predicted score between the item and the user via:

$$\hat{p}_{u_j,i_k} = \mathbf{h}_{uni}^{j\ T} \cdot \boldsymbol{W} \cdot \mathbf{h}^k, \tag{5.8}$$

where the trainable weight matrix $\boldsymbol{W} \in \mathbb{R}^h$. The matrix $\boldsymbol{W}$ is used to map the unified behavior pattern to the space where item embeddings are in for score prediction.

To train the model, the negative logarithm of the likelihood function [41] is adopted:

$$\begin{aligned} \mathscr{L}_{rec} = - \sum_{(u_j,i_k) \in \mathscr{Y} \cup \mathscr{Y}^-} & p_{u_j,i_k} \log \hat{p}_{u_j,i_k} \\ & + (1 - p_{u_j,i_k}) \log(1 - \hat{p}_{u_j,i_k}). \end{aligned} \tag{5.9}$$

To normalize the loss value of loss function on recommendation tasks, the following formula is adopted as the objective,:

$$\mathscr{L}_{ave\_rec} = \frac{\mathscr{L}_{rec}}{|\{(u_j,i_k)|(u_j,i_k) \in \mathscr{Y} \cup \mathscr{Y}^-\}|}, \tag{5.10}$$

where $\mathscr{Y}$ and $\mathscr{Y}^-$ denote postive interaction records and sampled negative interaction records, $p_{u_j,i_k} \in \{0,1\}$ represents if there is an interaction between user u_j and item i_k.

Finally, to train the model in an end-to-end manner, the contrastive objective and the recommendation objective are coupled as the overall training objective:

$$\mathscr{L} = (1 - \beta) \cdot \mathscr{L}_{contra} + \beta \cdot \mathscr{L}_{ave\_rec}, \tag{5.11}$$

where β is a hyperparameter controlling the significance of two objectives.

5.4 Experiments on HMG-CR

This section evaluates HMG-CR on recommendation tasks with two real-world datasets. The comparison experiment results of HMG-CR and baselines are first reported. Then, how GCL works in HMG-CR is analyzed. Lastly, ablation studies are conducted on the graph encoder and fusion layer in the model.

5.4.1 Experiment Setup

Detailed experiment setup is listed in this section, including datasets, baselines, and experimental settings to facilitate reproducibility.

Table 5.1: Statistics of datasets for HMG-CR experiments.

Dataset	Taobao	Tmall
#users	48946	9368
#items	1500839	302722
#pv (percentage)	7723217 (85.17%)	1510303 (92.14%)
#fav (percentage)	436715 (4.82%)	102419 (6.25%)
#cart (percentage)	527221 (5.81%)	24557 (1.50%)
#buy (percentage)	380877 (4.20%)	104360 (6.37%)
#total	9068030	1639220
#ave_pv	157.79	161.22
#ave_fav	8.92	10.93
#ave_cart	10.77	2.62
#ave_buy	7.78	11.14
#ave_total	185.27	174.98

5.4.1.1 Datasets

The proposed framework is evaluated on two real-world datasets, which are of high quality and widely used, including Taobao[1] and Tmall[2]. To ensure the quality of the datasets, the customary practice [95] is followed to discard users and items with less than five interactions of *buy*. The users with too many interactions of *page view* in Tmall are filtered out to discard noise. The statistics of the filtered datasets are shown in Table 5.1.

5.4.1.2 Baselines

To verify the effectiveness of the proposed framework, we compare it with three categories of baselines. The first category is conventional GNNs, including **GCN** [53] and **GraphSAGE** [32], which cannot distinguish different types of edges in the graph. They treat different user behaviors in the same way. The second category is edge types-aware GNNs, including **GAT** [103] and **RGCN** [86], which can process various types of edges in the graph explicitly or implicitly to capture the features of different user behaviors. The last category is novel multi-behavior recommendation frameworks, **NMTR** [26] and **EHCF** [15], which achieve state-of-the-art performances on multi-behavior recommendation tasks.

[1]https://tianchi.aliyun.com/dataset/dataDetail?dataId=649
[2]https://tianchi.aliyun.com/dataset/dataDetail?dataId=47

5.4.2 Settings

For reproducibility, the details of the hyperparameter settings of the proposed framework are introduced in this part. HMG-CR is trained on dataset Taobao with learning rate lr = 0.0001, weight decay wd = 0.000001, hidden dimension h = 16 advised by [139], and 3-layer TAGCN [22] as the graph encoder. As to dataset Tmall, HMG-CR is tuned with the same learning rate, weight decay and hidden dimension. A 3-layer GIN is adopted as the graph encoder for HMG-CR on dataset Tmall. To ensure fairness in the comparison studies, the widely used *leave-one-out* strategy [41] is followed to conduct comparison studies. The metrics adopted are Recall@K and NDCG@K, which show the quality of the recommendations of top-K items.

5.4.3 Comparison Experiment Results

Table 5.2: Comparison experiment results of HMG-CR.

Dataset	Taobao				Tmall			
Metrics / Methods	Recall@5	Recall@10	NDCG@5	NDCG@10	Recall@5	Recall@10	NDCG@5	NDCG@10
GCN	0.2577	0.3589	0.1842	0.2167	0.2544	0.3775	0.1763	0.2163
GraphSAGE	0.2751	0.3826	0.1965	0.2312	0.2588	0.3695	.1813	0.2170
GAT	0.2782	0.3921	.1972	0.2339	0.2561	0.3735	0.1777	0.2158
RGCN	0.2714	0.3767	0.1946	0.2285	0.2725	0.4144	0.1749	0.2215
NMTR	0.2215	0.3781	0.1513	0.2012	.2780	.4230	0.1798	.2265
EHCF	.2882	.4166	0.1945	.2359	0.2451	0.4115	0.1581	0.2113
HMG-CR(SG)	0.3050	0.4417	0.2162	0.2608	0.2943	0.4329	0.1863	0.2321
HMG-CR(GCN)	0.3039	0.4441	0.2154	0.2613	0.2954	0.4332	0.1869	0.2324
HMG-CR(GAT)	0.3460	0.4390	0.2443	0.2746	0.3163	0.4320	0.2224	0.2604
HMG-CR(GIN)	0.3141	0.3627	0.2029	0.2191	**0.3547**	0.4313	**0.2642**	**0.2891**
HMG-CR(TAGCN)	**0.3588**	**0.4464**	**0.2639**	**0.2926**	0.2964	**0.4350**	0.1902	0.2359
Improvement	24.50%	7.15%	33.82%	24.04%	27.59%	2.84%	45.73%	27.64%

Table 5.2 lists the comparison experiment results for all methods on two datasets. Overall, the proposed framework HMG-CR with different graph encoders consistently and significantly outperforms all baselines in terms of all metrics. Particularly, our proposed framework has more significant improvement on the metric NDCG, which shows that our proposed framework pays more attention to sorting recommended items. Note that HMG-CR on dataset Taobao slightly outperforms that on dataset Tmall. According to the statistics of the two datasets, as shown in Table 5.1, we note that the average numbers of total interactions for each user are close in the two datasets, but there are differences among the distribution of numbers of different user behaviors. The ratio of *add to cart* in dataset Tmall

is much less than that in dataset Taobao. Each user in both datasets has four hyper meta-graphs since there are four different types of user behaviors. Due to the lack of *add to cart* in dataset Tmall, the third hyper meta-graph for a user, including *page view*, *add to cart*, and *buy*, is similar to the second hyper meta-graph for the user, including *page view* and *buy*. Under such a scenario, it is hard for graph contrastive learning to maximize the dissimilarities between the second hyper meta-graph and the third hyper meta-graph. Hence, the user behavior pattern embedding generated in this part would be misleading for unified user behavior pattern embedding generation.

Graph neural network-based methods performed unsatisfyingly in the comparison experiment. The interaction graphs for each user in the recommendation systems have simple structures (*e.g.,* bipartite graphs). Conventional GNN models, like GCN and GraphSAGE, may be insufficient to capture user behavior pattern embeddings on such simple graph structures. Edge types-aware GNN models, like GAT and RGCN, slightly outperform GCN and GraphSAGE since they integrate fruitful side information regarding different types of user behaviors. Overall, two categories of GNN models have no significant gaps because *page view* takes the most place in the datasets. Message passing and aggregation are not capable of capturing sophisticated relations among different types of user behaviors since the semantics of *page view* would conceal other information.

NMTR and EHCF are state-of-the-art multi-behavior recommendation frameworks. They leverage the well-designed recommendation models and multi-task learning strategy to utilize the supervision signals from all types of user behaviors. However, there is a limitation for both frameworks. Both of them have an assumption that each type of user behavior has strong connections with precedent types of user behaviors. This assumption is not solid because users' behavioral patterns are complex. The proposed HMG-CR adopts a more flexible manner to utilize GCL to capture the dependencies among different types of user behaviors instead of assuming there are strong connections between a behavior and the precedent one. Because of this, even without multi-task supervision signals and well-designed recommendation models, the proposed HMG-CR still outperforms NMTR and EHCF by leveraging the advantages of hyper meta-graphs and GCL.

5.4.4 Analysis of GCL in HMG-CR

In this part, the detailed mechanism of GCL in HMG-CR is introduced.

First, as shown in Figure 5.3, the training loss of the proposed framework on two datasets during the training process is demonstrated. The training loss is twofold, contrastive loss and recommendation loss. A clear tendency can be observed that contrastive loss drops

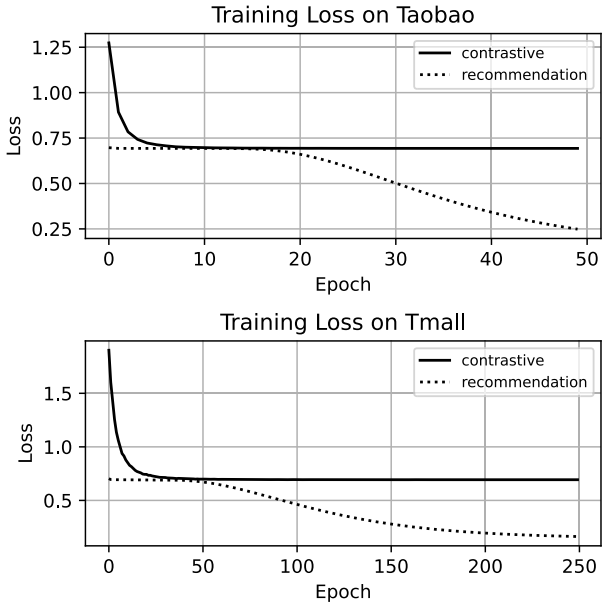
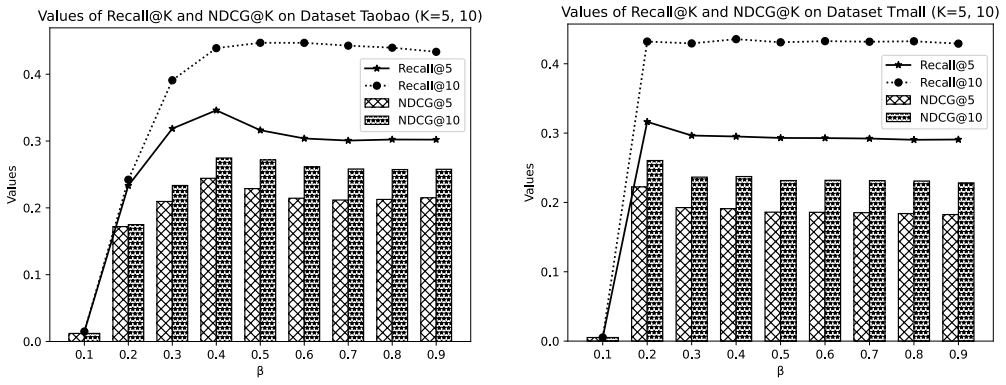


Figure 5.3: The contrastive loss and recommendation loss of HMG-CR on both datasets.

first and remains stabilized, and then the recommendation loss starts to decrease. This phenomenon reflects that the proposed framework first maximizes the dissimilarity among hyper meta-graphs to obtain user behavior pattern embeddings and update parameters on the recommendation task. The contrasting loss among hyper meta-graphs is maintained throughout the remaining training process.



(a) (b)

Figure 5.4: Performance of HMG-CR with different β.

To further illustrate the impact of GCL on HMG-CR, hyperparameter studies are conducted on β, which controls the relative significance of GCL tasks and recommendation tasks. The experimental results of hyperparameter studies are shown in Figure 5.4. Overall, HMG-CR is not that sensitive to β as long as β is not too small. However, it is worth

noting that HMG-CR has worse results when β takes the boundary values (*e.g.,* $\beta = 0.1$). When β is too low, the model pays less attention to the recommendation tasks. The model cannot acquire sufficient supervision signals from training data to update the parameters. Under this scenario, it is difficult for our model to converge quickly and precisely on recommendation tasks. With β increasing, the performances of the proposed framework increase accordingly. When β is larger than some specific values, *e.g.,* $\beta = 0.4$ for dataset Taobao and $\beta = 0.2$ for dataset Tmall, the performances start to decrease slightly. Large β values will neglect GCL tasks, which would undermine the ability of the model to acquire user behavior pattern embeddings from sophisticated hyper meta-graphs. This phenomenon also verifies that GCL is helpful to HMG-CR. In summary, GCL task and recommendation task should have a relatively balanced significance, and β should not be too large in our proposed framework to avoid decreasing model performances and cannot be too small, in which case the framework may not work.

5.4.5 Ablation Studies

Table 5.3: Ablation study of HMG-CR regarding the graph encoder.

Dataset	Taobao				Tmall			
Metrics / Methods	Recall@5	Recall@10	NDCG@5	NDCG@10	Recall@5	Recall@10	NDCG@5	NDCG@10
GIN	0.2682	0.3779	0.1892	0.2246	0.2817	0.4236	0.1878	0.2340
TAG	0.2784	0.3863	0.1994	0.2342	0.2845	0.4235	0.1869	0.2323
HMG-CR(GIN)	0.3141	0.3627	0.2029	0.2191	**0.3547**	0.4313	**0.2642**	**0.2891**
HMG-CR(TAG)	**0.3588**	**0.4464**	**0.2639**	**0.2926**	0.2964	**0.4350**	0.1902	0.2359

Table 5.4: Ablation study of HMG-CR regarding the fusion layer.

