A Study on

User Behavior Analysis with Graph-Structured Representations

Thesis submitted in fulfilment of the requirements
for the degree of

Doctor of Philosophy
in
Analytics

by

LI HE

Under the supervision of Professor Guandong Xu and Dr. Xianzhi Wang

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

August 2023

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, *Li He*, declare that this thesis is submitted in fulfilment of the requirements for the award of Philosophy, in the *School of Computer Science*, *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.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

SIGNATURE: _____

DATE: 17th August, 2023

PLACE: Sydney, Australia

ACKNOWLEDGMENTS

I acknowledge Completing a doctoral thesis is a challenging and demanding endeavor that requires the support and encouragement of numerous individuals and organizations. With deep appreciation and sincere gratitude, I would like to acknowledge and express my heartfelt thanks to those who have contributed to the successful completion of my thesis.

First and foremost, I would like to express my profound gratitude to my principal supervisor Prof. Gaundong Xu and co-supervisor Dr. Xianzhi Wang. Their guidance, expertise, and unwavering support throughout this journey have been invaluable. I am truly fortunate to have had their mentorship, as they have not only provided me with invaluable insights and constructive feedback but have also challenged me to think critically, pushing me to perform at my best. Their dedication, patience, and genuine interest in my research have been a constant source of inspiration, and I am immensely grateful for their guidance.

I am deeply indebted to my dissertation committee members, Hongda Tian and Shoujin Wang, for their valuable time, expertise, and insightful feedback. They have provided me with additional perspectives and suggestions, refining my research and enhancing the quality of my thesis. Their commitment to academic excellence and their unwavering support have propelled me to new levels of understanding and knowledge in my field of study. I am tremendously grateful for their contributions and contributions to my research.

I would also like to extend my heartfelt thanks to the participants of my study, whose time, experiences, and insights have been instrumental in shaping the findings of my thesis. Their willingness to share their knowledge, stories, and expertise have been invaluable, and I am truly grateful for their contributions. Their cooperation and enthusiasm have made this research journey even more rewarding.

I would like to express my sincere appreciation to the department faculty and staff who have provided assistance and resources throughout my doctoral studies. Their continuous support, encouragement, and commitment to academic excellence have created a

nurturing and stimulating environment for learning and research. The access provided to research materials, databases, and other resources has been critical to the success of this thesis, and I am immensely grateful for their contributions.

Furthermore, I would like to extend my gratitude to the research institutions, libraries, and archives that have provided access to the resources and materials necessary for the completion of my thesis. Their commitment to preserving and sharing knowledge has been instrumental in the advancement of research within our field, and I am grateful for their dedication to academic excellence.

I would like to express my deep appreciation to my wife, my son, friends, and loved ones, without whom this journey would not have been possible. Their unwavering support, understanding, and patience have sustained me during the challenging periods of this journey. Their belief in my abilities and constant encouragement have given me the strength and motivation to persevere, even when faced with obstacles. I am truly blessed to have such an incredible support system, and I am grateful for their love and encouragement throughout this journey.

In conclusion, this doctoral thesis represents the culmination of years of hard work, dedication, and support from numerous individuals and organizations. I extend my deepest appreciation and gratitude to my advisor, committee members, research participants, department faculty and staff, research institutions and libraries, and my loved ones for their unwavering support throughout this journey. Each and every one of you has played a significant role in shaping my research and helping me to achieve this milestone. Your contributions, guidance, encouragement, and support have had a profound impact on my academic journey, and I am eternally grateful for your presence in my life.

Abstract

Nowadays, the development of Web 2.0 technology brings a huge change in the way of human's life styles. Variant e-commerce websites, e.g., Yelp, eBay and Amazon, provide internet user with a convenient, efficient and relatively reliable online trading environment. More and more merchants prefer to build their virtual shop through different online platforms. Meanwhile, an increasing number of consumers gradually get used to this way of shopping, and automatically share their shop experiences by using online platform which applied by the e-commerce website. This trend generates huge amount of user behavior information and product attributes during purchasing process. We define this online shopping scenario as a special kind of social network, named **e**-commerce **S**ocial **N**etworks (ESNs) in this thesis. Online ESNs poses an interesting problem: how to best characterize the different classes of user behavior. Traditionally, user behavior representation methods, based on user individual features, are not appropriate for online networking platforms. In these complex social networks, users interact with other users through multiple interfaces that allow them to upload multimedia content and have many other interactions. Different behavior patterns can be observed for different individuals and groups. In this thesis, we will propose graph-structured methodologies for characterizing and identifying user behaviors in online social networks. This thesis will help the achievement of more strategic objectives on large-scale node classification tasks in graph-structured social network datasets.

This thesis achieves research contributions as follows:

- It develops a novel graph-based model, namely Graph-aware Deep Fusion Networks (GDFN) that utilizes information from relevant metadata (review text, features of users, and items) and relational data (network) to capture the semantic information from their complex heterogeneous interactions via graph convolutional networks. Besides, GDFN also uses a novel fusion technique to synthesize low and high-order interactions with propagated information across multiple review-related sub-graphs. Extensive experiments on publicly available datasets show

that our proposed model is effective and outperforms several strong state-of-the-art baselines.

- It designs a Hypergraph Click-Through Rate prediction framework (HyperCTR) built upon the hyperedge notion of hypergraph neural networks, which can yield modal-specific representations of users and micro-videos to better capture user preferences. We construct a time-aware user-item bipartite network with multi-modal information and enrich the representation of each user and item with the generated interests-based user hypergraph and item hypergraph. Through extensive experiments on three public datasets, we demonstrate that our proposed model significantly outperforms various state-of-the-art methods.

- It further improves the GCN-based Collaborative Filtering (CF) models from two aspects. First, we remove non-linearities to enhance recommendation performance, which is consistent with the theories in simple graph convolutional networks. Second, we obtain the initialization of the embedding for each node in the graph by computing the network embedding on the condensed graph, which alleviates the over smoothing problem in graph convolution aggregation operation with sparse interaction data. The proposed model is a linear model that is easy to train, scalable to large datasets, and shown to yield better efficiency and effectiveness on four real datasets.

- It explores informative and controllable text using social media language by incorporating topic knowledge into a keyword-to-text framework. It is a novel Topic-Controllable Key-to-Text (TC-K2T) generator that focuses on the issues of ignoring unordered keywords and utilizing subject-controlled information from previous research. TC-K2T is built on the framework of conditional language encoders. In order to guide the model to produce an informative and controllable language, the generator first inputs unordered keywords and uses subjects to simulate prior human knowledge. Using an additional probability term, the model increases the likelihood of topic words appearing in the generated text to bias the overall distribution. The proposed TC-K2T can produce more informative and controllable senescence, outperforming state-of-the-art models, according to empirical research on automatic evaluation metrics and human annotations.

List of Publications

Conference Papers :

1. C-1. He, Li, et al. "Click-through rate prediction with multi-modal hypergraphs." Proceedings of the 30th ACM International Conference on Information & Knowledge Management. 2021. (Chapter-4)

2. C-2. He, Li, et al. "Simplifying Graph-based Collaborative Filtering for Recommendation." Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining. 2023. (Chapter-5)

3. C-3. He, Li, et al. "TagPick: A System for Bridging Micro-Video Hashtags and E-commerce Categories." Proceedings of the 30th ACM International Conference on Information & Knowledge Management. 2021.

4. C-4. Kaize Shi, Xueyao Sun, Li He, Dingxian Wang, Qing Li, and Guandong Xu. 2023. AMR-TST: Abstract Meaning Representation-based Text Style Transfer. In Findings of the Association for Computational Linguistics: ACL 2023, pages 4231-4243, Toronto, Canada. Association for Computational Linguistics.

Journal Papers :

1. J-1. He, Li, et al. "Graph-Aware Deep Fusion Networks for Online Spam Review Detection." IEEE Transactions on Computational Social Systems (2022). (Chapter-3)

2. J-2. He, Li, et al. "Online Spam Review Detection: A Survey of Literature." Human-Centric Intelligent Systems 2.1-2 (2022): 14-30.

3. J-3. He, Li, et al. "A topic,Äêcontrollable keywords,Äêto,Äêtext generator with knowledge base network." CAAI Transactions on Intelligence Technology (2023). (Chapter-6)

Table of Contents

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Over the years, User Behavioral Analysis (UBA) has been the focus of intense efforts in e-commerce applications [35]. Obviously, the objective is to adopt some new specific and efficient marketing strategies that are based on data, i.e., recorded information that represent the past activities of potential consumers. This is referred to as data-based behavioral marketing [37]. Behavioral analysis has also found its usefulness in the fight against fraud and in various other applications [3]. Recently, it is not a surprise to see that behavioral analysis can enhance information and communication technology, organize more efficiently production tools, detect internal threats like targeted attacks, adapt softwares to the users, accelerate some repetitive tasks, etc. However, it goes with a certain acceptability from the users [6].

The UBA is the discipline of analyzing user behaviors. In an operational way, it is essentially the collecting, monitoring and processing of user data. The datasets collected from the users are stored in databases, data log files, histories, directories, and furthermore any other systems recording the user behaviors. The purpose of this process is to provide parameters and to build reliable and usable models of users, in other words, that accurately characterize the users.

For example, the online social network has become a sharing content space for individuals and groups. Indeed, technologies are now mature, ready and spread out in order to collect and exploit the present and past behavior of Internet users in real time. The datasets generated through internet browsing enable us to deduce various aspects such as points of interest (POI), attendance patterns, factual details and gestures,

movements, attitudes, lifestyle preferences, and living standards. Obviously, the status of a user can evolve and change over time. Techniques make it possible to adapt the models on the basis of the experiment and according to the evolutions of the collected data in real time [12].

The UBA relies on three key components: Data analysis, data integration and data presentation. Actually, the analysis and processing the phenomenal amount of data is the most difficult challenge. The heterogeneity, volume and speed of data generation are increasing rapidly. This is exacerbated with the use of wireless networks, Internet of Things (IoT) sensors, smartphones and the increasing activities on the Internet. Therefore, real time UBA must be fast in processing the big amount of data and Machine Learning (ML) algorithms should be appropriate candidates [90]. For that purpose, ML algorithms must be run in real time, access to the whole datasets, adapt their own learning parameters. ML algorithms can also be interfaced with enterprise resource planning softwares to get additional information about the users and to combine them with their past and present activities while processing. The idea is to enable the establishment of self-adaptive models [113].

The thesis aims to solve two challenges of UBA, which are the user credibility analysis and the user preference learning. It will provide a deeper look at the online consumer behavior and quantify these behavior with in online content sharing platforms using machine learning, neural networks methods, text mining approaches and graph-based solutions. These are about combining graph-structured representation techniques with other deep learning models to quantify internet users behaviours within a mathematical model. The research methodology utilises deep learning algorithms to apply the group-based neural network for deep analytics.

1.1 Main Challenges

In this section, we will introduce our main challenges from four perspectives: online reviews, click-through rate, graph convolutional networks(GCN) simplification and natural language generation.

Online Reviews E-commerce companies such as Amazon and eBay have earned approximately $3.5 trillion in sales in 2019 and are anticipating an increase to $4.9 trillion by the end of 2021, according to Shopify.com.Online e-commerce has demonstrated its unique importance during the COVID-19 pandemic and enabled hundreds of millions

of consumers to purchase products anytime and anywhere around the world. These days customers can also share their shopping experiences by rating items, writing reviews and answering questions.

Online reviews play an important role in e-commerce as they impact the purchasing decisions of approximately 93% people, according to Ingniyte.co.uk. Unfortunately, online reviews can be deliberately injected (a.k.a., "spam reviews") to mislead potential customers [58] for various unethical reasons, such as unfair marketing or online brand attacks [71]. According to BrightLocal.com, 74% of consumers in 2019 have encountered spam reviews yet failed to recognize them. It has thus become very crucial to devise effective methods that can identify spam reviews automatically so that these platforms remain reliable [147].

Click-through Rate Advertising is central to many online e-commerce platforms such as e-Bay and Amazon. One of the important signals that these platforms rely upon is the click-through rate (CTR) prediction. The recent popularity of multi-modal sharing platforms such as TikTok has led to an increased interest in online micro-videos. It is, therefore, useful to consider micro-videos to help a merchant target micro-video advertising better and find users' favourites to enhance user experience. Existing works on CTR prediction largely exploit unimodal content to learn item representations. A relatively minimal effort has been made to leverage multi-modal information exchange among users and items.

CTR prediction has become one of the core components of modern advertising on many e-commerce platforms. The goal is to predict customers' click probability on wide range of items. Existing works on CTR prediction only focus on modeling pairwise interactions from uni-modal features which might not lead to satisfactory results. This existing gap leads to new opportunities where we can exploit the widely available multi-modal features which is largely unexplored. Besides, they can given complementary information to the model which alone cannot be obtained via uni-modal modeling.

Recently, the wide-spreading influence of micro-video sharing platforms, e.g., Tiktok [1] and Kuaishou [2] make them a popular platform for socialising, sharing and advertising as micro-videos. These videos are compact and come with rich multimedia content from multiple modalities, i.e., textual, visual, as well as acoustic information. Motivated by this, we propose a novel method which addresses the limitations in current methods

[1] https://www.tiktok.com/

[2] https://www.kuaishou.com/

and improve CTR prediction performance through micro-videos. However, modeling multi-modal features extracted from micro-videos for CTR prediction in a holistic way is not straightforward. First, in a typical setting of CTR prediction, the interactions between users and items are normally sparse, and the sparsity issue becomes even more severe (in magnitude of number of modalities) when taking into account multi-modal features. For example, compared to uni-modal feature space, the sparsity of a dataset is tripled when considering visual, acoustic and text features of a target item. Therefore, effectively mitigating the sparsity issues introduced by multi-modal features without compromising upon the performance of the model is the key to this problem.

GCN Simplification The field of "GCN Simplification" refers to the research area focused on simplifying Graph Convolutional Networks (GCNs) - a popular deep learning architecture for graph data. Graph data involves entities or nodes connected through edges, representing relationships between them such as social networks, citation networks, or biological interactions.

GCNs have gained significant attention due to their ability to effectively capture the complex relationships and dependencies inherent in graph data. However, their complexity and large parameter space pose challenges, limiting their usability, especially in resource-constrained scenarios.

The goal of GCN Simplification research is to address these challenges by devising techniques that reduce the complexity and size of GCNs while maintaining or even enhancing their performance. This involves exploring approaches such as network pruning, model compression, or knowledge distillation. By simplifying GCNs, researchers aim to make them more computationally efficient, memory-friendly, and deployable on edge devices or low-power hardware.

The outcomes of GCN Simplification research have the potential to open doors for widespread adoption of GCNs in various fields, including social network analysis, recommendation systems, bioinformatics, and drug discovery. By making GCNs more accessible and less resource-demanding, researchers envision democratizing the power of graph-based learning, enabling more efficient analysis and decision-making in a wide range of domains.

Natural Language Generation Natural Language Generation (NLG) is a research field within the realm of Artificial Intelligence (AI) and natural language processing that focuses on the automatic generation of human-like text or speech from structured data

or other non-linguistic sources. NLG aims to enable machines to produce coherent, contextually appropriate, and linguistically accurate narratives. This technology has diverse applications in chatbots, conversational agents, language translation, summarization systems, content generation, and personalized recommendation systems.

According to a study conducted by Dusek et al. (2019) [115], NLG techniques employ statistical models, deep learning architectures, and rule-based approaches to transform structured data into natural language text. These methodologies utilize advanced machine learning and computational linguistics techniques to generate text that mimics human-like language patterns, semantics, and stylistic variations. However, NLG research encounters challenges such as managing ambiguity, ensuring coherence, generating diverse and creative outputs, and decreasing grammar and meaning errors.

The advancements in NLG have shown promising results, with systems capable of generating high-quality narratives and engaging in real-time interactions with users. NLG technology has significant implications for various industries, including customer service, e-commerce, healthcare, and content creation, as it enables automated generation of context-aware and personalized content at scale.

In conclusion, ongoing research in NLG aims to advance the state-of-the-art in generating natural language text, making machines proficient in understanding and producing human-like narratives, and enhancing human-machine communication and collaboration.

1.2 Research Questions and Objectives

As the digital landscape continues to evolve, the credibility of user behavior in e-commerce systems and the effective utilization of graph neural networks for large-scale node classification tasks have become paramount research questions. Similarly, efficiently presenting user preferences in recommender systems and leveraging graph neural networks for prediction tasks and unsupervised node representation learning pose significant challenges. Furthermore, enhancing complex graph-based models with multimodal information to infer user behavior in sequential data, improving upon these models with simplified and accelerated forms of GCNs, and discovering effective representations of user behavior through natural language processing are critical areas of inquiry.

Research Question 1: How to accurately analyze the credibility of user behavior in e-commerce system and can we develop and efficiently implement

graph neural network models for large-scale node classification tasks in graph-structured datasets?

Our main contribution to address this question is a novel graph neural network model that we call the graph convolutional network (GCN) [64]. GCN improves upon earlier work in the community on so-called spectral graph convolutions. We extend this method to homogeneous and heterogeneous networks and demonstrate an application of this model on text classification task with large-scale nodes and edges.

Research Question 2: How to efficient to present user potential preferences in recommender system and can graph neural networks be utilized for prediction task and unsupervised node representation learning?

We introduce a extension to the GCN model to address prediction task in recommendation system. Our models can be trained on graphs in the absence of node labels, a setting often referred to as unsupervised node representation learning and introduced neural network architectures for explicitly graph-structured data in this thesis. We will investigate how models with structural and compositional inductive biases - such as graph neural networks - can be developed and applied to problems with implicit or hidden structure.

Research Question 3: How can we improve upon complex graph-based models with multimodal information that infer user behavior in sequential data?

Many graph-based methods are heavily rely on the pairwise relations and users are regarded as independent. However, these user behaviour models might not be a strong suit for the scenarios with multi-modal data, the situation for data correlation modelling could be more complex. Under such circumstances, traditional graph structure has the limitation to formulate the data correlation, which limits the application of graph convolutional neural networks. With the recent advancement in hypergraph neural networks (HGNN), quantifying the user behaviour with multimodal hypergraph neural networks might be a good approach to model e-commerce users,Äò behaviours.

Research Question 4: How can we improve upon complex graph-based models with simplified and accelerated form of GCNs?

The goal of the research is to develop a more efficient and effective method for graph-based modeling using a simplified and accelerated version of Graph Convolutional Networks (GCNs). GCNs are a popular technique for learning representations in graph-structured data, but they can be computationally expensive and challenging to implement in large-scale applications. Therefore, the research aims to simplify and accelerate GCNs

to make them more practical and accessible for complex graph-based models.

Research Question 5: How can we discover and build effective representations of user behavior by using natural language processing?

The purpose of the research is to explore the utilization of natural language processing techniques to discover and construct effective representations of user behavior. This involves analyzing textual data, such as user reviews, comments, or feedback, to extract meaningful information and patterns. By leveraging natural language processing algorithms, the research aims to develop techniques that can process and understand user-generated content, enabling the construction of accurate and comprehensive representations of user behavior. These representations can then be utilized for various applications, such as personalized recommendations, sentiment analysis, or user profiling.

To address these questions, we have formulated the following research objectives: Firstly, we aim to analyze the correlation between various factors and online review behaviors of e-commerce consumers, providing insights into user credibility. Secondly, we seek to develop an efficient click-through rate prediction model tailored to capture the interest behaviors of users on content sharing platforms. Thirdly, we intend to construct sequential graph-based networks that incorporate multiple modalities of information to accurately model user preferences. Finally, we explore the utilization of control language generation models to generate textual information and employ representation learning methods embedded in user behavior, aiming to discover and build effective representations of user behavior. Through these objectives, we hope to contribute significantly to the understanding and optimization of user behavior in e-commerce systems and recommender systems.