Dataset	Taobao				Tmall			
Metrics / Fusion	Recall@5	Recall@10	NDCG@5	NDCG@10	Recall@5	Recall@10	NDCG@5	NDCG@10
MEAN	**0.3460**	0.4390	**0.2443**	**0.2746**	**0.3163**	0.4320	**0.2224**	**0.2604**
SUM	0.3012	**0.4427**	0.2118	0.2580	0.2939	0.4345	0.1879	0.2343
MLP	0.3024	0.4344	0.2150	0.2579	0.2946	**0.4349**	0.1873	0.2336
PNLF	0.3046	0.4363	0.2157	0.2586	0.2944	0.4344	0.1865	0.2327

Graph Encoder. Choosing a proper graph encoder for the framework determines whether it can achieve good performance. Three common categories of GNNs are selected: conventional message passing-based GNNs including SG [112] and GCN, attention mechanism-based GNNs including GAT [103], and graph topological or geometric structure-aware GNNs

including TAGCN [22] and GIN [118]. The experiment results are shown in the Table 5.3. According to the results, the proposed HMG-CR with any graph encoders outperforms all baselines. Specifically, HMG-CR with SG or GCN slightly outperforms baselines since conventional message-passing-based GNNs are insufficient to capture complex user behavior features from the constructed hypermeta-graphs. Despite the user-item interactions in the hyper meta-graphs, there are also chronological dependencies among different user behaviors. With such sophisticated relations in the hyper meta-graphs, GAT leverages the attention mechanism to learn user behavior embeddings via adaptively distinguishing different relations (*i.e.,* edges) in the hyper-meta graphs. However, we replaced different types of edges, which represent user behaviors, with different types of nodes in the hyper meta-graphs. We explicitly add the information of interactions among users and items into the hyper meta-graph. This means that the improvement brought by the attention mechanism, which distinguishes different edges, is limited. Note that the hyper meta-graphs have a structure that is similar to tree topology. Hence, the hyper meta-graphs have not only fruitful semantic information but also excellent structure. HMG-CR with graph structure-aware graph encoders leverages the advantages of the hyper meta-graphs and achieves the best results in our experiments. To verify the improvement brought by our proposed framework instead of TAG or GIN solely, supplementary experiments of HMG-CR with TAG and GIN are conducted, which are also shown in Table 5.3.

Fusion Layer. The fusion layer is the output layer of the proposed framework. Ttwo categories of fusion layers are examined, linear fusion layer, *mean* and *sum*, and non-linear fusion layer, MLP and PNLF [90]. The experimental results are shown in the Table 5.4. According to the results, HMG-CR taking *mean* as the fusion layer achieves the best result. Overall, HMG-CR with a linear fusion layer performed better in our experiments. It is worth noting that there are mapping layers in MLP and PNLF. In this component, the mapping mechanism may disturb the user behavior pattern obtained in the space in which GCL was conducted. Hence, a linear fusion layer should be adopted to output the unified user behavior pattern embeddings for HMG-CR to avoid disturbing caused by conducting fusion in another embedding space.

5.5 Summary of HMG-CR

In this chapter, a novel concept of hyper meta-path and a novel framework are proposed, HMG-CR, which first utilizes GCL techniques in RS. It is a pioneering research work in the

related research domain and addresses RQ2.1, inspiring future works. Leveraging the advantages of hyper meta-path, HMG-CR achieves SOTA performances on the task of purchasing prediction on both datasets in the scenario of multi-behavior recommendation. An extensive analysis of HMG-CR is conducted, which fully demonstrates the details and properties of HMG-CR. The concept of hyper meta-path and the HMG-CR framework are flexible and can be applied to other heterogeneous graph mining tasks, improving the research progress in GCL, recommendations, and other related research domains.

The method presented in this chapter combines GCL objectives with downstream recommendation tasks and conduct joint training like many current methods [113, 134, 129]. However, during the experiments, it is revealed that joint training requires meticulous hyperparameter tuning to balance the weights of the GCL objectives and the recommendation objectives within the overall training framework. Without careful adjustment, the performance can be suboptimal.

Furthermore, it is important to note that GCL itself is fundamentally an unsupervised training paradigm, primarily aimed at pre-training. Such a gap between the characteristic of GCL and the joint training paradigm in current methods [113, 134, 129] suggests that there may be alternative training paradigms that could be more effective for recommendation systems. Inspired by these findings, next chapter tries to investigate and develop better training paradigms for GCL in recommendation scenarios to address RQ2.2. Such an exploration could lead to more robust and efficient models, ultimately enhancing the performance and applicability of GCL in RS.

CHAPTER 6

TRAINING PARADIGM OF GRAPH CONTRASTIVE LEARNING IN RECOMMENDATION SYSTEMS

GCL has emerged as an effective technology for various graph learning tasks. It has been successfully applied in real-world recommendation systems, where the contrastive loss and downstream recommendation objectives are combined to form the overall objective function. However, this approach deviates from the original GCL paradigm, which pre-trains graph embeddings without involving downstream training objectives. In this paper, we propose a novel framework called CPTPP, which enhances GCL-based recommendation systems by leveraging prompt tuning. This framework allows us to fully exploit the advantages of the original GCL protocol. Specifically, we first summarize user profiles in graph recommendation systems to automatically generate personalized user prompts. These soft prompts are then combined with pre-trained user embeddings for prompt tuning in downstream tasks. This helps bridge the gap between pre-training and downstream tasks. Our extensive experiments on three benchmark datasets confirm the effectiveness of CPTPP compared to state-of-the-art baselines. Additionally, a visualization experiment illustrates that user embeddings generated by CPTPP have a more uniform distribution, indicating improved modeling capability for user preferences.

This chapter examines GCL in RS at a higher level and presents an empirical study focusing on the training paradigm of GCL for recommendation. The empirical study indicates that the end-to-end training, *i.e.,* GCL objective and recommendation objective are combined together and trained in an end-to-end manner, may not be the optimal one.

Consequently, a novel paradigm, namely CPTPP, for GCL in RS is proposed with a GCL pre-training phase and a soft-prompting phase involved, addressing RQ2.2.

6.1 Brief Introduction to CPTPP

GCL has gained significant attention in the literature as a prominent self-supervised learning paradigm. Several recent studies have showcased the effectiveness of GCL in various general graph representation tasks [104, 78, 122, 149, 132], including node classification and link prediction. Moreover, GCL has also demonstrated its applicability in real-world domains[126], such as RS [121, 136, 64]. By introducing additional self-supervision signals, GCL provides recommendation systems with a means to address the challenge of lacking sufficient supervision signals.

Most recommendation methods based on GCL typically combine contrastive loss with recommendation objectives to optimize the model in an end-to-end manner. However, this training protocol does not align with the purpose of GCL, which is primarily designed for pre-training graph representations without involving downstream task objectives [104, 78]. In this approach, GCL first pre-trains embeddings, then they are fine-tuned to specific tasks using downstream models. Incorporating both GCL and recommendation objectives into the overall training objective can disrupt the embedding pre-training process and require careful control of the weight placed on contrastive loss. Additionally, previous studies on GCL-based recommendation methods [121, 64] have shown that the weights of contrastive loss in the overall objective are significantly smaller compared to the weight on the recommendation objective. This is done to ensure desired performance on recommendation tasks. Therefore, based on these observations, simply combining contrastive loss with downstream recommendation objectives may not be effective for recommendation tasks.

The disparity between the pre-training objective and downstream tasks hinders the effective extraction of useful information from pre-trained embeddings by downstream models [61, 115]. Consequently, researchers often opt to combine GCL with recommendation objectives. However, it is important to note that GCL pre-training targets primarily assess the agreement of mutual information among graph elements, such as nodes, edges, and sub-graphs. This differs from conventional graph learning tasks like node classification and link prediction. Consequently, the pre-training targets of GCL also significantly diverge from downstream recommendation objectives that involve interaction (link) prediction between users and items. Consequently, the reduction of such dissimilarities is essential to enhance the performance of GCL-based recommendation approaches.

In this section, the CPTPP framework is presented as an extension of recent advancements in prompt tuning for enhancing recommendation performance [111, 137] utilizing user embeddings pre-trained by GCL. The technique of prompt tuning has emerged as a prominent method for fine-tuning pre-trained models. By constructing appropriate prompts for downstream learning modules, this approach effectively reformulates downstream tasks, thereby reducing disparities [115, 61, 77, 92]. By incorporating prompt-tuning, existing GCL-based recommendation methods can be modified to align with the original GCL protocol involving pre-training and prompt-tuning. Previous endeavors have also explored the integration of prompt learning into conventional recommendation models [115, 28]. Despite their advantages, applying the prompt mechanism directly to GCL-based recommendation methods is still difficult and not straightforward, *i.e., how to generate personalized user prompts using only the user-item interaction graph without side information (*e.g., *age and occupation)*? To address this issue, three methods are summarized to produce different user profiles, including *historical interaction records*, *adjacency matrix factorization*, and *high-order user relations*, based on the user-item interaction graph for the personalized user prompt generation, which is applicable in situations devoid of side information. Comprehensive experiments conducted on three publicly available datasets illustrate the effectiveness of the proposed method with different types of prompts.

In short, the contributions of this work are three-fold: (1) A reformulation of existing GCL-based recommendation methods is proposed by incorporating the prompt tuning mechanism. This allows us to fully leverage the advantages of GCL during the pre-training phase, rather than relying on the combination of contrastive loss with downstream objectives. (2) Three user profiles derived from the user-item interaction graph as inputs for the prompt generator are summarized. By using these profiles, CPTPP is able to generate personalized prompts that enhance the quality of user embeddings in graph-based recommendation systems. (3) Extensive experiments are conducted on three publicly available benchmark datasets to validate the effectiveness of our model. Through these experiments, the important components and hyper-parameters of CPTPP are analyzed and the impact of different personalized prompts generated by our method is also investigated.

6.2 Methodology

In this part, graph **C**ontrastive **P**re-**T**raining with **P**rom**P**t-tuning for recommendation (CPTPP), is introduced to reveal the intuitions and the technical details.

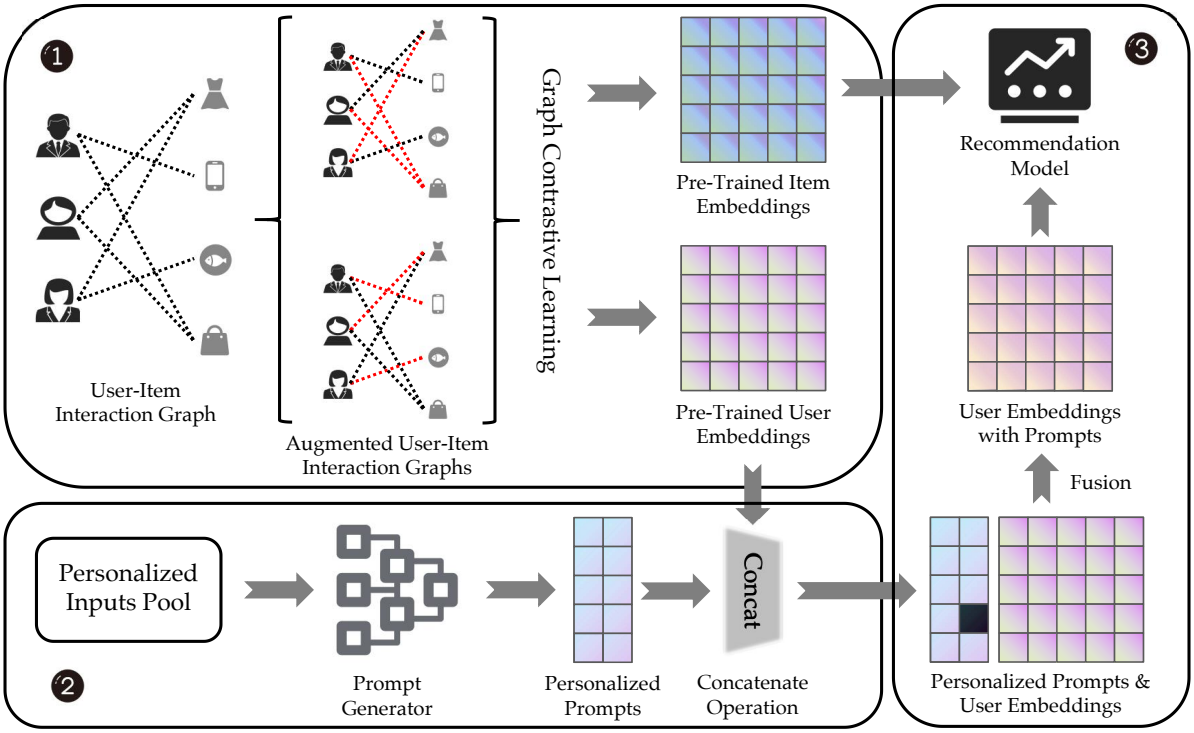


Figure 6.1: The overview of the proposed CPTPP framework.

6.2.1 Framework Overview

There are three modules in the proposed CPTPP method, as shown in Figure 6.1: (1) GCL pre-training module, leveraging the advantages of GCL to the pre-train user and item embeddings, (2) personalized prompts generation module, applying prompt mechanism to generate personalized prompts for users, and (3) recommendation module, fusing the generated personalized prompts and pre-trained user embeddings to conduct prompt-tuning for the downstream recommendation task.

6.2.2 Graph Contrastive Learning Module

In order to achieve optimal performance in downstream tasks, the selection of a suitable pre-training strategy is crucial for generating high-quality inputs for downstream modules. GCL has been demonstrated as a powerful technique for graph pre-training [104, 78, 132, 149] and has emerged as an effective tool for leveraging self-supervised signals to enhance graph-based recommendation models [121, 64, 113, 136]. In the case of graph-based recommender systems, GCL represents a viable option for pre-training embeddings. Furthermore, our work specifically focuses on reforming and improving existing GCL-based recommendation methods. Therefore, it is imperative that we formulate a GCL module within

our proposed method. Current GCL-based recommendation methods [64, 113, 121, 136] have explored various graph augmentation techniques on user-item interaction graphs in order to generate augmented graphs for GCL, enabling the extraction of informative semantics from the graph structures. Alternatively, some studies [135, 64] have designed context embeddings tailored for GCL in recommended settings. Although different approaches for constructing contrasting samples exist, they have the same GCL training backbone.