Objective 1 (in answer to RQ1)Analyse the correlation between different factors and different online review behaviours of e-commerce consumers.

We hypothesize that modelling the information gathered from reviews, users and items could help substantially improve the performance and generalizability of online spam review detection systems. We consider the user-review-item network to formulate the problem as a graph-based classification task, in which reviews are labelled as spam or non-spam.

Objective 2 (in answer to RQ2 and RQ3) Build a efficient click-through rate prediction model for interest behaviours of sharing content platform users.

The existing deep learning approaches proposed for CTR prediction target without considering data aggregation and correlation. In this thesis, the complex data correla-

tion is formulated in a hypergraph structure, and we design a hyperedge convolution operation to better exploit the high-order data correlation for representation learning.

Objective 3 (in answer to RQ4) Construct a sequential graph-based networks to model the user preference with multiple modalities information upon the hyperedge notion.

Motivated by rich multimedia content from multiple modalities, i.e., textual, visual, as well as acoustic information, we address this problem with multi-modal features from user aspect. We first build modality-originated hypergraphs which can be treated as data argumentation technique.

Objective 4 (in answer to RQ5) Explore the control language generation model to generate the text information, and use the representation learning method embedded in the user behavior.

This research object lies in its potential to bridge the gap between user behavior data and text generation. By exploring the use of a control language generation model, the research aims to unlock the capability of generating coherent and contextually appropriate text based on user behavior.

1.3 Research Significance

Understanding user behavior is essential for various domains, including recommender systems, fraud detection, social network analysis, and personalized marketing. The research significance of user behavior analysis with graph-structured representations lies in its ability to uncover hidden connections, enhance recommender systems, strengthen fraud detection, unveil social network dynamics, and contribute to personalized marketing efforts.

Graph-structured representations offer a powerful framework for capturing complex relationships among users, items, and their interactions. By representing users, items, and their interactions as nodes and edges in a graph, this approach enables the exploration of hidden connections that may not be evident in traditional tabular or sequential data formats. This allows researchers to uncover latent user behavior patterns, community structures, and influential nodes within the network.

In the realm of recommender systems, the integration of graph-structured representations opens up new possibilities for collaborative filtering, item similarity analysis, and knowledge propagation across the user-item graph. This leads to improved recom-

mendation accuracy, diversity, and serendipity, ultimately resulting in enhanced user satisfaction and engagement.

Moreover, the use of graph-based models strengthens fraud detection systems by identifying anomalous behavior patterns, suspicious networks, and coordinated attacks. By leveraging the holistic view of user interactions provided by graph-structured representations, researchers can develop robust fraud detection systems that effectively identify and prevent fraudulent activities.

Understanding social network dynamics is crucial for analyzing information diffusion, social influence, and community structures. Graph-structured representations enable researchers to study the complex interplay between individuals, groups, and information flow within social networks. This provides valuable insights into collective behavior, the emergence of communities, and information propagation mechanisms.

Lastly, in personalized marketing, graph-structured representations allow for the creation of personalized customer profiles, identification of similar user clusters, and targeted marketing campaigns. By leveraging graph-based models, researchers can enhance personalized marketing efforts, resulting in improved customer satisfaction, increased conversion rates, and higher marketing return on investment.

In conclusion, the research significance of user behavior analysis with graph-structured representations lies in its ability to uncover hidden connections, enhance recommender systems, strengthen fraud detection, unveil social network dynamics, and contribute to personalized marketing efforts. By harnessing the power of graphs, this research aims to advance the field of user behavior analysis and provide valuable insights for various domains, ultimately leading to more effective decision-making, enhanced user experiences, and improved system performance.

1.4 Thesis Structure

The logical structure of this thesis and relationship between the chapters are shown in Figure 1.1, including the chapters and the corresponding research questions. The main contents of each chapter are summarized as follows:

Chapter 2 presents a literature review of research related to this study. In this chapter, we review related studies from four perspectives: online spam review detection, click-through rate prediction, GCN simplification and natural language generation. Next, we introduce the public datasets commonly used in social media analysis research works. Finally, we review popular methodology of graph-based deep learning.

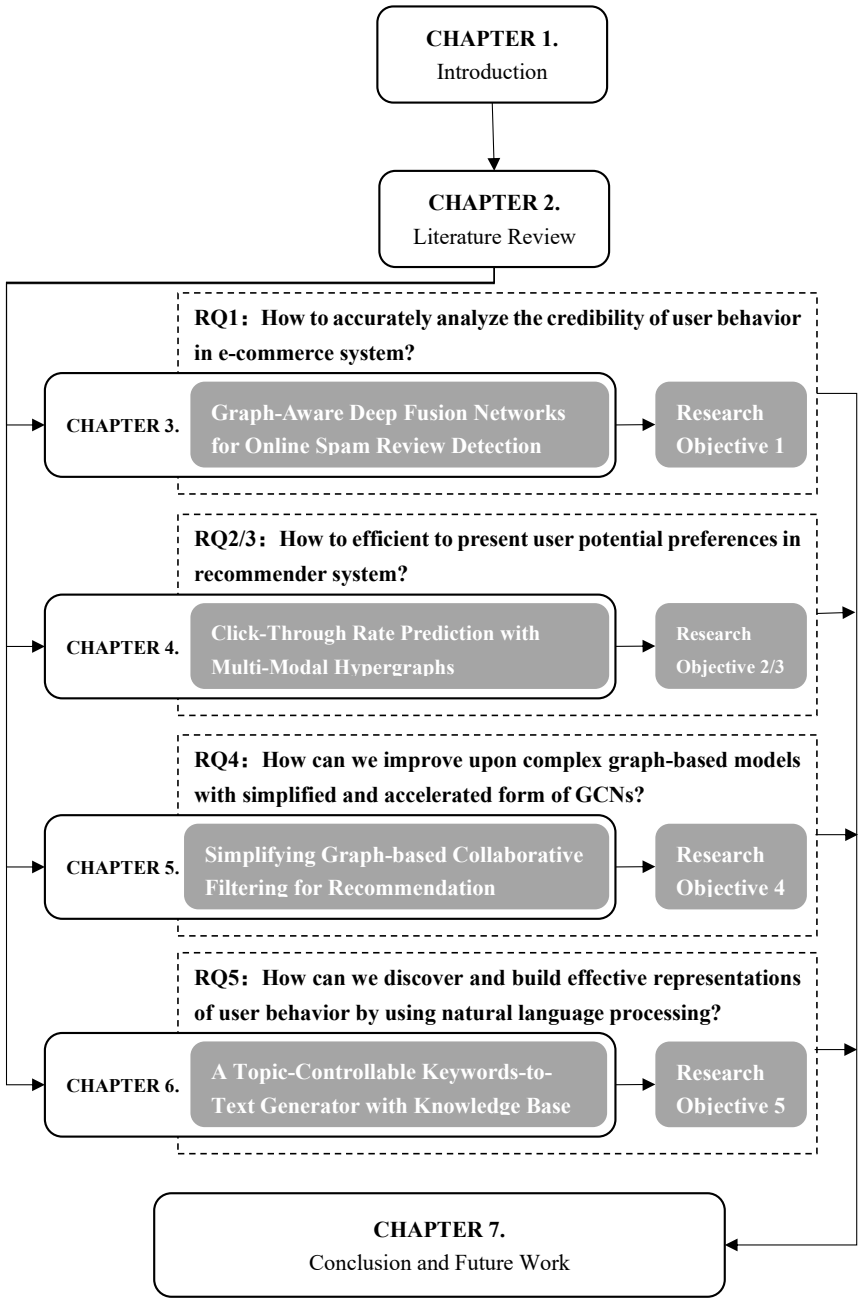

Figure 1.1: The Structure of Thesis

Chapter 3 introduces a new model called GDFN, which predicts spam reviews by using a unimodal graph to group similar review text and extract aggregation-based semantic features. Additionally, we incorporate user (item)-level information to enhance the overall representation. It also employs a fusion mechanism to capture the underlying relationship between users, reviews, and items. Through experiments on two public datasets, it showcase the superior performance of GDFN when compared to other state-of-the-art models.

Chapter 4 improves the accuracy of CTR prediction, this chapter utilizes temporal user preferences and multi-modal item attributes. A novel framework, called HyperCTR, leverages the interaction between users and micro-videos by considering different modalities, using an HGCN-based approach. We enhance the user representation by incorporating time-aware and group-aware features. By stacking hypergraph convolution networks, self-attention, and fusion layers, the model effectively captures user preferences and achieves improved performance.

Chapter 5 conducts a review of existing recommendation models based on Graph Convolution Networks (GCNs) and introduced a new model called Simplifying Graph-based Collaborative Filtering(SGCF) for Collaborative Filtering (CF)-based recommendation. SGCF consists of two primary components. Firstly, it empirically removes non-linear transformations in GCNs, opting for linear embedding propagation, which has shown promising progress. Secondly, it devises a condensed graph learning process for the input network to mitigate the excessive smoothing effect usually caused by higher layers of graph convolutions. The extensive experiments demonstrated the effectiveness and efficiency of its proposed model.

Chapter 6 proposes a new text generator called TC-K2T model is introduced for the task of generating text from keywords. Unlike previous models, TC-K2T incorporates topic knowledge networks and features an improved decoder. This is the first attempt at addressing this challenging task. Through a series of experiments on a publicly available dataset, the performance of TC-K2T model is evaluated, demonstrating its superiority over state-of-the-art models.

Chapter 7 summarizes the contributions of this research and discusses research issues for further study.

The justification for each chapter from Chapter 3 to Chapter 6 of the thesis lies in its focus on a specific objective. Moreover, the selection of methods and datasets in each chapter is meticulous to ensure the best achievement of that objective. Additionally, the datasets used in each chapter are selected based on their reliability and validity,

after carefully assessing their sources and collection methods. Only high-quality and relevant datasets are utilized, ensuring a solid foundation for addressing the research question. In essence, the utilization of diverse datasets across various chapters facilitates a comprehensive analysis of the research questions, thereby bolstering the credibility of the thesis' conclusions.

CHAPTER 2

LITERATURE REVIEW

This Literature Review delves into four crucial areas of research: Online Spam Review Detection, Click-through Rate Prediction, GCN Simplification, and Natural Language Generation. By exploring the wealth of existing literature in these fields, we aim to establish a solid foundation for our study, identify key advancements, and highlight potential areas for further exploration. Additionally, this review also incorporates a discussion on Data and Research Ethics, emphasizing the importance of ethical considerations in conducting research in this domain.

2.1 Online Spam Review Detection

Feature-centric Methods Traditional statistical methods rely on extracting different features from textual reviews, followed by learning a language model. [58] first identified three types of spam reviews, which are untruthful opinions, reviews on brands only and non-reviews, and then analyzed real-world datasets from Amazon. They extracted review-centric, reviewer-centric and product-centric features, and used them as input to a logistic regression (LR) model. Recently, [142] summarized eleven platform-independent features from the word level, the semantic level and the structural level to discriminate fraud and normal items. They selected Xgboost as a binary classifier, and their evaluation results indicated that CATS achieves both high precision and recall. [93] approached the problem using three strategies as features in naive Bayes and SVM classifier. Further, [135] attempted to use Long Short-Term Memory (LSTM) framework to detect spam

reviews. They established three types of layers to predict spam reviews, the input layer for receiving data, hidden layer of LSTM and output layer, respectively.

Graph-based Methods Graph-based methods have been popularly applied to capture text features among different entities. The first Graph Neural Network [114] (GNN)-based spam review detection method was proposed by [136], who built a heterogeneous "review graph" to represent the relationship among reviewers, reviews and online sellers. [117] utilized spam features for modelling review datasets as heterogeneous information networks to map spam review detection procedure into a classification problem in such networks. In the classification step, they proposed meta-path concepts to find features importance and calculate the weight. [82] presented a neural network-based graph model, named Graph Embeddings for Malicious accounts (GEM), which both considered "device aggregation" and "activity aggregation" in heterogeneous graphs. Until present, these methods have focused on shallow encoders, i.e., matrix factorization. There is no parameter sharing, and every node has its unique embedding vector and the inherent "transductive" features are impossible to generate embeddings for unseen nodes during training and do not incorporate node features.

Recent years have witnessed a growing interest in utilizing "message-passing" methodology in graphs [163], which learns how to aggregate information from each type of neighbour using Markov Random Field (MRF) techniques implicitly. [40] presented the GraphSAGE model, which achieves significant improvements compared with previous methods such as DeepWalk [98] and SemiGCN [64]. This method overcomes the limitation of applying GCN in transductive settings with specified Laplacian matrix. A model-based Graph Convolutional Neutral Networks (GCNN) for spam-bot detection is proposed in [1], which proposed an inductive representation learning approach for spam review detection based on the reviewer profile information and the social network graph on Twitter datasets, and the inductive representation learning method used in their approach is similar to that of GraphSAGE. In short, GCN-based methods have been applied in various domains, such as spam advertisement identification [67], recommendation system [154], social spammer detection [146], rumor detection [8] and so no. However, the above methods depend only on the local information of surrounding neighbourhoods of a target node, making the model sometimes noisy and thus ineffective.

2.2 Click-through Rate Prediction

CTR prediction Learning the effect of feature interactions seems to be crucial for accurate CTR prediction. Factorization Machines (FMs) [9, 109] are proposed to model pairwise feature interactions in terms of the vectors corresponding to the involved features. AutoFIS [78] and UBR4CTR [101] further improve FM by removing the redundant feature interactions and retrieving a limited number of historic behavior that are most useful for each CTR prediction target. However, a FM-based model considers learning shallow representation, and it thus is unable to model the features faithfully. Deep Neural Networks (DNNs) are exploited for CTR prediction in order to automatically learn feature representations and higher-order feature interactions. DSTN [94] integrates heterogeneous auxiliary data (i.e., contextual, clicked and unclicked ads) in a unified framework based on the DNN model. Further, the other stream of models focus more on mining temporal patterns from sequential user behavior. GRU4Rec [51] is based on RNN. It is the first work which uses the recurrent cell to model sequential user behavior. MIMN [99] applies the LSTM/GRU operations for modeling users' lifelong sequential behavior.

Exploiting multi-modal representation Some works focus on the multi-modal representation in the area of multi-modal CTR prediction. Existing multi-modal representations have mostly been applied to recommender systems and have been grouped into two categories: joint representations and coordinated representations [141]. Joint representations usually combine the single-modal information and project into the same representation space [22, 23, 160], but they are suited for situations where all of the modalities are present during inference, which is hardly guaranteed in social platforms. Different from the joint representations, the coordinated models learn separate representations for each modality but coordinate them with constraints [141]. Since the modal-specific information is the factor for the differences in each modality signals, the model-specific features are inevitably discarded via similar constrains. In contrast, we introduce a novel model which respectively models the information augmentation and group-aware network problems to address the limitations in existing works.

Graph Convolution Network Our proposed model uses the GCNs technique to represent the users and items, which has been popularly used for modeling the social media data. In [40] the authors proposed a general inductive framework which leverages the content information to generate node representation for unseen data. In [154] the

authors developed a large-scale deep recommendation engine on Pinterest for image recommendation. In their model, graph convolutions and random walks are combined to generate the representations of nodes. In [7] the authors proposed a graph auto-encoder framework based on message passing on the bipartite interaction graph. However, these methods cannot model the multi-modal data including cases where data correlation modeling is not straightforward [33].

2.3 GCN Simplification

Graph Convolutional Networks (GCNs) have become a popular deep learning technique for extracting meaningful representations from graph-structured data. However, the complexity and computational cost of GCNs make them challenging to implement in resource-constrained environments. Consequently, simplification techniques for GCNs have been explored to reduce their complexity while maintaining their essential capabilities. This literature review aims to provide an overview of existing approaches for GCN simplification and their effectiveness in achieving simplified yet efficient graph convolution operations.

Graph Convolutional Network (GCN) Kipf and Welling (2017) [64] introduced the GCN neural network architecture, which generalizes the convolutional operations from grid-like data to graph-structured data. They proposed a simplified version of spectral graph convolutions that leverages the localized first-order approximation of spectral filters. GCNs have shown promising results in various applications, including node classification and link prediction.

Faster GCN Training To simplify the training process of GCNs, Chen et al. (2018) [15] proposed a simplified variant called FastGCN. They introduced an edge sampling technique that approximates global neighborhood aggregation, reducing both the training time and memory overhead while maintaining competitive accuracy. FastGCN achieves similar performance to standard GCNs but with significantly improved efficiency.

GraphSAGE Hamilton et al. (2017) [40] presented GraphSAGE, a simplified variation of GCNs. GraphSAGE defines a generalized framework for generating node representations using neighborhood aggregation. It employs a simple yet effective sampling

strategy, which scales well with large graphs. GraphSAGE demonstrates competitive performance in various graph-based tasks while simplifying the overall architecture.

Graph Attention Networks (GAT) While not a direct simplification of GCNs, Velickovic et al. (2018) [133] introduced Graph Attention Networks (GAT), which offer an alternative, simplified mechanism for aggregating node information. GATs utilize self-attention mechanisms to learn weights for aggregating neighbors' information, avoiding the need for carefully designed filter functions. GATs achieve competitive performance with GCNs while simplifying the convolutional operation.

GCN simplification techniques aim to reduce the complexity and computational cost of traditional GCNs while preserving their representational power. Several approaches, including FastGCN, GraphSAGE, low-rank approximation, and GAT, have been proposed to achieve this goal. These techniques simplify various aspects of the GCN architecture, such as training, weight approximation, and aggregation mechanisms. By striking a balance between efficiency and accuracy, these simplification techniques make GCNs more accessible and practical for resource-constrained environments.

2.4 Natural Language Generation

Natural Language Generation (NLG) refers to the process of automatically producing human-like text or speech from structured data or other non-linguistic inputs. NLG has become a crucial area of research and application, enabling various tasks such as machine translation, chatbots, summarization, and data storytelling. This literature review aims to provide an overview of the recent advancements in NLG techniques and their applications.

Traditional Rule-Based Approaches Early NLG systems primarily relied on rule-based approaches, where predefined templates and heuristics were used to generate sentences. These systems were limited in their ability to handle complex and diverse inputs and often required manual intervention for rule creation. Despite their limitations, rule-based NLG systems paved the way for subsequent advancements.

Statistical and Machine Learning Approaches With the rise of machine learning and statistical techniques, NLG witnessed significant advancements. Techniques such as Hidden Markov Models (HMMs), Conditional Random Fields (CRFs), and Recurrent

Neural Networks (RNNs) were employed to capture the statistical properties of text and perform language generation. [92] These approaches allowed for more dynamic and adaptable NLG systems.

Neural Language Models The emergence of neural network architectures revolutionized NLG. Models like LSTM (Long Short-Term Memory) and Transformer have shown exceptional performance in various NLG tasks, including language translation, text summarization, and image captioning. [128, 132] These models leverage the power of deep learning and attention mechanisms, allowing for more accurate and context-aware text generation.

Variations of NLG Models Several variations of NLG models have been proposed to address specific challenges. For example, Pointer-Generator Networks combine the benefits of traditional statistical approaches and neural networks to generate text by copying words from the input. GPT (Generative Pre-trained Transformer) models use large-scale unsupervised training to generate coherent and contextually relevant text. These variations have led to significant improvements in the quality and fluency of generated text.

NLG for Conversational Agents Recent research has focused on NLG for conversational agents such as chatbots and virtual assistants. [103, 115] Neural approaches, coupled with reinforcement learning techniques, have been used to train chatbots to generate contextually appropriate and engaging responses. Dialog systems have also seen advancements in NLG, incorporating techniques such as dialogue state tracking and adaptive responses.