Here, a formal description of the GCL training protocol is given. Let u_i denote the target graph element (*e.g.,* user node), u_i^+ represent the positive sample generated from u_i (*e.g.,* the neighbor node of the target node), and $\mathscr{U}^- = \{u_{i,0}^-, u_{i,1}^-, \cdots, u_{i,t}^-\}$ be the set containing t contrasting samples of u_i (*e.g.,* non-neighbour nodes of the target node). Considering the settings of the recommendation task, we use G to represent the overall user-item graph, and all the target, positive sample, and contrasting samples are within graph G. To acquire embeddings of these graph elements, we adopt $f(*)$ as the graph encoder to process them, and the target embedding is denoted by $\mathbf{u}_i = f(u_i; G)$, $\mathbf{u}^+ = f(u_i^+; G)$ is the embedding of the positive sample, and $\{\mathbf{u}_{i,0}^-, \mathbf{u}_{i,1}^-, \cdots, \mathbf{u}_{i,t}^-\}$ are the list of embeddings of the negative contrasting samples. Following the settings of InfoNCE [102], the self-supervised learning objective can be formulated as follows:

$$\mathscr{L}_{contra} = -\log \frac{\exp(sim(\mathbf{u}_i, \mathbf{u}_i^+)/\tau)}{\sum_{t=0}^{|\mathscr{U}^-|} \exp(sim(\mathbf{u}_i, \mathbf{u}_{i,t}^-)/\tau)}, \tag{6.1}$$

where τ is the temperature hyper-parameter and sim$(\cdot, \cdot)$ is the similarity metric. In existing research works [113, 121, 136, 135, 64], researchers usually combine the aforementioned contrastive learning loss function with the recommendation objectives to formulate an overall objective function to train the model in an end-to-end manner:

$$\mathscr{L}_{overall} = \mathscr{L}_{rec} + \lambda \cdot \mathscr{L}_{contra}, \tag{6.2}$$

where λ is a hyper-parameter that controls the weight of the contrastive learning objective. However, as mentioned in the brief introduction, the proposed CPTPP adopts a *pre-training and prompt-tuning* manner to train the model and treats GCL as a pre-training task instead of combining the contrastive loss with recommendation objectives. To leverage recent research progress in GCL, various GCL learning methods tailored for the recommendation task can be adopted here, like NCL [64], SGL [113], and SimGCL [136], to obtain high-quality user and item embeddings. Then, the pre-trained user and item embeddings will be processed using the prompt mechanism.

6.2.3 Prompts Generation Module

Following the pre-training phase, our method, named CPTPP, incorporates a personalized prompts generation module to utilize the pre-trained user and item embeddings effectively. The primary objective of this module is to address the limitations present in existing prompt and recommendation research. Prior studies [92, 77, 115] have highlighted the triviality and resource-intensive nature of hard prompt design, making it impractical for real-world scenarios. Additionally, it is worth noting that most current approaches rely on side information (*e.g.,* user descriptions) to generate prompts and lack a specific paradigm for prompting in graph-based recommendation scenarios. To overcome these limitations, the integration of a prompt generator [115] is proposed, which generates personalized prompts tailored specifically for graph-based recommendation contexts.

6.2.3.1 Personalized Prompt Generator

The main scope of the generated prompts lies in narrowing the gap between the pre-training targets and the downstream objectives to utilize the pre-trained models or embeddings better. Some research works designed hard prompts tailored for recommendation tasks, converting recommendation tasks into NLP tasks [28], which unifies multiple recommendation tasks in a single framework. For example, a convention recommendation task can be converted to a sentence, '*User 123 will purchase item [id_token]*'. Then, NLP techniques will be applied to predict the token. However, PPR [115] argued that such an NLP-style hard prompt designing method has two major limitations: (i) It is difficult to apply NLP techniques to predict the designated tokens since these tokens could be a user ID, item ID, or ratings, which lack meaningful semantics. (ii) The designed hard prompts are universal and cannot be customized for various users or items.

To address the challenges, a method to construct personalized prompts from user profiles in a soft prompt automatic generation manner [61, 77, 115] is adopted. Let $\mathbf{x}_i^u \in \mathbb{R}^{d\times 1}$ denote the profile of user i. Then, all the users' profiles can be concatenated to form the user profile matrix $\mathbf{X}^u = [\mathbf{x}_1^u, \mathbf{x}_2^u, \cdots, \mathbf{x}_n^u] \in \mathbb{R}^{d\times n}$, where n is the number of users. This matrix will be fed into a two-layer MLP $f(\cdot)$ to acquire personalized prompts for each user $\mathbf{P}^u = [\mathbf{p}_1^u, \mathbf{p}_2^u, \cdots, \mathbf{p}_n^u] \in \mathbb{R}^{p\times n}$ as follows:

$$\mathbf{P}^u = f(\mathbf{X}^u) = \mathbf{W}_2 \cdot \alpha(\mathbf{W}_1 \cdot \mathbf{X}^u + b_1) + b_2 \in \mathbb{R}^{p\times n}, \tag{6.3}$$

where p is the prompt size, $\mathbf{W}_1 \in \mathbb{R}^{h\times d}$ and $\mathbf{W}_2 \in \mathbb{R}^{p\times h}$ are trainable weights, $b_1 \in \mathbb{R}^{h\times n}$ and $b_2 \in \mathbb{R}^{d\times n}$ are biases, and $\alpha(\cdot)$ is the activation function, d and h represent the dimensions

of the pre-trained embeddings and hidden dimensions, and p denotes the dimension of the generated prompts, respectively.

The generated prompts will be concatenated with the pre-trained user embeddings in a pre-fixed manner [61] and tuned by the downstream objectives in the recommendation module to fulfill the process of prompt-tuning. Specifically, let $\mathbf{U}_{pre\_train} \in \mathbb{R}^{d \times n}$ denote pre-trained user embeddings. Then, the inputs from the user side for the recommendation module can be obtained:

$$\mathbf{U}_{concat} = \begin{bmatrix} \mathbf{P}^u \\ \mathbf{U}_{pre\_train} \end{bmatrix}^T \in \mathbb{R}^{n \times (p+d)}. \tag{6.4}$$

6.2.3.2 Personalized Inputs for Prompt Generation

The quality of the generated prompts depends on the personalized inputs for the generator. Current research on prompt learning for recommendation mainly focuses on utilizing existing user features (*e.g.,* age and occupation) and historical interaction records as the inputs to generate personalized prompts [115, 59, 114]. However, these methods are designed for conventional and sequential recommendations, which are not entirely aligned with graph recommendations. It is necessary to summarize and explore how to generate personalized prompts from the perspective of the graph recommendation system. In this section, three inputs for the generator to generate personalized prompts are summarized: historical interaction records, adjacency matrix factorization, and high-order user relations.

Historical Interaction Records. It is a common and widely-used method to illustrate users' features or preferences via aggregating his/her historical interaction records, which is feasible in various scenarios in recommendation systems. Let $\mathscr{I}_k^u = \{i_{k,1}, i_{k,2}, \cdots, i_{k,m}\}$ denote the item set which are purchased by user k. We use $\mathbf{i}_{k,j} \in \mathbb{R}^d$ to represent the embedding of the j-th item in user k's purchase history. Then, the profile of user k can be acquired by aggregating embeddings of those items purchased by the user k:

$$\mathbf{x}_k^u = Aggregation(\mathbf{i}_{k,1}, \mathbf{i}_{k,2}, \cdots, \mathbf{i}_{k,m}), \tag{6.5}$$

where $Aggregation(*)$ is the aggregation function that reads out the user's profile.

The concatenation of all the user profiles can form the matrix $\mathbf{X}^u$ to be processed by the personalized prompt generator. Let $\mathbf{A} \in \mathbb{R}^{n \times q}$ denote the adjacency matrix for the recommendation system, which contains n users and q items. If the pre-trained item embeddings $\mathbf{I}_{pre\_train} = [\mathbf{i}_1, \mathbf{i}_1, \cdots, \mathbf{i}_q] \in \mathbb{R}^{d \times q}$ was obtained, then, user profile matrix can be acquired via:

$$\mathbf{X}^u = (\mathbf{A} \cdot \mathbf{I}_{pre\_train}^T)^T. \tag{6.6}$$

Adjacency Matrix Factorization. The adjacency matrix is an effective tool for demonstrating user-item relations in the recommendation system. However, the adjacency matrix usually suffers from sparsity problems and thus cannot be smoothly applied in many real-world recommendation scenarios. To address this problem, researchers proposed several Matrix Factorization (MF) methods [93, 42] to decompose the adjacency matrix to obtain two matrices, $\mathbf{U}$ and $\mathbf{V}$, denoting the latent embeddings for users and items, which are much denser than the adjacency matrix $\mathbf{A}$ itself. The process of MF can be formulated as:

$$\arg\min_{\mathbf{U},\mathbf{V}} \sum_{i=1}^{n} \sum_{j=1}^{q} (\mathbf{A}_{i,j} - \hat{\mathbf{A}}_{i,j}), \tag{6.7}$$

where $\hat{\mathbf{A}}_{i,j} = \sum_k \mathbf{U}_{i,k} \cdot \mathbf{V}^T_{k,j} = \mathbf{U}_i \mathbf{V}^T_j$. After the MF process, the latent matrix of users, $\mathbf{U} \in \mathbb{R}^{n \times d}$, can be obtained, serving as the user profile matrix $\mathbf{X}^u$ after transposed $\mathbf{X}^u = \mathbf{U}^T$ and can be fed into a personalized prompt generator to produce personalized prompts $\mathbf{P}^u$ for each user. Specifically, the size of latent embeddings is set to d, which is the same as the dimension of pre-trained embeddings.

High-Order User Relations. Learning informative embeddings from a 1-hop user-item interaction graph is challenging when there is no side information. To address this limitation, we propose to leverage high-order user relations to enrich the learned embeddings via constructing 2-ego graphs for each user node to find the links between the other users and itself [109]. Then, we fuse the target user's purchase history and high-order neighbor embeddings to represent the target user profile.

The high-order connectivity matrix must be first constructed to achieve the goal. Let $\mathbf{A}^* = \bar{\mathbf{A}} \cdot \bar{\mathbf{A}} \in \mathbb{R}^{(n+q)\times(n+q)}$ denote the high-order connectivity matrix, where $\bar{\mathbf{A}} = \begin{bmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{(n+q)\times(n+q)}$, recording all the users and items to which a user or item node is connected. Then, we build the matrix $\mathbf{E} = [\mathbf{U}_{pre\_train}, \mathbf{I}_{pre\_train}]^T \in \mathbb{R}^{(n+q)\times d}$ to store pre-trained embeddings. Next, we can acquire matrix $\mathbf{Q} \in \mathbb{R}^{n \times d}$, which are the users' personalized profiles about high-order user relations, via:

$$\begin{bmatrix} \mathbf{Q} \\ \mathbf{M} \end{bmatrix} = \mathbf{A}^* \cdot \mathbf{E} \in \mathbb{R}^{(n+q)\times d}, \tag{6.8}$$

where $\mathbf{M} \in \mathbb{R}^{q \times d}$, denoting the high-order item relations. Then, a matrix transpose operation is required to obtain the user profile matrix $\mathbf{X}^u = \mathbf{Q}^T$.

6.2.4 Recommendation Module

After the pre-training and the personalized prompts generation phase, a recommendation module is equipped to conduct recommendation tasks so that the prompt-tuning module can be verified if it can rectify the pre-trained embeddings by GCL and makes them be adapted to the downstream recommendation tasks better. In this module, the inner product of user and item embeddings is adopted as the predicted score for the recommendation. Bayesian Personalized Ranking (BPR) [82] is adopted as the training objective to tune the pre-trained embeddings based on the predicted scores. The motivation for formulating such a simple recommendation module is to avoid the performance gain brought by the delicate designs of those advanced recommendation models, which could affect the observations on the proposed CPTPP method.

6.2.4.1 Prompts and Pre-Trained Embeddings Fusion

The generated personalized prompts and the pre-trained user embeddings in the previous step need to be concatenated to have $\mathbf{U}_{concat} \in \mathbb{R}^{n\times(p+d)}$, whose dimensionality is different from the pre-trained embeddings $\mathbf{I}_{pre\_train} \in \mathbb{R}^{n\times d}$. Hence, the first step requires to fuse the personalized prompts and the pre-trained user embeddings for the objective training of recommendation. Specifically, a MLP $g(\cdot)$ is adopted as the mapping function that is $g : \mathbb{R}^{n\times(p+d)} \rightarrow \mathbb{R}^{n\times d}$. Then, dimensionality-reduced user representations can be obtained $\mathbf{U}^* = g(\mathbf{U}_{concat}) \in \mathbb{R}^{n\times d}$, enhanced by the personalized prompts. After that, we can apply the inner product to predict how likely the user i would interact with the item j by $\hat{y}_{i,j} = \mathbf{u}_i^* \cdot \mathbf{i}_j$, where $\mathbf{u}_i^*$ is the i-th row of $\mathbf{U}^*$.

6.2.4.2 Training Objective for Recommendation Task

For simplicity and fair comparison, BPR [82] loss is adopted as the training objective for the recommendation task. For each user i:

$$\mathscr{L}_{rec}^{i} = \sum_{j^+\in\mathscr{I}_i^u} \sum_{j^-\in\mathscr{I}\setminus\mathscr{I}_i^u} -\log\sigma(\hat{y}_{i,j^+} - \hat{y}_{i,j^-}). \tag{6.9}$$

However, it is unaffordable to consider all the unobserved interactions of the user i. Therefore, several negative items $\mathscr{N}_i^u$ are sampled, where $|\mathscr{N}_i^u| << |\mathscr{I}\setminus\mathscr{I}_i^u|$, in practice.