Evaluation of NLG Systems Evaluating the quality and effectiveness of NLG systems is a challenging task. Traditional metrics like BLEU and ROUGE have limitations in capturing the semantic quality and coherence of generated text. Recent research has focused on developing better evaluation metrics, such as METEOR and BERTScore, which consider both lexical and semantic aspects to assess the performance of NLG systems. [28]

The field of Natural Language Generation has seen significant advancements in recent years, driven by the adoption of neural network architectures and the availability of large-scale datasets. The combination of deep learning techniques, such as LSTM and Transformer, with innovative variations like Pointer-Generator Networks and GPT

Table 2.1: Dataset statistics.

Dataset	Yelp			Op Spam	
	CHI	NYC	ZIP	Positive	Negative
#Users	38,063	160,225	260,277	-	-
#Products	201	923	5044	20	20
#Spam Reviews	8,919	36,885	80,466	400	400
#Non-spam Reviews	58,477	322,167	528,133	400	400
%Spam	13.23%	10.27%	13.22%	-	-

models, has led to improved text generation quality and fluency. NLG has found applications in various domains, including conversational agents, summarization, and data storytelling. Further research is needed to address challenges in evaluation, adaptability to specific domains, and ethical considerations in NLG systems.

2.5 Data and Research Ethics

The thesis research will utilise both publicly available datasets from other literatures. Even though the below datasets are sufficient for the empirical works within the scope of this thesis, the author will actively seek to expand the datasets in order to provide a broader and more comprehensive analysis for user bevioural modelling of e-commerce consumers. All the datasets have been processed to remove any personal identification, including anonymizing names and personal contact details used within the text of datasets. Datasets will be integrated using customer number only.

2.5.1 Public Datasets

Unimodal Datasets We evaluate our proposed method on two benchmark publicly available datasets, which are: **Yelp** from [105] (Table 5.1), contains three public spam review datasets crawled from Yelp website: YelpChi, YelpNYC, and YelpZip. The dataset comprises of binary labels: N representing genuine review and Y representing spam reviews. **Op_spam_v1.4** from [21] (Table 5.1), consists of truthful and deceptive hotel reviews of 20 Chicago hotels. The label of each review in Op_spam_v1.4 was gathered from Amazon's popular Mechanical Turk crowdsourcing service and five popular online review communities: Expedia, Hotels.com, Orbitz, Priceline and TripAdvisor.

Multimodal Datasets Existing CTR prediction models mostly utilize unimodal datasets [77, 81, 101, 121]. In contrast, we introduce multiple modalities into CTR prediction. As

Table 2.2: Statistics of the dataset. (v, a and t denote the dimensions of visual, acoustic, and textual modalities, respectively.)

Dataset	#Items	#Users	#Interactions	Sparsity	v.	a.	t.
Kuaishou	3,239,534	10,000	13,661,383	99.98%	2048	-	128
MV1.7M	1,704,880	10,986	12,737,619	-	128	128	128
Movielens	10,681	71,567	10,000,054	99.63%	2048	128	100

mentioned above, micro-video datasets contain rich multimedia information and include multiple modalities - visual, acoustic and textual. We experimented with three publicly available datasets: Kuaishou, MV1.7M and MovieLens 10M which are summarized in Table 4.1.

Kuaishou: This dataset is released by the Kuaishou [75]. There are multiple interactions between users and micro-videos. Each behaviour is also associated with a timestamp, which records when the event happens. The timestamp has been processed to modify the absolute time, but the sequential temporal order is preserved w.r.t to the timestamp.

Micro-Video 1.7M: This dataset was proposed in [20]. The interaction types include “click” and “unclick”. Each micro-video is represented by a 128-dimensional visual embedding vector of its thumbnail. Each user’s historical interactions are sorted in chronological order.

MovieLens: The MovieLens dataset is obtained from the MovieLens 10M Data[1]. We assume that a user has an interaction with a movie if the user gives it a rating of 4 or 5. We use the pre-trained ResNet[42] models to obtain the visual features from key frames extracted from micro-video. For acoustic modality, we separate audio tracks with FFmpeg6 and adopt VGGish [50] to learn the acoustic deep learning features. For textual modality, we use Sentence2Vector [84] to derive the textual features from micro-videos’ descriptions.

2.6 Methodology

In this thesis, we utilize graph neural networks approaches to represent complex datasets for extracting inductive and aggregation-based semantic features. We build a unimodal graph to cluster similar review text and a multimodal hypergraph to learning user interests.

[1]http://files.grouplens.org/datasets/movielens/

2.6.1 Graph-based Methods

In this section, we introduce the graph related methods that are utilized in this thesis : the graph convolutional network (GCN) and a hypergraph neural networks (HGNN) framework, respectively.

Graph-aware Representation In online spam review detection task, our goal is to learn a graph to model the interaction among similar review source r_i from individuals u_j and apply to different item p_k. Our motivation is that some correlations between reviews with particular semantics can reveal the possibility that the source review is spam. To achieve our objective, a graph $\mathcal{G}_{c_i} = (V_i, E_i)$ is constructed for depicting different review sets with same content i.e., V_i, where E_i is the corresponding edge set. To unify the review text input, the given source review is represented by a word level encoder. The input is the embedding of each word in review text r_i. Due to the difference in length of each review, we perform zero padding appending to the tail by setting a fixed length l. Since the edge set among reviews is unknown, we consider graph-based clustering algorithm to generate relationship R by connecting comments with similar contents i.e., $\forall e_{\alpha\beta} \in \mathcal{E}\left(\mathcal{G}_{C_i}\right), v_\alpha \in R_i, v_\beta \in R_i$, and $v_\alpha \neq v_\beta, |\varepsilon(C_i)| = \frac{k\cdot(k-1)}{2}$, where α and β denote two linked nodes, k denotes the number of v in $\mathcal{G}_{c_i}$ graph, and $\mathcal{E}$ and ε used to denote the graph edge sets. Then, let the affinity $\mathbf{M}$ incorporate the similarity between review node embeddings given by $\mathbf{M}_{C_i}^{(v_\alpha, v_\beta)} = \mathrm{D}\left(\left\{r_\alpha \mid w_1^\alpha, w_2^\alpha, \cdots, w_{e(i)}^\alpha\right\}, \left\{r_\beta \mid w_1^\beta, w_2^\beta, \cdots, w_{e(i)}^\beta\right\}\right)$, where r_α and r_β can be seen as embedding vectors of each review text sequence and D denotes the vector distance. We use matrix $\mathbf{R} = [w_{v\alpha, v\beta}] \in \mathbb{R}^{n\times n}$ to represent the relationship between any pair of nodes v_α and v_β in graph $\mathcal{G}_{c_i}$. After the clustering operation, the propagation features is obtained by GCN-based methods. As mentioned above, GCN can capture information from a node's one-hop and multi-hop neighbours through stacking layer-wise convolution. Given the matrix $\mathbf{R}$ depicting the matrix of relationship for review nodes in graph $\mathcal{G}_{c_i}$, the new d-dimensional node feature matrix $\mathbf{H}_l \in \mathbb{R}^{n\times d}$ represents the output clustering review embeddings: $\mathbf{X}_r$.

$$\mathbf{H}_{\mathbf{N}(v)}^{(l)} = \sigma\left(\tilde{A}\mathbf{H}^{(l-1)}\mathbf{W}^{(l)} \cdot \mathbf{R}_{C_i}^{\mathbf{N}(v)}\right) \tag{2.1}$$

where l is the layer number, $\mathbf{W}^{(l)}$ is a trainable matrix shared among all nodes at layer l. Then, we choose to stack two sub-layers to derive the propagation learning representation, denoted $AGG(.)$ and $CONCAT(.)$. Note here an edge is associated with relationship R, and the hidden state is updated as the concatenation of previous hidden

states of two nodes it links to. So the $AGG(.)$ function can be defined as:

$$\mathbf{h}_{\mathbf{N}(v)}^{(l)} \leftarrow AGG_{(l)}\left(\left\{\mathbf{h}_r^{(l-1)}, \forall r \in \mathbf{N}(v) \mid \mathbf{R}_{C_i}^{\mathbf{N}(v)} = 1\right\}\right) \tag{2.2}$$

After aggregating the neighbors' information, we follow a combination strategy in [40] for the homogeneous graph as:

$$\mathbf{h}_v^{(l)} \leftarrow \sigma\left(\mathbf{W}^{(l)} \cdot CONCAT\left(\mathbf{h}_v^{(l-1)}, \mathbf{h}_{\mathbf{N}(v)}^{(l)}\right)\right) \tag{2.3}$$

Hypergraph Convolution Network (HGCN) In CTR prediction research task, we aim to exploit the correlations among users and items for their high-order rich embeddings, in which the correlated users or items can be more complex than pairwise relationship, which is difficult to be modeled by a graph structure. On the other hand, the data representation tends to be multi-modal, such as the visual, text and social connections. To achieve that, each user should connect with multiple items with various modality attributes, while each item should correlated with several users. This naturally fits the assumption of the hypergraph structure for data modeling. Compared with simple graph, on which the degree for all edges is mandatory to be 2, a hypergraph can encode high-order data correlation using its degree-free hyperedges [33]. In our work, we construct a $\mathcal{G}(u,i)$ to present user-item interactions over different time slots. Then, we aim to distill some hyperedges to build user interest-based hypergraph $\mathcal{G}_g^{t_n}$ and item hypergraph $\mathcal{G}_i^{t_n}$ to aggregate high-order information from all neighborhood. We concatenate the hyperedge groups to generate the hypergraph adjacent matrix $\mathbf{H}$. The hypergraph adjacent matrix $\mathbf{H}$ and the node feature are fed into a convolutional neural network (CNN) to get the node output representations. We build a hyperedge convolutional layer $f(\mathbf{X}, \mathbf{W}, \Theta)$ as follows:

$$\mathbf{X}^{(l+1)} = \sigma\left(\mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^\top\mathbf{D}_v^{-1/2}\mathbf{X}^{(l)}\Theta^{(l)}\right) \tag{2.4}$$

where define $\mathbf{X}, \mathbf{D}_v, \mathbf{D}_e$ and Θ is the signal of hypergraph at l layer, σ denotes the nonlinear activation function. The GNN model is based on the spectral convolution on the hypergraph.

2.6.2 Unsupervised Learning

Unsupervised learning is a popular machine learning methods with less labelled datasets. In this thesis, we first utilize the spectral clustering to build unsupervised learning modules for learning the review data similarity matrix. Further, inspired by the recent

success of self-supervised learning (SSL) [81], we utilize the mutual information maximization principle to learn the intrinsic data correlation [164] to help construct the interests-based hypergraph where we represent a group of users with common preference on modal-specific content.

Graph Clustering Inspired by graph-based clustering approaches, we use relationships from graphs, such as spectral clustering technique [91] to transform the data into a weighted, undirected graph based on pairwise similarities. The graph clustering methods generally build k-means graphs with unlabeled data D_u as input and extract features $F(D_u)$. With these features, they find k-means for each sample D_u using cosine similarity. We obtain two different versions of k-means graphs, which are: 1) The relationship R, between the two nodes. Intuitively, it can be understood as whether two nodes are neighbors in the view of each k-means graph.

$$R_{C_i}^{(n_0,n_1)} = \begin{cases} 1 & \text{if } (n_0,n_1) \in \mathscr{E}\left(\mathscr{G}_{c_i}\right),\ i = 1,2,\ldots,N, \\ 0 & \text{otherwise.} \end{cases} \tag{2.5}$$

where $\mathscr{G}_{c_i}$ denote to the k-means graph of i-th clustering, and $\mathscr{E}$ denotes all edges of a graph. Here, n_0 and n_1 represent two nodes in the graph. 2) The affinity $\mathbf{M}$ is defined as the Euclidean Distance measured in the feature space,

$$\mathbf{M}_{C_i}^{(n_0,n_1)} = \text{Distance}\left(\left\langle F_{C_i}(n_0), F_{C_i}(n_1)\right\rangle\right), \quad i = 1,2,\ldots N \tag{2.6}$$

Here, n_0 and n_1 connected by the affinity vector $\mathbf{M}_{C_i}$ in $\mathscr{C}_i$ clustering graph.

Self-supervised Learning We aim to utilize self-supervised learning for the user-interest matrix $\mathbf{F} \in \mathbb{R}^{L\times d}$, where L denote the user counts and d denote the number of multi-modalities according to items. We trained the weights $\{\theta_a, \theta_b, \theta_c\}$ for each modalities. We define $\{\alpha, \beta, \gamma\}$ to denote the degree of interest of each modalities from the item features. A threshold δ was applied to measure which modality contributes the most for user-item interaction. We first maximize the mutual information between users u and item's multi-modal attributes $M_{i_n}^{t_n}$. For each user and item, the metadata and attributes provide fine-grained information about them. We aim to fuse user and multimodal-level information through modeling user-multimodal correlation. It is thus expected to inject useful multi-modal information into user group representation. Given an item i and the multi-modal attributes embedding matrix $\mathbf{M}_{i_i}^{t_n} \in \mathbb{R}^{|\mathscr{A}|\times d}$, we treat user, item and its associated attributes as three different views denoted as $\mathbf{E}_U$, $\mathbf{E}_I^{t_n}$ and $\mathbf{E}_A^{t_n}$. Each $\mathbf{E}_A^{t_n}$ is

associated with a embedding matrix $M_k \in M_{i_n}^{t_n} = \{v_{i_n}^{t_n}, a_{i_n}^{t_n}, x_{i_n}^{t_n}\}$. We design a loss function by the contrastive learning framework that maximizes the mutual information between the three views. Following Eq 4.8, we minimize the User Interest Prediction (UIP) loss by:

$$L_{UIP}\left(u,i,\mathbf{E}_{A_i}\right) = \mathbb{E}_{a_j \in \mathbf{E}_{A_i}}\left[f\left(u,i,a_j\right) - \log \sum_{\tilde{a} \in \mathbf{E}_A \setminus \mathbf{E}_{A_i}} \exp(f(u,i,\tilde{a}))\right] \tag{2.7}$$

where we sample negative attributes $\tilde{a}$ that enhance the association among user, item and the ground-truth multi-modal attributes, "\" defines set subtraction operation. The function $f(\cdot,\cdot,\cdot)$ is implemented with a simple bilinear network:

$$f\left(u,i,a_j\right) = \sigma\left[\left(\mathbf{E}_I^{\top} \cdot \mathbf{W}_{UIP} \cdot \mathbf{E}_{A_j}\right) \cdot \mathbf{E}_U\right] \tag{2.8}$$

where $\mathbf{W}_{UIP} \in \mathbb{R}^{d \times d}$ is a parameter matrix to learn and $\sigma(.)$ is the sigmoid function. We define the loss function L_{UIP} for a single user, which will can be extended over the user set in a straightforward way. The outcome from $f(.)$ for each user can be constructed as a user-interest matrix $\mathbf{F}$ and compared with the threshold δ to output the L-dimensions vector $\mathbf{v} \in \mathbb{R}^{1 \times L}$.

2.6.3 User behaviour Model

In order to answer the research questions from one to three, we model the online user reviews detection and user CTR prediction with some embeddings and deep learning techniques: hybrid fusion strategy, self-attention methods and multi-layer perceptron.

Hybrid Fusion Strategy Existing works on multiple embeddings research use concatenation as fusion [67], resulting in suboptimal interactions. To tackle multiple types of interactions effectively, we utilize an fusion process that transforms the input representations into a heterogeneous tensor [88]. We use three unimodal information vectors denoted as $\mathbf{X}_r$, $\mathbf{X}_u$ and $\mathbf{X}_p$, according to the encoded representations $\mathbf{H}$, $\mathbf{v}_j$ and $\mathbf{x}_k$ respectively. Each vector $\mathbf{X}$ is augmented with an additional feature of constant values equal to 1, denoted as $\mathbf{X} = (\mathbf{X}, 1)^T$. Then the augmented matrix $\mathbf{X}$ is projected into a multiple dimensional latent vector space by a parameter matrix $\mathbf{W}$, denoted as $\mathbf{W}^T\mathbf{X}_m$. Therefore, each possible multiple feature interaction among review-user-item is computed via outer product, $\mathscr{T} = f\left(\mathbf{W}^T \cdot \tilde{\mathbf{X}}_m\right)$, expressed as:

$$\mathscr{T} = \mathbf{W}^T \cdot \left(\mathbf{X}_r \otimes \mathbf{X}_u \otimes \mathbf{X}_p\right) \tag{2.9}$$

Here $\otimes$ denotes outer product, $\tilde{\mathbf{X}}$ is the input representation from review, user and item level. It is a three-fold heterogeneous tensor, modeling all possible interrelation,

i.e., review graph-aware aggregation features $\mathbf{H}$, and user-item interaction outcome $\mathbf{X}_u$ and $\mathbf{X}_p$. These operations result in two benefits: 1) different from simple concatenation, making use of normal vector among multiple vectors enables learning the different impacts of elements in different modalities, 2) it can also reduce the dimensionality by compressing the fusion feature along three directions.

Self-Attentions We develop the sequential user behavior encoder by utilizing attention mechanism. We proposed to use self-attention layer, i.e., transformer which has also been applied in time series prediction [104]. In constrast to CNN, RNN-based approaches and Markov Chains-based models [60], we adopt self-attention as the basic model to capture the temporal pattern in user-items interaction sequence. A self-attention module generally consists of two sub-layers, i.e., a multi-head self-attention layer and a point-wise feed-forward network. The multi-head self-attention mechanism has been adopted for effectively extracting the information selectively from different representation subspaces [164]. The multi-head self-attention is defined as:

$$MultiHead(Q,K,V) = \text{ Concat }(\text{ head }_1,\ldots,\text{ head }_\text{h})W^O \tag{2.10}$$

$$\text{head }_\text{i} = \text{ Attention }\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{2.11}$$

where the projections are parameter matrices $W_i^Q \in \mathbb{R}^{d\times d_k}$, $W_i^K \in \mathbb{R}^{d\times d_k}$, $W_i^V \in \mathbb{R}^{d\times d_v}$ and $W^O \in \mathbb{R}^{hd_v\times d}$. The attention function is implemented by scaled dot-product operation:

$$Attention(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.12}$$

where $(Q = K = V) = \mathbf{E}$ are the linear transformations of the input embedding matrix, and $\frac{1}{\sqrt{d_k}}$ is the scale factor to avoid large values of the inner product, since the multi-head attention module is mainly build on the linear projections. In addition to attention sub-layers, we applied a fully connected feed-forward network, which contains two linear transformations with a ReLU activation in between.

$$\text{FFN}(x) = ReLU(0, xW_1 + b_1)W_2 + b_2 \tag{2.13}$$

where W_1, b_1, W_2, b_2 are trainable parameters.

Multi-layer Perceptron Prediction Mudules We want to incorporate both user sequential embeddings and group-aware high-order information for a more expressive

representation of each user in the sequence. We propose the fusion layer to generate the representation of user u at t_n. Existing works on multiple embeddings use concatenation as fusion [67], resulting in suboptimal interactions. We utilize the fusion process that transforms the input representations into a heterogeneous tensor [88]. We use the user sequential embedding $\mathbf{E}^{t_n}$ and group-aware hypergraph embedding $\mathbf{E}_g^{t_n}$. Each vector $\mathbf{E}$ is augmented with an additional feature of constant value equal to 1, denoted as $\mathbf{E} = (\mathbf{E}, 1)^T$. The augmented matrix $\mathbf{E}$ is projected into a multi-dimensional latent vector space by a parameter matrix $\mathbf{W}$, denoted as $\mathbf{W}^T\mathbf{E}_m$. Therefore, each possible multiple feature interaction between user and group-level is computed via outer product, $\mathscr{T} = f\left(\mathbf{W}^T \cdot \tilde{\mathbf{E}}_m\right)$, expressed as:

$$\mathscr{T}_U = \mathbf{W}^T \cdot \left(\mathbf{E}^{t_n} \otimes \mathbf{E}_g^{t_n}\right) \tag{2.14}$$

Here $\otimes$ denotes outer product, $\tilde{\mathbf{E}_m}$ is the input representation from user and group level. It is a two-fold heterogeneous user-aspect tensor $\mathscr{T}_U$, modeling all possible interrelation, i.e., user-item sequential outcome embeddings $\mathbf{E}^{t_n}$ and group-aware aggregation features $\mathbf{E}_g^{t_n}$.