Moreover, $L2$-norm is introduced into the training objective to regularize the parameters in the model to address the overfitting problem and improve generalization ability.

Algorithm 2: CPTPP algorithm

Input: User embedding table $\mathbf{U}_E$; Item embedding table $\mathbf{I}_E$; User-item interaction graph adjacency matrix $\mathbf{A}$; Graph contrastive learning model $f(*)$; User profile $\mathbf{X}^u$; Prompt generator $g(\cdot)$; Multi-layer perceptron $\texttt{MLP}(\cdot)$; Pre-train epoch i; Prompt-tune epoch j.

Output: User and item embedding tables $\mathbf{U}_E^*$ and $\mathbf{I}_E^*$.

Pre-train phase:
Initialize $\mathbf{U}_E, \mathbf{I}_E$; $\mathbf{U}_E^{'}, \mathbf{I}_E^{'} \leftarrow \mathbf{U}_E, \mathbf{I}_E$;
$count = 0$;
while $count < i$ **do**
    `// Update user and item embedding tables.`
    $\mathbf{U}_E^{'}, \mathbf{I}_E^{'} = f(\mathbf{U}_E^{'}; \mathbf{I}_E^{'}; \mathbf{A})$;
    $count = count + 1$;
end
Prompt-tune phase:
$\mathbf{U}_E^* \leftarrow \mathbf{U}_E^{'}$; $\mathbf{I}_E^* \leftarrow \mathbf{I}_E^{'}$;
$count = 0$;
while $count < j$ **do**
    `// Personalized prompt generation.`
    $\mathbf{P}^u = g(\mathbf{X}^u)$;
    `// Concatenate & fusion.`
    $\mathbf{U}_E^* = \texttt{MLP}([\mathbf{P}^u; \mathbf{U}_E^*]^T) \in \mathbb{R}^{n \times d}$;
    Optimise $\mathscr{L} = \sum_{i \in \mathscr{U}} \mathscr{L}_{rec}^i + \lambda ||\Theta||_2^2$;
    Update $\mathbf{U}_E^*, \mathbf{I}_E^*$;
    $count = count + 1$;
end
return $\mathbf{U}_E^*, \mathbf{I}_E^*$

Therefore, the overall objective function can be formulated as:

$$\mathscr{L} = \sum_{i \in \mathscr{U}} \mathscr{L}_{rec}^i + \lambda ||\Theta||_2^2. \tag{6.10}$$

6.2.5 CPTPP Algorithm Summary

After the training process ends, the model can be used to conduct inference. For the inference phase, pre-train and prompt-tune will not be performed again. What need to do is to extract target user and item embeddings from the trained embedding tables. Then, their embeddings' inner product can be calculated to predict the probability if the user will interact with the item in the future.

The complete training procedure of CPTPP is illustrated by Algorithm 2. The user and item embedding tables are first initialized (line 2). Then, a GCL model is applied to conducting embedding pre-training (line 4 ~ 7). Next, the prompt-tuning phase assigns the pre-trained embeddings to $\mathbf{U}_E^*$ and $\mathbf{I}_E^*$ (line 9). Following, the user profiles are input into the prompt generator to produce the personalized prompts (line 12) and combine them with $\mathbf{U}_E^*$ (line 13). Finally, $\mathbf{U}_E^*$ and $\mathbf{I}_E^*$ are used to calculate the loss and update them accordingly (line 14 ~ 15). The update procedure will repeat until the termination condition is achieved (line 11 ~ 17). The procedures above are all the workflows of CPTPP.

6.3 Experiment on CPTPP

To verify the effectiveness of the proposed method, CPTPP, extensive experiments are conducted to demonstrate the results with insightful analysis in this section.

Table 6.1: Dataset statistics in CPTPP experiments.

Dataset	#Users	#Items	#Interactions	Density
Douban	2,848	39,586	894,887	0.794%
ML-1M	6,040	3,900	1,000,209	4.246%
Gowalla	29,858	40,981	1,027,370	0.084%

6.3.1 Experimental Setup

This section introduces the experimental settings, including datasets and baselines used, performance metrics, and hyper-parameter settings for CPTPP.

6.3.1.1 Datasets

To verify the performance of CPTPP in the recommendation task, three popular datasets are selected: **Douban** [140], **MovieLens-1M** [33], and **Gowalla** [63]. The detailed statistics about the three datasets are listed in Table 6.1. For each dataset, 80% of historical user-item interactions are randomly selected as the training set, and the rest 20% records will serve as the testing set. Following the settings widely adopted by the research community [109, 40], each user-item interaction record is treated as a positive instance and it will be coupled with negative instances generated by negative sampling, which are unobserved user-item interactions in the dataset.

6.3.1.2 Baselines

Several baselines are selected for comparison experiments: **BPR-MF** [55], **BUIR** [58], **SelfCF** [147], **NCL** [64], and **SimGCL** [136]. For CPTPP, we have three variations, which are CPTPP-H, CPTPP-M, and CPTPP-R, respectively. -*H* takes historical interaction records for personalized prompt generation. -*M* indicates that we take adjacency matrix factorization for personalized prompts generation. Furthermore, -*R* takes high-order user relations for the personalized prompt generation.

6.3.1.3 Metrics

To evaluate the quality of top-K recommendation, three popular metrics are adopted, which are *Hit Ratio@K*, *Precision@K*, and *NDCG@K*, respectively. In the settings of this work, the value of K is set to 5 and 20. Following the evaluation protocol in [64, 136], the full ranking strategy [146] is adopted.

6.3.1.4 Hyper-parameter Settings

To ensure reproducibility, the comprehensive hyper-parameter settings for implementing our proposed CPTPP are disclosed. Detailed hyper-parameter settings are listed in Table 6.2 for reproducibility. The dimensionality of the representation embeddings of users and items is set to 64, and the personalized prompt size is chosen from $\{8, 16, 32, 64, 128\}$. For the pre-train phase, the maximum training epoch is 10, and for the prompt-tune stage, the training epoch is set to 100. The training batch size is 512 for the relatively smaller datasets, including Douban and ML-1M. For Gowalla, it is set to 2048. The learning rate and λ are set to $1e^{-3}$ and $1e^{-4}$, where λ is the weight for the $l2$-norm term in the overall training objective. The default number of layers of GNN used in the models is set to 2.

6.3.2 Experiment Results

Extensive experiments are conducted and related analysis is provided in this section, including comparison experiment, hyperparameter study, and ablation study, respectively.

6.3.2.1 Overall Comparison Studies

Table 6.3 shows the comparison results among all the baselines and different versions of the proposed methods. (i) It can first observe that the traditional method BPR-MF is outperformed by all the other methods as they utilize contrastive learning to introduce extra

Table 6.2: Summary of hyper-parameter settings for CPTPP.

Hyper-parameter	Notation	Dataset		
		Douban	ML-1M	Gowalla
Hidden dimension size	d	64	64	64
Pre-train epoch	-	10	10	10
Prompt-tune epoch	-	100	100	100
Batch size	-	512	512	2048
Learning rate	-	0.003	0.001	0.001
Regularizer weight	λ	0.0001	0.0001	0.0001
Number of GNN layers	-	2	2	2
Dropout rate	-	0.1	0.1	0.1
Temperature parameter	τ	0.2	0.2	0.2
Prompt size	p	{8, 16, 32, 64, 128, 256}	{8, 16, 32, 64, 128, 256}	{8, 16, 32, 64, 128, 256}

unsupervised training signals. (ii) Among all the baselines, GCL-based recommendation methods, including NCL and SimGCL, significantly and consistently outperform those self-supervised recommendation methods without graph learning module equipped, BUIR and SelfCF. It is because those GCL-based methods adopt graph neural networks, leveraging the sophisticated structure semantics in user-item interaction graphs to enrich learned user embeddings and item embeddings. (iii) But it worth noting that SimGCL only outperforms NCL on dataset Gowalla, which has a much larger scale than the others, probably because SimGCL adopts a simplified GCL method that relieves the model overfitting problem on a large-scale dataset. It is the potential reason NCL outperforms SimGCL on smaller datasets, as the simplified GCL method may not provide sufficient self-supervised training signals. (iv) Though the proposed CPTPP solely adopts BPR loss, which is significantly different from the pre-training procedure, for the recommendation task training, the prompt learning mechanism is utilized to better adapt the embeddings pre-trained by the GCL method to the downstream task, expecting to improve the recommendation performance. According to the experiment results, all versions of our proposed method achieve competitive results. Such results reflect prompt-tuning's effectiveness in narrowing the gap between the pre-train objective and the downstream tasks.

To further evaluate the performance of the GCL-based recommendation methods, the user embeddings produced by CPTPP and baselines are processed by t-SNE and Gaussian Kernel Density Estimation (KDE) to conduct visualization.

According to Figure 6.2, it can observe that (i) embeddings learned by SimGCL fall into several hot areas on dataset ML-1M, and they are centralized in a small area on datasets Douban and Gowalla. (ii) NCL exhibits better performance as the distribution of the user

Table 6.3: Comparison experiment results of CPTPP and baselines.

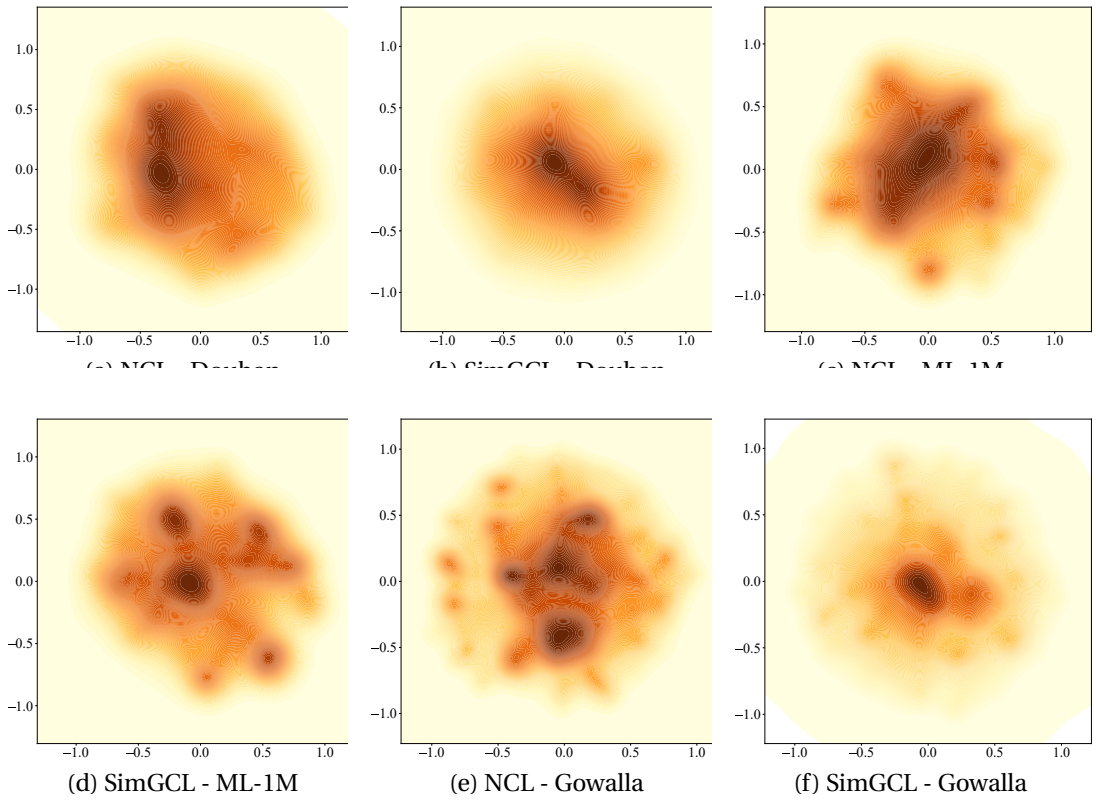
Datasets	**Metrics**	**Methods**							
		BPR-MF	BUIR	SelfCF	NCL	SimGCL	CPTPP-H	CPTPP-M	CPTPP-R
Douban	Hit Ratio@5	0.0134	0.0156	.0161	.0161	.0161	0.0164	**0.0165***	0.0164
	Hit Ratio@20	0.0446	0.0492	0.0502	.0507	0.0489	0.0521	**0.0528***	0.0523
	Precision@5	0.1812	0.2113	0.2185	.2187	0.2182	0.2221	**0.2235***	0.2224
	Precision@20	0.1512	0.1667	0.1699	.1717	0.1657	0.1766	**0.1790***	0.1772
	NDCG@5	0.1904	0.2209	0.2264	0.2313	.2370	0.2359	**0.2378***	0.2355
	NDCG@20	0.1749	0.2019	.2058	0.1958	0.2020	0.2065	**0.2098***	0.2070
ML-1M	Hit Ratio@5	0.0469	0.0617	0.0624	.0655	0.0631	**0.0676***	0.0674	0.0672
	Hit Ratio@20	0.1454	0.1519	0.1643	.1796	0.1698	0.1851	**0.1861***	0.1845
	Precision@5	0.1800	0.2368	0.2396	.2513	0.2420	**0.2592***	0.2585	0.2577
	Precision@20	0.1395	0.1457	0.1576	.1723	0.1629	0.1776	**0.1785***	0.1770
	NDCG@5	0.1968	0.2722	0.2689	.2818	0.2767	**0.2919***	0.2895	0.2878
	NDCG@20	0.2103	0.2367	0.2508	.2683	0.2670	0.2781	**0.2782***	0.2756
Gowalla	Hit Ratio@5	0.0429	0.0479	0.0497	0.0488	.0513	0.0518	0.0512	**0.0519***
	Hit Ratio@20	0.1039	0.0993	0.1042	0.1040	.1065	0.1115	0.1103	**0.1120***
	Precision@5	0.0624	0.0698	0.0723	0.0711	.0746	0.0754	0.0745	**0.0755***
	Precision@20	0.0378	0.0361	0.0379	0.0378	.0387	0.0406	0.0401	**0.0407***
	NDCG@5	0.0770	0.0911	0.0939	0.0894	.0963	**0.0963**	0.0953	0.0961
	NDCG@20	0.0939	0.0990	0.1036	0.1005	.1126	**0.1092**	0.1083	**0.1092**

"*" indicates that CPTPP outperforms the best baseline significantly (i.e., two-sided t-test with $p < 0.05$).

embeddings expands to a relatively larger area than that of SimGCL. Compared to our proposed method CPTPP, we can observe that CPTPP has a more uniform distribution of the produced user embeddings, illustrated by the uniformity of the color maps, especially on dataset ML-1M and Gowalla. As suggested in [64], the more uniform the embedding distribution is, the more capability to model the diverse preferences of users the method has, which reflects CPTPP's superiority. Hence, CPTPP has a more uniform distribution of the produced user embeddings, illustrated by the uniformity of the color maps, especially on dataset ML-1M and Gowalla. As suggested by Z. Lin *et al.* [64], the more uniform the embedding distribution is, the more powerful the capability to model the diverse preferences of users the produced embeddings will have, which reflects the superiority of CPTPP.