When predicting the CTR of user for items, we take both sequential user embedding and item embedding into consideration. We calculate the user-level probability score y to a candidate item i, to clearly show how the function f works. The final estimation for the user CTR prediction probability is calculated as:

$$\hat{y} = f(\boldsymbol{e}_u, \boldsymbol{e}_i; \boldsymbol{\Theta}) \tag{2.15}$$

where $\boldsymbol{e}_u$ and $\boldsymbol{e}_i$ denote user and item-level embeddings, respectively. f is the learned function with parameter $\boldsymbol{\Theta}$ and implemented as a multi-layer deep network with three layers, whose widths are denoted as $\{D_1, D_2, \ldots, D_N\}$ respectively. The first and second layer use $ReLU$ as activation function while the last layer uses sigmoid function as $\text{Sigmoid}(x) = \frac{1}{1+e^{-x}}$. As for the loss function, we take an widely used end-to-end training approach, Cross Entropy Loss[32, 106, 162], and it is formulated as:

$$L(\boldsymbol{e}_u, \boldsymbol{e}_i) = y \log \sigma(f(\boldsymbol{e}_u, \boldsymbol{e}_i)) + (1-y)\log(1 - \sigma(f(\boldsymbol{e}_u, \boldsymbol{e}_i))) \tag{2.16}$$

where $y \in \{0, 1\}$ is ground-truth that indicates whether the user clicks the micro-video or not, and f represents the multi-layer deep network.

CHAPTER 3

GRAPH-AWARE DEEP FUSION NETWORKS FOR ONLINE SPAM REVIEW DETECTION

3.1 Introduction

E-commerce companies such as Amazon and eBay have earned approximately $3.5 trillion in sales in 2019 and are anticipating an increase to $4.9 trillion by the end of 2021, according to Shopify.com. Online e-commerce has demonstrated unique importance during the COVID-19 pandemic and enabled hundreds of millions of consumers to purchase products anytime and anywhere around the world. Currently, customers can also share their shopping experiences by rating items, writing reviews, and answering questions related to the products that they have used in the past or recently purchased online.

Online reviews play an important role in e-commerce as they impact the purchasing decisions of approximately 93% of people, according to Ingniyte.co.uk. Unfortunately, online reviews can be deliberately injected (a.k.a., "spam reviews") to mislead potential customers [58] for various unethical reasons, such as unfair marketing or online brand attacks [71]. According to BrightLocal.com, 74% of consumers in 2019 have encountered spam reviews yet failed to recognize them. It has thus become very crucial to devise effective methods that can identify spam reviews automatically so that these platforms remain reliable [147].

Despite various efforts on automatic spam review detection, most of them largely

rely on learning from engineered features and lack generalizability. For example, traditional statistical learning methods usually use supervised classifiers, e.g., support vector machines [93] (SVM), logistic regression [58], and Naïve Bayes [68], to detect unusual patterns based on extracting review-specific semantic information [156]. Such feature-centric methods usually ignore correlations among reviews, users, and items. As shown in Figure 3.1(a), reviews on Yelp are useful and can be used as a reliable guide for users to make a choice. However, experience tells us that only looking at the review may mislead us into making an unwise decision, and we may need to double-check the information (e.g., credibility, tastes, biases, and beliefs) about reviewers. Similarly, only leveraging review text as features can be problematic as they are sometimes ambiguous, and credibility cannot be guaranteed at all times.

To address the limitations of existing methods, we hypothesize that modeling the information gathered from reviews, users, and items could help substantially improve the performance and generalizability of online spam review detection systems. We thus develop a novel model **Graph-aware Deep Fusion Networks** (**GDFN**), which is capable of capturing the heterogeneity and complex interactions among different features obtained from users, and their reviews on the items. GDFN considers the user-review-item network to formulate the problem as a graph-based classification task, in which reviews are labeled as spam or non-spam. At the local feature space level, GDFN can distill the graph's structural information from different types of features (i.e., user-item bipartite graph, review text graph, user-review graph, and item-review graph). For example, GDFN extracts structured information networks from the original unstructured textual information of review data, which is potentially helpful for learning strong discriminative features in spam review detection. At the global level, GDFN can also learn the macro view of the heterogeneous information network that is aggregated from the extracted individual graphs. GDFN then learns reinforced cross-graph features that depict the useful correlations among all available metadata under a unified framework to detect spam reviews.

Existing methods either concatenate multiple vectors or use selected pooling methods to obtain a fixed dimensional vector. The limitation of such methods is that they may result in information loss, especially under heterogeneous feature scenarios. As a result, we propose a new fusion method to allow flexible information exchange and the interplay between different local views of graph structural information. Instead of applying concatenation of embeddings of various graph views, we adopt the outer product between subgraph-specific embeddings to obtain the fused features. The reason is that the outer

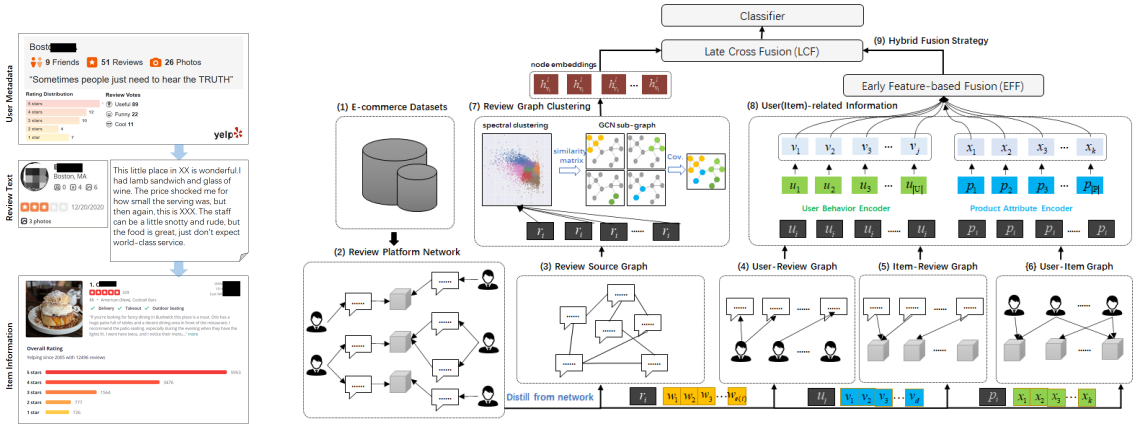

Figure 3.1: **(a):** Yelp Review Platform: the user metadata page (top) and the item information page (bottom). The comment area (middle) with rating and raw review text, then the post review can be treated as a bridge between user and item. **(b)**: Our proposed framework comprises of the following parts: (1) Training datasets from e-commerce platforms. (2) Heterogeneous Network from review platforms. (3)-(6) Review source homogeneous graph and User-Review/Item-Review/User-Item bipartite graph distilled from component (2). (7) Review Graph Clustering. (8) User (Item)-related feature embeddings. (9) Hybrid Fusion Strategy.

product kernel outputs an N-way tensor that favors the strong expressiveness of both lower and higher-order feature interactions. When modeling the above information together, there are several underlying challenges. For instance, a usual representation learning approach is not universal to different graph structures distilled from distinct features. Besides, a single general graph convolutional network (GCN) is not adequate to capture the unique characteristics of different graphs constructed from multiple feature spaces in a complex heterogeneous environment of online review platforms.

In summary, our key contributions are as follows:

- We propose a novel GCN-based heterogeneous graph-aware spam review detection framework that is more expressive than existing text-based methods as it seamlessly captures relevant metadata and relational data to strengthen the review embedding for the underlying task.
- We exploit unsupervised approaches to learn the constructed review graph, which effectively resolves the problem of lack of labeled data. We also develop a novel fusion strategy to model multiple types of interaction information effectively.
- Extensive experiments with large-scale reviews from two real-world datasets demonstrate that our framework achieves consistent improvements over state-

of-the-art methods. Our ablation study demonstrates the effectiveness of novel components of GDFN.

3.2 Related Work

3.2.1 Feature-centric Methods

Traditional statistical methods rely on extracting different features from textual reviews, followed by learning a language model. In [58], the authors first identified three types of spam reviews, which are untruthful opinions, reviews on brands only, and non-reviews, and then analyzed real-world datasets from Amazon. They extracted review-centric, reviewer-centric, and product-centric features, and used them as input to a logistic regression (LR) model. Recently, in [142], the authors summarize eleven platform-independent features from the word level, the semantic level, and the structural level to discriminate between fraud and normal items. They used Xgboost as a binary classifier, and their evaluation results indicated that CATS achieves both high precision and recall. In [93], the authors approached the problem using three strategies as features in Naïve Bayes and SVM classifier. The authors in [135] attempted to use Long Short-Term Memory (LSTM) framework to detect spam reviews. They established three types of layers to predict spam reviews, the input layer for receiving data, the hidden layer of LSTM, and the output layer, respectively.

3.2.2 Graph-based Methods

Graph-based methods have been popularly applied to capture text features among different entities. The first Graph Neural Network [114] (GNN)-based spam review detection method was proposed by [136], who built a heterogeneous "review graph" to represent the relationship among reviewers, reviews, and online sellers. In [117], the authors utilized spam features for modeling review datasets as heterogeneous information networks to map spam review detection procedure into a classification problem in such networks. In the classification step, they proposed meta-path concepts to find feature importance and calculate the weight. The authors in [82] presented a neural network-based graph model, named Graph Embeddings for Malicious accounts (GEM), which both considered "device aggregation" and "activity aggregation" in heterogeneous graphs. So far, these methods have focused on shallow encoders, i.e., matrix factorization. There is no parameter sharing, and every node has its unique embedding vector and

the inherent "transductive" features are impossible to generate embeddings for unseen nodes during training and do not incorporate node features.

Recent years have witnessed a growing interest in utilizing the "message-passing" methodology in graphs [163], which learns how to aggregate information from each type of neighbor using Markov Random Field (MRF) techniques implicitly. In [40], the authors presented the GraphSAGE model, which achieves significant improvements compared with previous methods such as DeepWalk [98] and SemiGCN [64]. This method overcomes the limitation of applying GCN in transductive settings with a specified Laplacian matrix. A model-based Graph Convolutional Neutral Networks (GCNN) for spam-bot detection is proposed in [1], which proposed an inductive representation learning approach for spam review detection based on the reviewer profile information and the social network graph on Twitter datasets, and the inductive representation learning method used in their approach is similar to that of GraphSAGE. In short, GCN-based methods have been applied in various domains, such as spam advertisement identification [67], recommendation system [43, 44, 154], social spammer detection [146], rumor detection [8] and so no.

The above methods depend only on the local information of surrounding neighborhoods of a target node, making the model sometimes noisy and thus ineffective. The multiple convolutional layers may cause an over-fitting and over-smoothing problem. To overcome the limitations inherent in existing methods, we design a novel model that exploits review-user-item three-fold information and distillsreview clustering and user-item level information.

3.2.3 Preliminaries

In this section, we introduce some fundamental concepts that are necessary to understand our model. The notations used in this paper are summarized in Table 1.

3.2.3.1 Graph Convolutional Networks (GCN)

Recently, there is an increasing interest in utilizing convolutions in graph-based methods. GCN is one of the most effective graph-aware models, whose convolution operation is considered as a general layer-wise propagation architecture as follows:

$$\mathbf{H}^{(l+1)} = \sigma(\tilde{\mathbf{A}}^{(l)}\mathbf{W}^{(l)}) \tag{3.1}$$

The input is an adjacency matrix $\mathbf{A}$ and a feature matrix $\mathbf{W} \in \mathbb{R}^{N \times E}$, where $\tilde{\mathbf{A}} = \tilde{D}^{-\frac{1}{2}}\mathbf{A}\tilde{D}^{-\frac{1}{2}}$, $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$ is the adjacency matrix of graph $\mathscr{G}$ with added self-connections and $\tilde{\mathbf{D}}_{ii} =$

Table 3.1: Notations

Notation	Description
R	The set of review source, $\{r_1, r_2, \cdots, r_{\|R\|}\}$
U	The set of users, $\{u_1, u_2, \cdots, u_{\|U\|}\}$
P	The set of items, $\{p_1, p_2, \cdots, p_{\|P\|}\}$
r_i	e_i words $\{w_1^i, w_2^i, \cdots, w_{e_i}^i\}$,$r_i \in R$
Y_i	The tuple formula, denoted as $\{u_j, r_i, p_k\}$
$\mathcal{G}_i$	A undirected graph of each review cluster
$< V_i, E >$	The node and edge set of $\mathcal{G}_i$
y_i	The ground-truth label, $y_i \in \{Y, N\}$
$\mathcal{T}$	The outcome fusion tensor
$f(.)$	The classifier function

$\sum_j \tilde{\mathbf{A}}_{ij}$. σ is a non-linear activation function, such as the $ReLU(.) = max(0, \neg\Sigma)$. In [64], a propagation structure is proposed that can be separated into two components: aggregation and combination. In general, for a GCN with L layer, aggregation and combination sub-layers at l^{th} layer ($l = 1, \cdots, L$) can be written as:

$$\mathbf{H}_{N(v)}^{(l)} = \sigma\left(\mathbf{W}^{(l)} \cdot AGG\left(\left\{\mathbf{H}_v^{(l-1)}, \forall v \in \mathbf{N}(v)\right\}\right)\right) \tag{3.2}$$

$$\mathbf{H}_v^{(l)} = CONCAT\left(\mathbf{H}_v^{(l-1)}, \mathbf{H}_{\mathbf{N}(v)}^{(l)}\right) \tag{3.3}$$

where $\mathbf{N}(v)$ is a set of nodes adjacent to $\mathbf{v}$, $AGG(.)$ is a function used for aggregating embeddings from neighbors of node $\mathbf{v}$. This function can be customized for specific models, e.g., mean aggregator, LSTM aggregator and pooling aggregator. The notation $\mathbf{H}_{\mathbf{N}(v)}^{(l)}$ denotes the aggregated feature of node $\mathbf{v}$'s neighborhood at l^{th} layer. $CONCAT(.)$ function is used to combine self embedding and the aggregated embeddings of neighbors, which is also a customized setup for different graph models, e.g., concatenation as in GraphSAGE [40].

3.2.3.2 Graph-based Clustering

Inspired by graph-based clustering approaches, we use relationships from graphs, such as the spectral clustering technique [91] to transform the data into a weighted, undirected graph based on pairwise similarities. The graph clustering methods generally build k-means graphs with unlabeled data D_u as input and extract features $F(D_u)$. With these features, they find k-means for each sample D_u using cosine similarity. We develop two different versions of k-means graphs, which are:

- The relationship R between the two nodes. Intuitively, it can be understood as whether two nodes are neighbors in the view of each k-means graph.

$$R_{C_i}^{(n_0,n_1)} = \begin{cases} 1 & \text{if } (n_0,n_1) \in \mathscr{E}\left(\mathscr{G}_{c_i}\right),\ i = 1,2,\ldots,N, \\ 0 & \text{otherwise.} \end{cases} \tag{3.4}$$

  where $\mathscr{G}_{c_i}$ denote to the k-means graph of i-th clustering, and $\mathscr{E}$ denotes all edges of a graph. Here, n_0 and n_1 represent two nodes in the graph.

- The affinity $\mathbf{M}$ is defined as the Euclidean Distance (denoted as Dist) measured in the feature space,

$$\mathbf{M}_{C_i}^{(n_0,n_1)} = \text{Dist}\left(\left\langle F_{C_i}(n_0), F_{C_i}(n_1)\right\rangle\right), i = 1,2,\ldots N \tag{3.5}$$

  Here, n_0 and n_1 are connected by the affinity vector $\mathbf{M}_{C_i}$ in $\mathscr{C}_i$ clustering graph.

3.3 Methodology

In a nutshell, in our model, well-tailored representation learning models for each subgraph are elaborated to preserve the uniqueness of the derived features. We first utilize spectral clustering to build unsupervised learning modules for learning the review data similarity matrix. A multi-layer convolutional neural network is constructed to capture information from similar neighborhoods of a node, where the convolutions are defined on a graph structure. We then employ a hybrid fusion strategy [5] to obtain discrete values from user behavior and item attribute information. Specifically, we first adopt the idea of "early feature-level fusion" to exploit latent relation among attributes, then apply a "late cross fusion" method to exploit the correlation and interaction among processed modalities.

3.3.1 Model Overview

An online review instance is defined as an ensemble representing three types of information $A = \{R, U, P\}$, where R is a set of review text, U is the user metadata and profiles, and P is the corresponding item attributes. By leveraging multi-level features to obtain a fusion tensor $\mathscr{T}$, we build a classifier to learn the mapping relation from input tensor to output prediction labels. Our novel model Graph-aware Deep Fusion Networks (GDFN), as illustrated in Figure 3.1(b), automatically predicts spam reviews based on a

unimodal graph to cluster similar review texts for extracting aggregation-based semantic features. We then encode user (item)-level information to strengthen the final tensor representation. By modeling each level of information in A using relatively independent processes, the output of each encoder becomes the specific individual embeddings. The graph-aware representation learns semantic correlations from the cluster network and aggregates neighbor information from multiple sub-graphs. The fusion module is to explicitly model interactions among reviews, users and items, denoted by fusion tensor $\mathcal{T}$, including three types of combinations: shallow-level (review text only), medium-level (two-dimensional matrix) and top-level (three-dimensional tensor). The fused tensor is fed into fully-connected layers with a softmax layer to perform review classification.

3.3.2 Graph-aware Representation

Our goal is to learn a novel graph to model the interaction among similar review source r_i from individuals u_j and apply it to a different item p_k. Our motivation is that some correlations between reviews with particular semantics can reveal the possibility that the source review is spam.

To achieve our objective, a graph $\mathcal{G}_{c_i} = (V_i, E_i)$ is constructed for depicting different review sets with same content i.e., V_i, where E_i is the corresponding edge set. To unify the review text input, the given source review is represented by a word-level encoder. The input is the embedding of each word in review text r_i. Due to the difference in length of each review, we perform zero-padding, appending to the tail by setting a fixed length l. Since the edge set among reviews is unknown, we consider a graph-based clustering algorithm to generate relationship R by connecting comments with similar contents. We depict this in the following two equations:

$$\forall e_{\alpha\beta} \in \mathcal{E}\left(\mathcal{G}_{C_i}\right), v_\alpha \in R_i, v_\beta \in R_i. \tag{3.6}$$

and,

$$v_\alpha \neq v_\beta, |\varepsilon(C_i)| = \frac{k \cdot (k-1)}{2} \tag{3.7}$$

where α and β denote two linked nodes, k denotes the number of v in $\mathcal{G}_{c_i}$ graph, and $\mathcal{E}$ and ε used to denote the graph edge sets. Let the affinity $\mathbf{M}$ incorporate the similarity between review node embeddings given by the following equation:

$$\mathbf{M}_{C_i}^{(v_\alpha, v_\beta)} = \mathrm{D}\left(\left\{r_\alpha \mid w_1^\alpha, w_2^\alpha, \cdots, w_{e(i)}^\alpha\right\}, \left\{r_\beta \mid w_1^\beta, w_2^\beta, \cdots, w_{e(i)}^\beta\right\}\right) \tag{3.8}$$

where r_α and r_β are seen as embedding vectors of each review text sequence and D denotes the vector distance. We use matrix $\mathbf{R} = [w_{v\alpha, v\beta}] \in \mathbb{R}^{n \times n}$ to represent the relationship between any pair of nodes v_α and v_β in graph $\mathcal{G}_{c_i}$.

After the clustering operation, the propagation features are obtained by GCN-based methods. As mentioned above, GCN can capture information from a node's one-hop and multi-hop neighbors through stacking layer-wise convolution. Given the matrix $\mathbf{R}$ depicting the matrix of relationship for review nodes in graph $\mathcal{G}_{c_i}$, the new d-dimensional node feature matrix $\mathbf{H}_l \in \mathbb{R}^{n \times d}$ represents the output clustering review embeddings: $\mathbf{X}_r$.

$$\mathbf{H}_{\mathbf{N}(v)}^{(l)} = \sigma\left(\tilde{A}\mathbf{H}^{(l-1)}\mathbf{W}^{(l)} \cdot \mathbf{R}_{C_i}^{\mathbf{N}(v)}\right) \tag{3.9}$$

where l is the layer number, $\mathbf{W}^{(l)}$ is a trainable matrix shared among all nodes at layer l. We then choose to stack two sub-layers to derive the propagation learning representation denoted $AGG(.)$ and $CONCAT(.)$. An edge is associated with relationship R and the hidden state is updated as the concatenation of previous hidden states of the two nodes it links to. As a result, the $AGG(.)$ function can be written as:

$$\mathbf{h}_{\mathbf{N}(v)}^{(l)} \leftarrow AGG_{(l)}\left(\left\{\mathbf{h}_r^{(l-1)}, \forall r \in \mathbf{N}(v) \mid \mathbf{R}_{C_i}^{\mathbf{N}(v)} = 1\right\}\right) \tag{3.10}$$

After aggregating the neighbors' information, we follow a combination strategy described in [40] for the homogeneous graph as shown below:

$$\mathbf{h}_v^{(l)} \leftarrow \sigma\left(\mathbf{W}^{(l)} \cdot CONCAT\left(\mathbf{h}_v^{(l-1)}, \mathbf{h}_{\mathbf{N}(v)}^{(l)}\right)\right) \tag{3.11}$$

3.3.3 User (Item)-related Information

User (item)-related information has been popularly used in the past [105, 117, 140], where crucial social characteristic features have been used with faithful performance. For example, given more metadata and attributes about the user and item level, the model will focus on the balanced arbitration if posts are with positive or negative emotions. We extract three types of objective features including account-based features and transduction-based features.

User Metadata To depict user behavior features, we use their metadata and profiles and define a feature vector $\mathbf{v}_j$ for each user $\mathbf{u}_j$. These features have also been used in [105]. These features are:

- name of u_j registered on the website
- date when u_j joined, the number of u_j's friends
- number of times u_j has posted reviews
- number of cool/funny/useful review posted by u_j
- location of u_j

Each user feature vector $\mathbf{v}_j \in \mathbb{R}^d$ is generated, where d is the number of features. It is known that users' behavior is crucial in detecting spam reviews, e.g., the average rating given by reviewer, the standard deviation in rating and a feature indicating whether the reviewer always gave only good, average or low rating [21].

Item Attributes To exploit item level features, we collect abundant attribute relativity information from an online review website (i.e., Yelp) to identify item vector $\mathbf{x}_k$. The collected attributes are listed as follows:

- number of reviews written for p_k
- average rating deviation of p_k
- which categories p_k belongs to
- location of p_k
- ratio of positive reviews against negative reviews on p_k

For each item attribute, we map all discrete values into the Gaussian space and represent them as the vector $\mathbf{x}_k \in \mathbb{R}^m$ based on the three-sigma rule to avoid the sparsity problem [13, 100].

Transduction We consider attributes related to the transductive pattern of datasets, such as the average number of comments or words. In most cases, spam reviews are propagated in several fixed patterns [21]. Therefore, we use some useful data, such as the average length of all reviews posted by u_j or the average sentiment score of each p_k. Eventually, we utilize the strategy of decision-based operation [4] that unimodal feature portions will be more predictive by a pre-trained model, as it can project the raw features into a specialized embedding space [4]. To extract the unique information from individual raw data fields, we employ Factorization Machine (FM) [109] to tackle the problem of sparse data. As a result, the latent relevance among varying user-item behavior and attributes is encoded in the embedding vector with linear complexity.

3.3.4 Fusion Module

Existing works on multiple embeddings research use concatenation as fusion [67], resulting in suboptimal interactions. To tackle multiple types of interactions effectively, we utilize an fusion process that transforms the input representations into a heterogeneous tensor [88]. We use three unimodal information vectors denoted as $\mathbf{X}_r$, $\mathbf{X}_u$ and $\mathbf{X}_p$, according to the encoded representations $\mathbf{H}$, $\mathbf{v}_j$ and $\mathbf{x}_k$, respectively. Each vector $\mathbf{X}$ is augmented with an additional feature of constant values equal to 1, denoted as $\mathbf{X} = (\mathbf{X}, 1)^T$. The augmented matrix $\mathbf{X}$ is then projected into a multiple dimensional latent vector space by a parameter matrix $\mathbf{W}$, denoted as $\mathbf{W}^T\mathbf{X}_m$. Therefore, each possible multiple feature interaction among review-user-item is computed via outer product, $\mathcal{T} = f\left(\mathbf{W}^T \cdot \tilde{\mathbf{X}}_m\right)$, expressed as:

$$\mathcal{T} = \mathbf{W}^T \cdot \left(\mathbf{X}_r \otimes \mathbf{X}_u \otimes \mathbf{X}_p\right) \tag{3.12}$$

where $\otimes$ denotes the outer product, $\tilde{\mathbf{X}}$ is the input representation from review, user, and item level. It is a three-fold heterogeneous tensor, modeling all possible interrelations, i.e., review graph-aware aggregation features $\mathbf{H}$, and user-item interaction outcome $\mathbf{X}_u$ and $\mathbf{X}_p$. These operations result in following two benefits:

- different from simple concatenation, making use of feature vector among multiple vectors enables learning the different impacts of elements in different modalities
- reducing the dimensionality by compressing the fusion feature along with at least three directions.

3.3.5 Classification Model

We have obtained the graph-aware representation from review clustering networks, user-level behavior, and item-level attribute embeddings. Each review with these modalities can be represented as a heterogeneous tensor $\mathscr{T}$ with multiple sets. One of the advantages of the fusion model $\mathscr{T}$ is that it can tackle the missing information problem in the absence of one or two modalities. We use the heterogeneous tensor $\mathscr{T}$ as feature to detect spam reviews. The fully connected layers are applied over $\mathscr{T}$, and the $Softmax(.)$ function is used to convert the output values into probabilities which is commonly done in the literature.

$$\hat{y} = \text{Softmax}\left(\text{Fusion}\left(\mathbf{H}_{C_i}^{k}, \mathbf{X}_u, \mathbf{X}_p\right)\right) \tag{3.13}$$

where $\hat{y} \in \mathbb{R}^{1\times c}$ is the vector of probabilities for all the classes used to predict the labels of the reviews. Here we apply two-class for our detection task. We then train all the parameters in the GDFN models by choosing the cross-entropy loss as the objective function to optimize the classification task. The overall loss is the weighted sum of classification loss.

3.3.6 GDFN Algorithm

We provide a detailed description of GDFN approach in Algorithm 1.

3.4 Experiments and Results

In this section, we evaluate the performance of our proposed GDFN model and compare our model with different strong comparative methods. By conducting ablation study, we demonstrate the preformance of the key components of our model.

3.4.1 Datasets

We evaluate our proposed method on two benchmark publicly available datasets. They are:

- **Yelp** from [105] (Table 5.1), which contains three public spam review datasets crawled from the Yelp website: YelpChi, YelpNYC, and YelpZip. The dataset com-

Algorithm 1 GDFN training Algorithm

Data: Review Source r_i, User metadata u_j and Item attributes p_k
Result: Prediction Label $\hat{y}$ (Train a fixed number of epochs on the initial labeled and unlabeled sets R,U and P)

for each stage k **do**
  Step 1: Review Deep Clustering
  Execute K-means and Laplacian calculation based on review source word
  embedding r_i and obtain affinity matrix $\mathbf{M}$, relation $\mathbf{R}$ and clustering graph $\mathscr{G}$.
  Step 2: Graph Convolutional Networks
  Compute each review node v of each review cluster C_i in unlabeled data.
  for each cluster C of unlabeled set **do**
    $AGG_{(l)}\left(\left\{\mathbf{h}_r^{(l-1)}, \forall r \in \mathbf{N}(v) \mid \mathbf{R}_{C_i}^{\mathbf{N}(v)} = 1\right\}\right)$
    $\mathbf{W}^{(l)} \cdot CONCAT\left(\mathbf{h}_v^{(l-1)}, \mathbf{h}_{\mathbf{N}(v)}^{(l)}\right)$
  end for
  Step 3: User(Item) Information
  Compute user metadata vector $\mathbf{u_j}$
  Compute user metadata vector $\mathbf{p_k}$
  Early Feature-level Fusion: $\{\mathbf{u_j}, \mathbf{p_k}\} \rightarrow \{\mathbf{x_j}, \mathbf{x_k}\}$
  Step 4: Late Cross Fusion
  Calculate Fusion Tensor:
  for each r_i of all cluster data **do**
    $\mathscr{T}_i = \mathbf{W}^T \cdot \left(\mathbf{x}_r \otimes \mathbf{x}_u \otimes \mathbf{x}_p\right)$
  end for
  Step 5: Classification
  Train a fixed number of epochs on the labeled spam review datasets R.
end for
return Prediction label and Accuracy based on Tensor $\mathscr{T}$

prises binary labels: N representing genuine reviews and Y representing spam reviews.

- **Op_spam_v1.4** from [21] (Table 5.1), consists of truthful and deceptive hotel reviews of 20 Chicago hotels. The label of each review in Op_spam_v1.4 was gathered from Amazon's popular Mechanical Turk crowdsourcing service and five popular online review communities: Expedia, Hotels.com, Orbitz, Priceline, and TripAdvisor. Note that reviewer features are not available for the Op_spam_v1.4 dataset.

Table 3.2: Dataset statistics.

Dataset	Yelp			Op Spam	
	CHI	NYC	ZIP	Positive	Negative
#Users	38,063	160,225	260,277	-	-
#Products	201	923	5044	20	20
#Spam Reviews	8,919	36,885	80,466	400	400
#Non-spam Reviews	58,477	322,167	528,133	400	400
%Spam	13.23%	10.27%	13.22%	-	-

Table 3.3: Spam detection results on whole Yelp and OpSpam Datasets in %. (Bold indicates improvement over 10%)

Method	YelpCHI		YelpNYC		YelpZIP		Positive OpSpam			Negative OpSpam		
	AP	AUC	AP	AUC	AP	AUC	Prec.	Rec.	F1	Prec.	Rec.	F1
SVM+Ngram+BF	36.12	69.97	51.47	71.76	52.11	64.87	56.68	68.01	61.83	75.18	58.72	65.94
SpEagle	32.36	78.87	24.60	76.95	33.19	79.42	71.41	53.61	52.18	64.53	75.77	57.40
CATS	58.51	74.43	59.37	75.72	53.53	73.77	62.46	78.51	69.57	60.50	83.17	70.05
NB+Ngram	70.89	71.41	67.88	60.90	66.81	61.11	72.57	76.17	74.33	76.91	75.95	76.42
CNN	65.32	75.91	63.34	76.18	62.25	76.67	73.73	78.80	67.54	62.12	75.13	65.72
HFAN	48.87	83.24	53.82	84.78	62.35	87.28	86.96	67.31	75.88	61.17	40.00	48.37
GAS	68.90	71.02	70.09	71.67	67.02	60.00	88.65	84.61	81.53	88.60	84.87	81.63
GDFN	81.35	85.35	81.78	86.42	80.24	87.67	88.67	90.45	90.28	88.72	93.78	90.18
GDFN (+BERT)	**82.39**	87.69	**82.46**	87.85	**82.91**	88.05	88.75	92.83	**90.75**	89.82	**94.90**	**91.62**
Improvement(%)	16.22	5.34	17.65	3.62	23.70	0.88	0.11	9.72	11.30	1.38	11.82	12.24

3.4.2 Baseline Models and Settings

We compare our proposed method, GDFN, with strong state-of-the-art baseline methods, including feature-centric and some recently proposed network-based models for spam review detection. The comparative models are:

- **NB** [68]: A naive Bayes classifier [112] based on four groups of features: content features, sentiment features, product features and meta data features
- **SVM+Ngram+BF** [93]: A standard n-gram (n=3) text categorization technique applied to detect negative deceptive opinion spam with SVM classifier
- **SpEagle** [105]: A pair-wise Markov Random Field model defined to tackle spam review detection task that utilized clues from metadata as well as relation data
- **CNN** [140]: A CNN method adopted to learn the textual information, and capture complex global semantic features for detecting spam reviews
- **CATS** [142]: A Xgboost [19] model as the classifier in the detector with multiple cross-platform independent features

- **HFAN** [156]: A Hierarchical Fusion Attention Network (HFAN) to automatically learn the semantics of reviews from user and product attribute
- **GAS** [67]: An end-to-end GCN-based Anti Spam (GAS) algorithm which incorporates the local context and the global context of comments with TextCNN classifier [62] to detect spam advertisements

We also use the pre-trained BERT-base model to exploit the information encoded in these pre-trained language models. We name this methods asGDFN (+BERT). To this end, we use the BERT-base multilingual cased pre-trained BERT model [1], which contains 104 languages, 12-layer, 768-hidden, 12-heads, 110M parameters. Since most of the review text contains multiple sentences, we use BERT-as-service [2] as a sentence encoding service, i.e., mapping a variable-length sentence to a fixed-length vector.

To compare our method with the traditional review mining methods, we have used commonly used evaluation metrics for this task, such as Average Precision (AP), Area Under Curve (AUC), Precision (Prec.), Recall (Rec.), and F1 measure (F_1). Specifically, for Yelp full datasets, AP and AUC are used as evaluation metrics. For Op_spam_v1.4 datasets, we evaluate Prec., Rec., and F_1 scores over two categories: negative and positive, respectively. For a fair comparison, we apply datasets with abundant metadata and profiles including conducting a five-fold cross-validation.

Data Preparation Most of our pre-processing strategy has been widely used in the literature [105, 156]. The maximum length of reviews in Yelp full datasets is set to 200, and for Op_spam_v1.4 dataset, the maximum length is set to 100. We also compute some additional features which usually have been shown to improve performance. These are listed as below:

- Word Count of the documents – total number of words in the documents
- Character Count of the documents – total number of characters in the documents
- Average Word Density of the documents – average length of the words used in the documents
- Puncutation Count in the Complete Essay – total number of punctuation marks in the documents

[1] https://storage.googleapis.com/BERT_models

[2] https://github.com/hanxiao/BERT-as-service#1-download-a-pre-trained-BERT-model

- Upper Case Count in the Complete Essay – total number of upper count words in the documents
- Title Word Count in the Complete Essay – total number of proper case (title) words in the documents
- Frequency distribution of Part of Speech Tags: Noun Count, Verb Count, Adjective Count, Adverb Count and Pronoun Count.

These features are applied as source input to the model during the training process.

Model Training In the feature-based baselines, we make use of text and label. Review text is transformed into feature vectors. Each word is first represented by a 300-dimensional GloVe[3] [97] embedding of the word. For the CNN-based model, we configure 200 hidden layers and "mean" aggregation operation. Moreover, the rate of dropout is 0.25, and the training iterations are set to 200 epochs, with early stopping when the validation loss stops decreasing by 20 epochs. In the training process of the GCN-based method, the dropout rate is 0.5, L_2 loss regression is $2.5e-4$. In our model training, we adopt unsupervised learning for the clustering module and convolutional operation for the GCN-based module.

For the clustering module, we select the top 10 clusters of the unsupervised learning of review text, to make enough nodes for each review cluster (here we set up the minimum node number of each cluster as 200). We provide a visualization for the distribution of the review clustering graph in the embedding space where the figure illustrates the embedding space learned by the spectral clustering method. In Figure 3.2, we represent the top 10 clusters by unsupervised learning from the review source. Moreover, we utilize k-means as our clustering method and compute the symmetric normalized Laplacian.

This visualization is conducted to prove the similarity of the review source. As shown in Figure 3.4.2, for unimodal interactions, obviously review text modality is the most predictive for the majority of samples, which is reasonable since the content is the most important clue for spam review analysis. Furthermore, we have found that a small defined number of clusters may increase the computational complexity of graph construction and then lead to a lower clustering precision, while a large number of clusters do not show the difference among clusters reliably, the cluster boundaries are not very distinctive.

[3]https://nlp.stanford.edu/projects/glove/

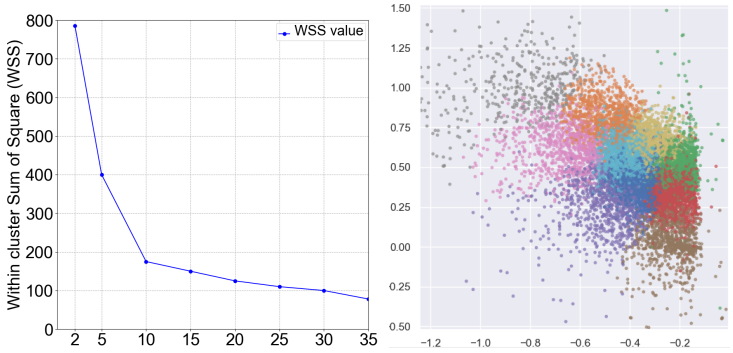

Figure 3.2: **(a):** Compactness measure via WSS. **(b):** Visualization for distributions of 10 clusters in learnt embedding space. The different colors dots represent different review text clusters.

We utilize **W**ithin cluster **S**um of **S**quare (WSS) [27] technique as our metric for deciding the number of clusters. As shown in Figure 3.4.2, we notice that our WSS measure drops considerably when the number of clusters is increased from 2 to 5, and again from 5 to 10, but the performance drop is comparatively lesser. Once we reach 10 clusters, the algorithm generally finds reliable groupings.

3.4.3 Results

Our experimental results are reported in Table 3.3. We can see that the graph-based methods outperform feature engineering methods since the graph-based methods better capture intricate representations of spam reviews. They are also suitable to capture generalized features and interaction among multiple modalities.

Our method outperforms GraphSAGE [40] model, GAS [67], which justifies the advantage of combining graph structure and hybrid fusion strategy for spam detection. Additionally, CNN-based method cannot capture data with the graph structure, whereas HFAN, the hierarchical fusion network ignores important propagation features for unseen data prediction. This shows that obtaining graph structure information and fusion strategy separately, results in lower performance on spam detection.

CNN only uses the convolutional hidden layer to capture feature vectors from Euclidean structure data so it is dependent on data samples. However, the review platform is similar to a social network. Unlike the CNN, GCNs enable the proposed model to pay more attention to the non-Euclidean structural information of the review posts, which helps improve our model's performance. Further, the experiment result of the

Table 3.4: Results of ablation study of GDFN on spam detection performance (Average Precision in %).

GDFN variants	YelpCHI	YelpNYC	YelpZIP
GDFN_{ur}	75.32	73.25	76.42
GDFN_{ir}	70.49	71.59	74.50
GDFN_{ro}	69.01	67.18	69.55
GDFN(+BERT)_{ur}	78.25	78.28	79.02
GDFN(+BERT)_{ir}	74.21	73.80	78.42
GDFN(+BERT)_{ro}	73.48	68.55	73.21

GCN-based model, GAS, has shown a significant fluctuation on different datasets, which makes it less ideal and overfits in case of some input samples, e.g., GAS obtains a better performance on OpSpam positive datasets. The proposed fusion strategy fuses extra information from user-item level to influence the final prediction, which helps us get a relatively stable result.

3.4.4 Ablation Study

To analyze the effect of the individual components of GDFN, we conduct an ablation study where we consider three different components: GDFN_{ur} which includes user-review text only, GDFN_{ir} which includes item-review only, and GDFN_{ro} which includes review text without user and item information.