6.3.2.2 Hyperparameter Studies

To investigate the properties of CPTPP, hyperparameter studies are conducted on an important term, the dimension size of the personalized prompt. By fixing all the other hyperparameters, the performance of three versions of the proposed CPTPP are comprehensively examined on all the datasets with different prompt sizes. Specifically, the size of the personalized prompt is selected from $\{8, 16, 32, 64, 128, 256\}$. Two metrics are chosen, *Precision*@5



(d) SimGCL - ML-1M (e) NCL - Gowalla (f) SimGCL - Gowalla

Figure 6.2: The visualization results of the user embeddings generated by baselines.

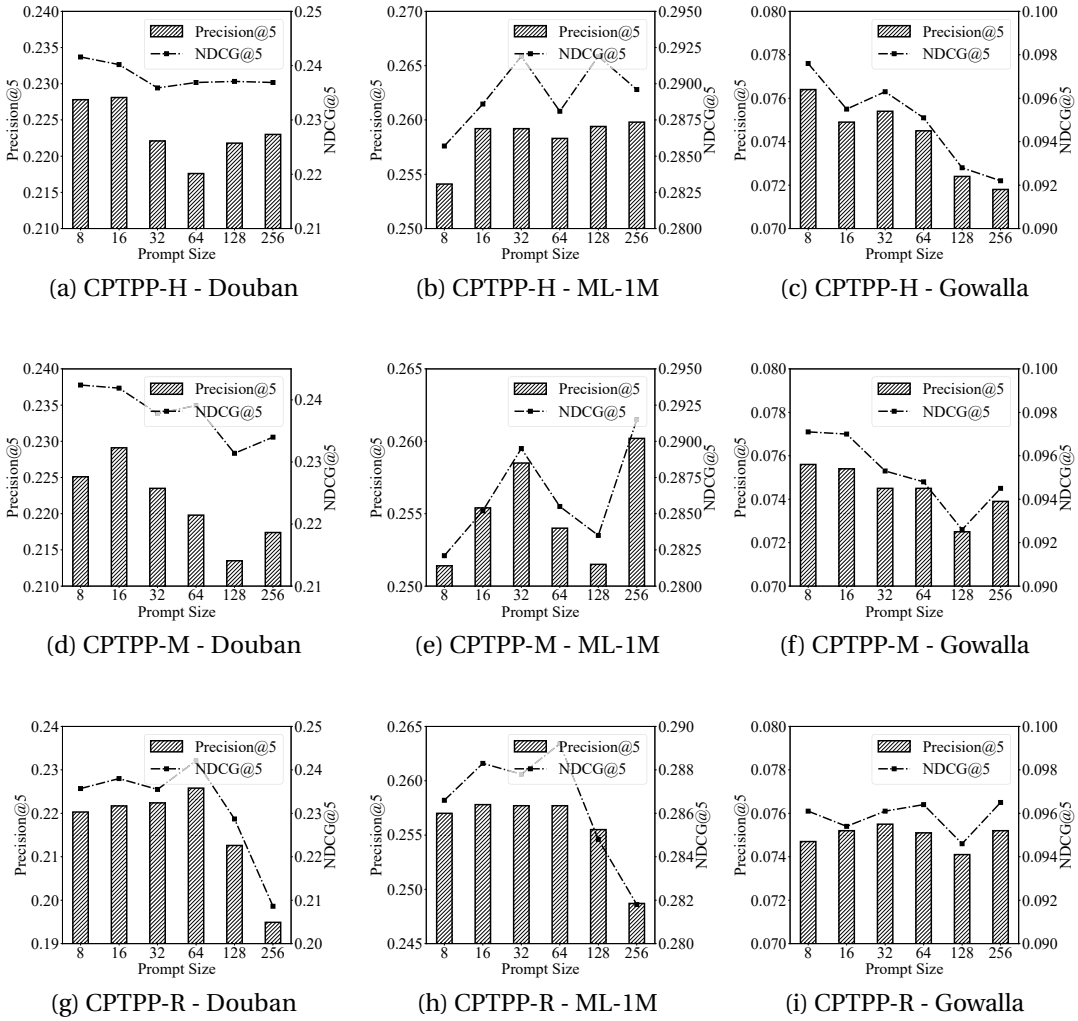
and *NDCG*@5, to demonstrate CPTPP's performance variations with regard to different prompt sizes.

All the experiment results are shown in Figure 6.3. Two insightful findings are listed:

- The first finding is that, in most cases, CPTPP has the best performance when the prompt size is not larger than the dimensionality of user embeddings, *i.e.*, 64. A potential reason is that the prompt is usually less informative than the pre-trained embeddings, so a sizeable prompt dimension would introduce too much noise to disturb the structural semantics contained in the pre-trained user embeddings.

- It is worth noting that a significant performance improvement occurs when prompt size is 256 in several cases, such as CPTPP-M on dataset ML-1M and CPTPP-R on dataset Gowalla. Such outlier performance could be caused by random factors during the overall training process. However, they still fail to significantly outperform the CPTPP model, which has a much smaller prompt size.

Therefore, small prompt size for prompt-tuning is a better option in practice as they achieve



(a) CPTPP-H - Douban (b) CPTPP-H - ML-1M (c) CPTPP-H - Gowalla

(d) CPTPP-M - Douban (e) CPTPP-M - ML-1M (f) CPTPP-M - Gowalla

(g) CPTPP-R - Douban (h) CPTPP-R - ML-1M (i) CPTPP-R - Gowalla

Figure 6.3: The hyperparameter study of CPTPP regarding the size of prompt.

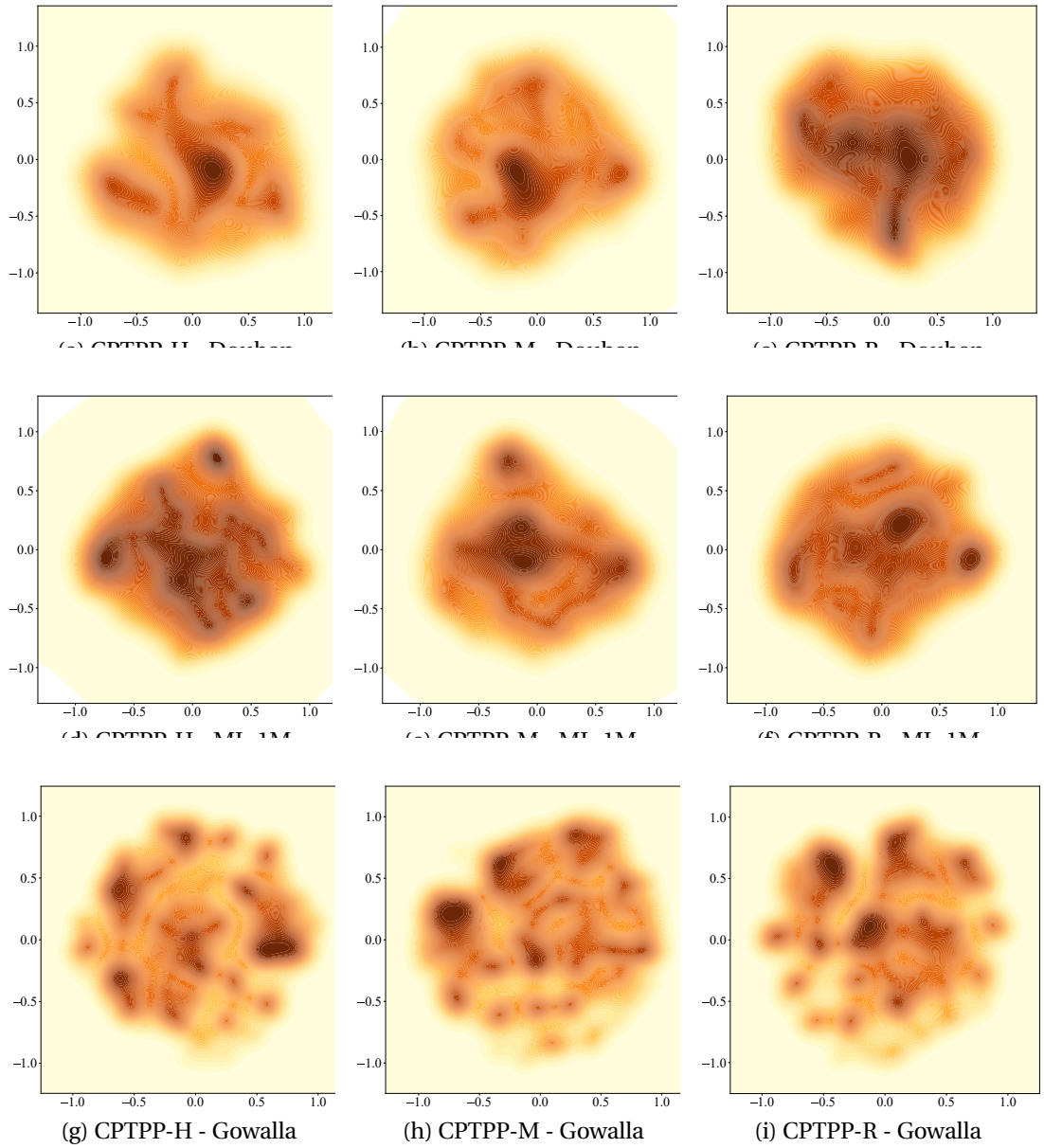
a relatively better recommendation quality and higher efficiency.

6.3.2.3 Ablation Studies

As discussed previously, three strategies are summarized to generate personalized prompts for users, two ablation studies are conducted in this part to illustrate the performance of three variations of the CPTPP method.

The first ablation study is about the overall evaluation of recommendation quality reflected in Table 6.3, whose analysis is listed below.

It is worth noting that (i) CPTPP-M achieves the best performance on dataset Douban. Nevertheless, the performance of CPTPP-M degrades on dataset ML-1M and is the worst



(g) CPTPP-H - Gowalla (h) CPTPP-M - Gowalla (i) CPTPP-R - Gowalla

Figure 6.4: The visualizations of user embeddings generated by CPTPP variations.

case on dataset Gowalla. Considering the number of users reflected in Table 6.1, we find that the performance of CPTPP-M drops as the dataset's number of users increases. So, CPTPP-M performs well if the number of users in the dataset is relatively small. It may be because matrix factorization, as a naive method, cannot fully reveal user preferences in a complex user-item interaction graph with too many user nodes. (ii) CPTPP-R utilizes high-order relationships among users to enrich the generated personalized prompts for users. In such settings, the item information would also be aggregated due to the message-passing mechanism in GNNs. Therefore, it achieves the best performance on the dataset Gowalla, having the most users and the most complex user-user relation among all the datasets. (iii) CPTPP-H has moderate performance. CPTPP-H adopts historical interaction records, formed by trainable item embeddings, to generate personalized prompts. Those trainable elements endow CPTPP-H with a more robust capability to represent user preferences than matrix factorization. It is also reasonable that CPTPP-R outperforms CPTPP-H as CPTPP-H lacks consideration of high-order user relations.

The second one is about the embedding visualizations of different variations of CPTPP. The impacts of different personalized prompts on CPTPP are investigated. The user embeddings produced by all three variations of the proposed CPTPP are visualized as shown in Figure 6.4. It can observe that both CPTPP-H and CPTPP-R have a more uniform distribution, especially on datasets Douban and ML-1M. Such an observation indicates that personalized prompts generated from trainable user profiles can produce user embeddings that have more uniform distributions to demonstrate diverse user preferences better.

6.4 Summary of CPTPP

In this section, an empirical study is conducted to reveal the limitations in current GCL-based recommendation methods. Based on the findings of the empirical study, CPTPP is proposed to adopt a prompt-tuning technique to reform and improve current GCL-based recommendation methods, addressing RQ2.2. To better accommodate prompt learning to graph recommendation scenarios, several graph-oriented user profiles are summarized to generate personalized user prompts to conduct prompt-tuning for downstream recommendation tasks. Comprehensive experiments have shown the effectiveness and superiority of the proposed CPTPP method. The future research directions about prompt-tuning in GCL-based recommendation may be two-fold: how to (i) generate personalized prompts and (ii) integrate prompt-tuning strategy into GCL protocols.

CHAPTER 7

CONCLUSIONS & FUTURE WORKS

This chapter draws conclusions of this thesis and discusses the potential future works of GCL and its applications in recommendations. Specifically:

- Sec. 7.1 summarizes the conclusions and contributions of the research works within this thesis, demonstrating that the research questions are all properly addressed.
- Sec. 7.2 discusses the potential future works of GCL and its applications in RS, focusing on the theoretical aspects of the methodology like improving theoretical understanding and practical aspects of GCL in real-world applications like interpretability.

This chapter encapsulates the significant findings and contributions of the thesis, affirming that the research questions have been comprehensively addressed. It also opens avenues for future exploration, emphasizing both theoretical advancements of GCL methodology and practical implementations of GCL in recommendation systems.

7.1 Conclusions

Focusing on the thorough literature review and the four summarized research questions, four research works are conducted within this thesis to address each of them.