As is shown in Table 3.4, we have observed that GDFN_{ur}, GDFN_{ir} and GDFN_{ro} cannot outperform our main model (results in Table 3.3. Meanwhile, GDFN_{ur}'s performance is close to that of GDFN, demonstrating that user-level information plays an important role in spam detection. We also observe that the worst results obtained from the variant GDFN_{ro}, but these results are still better than most of the other baseline methods, showing the superiority of our proposed framework for spam review detection.

3.5 Conclusions and Future Work

In this chapter, we proposed a novel model named GDFN, to predict spam reviews based on a unimodal graph to cluster similar review text for extracting aggregation-based semantic features and then encode user (item)-level information to strengthen the final representation. We also utilize the fusion mechanism to obtain the inherent relationship among users, reviews, and items. To evaluate the performance of GDFN, we conducted a

series of experiments on two public datasets, to demonstrate the superiority of the model in comparison with state-of-the-art models.

Given the recent success of multimedia sharing platforms, the items posted on these online social media websites contain rich multimedia information (e.g., visual and acoustic). Exploiting these multi-modality features is an interesting future direction. Moreover, data connections can be more complex than a pairwise relationships on the social networks. Addressing this problem in hypergraph networks can be considered a new research line in this field.

CHAPTER 4

CLICK-THROUGH RATE PREDICTION WITH MULTI-MODAL HYPERGRAPHS

4.1 Introduction

Click-Through Rate (CTR) prediction has become one of the core components of modern advertising on many e-commerce platforms. The goal is to predict customers' click probability on wide range of items. Existing works on CTR prediction only focus on modeling pairwise interactions from uni-modal features which might not lead to satisfactory results. This existing gap leads to new opportunities where we can exploit the widely available multi-modal features which is largely unexplored. Besides, they can give complementary information to the model which alone cannot be obtained via uni-modal modeling. AutoFIS [78] and UBR4CTR [101] are recent Factorization Machine (FM) [109] based models with multi-layer perceptron (MLP) which mainly utilize user-item interactions features. To supplement the lack of additional information, deep neural networks (DNNs) are also explored with automated feature engineering. For example, DSTN [94] leverages DNNs-based method to fuse additional auxiliary data and item information to further uncover hidden information. Although these representative works have achieved good performance, there are still limited exploration on modeling multi-modal features and how they could contribute towards the model performance.

Recently, the wide-spreading influence of micro-video sharing platforms, e.g., Tiktok [1]

[1] https://www.tiktok.com/

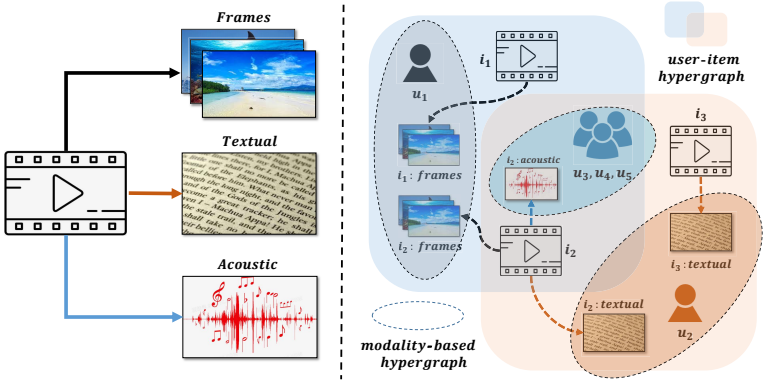

Figure 4.1: An illustration of multi-modal user preferences.

and Kuaishou [2] make them a popular platform for socialising, sharing and advertising as micro-videos. These videos are compact and come with rich multimedia content from multiple modalities, i.e., textual, visual, as well as acoustic information. Motivated by this, we propose a novel method which addresses the limitations in current methods and improve CTR prediction performance through micro-videos. However, modeling multi-modal features extracted from micro-videos for CTR prediction in a holistic way is not straightforward. First, in a typical setting of CTR prediction, the interactions between users and items are normally sparse, and the sparsity issue becomes even more severe (in magnitude of number of modalities) when taking into account multi-modal features. For example, compared to uni-modal feature space, the sparsity of a dataset is tripled when considering visual, acoustic and text features of a target item. Therefore, effectively mitigating the sparsity issues introduced by multi-modal features without compromising upon the performance of the model is the key to this problem.

We rely on hypergraphs to address some of the challenges. Hypergraph [10, 126] extends the concept of an edge in a graph and can connect more than two nodes. Inspired by the flexibility and expressiveness of hypergraphs, we use the concept to multi-modal feature modeling, and propose a new model based on modality-originated hypergraphs by which the sparsity issues between users and items under each modality can be alleviated. Figure 4.1 is an example of the proposed modality-originated hypergraphs, where user u_1 and user u_2 both have interactions with multiple micro-videos, e.g., i_1 and i_2, in which each hyperedge can connect multiple item nodes on a single edge. Compared with a simple graph on which the degree of all edges is set to be 2, a hypergraph can encode high-

[2] https://www.kuaishou.com/

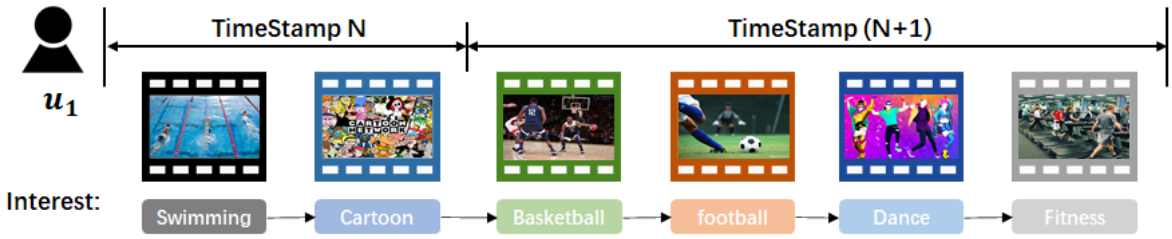

Figure 4.2: Illustration of user u_1's historical view records with micro-videos, which reflects the user's global view interests.

order data correlation (beyond pairwise connections) using its degree-free hyperedges. Different from various modalities, we incorporate different multi-modal information, e.g., frames, acoustic, textual into user-item hypergraphs to help establish an in-depth understanding of user preferences. The reason for considering using hypergraphs in our work is due to the purpose of building modality-originated hypergraphs which can be treated as data argumentation technique.

We also construct hypergraphs considering both user and item. In Figure 4.1, user u_1 cares more about frames of micro-video i_2, whereas user u_2 might be fond of the text content. Hence, different users might have different tastes on modalities of a micro-video. A group of users u_3, u_4 and u_5 click micro-video i_2 due to the intriguing sound tracks. Such signals can be utilized to construct a group-aware hypergraph which is comprised of multiple users who share the same interest for the item. Inspired by the recent success of self-supervised learning (SSL) [81], we utilize the mutual information maximization principle to learn the intrinsic data correlation [164] to help construct the interests-based hypergraph where we represent a group of users with common preference on modal-specific content. Hence, in each modality (e.g., visual), we aggregate information from the group-aware hypergraph and incorporate them into user representations. According to group-aware hypergraph, each user has interactions with one of the item's modalities, while different items can be interacted with the same user. For example, user u_1 likes i_1's frames, and u_1 will pay more attention to the visual-aspect of other items. Under such circumstances, we can also construct a homogeneous item-level hypergraph comprising of multiple items who have certain potential modality that appeal to the same user.

Generally, user preference evolves over time, and it is hence a sequential phenomenon. As shown in Figure 6.1, user u_1 has watched swimming and cartoon videos at timestamp N, indicating that the user has two very different interests and we cannot capture

the user's interests at the single time point. If at a new timestamp $N+1$, basketball, football, dance and fitness videos have been selected by the same user. Then, we can infer that this user has more interests in sports than comedy. Under such circumstances, some researches consider users' interest as dynamic when designing CTR systems and have better users' interest models such as THACIL [20]. Therefore, more user-behavior modeling methods are proposed for tacking this problem. There are RNN-based models [51, 75], CNN-based models [130], transformer-based models [99] and memory network-based models [30, 159].

To tackle the aforementioned problem, we propose HyperCTR, a novel temporal framework with user and item level hypergraphs to enhance CTR prediction. To explore the sequential correlations at different time slots, HyperCTR truncates the user interactions based on the timestamp to construct a series of hypergraphs. With a hypergraph convolutional network (HGCN), HyperCTR can aggregate the correlated users and items with direct or high-order connections to generate the dynamic embedding at each time slot. With change happening both over time and across users, the temporal and group-aware user embeddings are fed into a fusion layer to generate the final user representation. The prediction of an unseen interaction can be calculated as probability between the user and micro-video representations after MLP. We show the effectiveness of our framework on three publicly available datasets ‚Äì Kuaishou, Micro-Video 1.7M (MV1.7M) and MovieLens. Our **key contributions** are: 1) We study the dynamics of user preference from two perspectives - time-aware and group-aware - and uncover the importance in exploiting the information interchange on various modalities to reflect user interests and affect CTR performance. 2) We propose a novel method HyperCTR framework with two types of modality-originated hypergraphs to generate users and items embeddings. Three of the unique aspects of the framework are a self-attention layer to capture the dynamic pattern in user-item bipartite interaction networks, a fusion layer to encode each interaction with both the temporal individual embeddings and group-level embeddings for final user pattern modeling and the CTR probability will be calculated by a MLP layer with the input of user- and item-level embeddings. 3) Extensive experiments on three public datasets demonstrate that our proposed model outperforms several state-of-the-art models. Due to anonymous requirements, the code link is invisible until paper acceptance.

4.2 Our Novel HyperCTR Model

4.2.1 Preliminaries

Our goal is learning user preferences from the hypergraph structure and predicting the probability that a user clicks the recommended entities. We denote U to represent the set of users and I represents the set of P items in an online platform. The item is characterised by various modalities, which are visual, acoustic, and textual. We also have historical interactions, such as "click" between users and items. We represent this interaction as a hypergraph $\mathcal{G}(u,i)$, where $u \in U$ and $i \in I$ separately denote the user and item sets. A hyperedge, $\mathcal{E}(u,i_1,i_2,i_3,...,i_n)$ indicates an observed interaction between user u and multiple items $(i_1,i_2,i_3,...,i_n)$ where hyperedge is assigned with a weight by $\mathbf{W}$, a diagonal matrix of edge weights. We also have multi-modal information associated with each item, such as visual, acoustic and textual features. For instance, we denote $M = \{v,a,x\}$ as the multi-modal tuple, where v, a, and x represent the visual, acoustic, and textual modalities, respectively.

Our hypothesis is that user preference also plays an important role. A user group y is associated with a user set $C_y \in U$ which can be used to represent a N-dimensional group-aware embedding. The member of groups might change over time. For each user u, we denote the user's temporal behavior as B_u^c responding to the current time, and sequential view user behavior as B_u^s according to a time slot. We further utilize $\mathcal{K}(B_u^c)$ and $\mathcal{K}(B_u^s)$ to represent the set of items in the sequential behavior, respectively.

We explain some important terminologies below which includes temporal user-item interaction representation, group-aware hypergraph and item hypergraphs.

- *Definition 1 (Temporal User-item Interaction Representation)*

  Let a sequence $\mathcal{S}(u,i_1,i_2,i_3,...)$ indicate an observed interaction between user u and multiple items $(i_1,i_2,i_3,...)$ occurring during a time slot t_n. We denote $\mathbf{E}_I = [\mathbf{e}_1,\mathbf{e}_2,...]$ as the set of items' static latent embeddings, which represent the set of items a user interacts with during this time slot. Each item in current sequence is associated with multi-modal features, which utilize M_{i_n} and it contains three-fold information about visual, acoustic and textual, denoted as v_{i_n}, a_{i_n} and x_{i_n}, respectively.

- *Definition 2 (Group-aware Multi-Modal Hypergraph)*

  Let $\mathcal{G}_g^{t_n}$ represent a hypergraph associated with i-th item at time slot t_n. $\mathcal{G}_g^{t_n} = \{V_g^{t_n},\mathcal{E}_g^{t_n},\mathbf{W}_g^{t_n},\mathbf{H}_g^{t_n}\}$ is constructed based on the whole user-item interactions with

multi-modal information. $V_g^{t_n}$ represents the nodes of individual and the correlated item in $\mathcal{G}_g^{t_n}$, $\mathcal{E}_g^{t_n}$ denoted as a set of hyperedges. We are thus creating a link to users who have interactions with multiple modal list of items. Each $\mathcal{G}_g^{t_n}$ is associated with an incidence matrix $\mathbf{H}_g^{t_n}$ and it is also associated with a matrix $\mathcal{W}_g^{t_n}$, which is a diagonal matrix representing the weight of the hyperedge $\mathcal{E}_g^{t_n}$.

- *Definition 3 (Item Homogeneous Hypergraph)*

  There are three hyperedges in each $\mathcal{G}_g^{t_n}$, which was defined in Definition 2. Let $\mathcal{G}_i^{t_n}(\mathcal{G}_i^{t_n} \supseteq \{\mathbf{g}_v^{t_n}, \mathbf{g}_a^{t_n}, \mathbf{g}_x^{t_n}\})$ represent a series of item homogeneous hypergraphs for each user group member. $\mathcal{G}_i^{t_n} = \{V_i^{t_n}, \mathcal{E}_i^{t_n}, \mathbf{W}_i^{t_n}, \mathbf{H}_i^{t_n}\}$ is constructed based on each $\mathcal{G}_i^{t_n}$ and describes a set of items that a user interacts with generated in the time slot t_n. $V_i^{t_n}$ represents the nodes of items and $\mathcal{E}_i^{t_n}$ denotes a set of hyperedges, which is creating the link to items which have interactions with a user.

The group-aware hypergraph capture group member's preference, while item hypergraphs pay more attention to item-level high-order representation. Two types of hypergraphs are the fundamental for our temporal user-item interaction representation. We define our multi-modal hypergraph CTR problem as follow:

- *Problem 1* Click-Through Rate Prediction Given a target user intent sequence $\mathcal{S}$, and its group-aware hypergraph $\mathcal{G}_g^{t_n}$ and item hypergraph $\mathcal{G}_i^{t_n}$, both of them depending on the time sequence T, this problem can be formulated as a function $f(u, \mathcal{G}_g^{t_n}, \mathcal{G}_i^{t_n}, i) \rightarrow y$ for a recommended item i, where denotes y the probability that user clicks or not.

4.2.2 HYPERCTR Framework

HyperCTR framework is illustrated in Figure 6.2. The framework can be divided into four components: temporal user behavior attention module, interests-based user hyperedge generation module, item hypergraph construction module and prediction module. We illustrate the sequential user-item interactions in different timestamps from short-term and long-term granularity. The figure also shows that the target user has a pairwise relation with one item, while the item has multi-modal features such as visual, acoustic and textual. A user might have different tastes on modalities of an item, for example, a user is attracted by the frames, but might turn out to be disappointed with its poor sound tracks. Multiple modalities have varying contributions to user preferences. Each item can be treated as most current interactions from target user and the time-aware

selection windows capture a time slot user behavior interacting on various items. All the short and long-term user intent and item embedding are fed into attention layer to represent each target user preference.

From group-level aspect, most item own more than one user. We extract item information from user-items sequential historical records and generate group-aware hyperedges. We can see in Figure 6.2 that there are three different colored areas. Every area denotes a hyperedge and a group of users connected by one unimodal feature in each hyperedge. We call this hyperedge Interest-based user hyperedge, and our task is to learn a user-interest matrix, leading to construct these hyperedges. Each hypergraph in the figure represents a group of users interacting with same item in the current time altogether and have different tendencies. We can then easily learn the group-aware information to enhance individual's representation. Besides, we have the opportunity to infer the preference of each user to make our prediction more accurate.

According to the group-level hyperedges, we can naturally find that each item can map to several users, while each user also has multiple interactions with various items. Here we cluster item information to build item hyperedges. There are several layers for each modality which extends from interests-based user hyperedges. The generation model will then go through the whole time period. We can now easily capture each higher-order structural relationship among items and enrich the representation of each items.

We leverage hypergraph convolutional operators to learn rich representation capturing local and higher-order structural relationships [33, 74]. In the prediction module, we fuse group-aware user representation and sequential user representation. We then feed into a multi-layer perceptron and output the click-through rate prediction.

4.2.2.1 Temporal User behavior Attention Module

One user's historical interaction with items can span multiple times. A straightforward way is to apply RNN-type methods to analyze the sequence $\mathscr{S}(u, i_1, i_2, i_3, \ldots)$. However, these models fail to capture both short-term and long-term dependencies. We thus perform a sequential analysis using the proposed temporal user behavior attention mechanism.

Embedding Layer As depicted in Figure 6.2, the long-term user interaction can be represented by all the items the user has interacted with in a certain time slot t_n. In the user embedding mapping stage, to depict user behaviour features, we use their metadata and profiles and define an embedding matrix $\mathbf{E}_U$ for each user $\mathbf{u}_j$. We also maintain

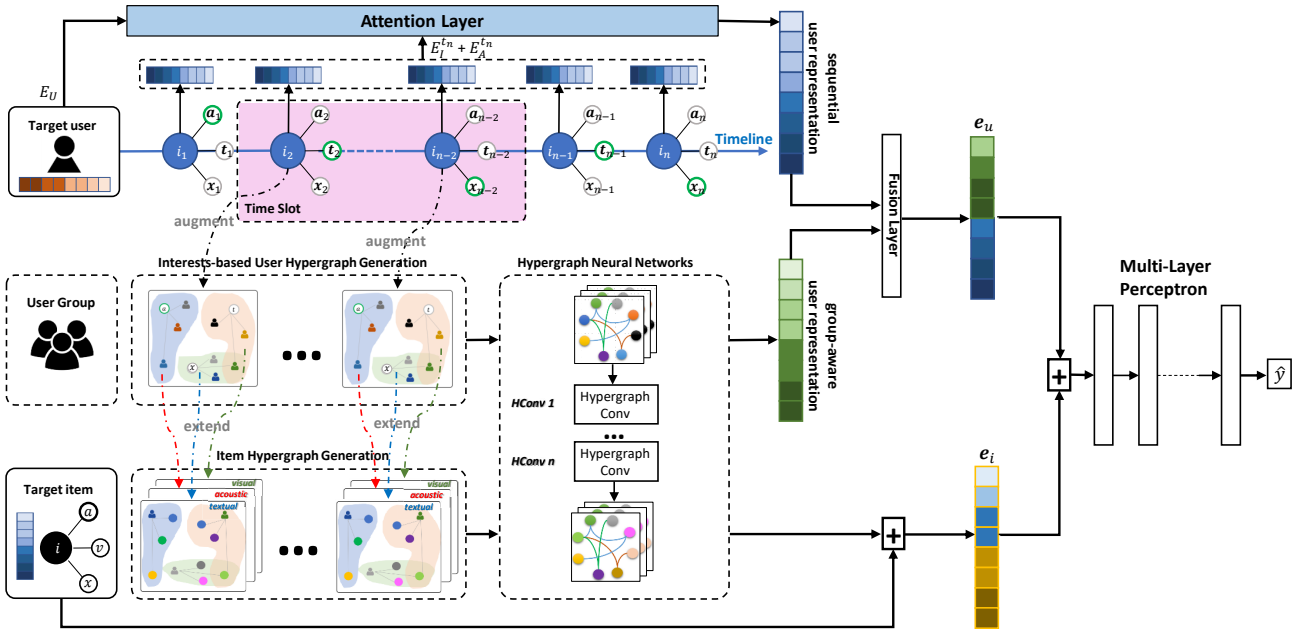

Figure 4.3: The structure of HyperCTR: two views of hypergraphs are constructed based on user-item correlations at different time slot and the Hypergraph Neural Networks is able to capture the correlations in multi-hop connections. The attention layer can capture dynamic pattern in interaction sequences. Both the group-aware and sequential user embedding fuse to represent each individual, meanwhile, the target item embedding and a set of homogeneous item-item hypergraph embeddings are considered to learn the final prediction with the multi-layer perceptron.

an item embedding matrix $\mathbf{M}_I \in \mathbb{R}^{|\mathscr{I}| \times d}$ and a multi-modal attribute embedding matrix $\mathbf{M}_A \in \mathbb{R}^{|\mathscr{A}| \times d}$. The two matrices project the high-dimensional one-hot representation of an item or multi-modal attribute to low-dimensional dense representations. Given a l-length time granularity sequence, we apply a time-aware slot window to form the input item embedding matrix $\mathbf{E}_I^{t_n} \in \mathbb{R}^{l \times d}$. Besides, we also form an embedding matrix $\mathbf{E}_A^{t_n} \in \mathbb{R}^{k \times d}$ for each item from the entire multi-modality attribute embedding matrix $\mathbf{M}_A$, where k is the number of item modalities. The sequence representation $\mathbf{E}^{t_n} \in \mathbb{R}^{n \times d}$ can be obtained by summing three embedding matrices: $\mathbf{E}^{t_n} = \mathbf{E}_U + \mathbf{E}_I^{t_n} + \mathbf{E}_A^{t_n}$.

Attention Layer We develop the sequential user behavior encoder by utilizing attention mechanism. We proposed to use self-attention layer, i.e., transformer which has also been applied in time series prediction [104]. In constrast to CNN, RNN-based approaches and Markov Chains-based models [60], we adopt self-attention as the basic model to capture the temporal pattern in user-items interaction sequence. A self-attention module

generally consists of two sub-layers, i.e., a multi-head self-attention layer and a point-wise feed-forward network. The multi-head self-attention mechanism has been adopted for effectively extracting the information selectively from different representation sub-spaces [164] and defined as:

$$MultiHead(Q,K,V) = \text{ Concat } (\text{ head }_1,\ldots,\text{ head }_h)W^O \tag{4.1}$$

$$\text{head }_i = \text{ Attention } \left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{4.2}$$

where the projections are parameter matrices $W_i^Q \in \mathbb{R}^{d\times d_k}$, $W_i^K \in \mathbb{R}^{d\times d_k}$, $W_i^V \in \mathbb{R}^{d\times d_v}$ and $W^O \in \mathbb{R}^{hd_v\times d}$. The attention function is implemented by scaled dot-product operation:

$$Attention(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{4.3}$$

where $(Q = K = V) = \mathbf{E}$ are the linear transformations of the input embedding matrix, and $\frac{1}{\sqrt{d_k}}$ is the scale factor to avoid large values of the inner product, since the multi-head attention module is mainly build on the linear projections.

In addition to attention sub-layers, we applied a fully connected feed-forward network, denoted as FFN(.), which contains two linear transformations with a ReLU activation in between.

$$\text{FFN}(x) = ReLU(0, xW_1 + b_1)W_2 + b_2 \tag{4.4}$$

where W_1, b_1, W_2, b_2 are trainable parameters.

4.2.2.2 Hypergraph Convolution Network (HGCN)

At each time slot, we aim to exploit the correlations among users and items for their high-order rich embeddings, in which the correlated users or items can be more complex than pairwise relationship, which is difficult to be modeled by a graph structure. On the other hand, the data representation tends to be multi-modal, such as the visual, text and social connections. To achieve that, each user should connect with multiple items with various modality attributes, while each item should correlated with several users. This naturally fits the assumption of the hypergraph structure for data modeling. Compared with simple graph, on which the degree for all edges is mandatory to be 2, a hypergraph can encode high-order data correlation using its degree-free hyperedges [33].

In our work, we construct a $\mathcal{G}(u,i)$ to present user-item interactions over different time slots. Then, we aim to distill some hyperedges to build user interest-based hypergraph $\mathcal{G}_g^{t_n}$ and item hypergraph $\mathcal{G}_i^{t_n}$ to aggregate high-order information from all neighborhood. We concatenate the hyperedge groups to generate the hypergraph adjacent matrix $\mathbf{H}$. The hypergraph adjacent matrix $\mathbf{H}$ and the node feature are fed into a convolutional neural network (CNN) to get the node output representations. We build a hyperedge convolutional layer $f(\mathbf{X},\mathbf{W},\Theta)$ as follows:

$$\mathbf{X}^{(l+1)} = \sigma\left(\mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^\top\mathbf{D}_v^{-1/2}\mathbf{X}^{(l)}\Theta^{(l)}\right) \tag{4.5}$$

where define $\mathbf{X},\mathbf{D}_v,\mathbf{D}_e$ and Θ is the signal of hypergraph at l layer, σ denotes the nonlinear activation function. The GNN model is based on the spectral convolution on the hypergraph.

4.2.2.3 Prediction Module and Losses

We want to incorporate both user sequential embeddings and group-aware high-order information for a more expressive representation of each user in the sequence. We propose the fusion layer to generate the representation of user u at t_n. Existing works on multiple embeddings use concatenation as fusion [67], resulting in suboptimal interactions. We utilize the fusion process that transforms the input representations into a heterogeneous tensor [88, 125, 152]. We use the user sequential embedding $\mathbf{E}^{t_n}$ and group-aware hypergraph embedding $\mathbf{E}_g^{t_n}$. Each vector $\mathbf{E}$ is augmented with an additional feature of constant value equal to 1, denoted as $\mathbf{E} = (\mathbf{E},1)^T$. The augmented matrix $\mathbf{E}$ is projected into a multi-dimensional latent vector space by a parameter matrix $\mathbf{W}$, denoted as $\mathbf{W}^T\mathbf{E}_m$. Therefore, each possible multiple feature interaction between user and group-level is computed via outer product, $\mathcal{T} = f\left(\mathbf{W}^T\cdot\tilde{\mathbf{E}}_m\right)$, expressed as:

$$\mathcal{T}_U = \mathbf{W}^T\cdot\left(\mathbf{E}^{t_n}\otimes\mathbf{E}_g^{t_n}\right) \tag{4.6}$$

Here $\otimes$ denotes outer product, $\tilde{\mathbf{E}_m}$ is the input representation from user and group level. It is a two-fold heterogeneous user-aspect tensor $\mathcal{T}_U$, modeling all possible interrelation, i.e., user-item sequential outcome embeddings $\mathbf{E}^{t_n}$ and group-aware aggregation features $\mathbf{E}_g^{t_n}$.

When predicting the CTR of user for items, we take both sequential user embedding and item embedding into consideration. We calculate the user-level probability score y to

a candidate item i, to clearly show how the function f works. The final estimation for the user CTR prediction probability is calculated as:

$$\hat{y} = f(\boldsymbol{e}_u, \boldsymbol{e}_i; \boldsymbol{\Theta}) \tag{4.7}$$

where $\boldsymbol{e}_u$ and $\boldsymbol{e}_i$ denote user and item-level embeddings, respectively. f is the learned function with parameter $\boldsymbol{\Theta}$ and implemented as a multi-layer deep network with three layers, whose widths are denoted as $\{D_1, D_2, \ldots, D_N\}$ respectively. The first and second layer use $ReLU$ as activation function while the last layer uses sigmoid function as $\text{Sigmoid}(x) = \frac{1}{1+e^{-x}}$. As for the loss function, we take an widely used end-to-end training approach, Cross Entropy Loss[32, 106, 162], and it is formulated as:

$$L(\boldsymbol{e}_u, \boldsymbol{e}_i) = y \log \sigma(f(\boldsymbol{e}_u, \boldsymbol{e}_i)) + (1-y)\log(1-\sigma(f(\boldsymbol{e}_u, \boldsymbol{e}_i))) \tag{4.8}$$

where $y \in \{0, 1\}$ is ground-truth that indicates whether the user clicks the micro-video or not, and f represents the multi-layer deep network.

4.2.3 Hypergraph Generation Modules

We aim to distill the user-level hypergraph group to enhance the representations of input data. We adopt a pre-training way to learn user group latent preference correlation to different modalities from items. However, as model trained is prone to suffer from unlabelled data problem, there is no explicit information to associate user and each item's modality. We further incorporate additional self-supervised signals with mutual information to learn the intrinsic data correlation [81, 164].

4.2.3.1 Interest-based User Hypergraph Generation Modeling

We aim to utilize self-supervised learning for the user-interest matrix $\mathbf{F} \in \mathbb{R}^{L \times d}$, where L denote the user counts and d denote the number of multi-modalities according to items. We trained the weights $\{\theta_a, \theta_b, \theta_c\}$ for each modalities. We define $\{\alpha, \beta, \gamma\}$ to denote the degree of interest of each modalities from the item features. A threshold δ was applied to measure which modality contributes the most for user-item interaction. We first maximize the mutual information between users u and item's multi-modal attributes $M_{i_n}^{t_n}$. For each user and item, the metadata and attributes provide fine-grained information about them. We aim to fuse user and multimodal-level information through modeling user-multimodal correlation. It is thus expected to inject useful multi-modal information into user group representation. Given an item i and the multi-modal attributes

embedding matrix $\mathbf{M}_{i_i}^{t_n} \in \mathbb{R}^{|\mathscr{A}| \times d}$, we treat user, item and its associated attributes as three different views denoted as $\mathbf{E}_U$, $\mathbf{E}_I^{t_n}$ and $\mathbf{E}_A^{t_n}$. Each $\mathbf{E}_A^{t_n}$ is associated with a embedding matrix $M_k \in M_{i_n}^{t_n} = \{v_{i_n}^{t_n}, a_{i_n}^{t_n}, x_{i_n}^{t_n}\}$. We design a loss function by the contrastive learning framework that maximizes the mutual information between the three views. Following Eq 4.8, we minimize the User Interest Prediction (UIP) loss by:

$$L_{UIP}\left(u, i, \mathbf{E}_{A_i}\right) = \mathbb{E}_{a_j \in \mathbf{E}_{A_i}}\left[f\left(u, i, a_j\right) - \log \sum_{\tilde{a} \in \mathbf{E}_A \setminus \mathbf{E}_{A_i}} \exp(f(u, i, \tilde{a}))\right] \tag{4.9}$$

where we sample negative attributes $\tilde{a}$ that enhance the association among user, item and the ground-truth multi-modal attributes, "\" defines set subtraction operation. The function $f(\cdot,\cdot,\cdot)$ is implemented with a simple bilinear network:

$$f\left(u, i, a_j\right) = \sigma\left[\left(\mathbf{E}_I^{\top} \cdot \mathbf{W}_{UIP} \cdot \mathbf{E}_{A_j}\right) \cdot \mathbf{E}_U\right] \tag{4.10}$$

where $\mathbf{W}_{UIP} \in \mathbb{R}^{d \times d}$ is a parameter matrix to learn and $\sigma(.)$ is the sigmoid function. We define the loss function L_{UIP} for a single user, which will can be extended over the user set in a straightforward way. The outcome from $f(.)$ for each user can be constructed as a user-interest matrix $\mathbf{F}$ and compared with the threshold δ to output the L-dimensions vector $\mathbf{v} \in \mathbb{R}^{1 \times L}$.

4.2.3.2 Item Hypergraph Construction

We exploit how to transform a sequential user-item interactions into a set of homogeneous item-level hypergraph. We construct a set of homogeneous hypergraphs $\mathscr{G}_I$, from node sets I as follow:

$$\mathscr{G}_I = \{\mathscr{G}_{I,\text{group}}, \mathscr{G}_{I,1}, \ldots, \mathscr{G}_{I,Q}\} \tag{4.11}$$

where $\mathscr{G}_{I,j} = \{I, \mathscr{E}_{I,j}\}$, and $\mathscr{E}_{I,j}$ denote hyperedges in $\mathscr{G}_{I,j}$. Note that all the homogeneous hypergraphs in $\mathscr{G}_I$ share the same node set I. For a node $i \in I$, a hyperedge introduced in $\mathscr{E}_{I,j}$ of $\mathscr{G}_{I,j}$, which connects to $\{i | i \in I, (u,i) \in \mathscr{E}_{T_n}\}$, i.e., the vertices in I that are directly connected to u by $\mathscr{E}_{T_n}$ in time period T_n. According to Figure 6.2, in the user-item sequential interaction network, the user u clicks three items v, which corresponds to a hyperedge that connects these three items in the homogeneous hypergraph $\mathscr{G}_I$. The special homogeneous hypergraph $\mathscr{G}_{I,\text{group}} \in \mathscr{G}_I$ are defined as $G\left(I, \bigcup_{j=1}^{k} \mathscr{E}_{I,j}\right)$. Note that the cardinalities of hyperedge sets in the constructed hypergraph are: $|\mathscr{E}_{I,j}| \leq |U|$ and $|\mathscr{E}_{I,\text{group}}| \leq k|U|$ for $j \leq k$. The total number of hyperedges in the homo-hypergraph is proportional to the number of nodes and edge types in the input sequence: $O(k(|I| + |V|))$. Thus, the transformation easily scales to large inputs.

4.2.3.3 Information Augmentation

The increasing data sparsity problem is one of our main motivations in tackling with CTR prediction task. To address the interaction sparsity problem, some information augmentation methods have been proposed [83, 149], however, they only consider in the case of single modality and cannot handle the scenarios with multi-modal features. We propose two data augmentation strategies, which use user behavior information and item multi-modal information to learn the subgraph embedding. We transform the initial user-item heterogeneous hypergraph into two homogeneous hypergraphs from the perspective of users and items respectively.

User Behavior Information Augment Strategy We have utilized temporal user interaction logs to represent user-level embedding. However, the heterogeneous nature between users and items aggravates the difficulty in network information fusion. A common observation is that the user usually interacts with only a small number of items while an item can only be exposed to a small number of users, which results in a sparse user-item network and limits the effectiveness of embedding representation. To mitigate the issue, we utilize the self-supervised user interest matrix $\mathbf{F}$ to build the user-user homogeneous graphs, which contains multiple hyperedges, and is regarded as hypergraph. It is denoted as $\mathcal{G}_g^{t_n}$ mentioned in Definition 2.

Item Multi-modal Information Augment Strategy It is a common observation that if two users both link to the same modality of items, then they have some common interest [158]. We are thus motivated to add an edge between them in $\mathcal{G}_g^{t_n}$. Similarly, if some items link to the same set of users, they share the same target user group. We thus add an hyperedge between them in $\mathcal{G}_i^{t_n}$.

According to the two information augmentation strategies, we transform the first-order neighbor relations of user-item to second-order neighbor relations of user-user and item-item, and represent the complex relationship as a multiple hypergraph structure. Compared with single hop neighbors, in our case nodes have more hop neighbors, which can be used to alleviate the problem of graph sparsity. The items in each hyperedge in $\mathcal{G}_i^{t_n}$ maintain some intrinsic attribute correlation due to which they connect with the same user preference. Adding edge information while aggregating information from neighbor nodes can exchange heterogeneous topology information between $\mathcal{G}_g^{t_n}$ and $\mathcal{G}_i^{t_n}$. The information fusion processes on the two graphs are interdependent.

Table 4.1: Statistics of the dataset. (v, a and t denote the dimensions of visual, acoustic, and textual modalities, respectively.)

Dataset	#Items	#Users	#Interactions	Sparsity	v.	a.	t.
Kuaishou	3,239,534	10,000	13,661,383	99.98%	2048	-	128
MV1.7M	1,704,880	10,986	12,737,619	-	128	128	128
Movielens	10,681	71,567	10,000,054	99.63%	2048	128	100

4.3 Experiments and Results

4.3.1 Experimental Settings

4.3.1.1 Datasets

Existing CTR prediction models mostly utilize unimodal datasets [77, 81, 101, 121]. In contrast, we introduce multiple modalities into CTR prediction. As mentioned above, micro-video datasets contain rich multimedia information and include multiple modalities - visual, acoustic and textual. We experimented with three publicly available datasets which are summarized in Table 4.1.

Kuaishou: This dataset is released by the Kuaishou [75]. There are multiple interactions between users and micro-videos. Each behaviour is also associated with a timestamp, which records when the event happens.

Micro-Video 1.7M: This dataset was proposed in [20]. The interaction types include "click" and "unclick". Each micro-video is represented by a 128-dimensional visual embedding vector of its thumbnail. Each user's historical interactions are sorted in chronological order.

MovieLens: The MovieLens dataset is obtained from the MovieLens 10M Data[3]. We assume that a user has an interaction with a movie if the user gives it a rating of 4 or 5. We use ResNet[42], VGGish [50] and Sentence2Vector [84] to handle the visual features, acoustic modality and textual information respectively.

4.3.1.2 Baseline Models

We compare our model with strong baselines from both sequential CTR prediction and recommendation. Our comparative methods are: 1) **GRU4Rec** [51] based on RNN. 2) **THACIL** [20] is a personalized micro-video recommendation method for modeling user's historical behaviors. 3) **DSTN** [94] learns the interactions between each type of auxiliary

[3]http://files.grouplens.org/datasets/movielens/

data and the target ad, to emphasize more important hidden information, and fuses heterogeneous data in a unified framework. 4) **MIMN** [99] is a novel memory-based multi-channel user interest memory network to capture user interests from long sequential behavior data. 5) **ALPINE** [75] is a personalized micro-video recommendation method which learns the diverse and dynamic interest, multi-level interest, and true negative samples. 6) **AutoFIS** [78] automatically selects important 2^{nd} and 3^{rd} order feature interactions. The proposed methods are generally applicable to many factorization models and the selected important interactions can be transferred to other deep learning models for CTR prediction. 7) **UBR4CTR** [101] has a retrieval module and it generates a query to search from the whole user behaviors archive to retrieve the most useful behavioral data for prediction.

4.3.1.3 Parameter Settings

We randomly split all datasets into training, validation, and testing sets with 7:2:1 ratio, and create the training triples based on random negative sampling. For testing set, we pair each observed user-item pair with 1000 unobserved micro-videos that the user has not interacted with before.

For our baseline methods, we use the implementation and settings provided in their respective papers. More details show as follow items and Table 4.2.

- **GRU4Rec** We applies GRU to model user click sequence for reproduce this model. We represent the items using embedding vectors rather than one-hot vectors.

- **THACIL** The number of micro-videos per user is set to 160. The temporal block size is set to 20. For users having more items than 160, we just preserve as much as 160 items. For users having less items, we pad all-zero vectors to augment.

- **DSTN** We set the dimension of the embedding vectors for each feature as 10, because the number of distinct features is huge. We set the number of fully connected layers in DSTN is 2, each with dimensions 512 and 256.

- **MIMN** Layers of FCN (fully connected network) are set by 200 × 80 × 2. The number of embedding dimension is set to be 16. The number of hidden dimension for GRU in MIU is set to be 32. We take AUC as the metric for measurement of model performance.

Table 4.2: Parameter Settings

Methods	#Batch size	#Dropout	#Learning rate
GRU4Rec	200	0.1	0.05
THACIL	128	0.2	0.001
DSTN	128	0.5	0.001
MIMN	200	0.2	0.001
ALPINE	2048	0.3	0.001
AutoFIS	2000	0.6	0.005
UBR4CTR	200	0.5	0.001

- **ALPINE** We utilized the 64-d visual embedding to represent the micro-video. The length of users‚Äô historical sequence is set to 300. If it exceeds 300, we truncated it to 300; otherwise, we padded it to 300 and masked the padding in the network.

- **AutoFIS** We implement the two-stage algorithm AutoFIS to automatically select important low-order and high-order feature interactions with FM-based model.

- **UBR4CTR** The datasets are processed into the format of comma separated features. A line containing user, item and context features is treated as a behavior document.