CGC introduces a novel method, which generates high-quality contrasting samples for GCL using a counterfactual mechanism. This learning-based GCL method can adaptively process diverse datasets with varying characteristics that tackles limitations in the current literature and proposing a flexible GCL approach, addressing RQ1.1.

LATEX-GCL addresses the limitation that current GCL methods cannot directly augment non-embedding features such as text. It presents a novel GCL framework, which leverages LLMs to augment text features on the graph, producing high-quality contrasting samples. Three novel and tailored prompts for text feature augmentation are introduced and examined through extensive experiments. This research work offers a promising solution to augment non-embedding features such as text, addressing RQ1.2.

HMG-CR presents a pioneering work by applying GCL to multi-behavior recommendation tasks, constructing contrasting samples in a rule-based manner by introducing the concept of hyper meta-paths to construct hyper meta-graphs. The proposed method outlines the implementation pipeline of GCL in recommendation systems and inspires future research in this domain, addressing RQ2.1.

CPTPP examines GCL in recommendation systems at a higher level, investigating the training paradigm of GCL for recommendations. An empirical study demonstrates the disadvantages of the current end-to-end training paradigm. Consequently, a novel framework for GCL in recommendation systems is proposed, which utilizes prompt learning to introduce an effective 'pre-training and prompt-tuning' paradigm, addressing RQ2.2.

In conclusion, both CGC and LATEX-GCL significantly advance the methodology of current GCL methods by focusing on graph augmentation strategies, thereby broadening the scope of GCL's applicability. Regarding GCL's applications in recommendation systems, HMG-CR introduces a novel concept for constructing contrasting samples in a rule-based manner, representing pioneering work in the literature. Additionally, CPTPP examines the training paradigm of GCL for recommendation tasks, proposing a novel 'pre-training and prompt-tuning' framework that further reveals GCL's potential in RS.

7.2 Future Works

Despite the promising progress in GCL-related research, several challenges remain that hinder the full realization of GCL methods' potential. The following contents outline and discuss several future research directions for further exploration.

Interpretability. Understanding the rationale and intuition behind graph learning methods is crucial for improving these methods and applying them to real-world problems. However, most research on interpretability in graph learning focuses on supervised learning scenarios [16, 59, 19], leaving interpretability in GCL largely underexplored. There is an urgent need to develop GCL methods with high interpretability, which would enable their application in critical industries such as finance [3] and healthcare [7].

Theoretical Analysis. Graph augmentation is a crucial component of GCL procedures. However, most graph augmentation strategies are designed based on intuitive understanding of specific scenarios rather than solid theoretical foundations [132, 123, 35, 94]. As a result, researchers may find it challenging to evaluate the quality of their designed augmentation strategies without empirical studies. Therefore, it is essential to explore and summarize the common properties in graph augmentation strategy design to guide the construction of contrasting samples in future research.

Real-world Applications. While this thesis discusses GCL's applications in recommendation systems, there are numerous other real-world scenarios to explore, such as finance [3], healthcare [7], and AI for science [14, 10]. The key to implementing GCL in these applications lies in designing augmentation strategies that effectively integrate domain-specific knowledge. Each application scenario inherently possesses specific priors that are crucial for models to understand the context. Relying solely on general graph augmentation strategies [132, 123, 35, 94] may prevent GCL methods from accurately capturing domain-specific semantics, leading to suboptimal performance. Therefore, it is essential to thoroughly investigate methods for integrating domain knowledge into GCL.

7.3 Chapter Summary

This chapter concludes the research works within this thesis by summarizing the contributions, findings, and conclusions of each one. Then, three future research directions are discussed, including interpretability, theoretical analysis, and real-world applications, to shed light on GCL-related research domains.

BIBLIOGRAPHY

[1] ADHIKARI, B., ZHANG, Y., RAMAKRISHNAN, N., AND PRAKASH, B. A.
Sub2vec: Feature learning for subgraphs.
In *Advances in Knowledge Discovery and Data Mining - 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, June 3-6, 2018, Proceedings, Part II* (2018), D. Q. Phung, V. S. Tseng, G. I. Webb, B. Ho, M. Ganji, and L. Rashidi, Eds., vol. 10938 of *Lecture Notes in Computer Science*, Springer, pp. 170–182.

[2] ANAND, A.
Contrastive self-supervised learning, 2020.
`https://ankeshanand.com/blog/2020/01/26/contrative-self-supervised-learning.html`.

[3] ANANDAKRISHNAN, A., KUMAR, S., STATNIKOV, A. R., FARUQUIE, T. A., AND XU, D.
Anomaly detection in finance: Editors' introduction.
In *Proceedings of the KDD 2017 Workshop on Anomaly Detection in Finance, ADF@KDD 2017, Halifax, Nova Scotia, Canada, August 14, 2017* (2017), A. Anandakrishnan, S. Kumar, A. R. Statnikov, T. A. Faruquie, and D. Xu, Eds., vol. 71 of *Proceedings of Machine Learning Research*, PMLR, pp. 1–7.

[4] BELGHAZI, M. I., BARATIN, A., RAJESWAR, S., OZAIR, S., BENGIO, Y., HJELM, R. D., AND COURVILLE, A. C.
Mutual information neural estimation.
In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018* (2018), J. G. Dy and A. Krause, Eds., vol. 80 of *Proceedings of Machine Learning Research*, PMLR, pp. 530–539.

[5] BIELAK, P., KAJDANOWICZ, T., AND CHAWLA, N. V.
Graph barlow twins: A self-supervised representation learning framework for graphs.
Knowl. Based Syst. 256 (2022), 109631.

[6] BONACICH, P.

Power and centrality: A family of measures.
American journal of sociology 92, 5 (1987), 1170–1182.

[7] BONGINI, P., BIANCHINI, M., AND SCARSELLI, F.
Molecular generative graph neural networks for drug discovery.
Neurocomputing 450 (2021), 242–252.

[8] BORGWARDT, K. M., AND KRIEGEL, H.
Shortest-path kernels on graphs.
In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM 2005), 27-30 November 2005, Houston, Texas, USA* (2005), IEEE Computer Society, pp. 74–81.

[9] BORGWARDT, K. M., ONG, C. S., SCHÖNAUER, S., VISHWANATHAN, S. V. N., SMOLA, A. J., AND KRIEGEL, H.
Protein function prediction via graph kernels.
In *Proceedings Thirteenth International Conference on Intelligent Systems for Molecular Biology 2005, Detroit, MI, USA, 25-29 June 2005* (2005), pp. 47–56.

[10] BORGWARDT, K. M., ONG, C. S., SCHÖNAUER, S., VISHWANATHAN, S. V. N., SMOLA, A. J., AND KRIEGEL, H.
Protein function prediction via graph kernels.
In *Proceedings Thirteenth International Conference on Intelligent Systems for Molecular Biology 2005, Detroit, MI, USA, 25-29 June 2005* (2005), pp. 47–56.

[11] BOSE, A. J., JAIN, A., MOLINO, P., AND HAMILTON, W. L.
Meta-graph: Few shot link prediction via meta learning.
CoRR abs/1912.09867 (2019).

[12] BROWN, T., MANN, B., RYDER, N., SUBBIAH, M., KAPLAN, J. D., DHARIWAL, P., NEELAKANTAN, A., SHYAM, P., SASTRY, G., ASKELL, A., AGARWAL, S., HERBERT-VOSS, A., KRUEGER, G., HENIGHAN, T., CHILD, R., RAMESH, A., ZIEGLER, D., WU, J., WINTER, C., HESSE, C., CHEN, M., SIGLER, E., LITWIN, M., GRAY, S., CHESS, B., CLARK, J., BERNER, C., MCCANDLISH, S., RADFORD, A., SUTSKEVER, I., AND AMODEI, D.
Language models are few-shot learners.
In *Advances in Neural Information Processing Systems* (2020), vol. 33, Curran Associates, Inc., pp. 1877–1901.

[13] BU, J., TAN, S., CHEN, C., WANG, C., WU, H., ZHANG, L., AND HE, X.

Music recommendation by unified hypergraph: combining social media information and music content.
In *Proceedings of the 18th International Conference on Multimedia 2010, Firenze, Italy, October 25-29, 2010* (2010), A. D. Bimbo, S. Chang, and A. W. M. Smeulders, Eds., ACM, pp. 391–400.

[14] CAI, H., ZHANG, H., ZHAO, D., WU, J., AND WANG, L.
FP-GNN: a versatile deep learning architecture for enhanced molecular property prediction.
Briefings Bioinform. 23, 6 (2022).

[15] CHEN, C., ZHANG, M., ZHANG, Y., MA, W., LIU, Y., AND MA, S.
Efficient heterogeneous collaborative filtering without negative sampling for recommendation.
In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020* (2020), AAAI Press, pp. 19–26.

[16] CHEN, H., LI, Y., SUN, X., XU, G., AND YIN, H.
Temporal meta-path guided explainable recommendation.
In *WSDM '21, The Fourteenth ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, March 8-12, 2021* (2021), L. Lewin-Eytan, D. Carmel, E. Yom-Tov, E. Agichtein, and E. Gabrilovich, Eds., ACM, pp. 1056–1064.

[17] CHEN, Z., MAO, H., LI, H., JIN, W., WEN, H., WEI, X., WANG, S., YIN, D., FAN, W., LIU, H., AND TANG, J.
Exploring the potential of large language models (llms)in learning on graphs.
SIGKDD Explor. 25, 2 (2023), 42–61.

[18] CLARK, K., LUONG, M., LE, Q. V., AND MANNING, C. D.
ELECTRA: pre-training text encoders as discriminators rather than generators.
In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020* (2020), OpenReview.net.

[19] CUI, Z., CHEN, H., CUI, L., LIU, S., LIU, X., XU, G., AND YIN, H.
Reinforced kgs reasoning for explainable sequential recommendation.
World Wide Web (06 2021).

[20] DEVLIN, J., CHANG, M., LEE, K., AND TOUTANOVA, K.
BERT: pre-training of deep bidirectional transformers for language understanding.

In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)* (2019), Association for Computational Linguistics, pp. 4171–4186.

[21] DONG, Y., CHAWLA, N. V., AND SWAMI, A.
metapath2vec: Scalable representation learning for heterogeneous networks.
In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017* (2017), ACM, pp. 135–144.

[22] DU, J., ZHANG, S., WU, G., MOURA, J. M. F., AND KAR, S.
Topology adaptive graph convolutional networks.
CoRR abs/1710.10370 (2017).

[23] EL MRABET, M. A., EL MAKKAOUI, K., AND FAIZE, A.
Supervised machine learning: A survey.
In *2021 4th International Conference on Advanced Communication Technologies and Networking (CommNet)* (2021), pp. 1–10.

[24] FAN, W., LIU, X., JIN, W., ZHAO, X., TANG, J., AND LI, Q.
Graph trend filtering networks for recommendation.
In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022* (2022), ACM, pp. 112–121.

[25] FENTON, N. E., NEIL, M., AND CONSTANTINOU, A. C.
The book of why: The new science of cause and effect, judea pearl, dana mackenzie. basic books (2018).
Artif. Intell. 284 (2020), 103286.

[26] GAO, C., HE, X., GAN, D., CHEN, X., FENG, F., LI, Y., CHUA, T., YAO, L., SONG, Y., AND JIN, D.
Learning to recommend with multiple cascading behaviors.
IEEE Trans. Knowl. Data Eng. 33, 6 (2021), 2588–2601.

[27] GAO, T., FISCH, A., AND CHEN, D.
Making pre-trained language models better few-shot learners.
In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Process-*

ing (Volume 1: Long Papers) (Online, Aug. 2021), Association for Computational Linguistics, pp. 3816–3830.

[28] GENG, S., LIU, S., FU, Z., GE, Y., AND ZHANG, Y.
Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5).
In *Proceedings of the 16th ACM Conference on Recommender Systems* (New York, NY, USA, 2022), RecSys '22, Association for Computing Machinery, p. 299–315.

[29] GOODFELLOW, I. J., POUGET-ABADIE, J., MIRZA, M., XU, B., WARDE-FARLEY, D., OZAIR, S., COURVILLE, A. C., AND BENGIO, Y.
Generative adversarial networks.
Commun. ACM 63, 11 (2020), 139–144.

[30] GU, Y., HAN, X., LIU, Z., AND HUANG, M.
PPT: Pre-trained prompt tuning for few-shot learning.
In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Dublin, Ireland, May 2022), Association for Computational Linguistics, pp. 8410–8423.

[31] GUI, J., CHEN, T., ZHANG, J., CAO, Q., SUN, Z., LUO, H., AND TAO, D.
A survey on self-supervised learning: Algorithms, applications, and future trends.
IEEE Transactions on Pattern Analysis and Machine Intelligence (2024), 1–20.

[32] HAMILTON, W. L., YING, Z., AND LESKOVEC, J.
Inductive representation learning on large graphs.
In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA* (2017), I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., pp. 1024–1034.

[33] HARPER, F. M., AND KONSTAN, J. A.
The movielens datasets: History and context.
ACM Trans. Interact. Intell. Syst. 5, 4 (dec 2015).

[34] HARRIS, Z. S.
Distributional structure.
WORD 10, 2-3 (1954), 146–162.

[35] HASSANI, K., AND AHMADI, A. H. K.
Contrastive multi-view representation learning on graphs.

In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event* (2020), vol. 119 of *Proceedings of Machine Learning Research*, PMLR, pp. 4116–4126.

[36] HE, K., FAN, H., WU, Y., XIE, S., AND GIRSHICK, R. B.
Momentum contrast for unsupervised visual representation learning.
In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020* (2020), Computer Vision Foundation / IEEE, pp. 9726–9735.

[37] HE, P., LIU, X., GAO, J., AND CHEN, W.
Deberta: decoding-enhanced bert with disentangled attention.
In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021* (2021), OpenReview.net.

[38] HE, R., AND MCAULEY, J. J.
Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering.
In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016* (2016), ACM, pp. 507–517.

[39] HE, X., BRESSON, X., LAURENT, T., AND HOOI, B.
Explanations as features: Llm-based features for text-attributed graphs.
CoRR abs/2305.19523 (2023).

[40] HE, X., DENG, K., WANG, X., LI, Y., ZHANG, Y., AND WANG, M.
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* (New York, NY, USA, 2020), SIGIR '20, Association for Computing Machinery, p. 639–648.

[41] HE, X., LIAO, L., ZHANG, H., NIE, L., HU, X., AND CHUA, T.
Neural collaborative filtering.
In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017* (2017), R. Barrett, R. Cummings, E. Agichtein, and E. Gabrilovich, Eds., ACM, pp. 173–182.

[42] HE, X., LIAO, L., ZHANG, H., NIE, L., HU, X., AND CHUA, T.-S.
Neural collaborative filtering.

In *Proceedings of the 26th International Conference on World Wide Web* (Republic and Canton of Geneva, CHE, 2017), WWW '17, International World Wide Web Conferences Steering Committee, p. 173–182.

[43] HERNÁNDEZ, J. M., AND VAN MIEGHEM, P.
Classification of graph metrics.
Delft University of Technology: Mekelweg, The Netherlands 1 (2011).

[44] INUWA-DUTSE, I., LIPTROTT, M., AND KORKONTZELOS, I.
A multilevel clustering technique for community detection.
Neurocomputing 441 (2021), 64–78.

[45] JAIN, N., COYLE, B., KASHEFI, E., AND KUMAR, N.
Graph neural network initialisation of quantum approximate optimisation.
Quantum 6 (2022), 861.

[46] JIANG, H., SONG, Y., WANG, C., ZHANG, M., AND SUN, Y.
Semi-supervised learning over heterogeneous information networks by ensemble of meta-graph guided random walks.
In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017* (2017), C. Sierra, Ed., ijcai.org, pp. 1944–1950.

[47] JIANG, Z., XU, F. F., ARAKI, J., AND NEUBIG, G.
How Can We Know What Language Models Know?
Transactions of the Association for Computational Linguistics 8 (07 2020), 423–438.

[48] JIN, B., LIU, G., HAN, C., JIANG, M., JI, H., AND HAN, J.
Large language models on graphs: A comprehensive survey.
CoRR abs/2312.02783 (2023).

[49] JIN, W., LIU, X., ZHAO, X., MA, Y., SHAH, N., AND TANG, J.
Automated self-supervised learning for graphs.
In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022* (2022), OpenReview.net.

[50] JU, W., WANG, Y., QIN, Y., MAO, Z., XIAO, Z., LUO, J., YANG, J., GU, Y., WANG, D., LONG, Q., YI, S., LUO, X., AND ZHANG, M.
Towards graph contrastive learning: A survey and beyond.
CoRR abs/2405.11868 (2024).

[51] KIM, S., CHEN, J., CHENG, T., GINDULYTE, A., HE, J., HE, S., LI, Q., SHOEMAKER, B. A., THIESSEN, P. A., YU, B., ET AL.
Pubchem 2019 update: improved access to chemical data.
Nucleic acids research 47, D1 (2019), D1102–D1109.

[52] KINGMA, D. P., AND WELLING, M.
Auto-encoding variational bayes.
In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings* (2014), Y. Bengio and Y. LeCun, Eds.

[53] KIPF, T. N., AND WELLING, M.
Semi-supervised classification with graph convolutional networks.
In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings* (2017), OpenReview.net.

[54] KLICPERA, J., WEISSENBERGER, S., AND GÜNNEMANN, S.
Diffusion improves graph learning.
In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada* (2019), H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., pp. 13333–13345.

[55] KOREN, Y., BELL, R., AND VOLINSKY, C.
Matrix factorization techniques for recommender systems.
Computer 42, 8 (2009), 30–37.

[56] KULLBACK, S., AND LEIBLER, R. A.
On Information and Sufficiency.
The Annals of Mathematical Statistics 22, 1 (1951), 79 – 86.

[57] LA CRUZ CABRERA, O. D., MATAR, M., AND REICHEL, L.
Edge importance in a network via line graphs and the matrix exponential.
Numer. Algorithms 83, 2 (2020), 807–832.

[58] LEE, D., KANG, S., JU, H., PARK, C., AND YU, H.
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* (New York, NY, USA, 2021), SIGIR '21, Association for Computing Machinery, p. 317–326.

[59] LI, L., ZHANG, Y., AND CHEN, L.
Personalized prompt learning for explainable recommendation.
ACM Trans. Inf. Syst. 41, 4 (mar 2023).

[60] LI, X., CAO, C. C., SHI, Y., BAI, W., GAO, H., QIU, L., WANG, C., GAO, Y., ZHANG, S., XUE, X., AND CHEN, L.
A survey of data-driven and knowledge-aware explainable AI.
IEEE Trans. Knowl. Data Eng. 34, 1 (2022), 29–49.

[61] LI, X. L., AND LIANG, P.
Prefix-tuning: Optimizing continuous prompts for generation.
In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (Online, Aug. 2021), Association for Computational Linguistics, pp. 4582–4597.

[62] LI, Y., CHEN, H., SUN, X., SUN, Z., LI, L., CUI, L., YU, P. S., AND XU, G.
Hyperbolic hypergraphs for sequential recommendation.
CoRR abs/2108.08134 (2021).

[63] LIANG, D., CHARLIN, L., MCINERNEY, J., AND BLEI, D. M.
Modeling user exposure in recommendation.
In *Proceedings of the 25th International Conference on World Wide Web* (Republic and Canton of Geneva, CHE, 2016), WWW '16, International World Wide Web Conferences Steering Committee, p. 951–961.

[64] LIN, Z., TIAN, C., HOU, Y., AND ZHAO, W. X.
Improving graph collaborative filtering with neighborhood-enriched contrastive learning.
In *Proceedings of the ACM Web Conference 2022* (New York, NY, USA, 2022), WWW '22, Association for Computing Machinery, p. 2320–2329.

[65] LUO, Y., HUANG, Z., CHEN, H., YANG, Y., YIN, H., AND BAKTASHMOTLAGH, M.
Interpretable signed link prediction with signed infomax hyperbolic graph.
IEEE Trans. Knowl. Data Eng. 35, 4 (2023), 3991–4002.

[66] MA, R., AND LUO, T.
Pi1m: a benchmark database for polymer informatics.
Journal of Chemical Information and Modeling 60, 10 (2020), 4684–4690.

[67] MCAULEY, J. J., TARGETT, C., SHI, Q., AND VAN DEN HENGEL, A.

Image-based recommendations on styles and substitutes.
In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015* (2015), ACM, pp. 43–52.

[68] MIKOLOV, T., CHEN, K., CORRADO, G., AND DEAN, J.
Efficient estimation of word representations in vector space.
In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings* (2013), Y. Bengio and Y. LeCun, Eds.

[69] MIKOLOV, T., SUTSKEVER, I., CHEN, K., CORRADO, G. S., AND DEAN, J.
Distributed representations of words and phrases and their compositionality.
In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States* (2013), C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger, Eds., pp. 3111–3119.

[70] MORRIS, C., KRIEGE, N. M., BAUSE, F., KERSTING, K., MUTZEL, P., AND NEUMANN, M.
Tudataset: A collection of benchmark datasets for learning with graphs.
CoRR abs/2007.08663 (2020).

[71] MORRIS, C., KRIEGE, N. M., KERSTING, K., AND MUTZEL, P.
Faster kernels for graphs with continuous attributes via hashing.
In *IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain* (2016), F. Bonchi, J. Domingo-Ferrer, R. Baeza-Yates, Z. Zhou, and X. Wu, Eds., IEEE Computer Society, pp. 1095–1100.

[72] NARAYANAN, A., CHANDRAMOHAN, M., VENKATESAN, R., CHEN, L., LIU, Y., AND JAISWAL, S.
graph2vec: Learning distributed representations of graphs.
CoRR abs/1707.05005 (2017).

[73] ORSINI, F., FRASCONI, P., AND RAEDT, L. D.
Graph invariant kernels.
In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015* (2015), Q. Yang and M. J. Wooldridge, Eds., AAAI Press, pp. 3756–3762.

[74] PAGE, L., BRIN, S., MOTWANI, R., AND WINOGRAD, T.

The pagerank citation ranking: Bringing order to the web.
Technical Report 1999-66, Stanford InfoLab, November 1999.
Previous number = SIDL-WP-1999-0120.

[75] PETRONI, F., ROCKTÄSCHEL, T., RIEDEL, S., LEWIS, P. S. H., BAKHTIN, A., WU, Y., AND MILLER, A. H.
Language models as knowledge bases?
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 2019, Hong Kong, China, November 3-7, 2019* (2019), Association for Computational Linguistics, pp. 2463–2473.

[76] PFADLER, A., ZHAO, H., WANG, J., WANG, L., HUANG, P., AND LEE, D. L.
Billion-scale recommendation with heterogeneous side information at taobao.
In *36th IEEE International Conference on Data Engineering, ICDE 2020, Dallas, TX, USA, April 20-24, 2020* (2020), IEEE, pp. 1667–1676.

[77] QIN, G., AND EISNER, J.
Learning how to ask: Querying LMs with mixtures of soft prompts.
In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (Online, June 2021), Association for Computational Linguistics, pp. 5203–5212.

[78] QIU, J., CHEN, Q., DONG, Y., ZHANG, J., YANG, H., DING, M., WANG, K., AND TANG, J.
GCC: graph contrastive coding for graph neural network pre-training.
In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020* (2020), ACM, pp. 1150–1160.

[79] RADFORD, A., WU, J., CHILD, R., LUAN, D., AMODEI, D., SUTSKEVER, I., ET AL.
Language models are unsupervised multitask learners.

[80] RAO, R., BHATTACHARYA, N., THOMAS, N., DUAN, Y., CHEN, X., CANNY, J., ABBEEL, P., AND SONG, Y. S.
Evaluating protein transfer learning with TAPE.
Curran Associates Inc., Red Hook, NY, USA, 2019.

[81] REN, X., WEI, W., XIA, L., AND HUANG, C.
A comprehensive survey on self-supervised learning for recommendation.
CoRR abs/2404.03354 (2024).

[82] RENDLE, S., FREUDENTHALER, C., GANTNER, Z., AND SCHMIDT-THIEME, L.
Bpr: Bayesian personalized ranking from implicit feedback.
UAI '09, AUAI Press, p. 452–461.

[83] RIESEN, K., AND BUNKE, H.
IAM graph database repository for graph based pattern recognition and machine learning.
In *Structural, Syntactic, and Statistical Pattern Recognition, Joint IAPR International Workshop, SSPR & SPR 2008, Orlando, USA, December 4-6, 2008. Proceedings* (2008), N. da Vitoria Lobo, T. Kasparis, F. Roli, J. T. Kwok, M. Georgiopoulos, G. C. Anagnostopoulos, and M. Loog, Eds., vol. 5342 of *Lecture Notes in Computer Science*, Springer, pp. 287–297.

[84] RIESEN, K., AND BUNKE, H.
IAM graph database repository for graph based pattern recognition and machine learning.
In *Structural, Syntactic, and Statistical Pattern Recognition, Joint IAPR International Workshop, SSPR & SPR 2008, Orlando, USA, December 4-6, 2008. Proceedings* (2008), vol. 5342 of *Lecture Notes in Computer Science*, Springer, pp. 287–297.

[85] ROBINSON, J. D., CHUANG, C., SRA, S., AND JEGELKA, S.
Contrastive learning with hard negative samples.
In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021* (2021), OpenReview.net.

[86] SCHLICHTKRULL, M. S., KIPF, T. N., BLOEM, P., VAN DEN BERG, R., TITOV, I., AND WELLING, M.
Modeling relational data with graph convolutional networks.
In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings* (2018), A. Gangemi, R. Navigli, M. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai, and M. Alam, Eds., vol. 10843 of *Lecture Notes in Computer Science*, Springer, pp. 593–607.

[87] SCHOMBURG, I., CHANG, A., EBELING, C., GREMSE, M., HELDT, C., HUHN, G., AND SCHOMBURG, D.
Brenda, the enzyme database: updates and major new developments.
Nucleic Acids Res. 32, Database-Issue (2004), 431–433.

[88] SHERVASHIDZE, N., SCHWEITZER, P., VAN LEEUWEN, E. J., MEHLHORN, K., AND BORGWARDT, K. M.

Weisfeiler-lehman graph kernels.
J. Mach. Learn. Res. 12 (2011), 2539–2561.

[89] SHERVASHIDZE, N., VISHWANATHAN, S. V. N., PETRI, T., MEHLHORN, K., AND BORGWARDT, K. M.
Efficient graphlet kernels for large graph comparison.
In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, AISTATS 2009, Clearwater Beach, Florida, USA, April 16-18, 2009* (2009), D. A. V. Dyk and M. Welling, Eds., vol. 5 of *JMLR Proceedings*, JMLR.org, pp. 488–495.

[90] SHI, C., HU, B., ZHAO, W. X., AND YU, P. S.
Heterogeneous information network embedding for recommendation.
IEEE Trans. Knowl. Data Eng. 31, 2 (2019), 357–370.

[91] SHI, C., KONG, X., HUANG, Y., YU, P. S., AND WU, B.
Hetesim: A general framework for relevance measure in heterogeneous networks.
IEEE Trans. Knowl. Data Eng. 26, 10 (2014), 2479–2492.

[92] SHIN, T., RAZEGHI, Y., LOGAN IV, R. L., WALLACE, E., AND SINGH, S.
AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts.
In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Online, Nov. 2020), Association for Computational Linguistics, pp. 4222–4235.

[93] SRA, S., AND DHILLON, I.
Generalized nonnegative matrix approximations with bregman divergences.
In *Advances in Neural Information Processing Systems* (2005), Y. Weiss, B. Schölkopf, and J. Platt, Eds., vol. 18, MIT Press.

[94] SUN, F., HOFFMANN, J., VERMA, V., AND TANG, J.
Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization.
In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020* (2020), OpenReview.net.

[95] SUN, F., LIU, J., WU, J., PEI, C., LIN, X., OU, W., AND JIANG, P.
Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer.

In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019* (2019), W. Zhu, D. Tao, X. Cheng, P. Cui, E. A. Rundensteiner, D. Carmel, Q. He, and J. X. Yu, Eds., ACM, pp. 1441–1450.

[96] SUN, X., YIN, H., LIU, B., CHEN, H., CAO, J., SHAO, Y., AND HUNG, N. Q. V.
Heterogeneous hypergraph embedding for graph classification.
In *WSDM '21, The Fourteenth ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, March 8-12, 2021* (2021), L. Lewin-Eytan, D. Carmel, E. Yom-Tov, E. Agichtein, and E. Gabrilovich, Eds., ACM, pp. 725–733.

[97] SUN, X., YIN, H., LIU, B., CHEN, H., MENG, Q., HAN, W., AND CAO, J.
Multi-level hyperedge distillation for social linking prediction on sparsely observed networks.
In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021* (2021), J. Leskovec, M. Grobelnik, M. Najork, J. Tang, and L. Zia, Eds., ACM / IW3C2, pp. 2934–2945.

[98] SUN, Y., AND HAN, J.
Mining Heterogeneous Information Networks: Principles and Methodologies.
Synthesis Lectures on Data Mining and Knowledge Discovery. Morgan & Claypool Publishers, 2012.

[99] THAKOOR, S., TALLEC, C., AZAR, M. G., AZABOU, M., DYER, E. L., MUNOS, R., VELICKOVIC, P., AND VALKO, M.
Large-scale representation learning on graphs via bootstrapping.
In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022* (2022), OpenReview.net.

[100] THANOU, D., DONG, X., KRESSNER, D., AND FROSSARD, P.
Learning heat diffusion graphs.
IEEE Trans. Signal Inf. Process. over Networks 3, 3 (2017), 484–499.

[101] TSITSULIN, A., PALOWITCH, J., PEROZZI, B., AND MÜLLER, E.
Graph clustering with graph neural networks.
J. Mach. Learn. Res. 24 (2023), 127:1–127:21.

[102] VAN DEN OORD, A., LI, Y., AND VINYALS, O.
Representation learning with contrastive predictive coding.
CoRR abs/1807.03748 (2018).

[103] VELICKOVIC, P., CUCURULL, G., CASANOVA, A., ROMERO, A., LIÒ, P., AND BENGIO, Y.
Graph attention networks.
In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings* (2018), OpenReview.net.

[104] VELICKOVIC, P., FEDUS, W., HAMILTON, W. L., LIÒ, P., BENGIO, Y., AND HJELM, R. D.
Deep graph infomax.
In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019* (2019), OpenReview.net.

[105] VISHWANATHAN, S. V. N., SCHRAUDOLPH, N. N., KONDOR, R., AND BORGWARDT, K. M.
Graph kernels.
J. Mach. Learn. Res. 11 (2010), 1201–1242.

[106] WACHTER, S., MITTELSTADT, B., AND RUSSELL, C.
Counterfactual explanations without opening the black box: automated decisions and the gdpr.
Harvard Journal of Law and Technology 31, 2 (2018), 841–887.

[107] WANG, C., LI, X., HAN, H., WANG, S., WANG, L., CAO, C. C., AND CHEN, L.
Counterfactual explanations in explainable AI: A tutorial.
In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021* (2021), ACM, pp. 4080–4081.

[108] WANG, J., DING, K., HONG, L., LIU, H., AND CAVERLEE, J.
Next-item recommendation with sequential hypergraphs.
In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020* (2020), J. Huang, Y. Chang, X. Cheng, J. Kamps, V. Murdock, J. Wen, and Y. Liu, Eds., ACM, pp. 1101–1110.

[109] WANG, X., HE, X., WANG, M., FENG, F., AND CHUA, T.-S.
Neural graph collaborative filtering.
In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2019), SIGIR'19, Association for Computing Machinery, p. 165–174.

[110] WANG, X., WANG, S., LIANG, X., ZHAO, D., HUANG, J., XU, X., DAI, B., AND MIAO, Q.
Deep reinforcement learning: A survey.

IEEE Transactions on Neural Networks and Learning Systems 35, 4 (2024), 5064–5078.

[111] WANG, Y., ZHAO, X., CHEN, B., LIU, Q., GUO, H., LIU, H., WANG, Y., ZHANG, R., AND TANG, R.
Plate: A prompt-enhanced paradigm for multi-scenario recommendations.
In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2023), SIGIR '23, Association for Computing Machinery, p. 1498–1507.

[112] WU, F., JR., A. H. S., ZHANG, T., FIFTY, C., YU, T., AND WEINBERGER, K. Q.
Simplifying graph convolutional networks.
In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA* (2019), K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97 of *Proceedings of Machine Learning Research*, PMLR, pp. 6861–6871.

[113] WU, J., WANG, X., FENG, F., HE, X., CHEN, L., LIAN, J., AND XIE, X.
Self-supervised graph learning for recommendation.
In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2021), SIGIR '21, Association for Computing Machinery, p. 726–735.

[114] WU, Y., XIE, R., ZHU, Y., ZHUANG, F., XIANG, A., ZHANG, X., LIN, L., AND HE, Q.
Selective fairness in recommendation via prompts.
In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2022), SIGIR '22, Association for Computing Machinery, p. 2657–2662.

[115] WU, Y., XIE, R., ZHU, Y., ZHUANG, F., ZHANG, X., LIN, L., AND HE, Q.
Personalized prompt for sequential recommendation.
IEEE Trans. Knowl. Data Eng. 36, 7 (2024), 3376–3389.

[116] WU, Z., XIONG, Y., YU, S. X., AND LIN, D.
Unsupervised feature learning via non-parametric instance discrimination.
In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018* (2018), Computer Vision Foundation / IEEE Computer Society, pp. 3733–3742.

[117] XIA, J., WU, L., CHEN, J., WANG, G., AND LI, S. Z.
Debiased graph contrastive learning.
CoRR abs/2110.02027 (2021).

[118] XU, K., HU, W., LESKOVEC, J., AND JEGELKA, S.
How powerful are graph neural networks?
In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019* (2019), OpenReview.net.

[119] YAN, H., LI, C., LONG, R., YAN, C., ZHAO, J., ZHUANG, W., YIN, J., ZHANG, P., HAN, W., SUN, H., DENG, W., ZHANG, Q., SUN, L., XIE, X., AND WANG, S.
A comprehensive study on text-attributed graphs: Benchmarking and rethinking.
In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023* (2023).

[120] YANG, C., XIAO, Y., ZHANG, Y., SUN, Y., AND HAN, J.
Heterogeneous network representation learning: Survey, benchmark, evaluation, and beyond.
CoRR abs/2004.00216 (2020).

[121] YANG, H., CHEN, H., LI, L., YU, P. S., AND XU, G.
Hyper meta-path contrastive learning for multi-behavior recommendation.
In *IEEE International Conference on Data Mining, ICDM 2021, Auckland, New Zealand, December 7-10, 2021* (2021), IEEE, pp. 787–796.

[122] YANG, H., CHEN, H., PAN, S., LI, L., YU, P. S., AND XU, G.
Dual space graph contrastive learning.
In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022* (2022), ACM, pp. 1238–1247.

[123] YANG, H., CHEN, H., ZHANG, S., SUN, X., LI, Q., ZHAO, X., AND XU, G.
Generating counterfactual hard negative samples for graph contrastive learning.
In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023* (2023), ACM, pp. 621–629.

[124] YANG, H., WANG, Y., ZHAO, X., CHEN, H., YIN, H., LI, Q., AND XU, G.
Multi-level graph knowledge contrastive learning.
IEEE Transactions on Knowledge and Data Engineering (2024), 1–14.

[125] YANG, H., ZHAO, X., HUANG, S., LI, Q., AND XU, G.
Latex-gcl: Large language models (llms)-based data augmentation for text-attributed graph contrastive learning, 2024.

[126] YANG, H., ZHAO, X., LI, M., CHEN, H., AND XU, G.

Mitigating the performance sacrifice in dp-satisfied federated settings through graph contrastive learning.
Inf. Sci. 648 (2023), 119552.

[127] YANG, H., ZHAO, X., LI, Y., CHEN, H., AND XU, G.
An empirical study towards prompt-tuning for graph contrastive pre-training in recommendations.
In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023* (2023).

[128] YANG, J., LIU, Z., XIAO, S., LI, C., LIAN, D., AGRAWAL, S., SINGH, A., SUN, G., AND XIE, X.
Graphformers: Gnn-nested transformers for representation learning on textual graph.
In *Advances in Neural Information Processing Systems* (2021), vol. 34, Curran Associates, Inc., pp. 28798–28810.

[129] YANG, Y., HUANG, C., XIA, L., AND LI, C.
Knowledge graph contrastive learning for recommendation.
In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2022), SIGIR '22, Association for Computing Machinery, p. 1434–1443.

[130] YIN, H., CHEN, H., SUN, X., WANG, H., WANG, Y., AND NGUYEN, Q. V. H.
SPTF: A scalable probabilistic tensor factorization model for semantic-aware behavior prediction.
In *2017 IEEE International Conference on Data Mining, ICDM 2017, New Orleans, LA, USA, November 18-21, 2017* (2017), V. Raghavan, S. Aluru, G. Karypis, L. Miele, and X. Wu, Eds., IEEE Computer Society, pp. 585–594.

[131] YOU, Y., CHEN, T., SHEN, Y., AND WANG, Z.
Graph contrastive learning automated.
In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event* (2021), vol. 139 of *Proceedings of Machine Learning Research*, PMLR, pp. 12121–12132.

[132] YOU, Y., CHEN, T., SUI, Y., CHEN, T., WANG, Z., AND SHEN, Y.
Graph contrastive learning with augmentations.

In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual* (2020).

[133] YU, J., XIA, X., CHEN, T., CUI, L., HUNG, N., AND YIN, H.
Xsimgcl: Towards extremely simple graph contrastive learning for recommendation.
IEEE Transactions on Knowledge and Data Engineering 36, 02 (feb 2024), 913–926.

[134] YU, J., YIN, H., LI, J., WANG, Q., HUNG, N. Q. V., AND ZHANG, X.
Self-supervised multi-channel hypergraph convolutional network for social recommendation.
In *Proceedings of the Web Conference 2021* (New York, NY, USA, 2021), WWW '21, Association for Computing Machinery, p. 413–424.

[135] YU, J., YIN, H., LI, J., WANG, Q., HUNG, N. Q. V., AND ZHANG, X.
Self-supervised multi-channel hypergraph convolutional network for social recommendation.
In *Proceedings of the Web Conference 2021* (New York, NY, USA, 2021), WWW '21, Association for Computing Machinery, p. 413–424.

[136] YU, J., YIN, H., XIA, X., CHEN, T., CUI, L., AND NGUYEN, Q. V. H.
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* (New York, NY, USA, 2022), SIGIR '22, Association for Computing Machinery, p. 1294–1303.

[137] ZHANG, C., CHEN, R., ZHAO, X., HAN, Q., AND LI, L.
Denoising and prompt-tuning for multi-behavior recommendation.
In *Proceedings of the ACM Web Conference 2023* (New York, NY, USA, 2023), WWW '23, Association for Computing Machinery, p. 1355–1363.

[138] ZHANG, M., AND CHEN, Y.
Link prediction based on graph neural networks.
In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada* (2018), S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., pp. 5171–5181.

[139] ZHANG, S., CHEN, H., MING, X., CUI, L., YIN, H., AND XU, G.
Where are we in embedding spaces?

In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021* (2021), F. Zhu, B. C. Ooi, and C. Miao, Eds., ACM, pp. 2223–2231.

[140] ZHAO, G., QIAN, X., AND XIE, X.
User-service rating prediction by exploring social users' rating behaviors.
IEEE Transactions on Multimedia 18, 3 (2016), 496–506.

[141] ZHAO, H., YANG, X., WANG, Z., YANG, E., AND DENG, C.
Graph debiased contrastive learning with joint representation clustering.
In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021* (2021), ijcai.org, pp. 3434–3440.

[142] ZHAO, H., YAO, Q., LI, J., SONG, Y., AND LEE, D. L.
Meta-graph based recommendation fusion over heterogeneous information networks.
In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017* (2017), ACM, pp. 635–644.

[143] ZHAO, J., QU, M., LI, C., YAN, H., LIU, Q., LI, R., XIE, X., AND TANG, J.
Learning on large-scale text-attributed graphs via variational inference.
In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023* (2023), OpenReview.net.

[144] ZHAO, T., JIN, W., LIU, Y., WANG, Y., LIU, G., GÜNNEMANN, S., SHAH, N., AND JIANG, M.
Graph data augmentation for graph machine learning: A survey.
IEEE Data Eng. Bull. 46, 2 (2023), 140–165.

[145] ZHAO, W., ZHONG, L., AND WANG, G.
Semi-contrans: Semi-supervised medical image segmentation via multi-scale feature fusion and cross teaching of cnn and transformer.
In *2024 IEEE International Symposium on Biomedical Imaging (ISBI)* (2024), pp. 1–5.

[146] ZHAO, W. X., CHEN, J., WANG, P., GU, Q., AND WEN, J.-R.
Revisiting alternative experimental settings for evaluating top-n item recommendation algorithms.
CIKM '20, Association for Computing Machinery, p. 2329–2332.

[147] ZHOU, X., SUN, A., LIU, Y., ZHANG, J., AND MIAO, C.
Selfcf: A simple framework for self-supervised collaborative filtering.
ACM Trans. Recomm. Syst. 1, 2 (jun 2023).

[148] ZHU, Y., DU, Y., WANG, Y., XU, Y., ZHANG, J., LIU, Q., AND WU, S.
A survey on deep graph generation: Methods and applications.
In *Learning on Graphs Conference, LoG 2022, 9-12 December 2022, Virtual Event* (2022), B. Rieck and R. Pascanu, Eds., vol. 198 of *Proceedings of Machine Learning Research*, PMLR, p. 47.

[149] ZHU, Y., XU, Y., YU, F., LIU, Q., WU, S., AND WANG, L.
Graph contrastive learning with adaptive augmentation.
In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021* (2021), ACM / IW3C2, pp. 2069–2080.