In HyperCTR and all its variants use Adam optimizer. For training, we randomly initialize model parameters with a Gaussian distribution and use the ReLU as the activation function. We then optimized the model with stochastic gradient descent (SGD). We search the batch size in 128, 256, 512, the the latent feature dimension in 32, 64, 128, the learning rate in 0.0001, 0.0005, 0.001.0.005, 0.01 and the regularizer in 0, 0.00001, 0.0001, 0.001, 0.01, 0.1. As the findings are consistent across the dimensions of latent vectors, we have shown the result of 64, a relatively large number that returns good performance whose details can be found sensitivity analysis.

4.3.1.4 Evaluation Metrics

We evaluate the CTR prediction performance using two widely used metrics. The first one is Area Under ROC curve (AUC) which reflects the pairwise ranking performance between click and non-click samples. The other metric is log loss. Log loss is to measure the overall likelihood of the test data and has been widely used for the classification tasks [107, 108].

Table 4.3: The overall performance of different models on Kuaishou, Micro-Video 1.7M and MovieLens datasets in %.

Method	Kuaishou		MV1.7M		MovieLens	
	AUC	Logloss	AUC	Logloss	AUC	Logloss
GRU4Rec	0.7367	0.5852	0.7522	0.6613	0.7486	0.6991
THACIL	0.6640	0.5793	0.6842	0.6572	0.6720	0.6791
DSTN	0.7722	0.5672	0.7956	0.6492	0.8008	0.6162
MIMN	0.7593	0.5912	0.7486	0.6862	0.7522	0.6751
ALPINE	0.6840	0.5632	0.7130	0.6591	0.7390	0.6163
AutoFIS	0.7870	0.5756	0.8010	0.5404	0.7983	0.5436
UBR4CTR	0.7520	0.5710	0.8070	0.5605	0.8050	0.5663
HYPERCTR	**0.8120**	**0.5548**	**0.8670**	**0.5160**	**0.8360**	**0.5380**
Improv.(%)	3.18%	1.49%	7.43%	4.51%	3.85%	1.03%

4.3.2 Quantitative Performance Comparison

Table 6.2 presents the AUC score and Logloss values for all models. When different modalities re used, all models show an improved performance when the same set of modalities containing visual, acoustic and textual features are used in MV1.7M and MoiveLens(10M). We also note that: (a) the performance of our model has improved significantly compared to the best performing baselines. AUC is improved by 3.18%, 7.43% and 3.85% on three datasets, respectively, and Logloss is improved by 1.49%, 4.51% and 1.03%, respectively; and (b) the improvement in our model demonstrates that the unimodal features do not embed enough temporal information which the baselines cannot exploit effectively. The baseline methods cannot perform well if the patterns that they try to capture do not contain multi-modal features in the user-item interaction sequence.

4.3.3 HyperCTR Component Analysis

4.3.3.1 Role of Multimodality

To explore the effect of different modalities, we compare the results on different modalities on the three datasets, as shown in Table 4.4. We make the following observations: 1) Our main method outperforms those with single-modal features on three datasets. It demonstrates that representing users with multi-modal information achieves a better performance. It also demonstrates that the construction of hyperedges can capture user's modal-specific preference from graph information. 2) The visual-modal is the most effective one among three modalities. It can be naturally understood that if a user clicks what to watch, one usually pays more attention to the visual information than

Table 4.4: Performance in terms of AUC & Logloss w.r.t different modalities on the three datasets in %.

Method	Kuaishou		MV1.7M		MovieLens	
	AUC	Logloss	AUC	Logloss	AUC	Logloss
multi-modal	**0.8120**	**0.5548**	**0.8670**	**0.5160**	**0.8360**	**0.5380**
visual-modal	0.8110	0.5560	0.8567	0.5167	0.8259	0.5376
acoustic-modal	-	-	0.8260	0.5171	0.8134	0.5373
textual-modal	0.7720	0.5756	0.8158	0.5175	0.8123	0.5364
(-) hypergraph	0.8034	0.5554	0.8137	0.5426	0.8064	0.5673

other modality. 3) The acoustic-modal shows more important information for user click compared with the textual features. This is expected as the background music is more attractive to users. 4) Textual modality contributes least towards click-through rate prediction. However, in MovieLens data corpus, this modality has smaller gap with the other modalities. This is because the text in MovieLens is highly related to the content. 4) Compared with GCN, our proposed model achieved better performance in all datasets. As shown in Table 4.4, based on Kuaishou datasets, when only two features are used for hypergraph, our model can still obtain slight improvement. With more features in the other two datasets, our model achieves much better performance compared with GCN. This phenomenon is consistent with our argument that when multi-modal features are available, hypergraph has the advantage of combining such multi-modal information in the same structure by its flexible hyperedges.

4.3.3.2 Role of HGCN Layers

To explore how the high-order connections in the hypergraph can help to uncover hidden item correlations and thus contribute to the final prediction. We compare the performance of HyperCTR by varying the number of hypergraph convolutional layers. As shown in Figure 4.4, when we apply only one convolution layer for our sequential model, each node embedding aggregates only information from others connected with them directly by the hyperedge. Our model performs poorly in all three datasets. By stacking three HGCN layers, it can bring in significant improvement compared with a model with just one convolution layer. We can infer that HGCN are useful options for extracting expressive item semantics and it is important to take the high-order neighboring information in hypergraph into consideration. On Kuaishou and MV1.7M, since the data is very sparse, it is not necessary to further increase the number of convolutional layers. Three HGCN layers are enough for extracting the user- and item-level semantics at different time slots. On MovieLens, more convolutional layers can further improve the embedding process.

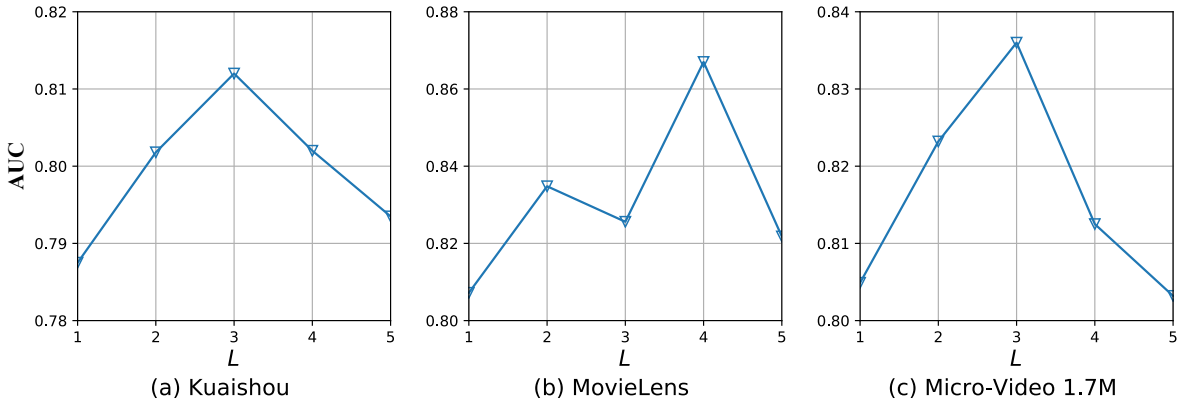

Figure 4.4: Performance comparison with different number of HGCN layers under AUC

This demonstrates the effectiveness of hypergraph and HGCN in modeling the temporal user and item correlations.

4.3.3.3 Role of Time Granularity

An important parameter which can effect the performance of HyperCTR is the granularity of the time slot. According to Figure 4.5, we show the performance of the proposed model by varying the granularity from 1 month to 18 months. When the granularity is small, we find that the model cannot achieve the best performance since the interactions are extremely sparse and not sufficient for building up a set of expressive user and item embeddings. While enlarging the granularity, we find that the performance of HyperCTR is increasing in all the datasets. In Kuaishou datasets, it reaches the best performance when the time granularity is set to half a year. However, for MovieLens, the optimized granularity is almost one year since the item in MovieLens is movie, it propagation speed is relatively slow and the impact time is relatively long. In MV1.7M datasets, the optimized granularity is around three months, which is smaller than that for the other datasets since the micro-video sharing platform attracts more interactions for each time slot for the temporal user preference representations. If we further enlarge the granularity, the performance will decrease since it underestimates the change of user preference and may introduce noise to the model.

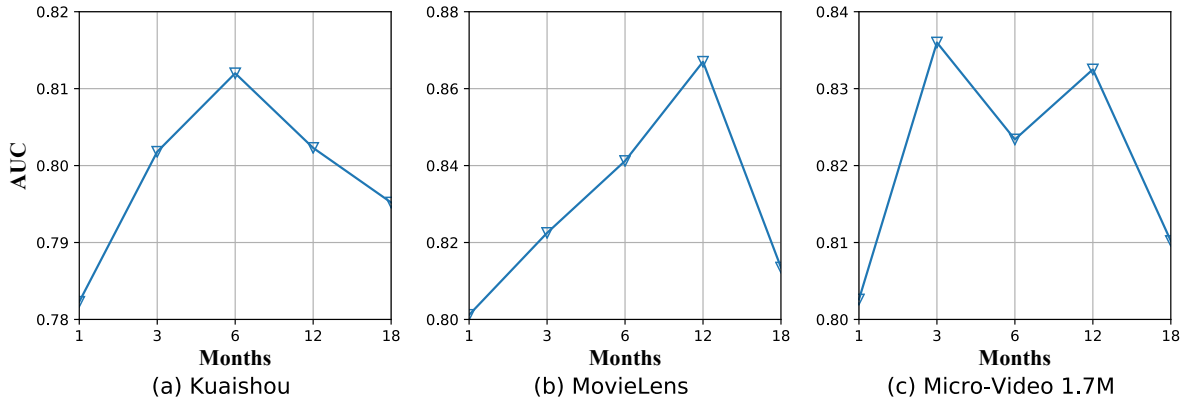

Figure 4.5: Performance comparison with various time granularity under AUC

4.3.4 HyperCTR Model Parameter Study

4.3.4.1 Hyperparametr Sensitivity Analysis

We study sensitivity of HyperCTR on the key hyperparameters using the three public datasets. The hyper-parameters play important roles in HGCN-based model, as they determine how the node embeddings are generated. We conduct experiments to analyze the impact of two key parameters which are the embedding dimension d and the size of sampled neighbors set for each node. According to Figure 4.6, we can note that: 1) When d varies from 8 to 256, all evaluation metrics increase in general since better representations can be learned. However, the performance becomes stable or slightly worse when d further increases. This may due to over-fitting. 2) When the neighbor size varies from 5 to 40, all evaluation metrics increase at first as suitable amount of neighborhood information are considered. When the size of neighbors exceeds a certain value, performance decreases slowly which may due to irrelevant neighbors. The most ideal neighbor size is in the range of 15 to 25.

4.3.4.2 Scalability Analysis

As GCN-based networks are complex and contain such a large number of nodes in the real world application scenario, it is necessary for a model being feasible to be applied in the large-scale datasets. We investigate the scalability of HyperCTR model optimized by gradient descent, which deploys multiple threads for parallel model optimization. Our experiments are conducted in a computer server with 24 cores and 512GiB memory. We run experiments with different threads from 1 to 24. We depict in Figure 5.4 the speedup

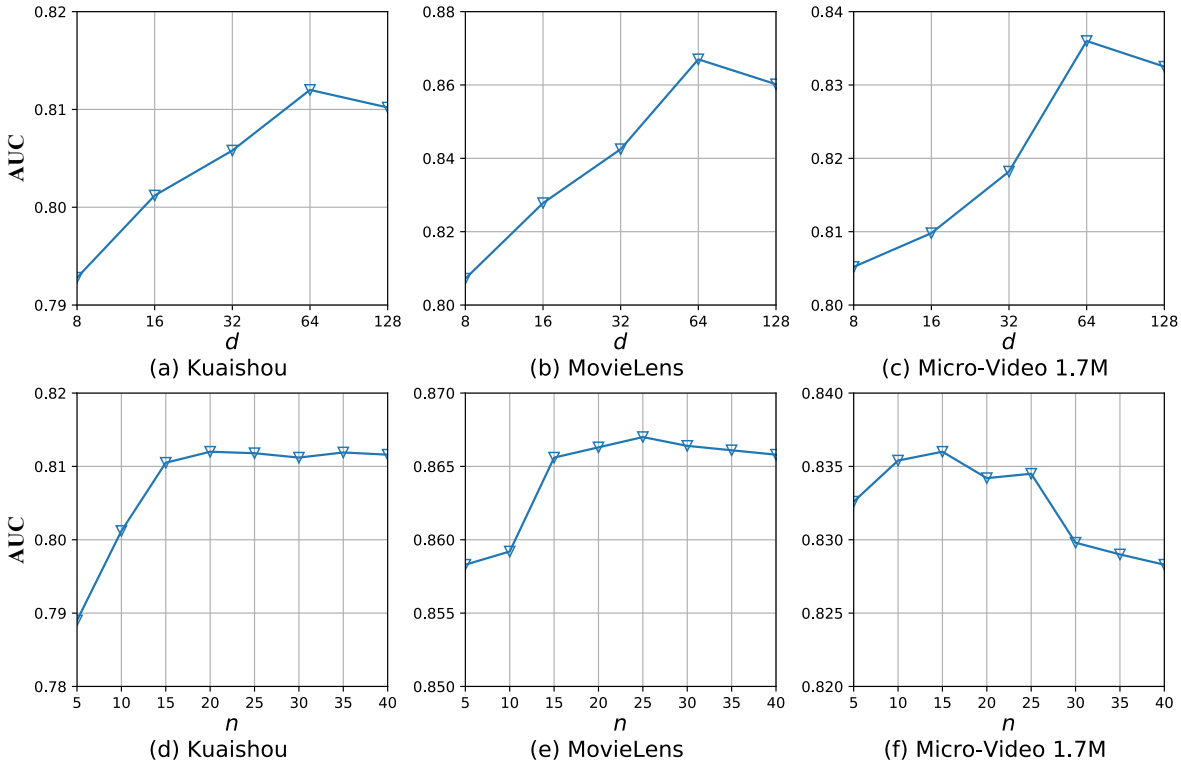

Figure 4.6: Impact of embedding dimension (top row) and sampled neighbor size (bottom row)

ratio vs. the number of threads. The speedup ratio is very close to linear, which indicates that the optimization algorithm of the HyperCTR is reasonably scalable.

4.3.4.3 Model Training

To depict our model training process, we plot the learning curves of HyperCTR, as shown in Figure 4.8. The three subfigures are the AUC curves of the multi-modal hypergraph framework when training on three datasets. Every epoch of the x-axis is corresponding to the iteration over 5% of the training set.

4.4 Related Work

CTR prediction Learning the effect of feature interactions seems to be crucial for accurate CTR prediction. Factorization Machines (FMs) [9, 109] are proposed to model pairwise feature interactions in terms of the vectors corresponding to the involved features. AutoFIS [78] and UBR4CTR [101] further improve FM by removing the redundant

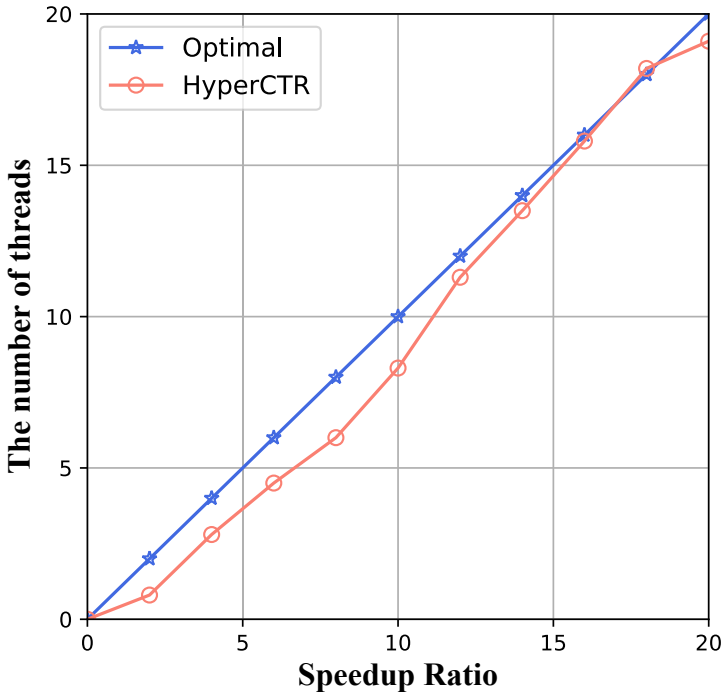

Figure 4.7: Scalability of HyperCTR

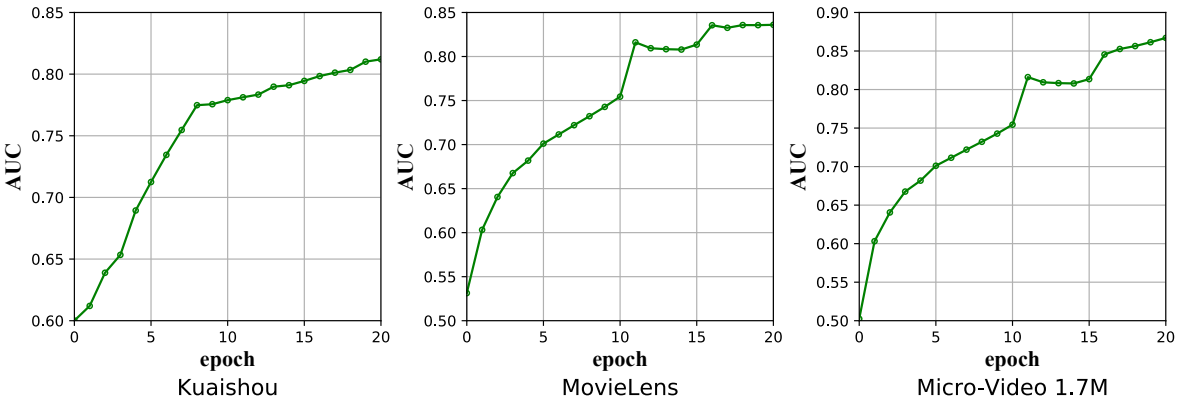

Figure 4.8: Learning process of HyperCTR.

feature interactions and retrieving a limited number of historic behavior that are most useful for each CTR prediction target. However, a FM-based model considers learning shallow representation, and it thus is unable to model the features faithfully. Deep Neural Networks (DNNs) are exploited for CTR prediction in order to automatically learn feature representations and higher-order feature interactions. DSTN [94] integrates heterogeneous auxiliary data (i.e., contextual, clicked and unclicked ads) in a unified

framework based on the DNN model. Further, the other stream of models focus more on mining temporal patterns from sequential user behavior. GRU4Rec [51] is based on RNN. It is the first work which uses the recurrent cell to model sequential user behavior. MIMN [99] applies the LSTM/GRU operations for modeling users' lifelong sequential behavior.

Exploiting multi-modal representation Some works focus on the multi-modal representation in the area of multi-modal CTR prediction. Existing multi-modal representations have mostly been applied to recommender systems and have been grouped into two categories: joint representations and coordinated representations [141]. Joint representations usually combine the uni-modal information and project into the same representation space [14, 22, 23, 25, 160]. Although, visual or textual data and are increasingly used in the multi-modal domain [72], they are suited for situations where all of the modalities are present during inference, which is hardly guaranteed in social platforms. Different from the joint representations, the coordinated models learn separate representations for each modality but coordinate them with constraints [141]. Since the modal-specific information is the factor for the differences in each modality signals, the model-specific features are inevitably discarded via similar constrains. In contrast, we introduce a novel model which respectively models the information augmentation and group-aware network problems to address the limitations in existing works.

Graph Convolution Network Our proposed model uses the GCNs technique to represent the users and items, which has been popularly used for modeling the social media data. In [40] the authors proposed a general inductive framework which leverages the content information to generate node representation for unseen data. In [154] the authors developed a large-scale deep recommendation engine on Pinterest for image recommendation. In their model, graph convolutions and random walks are combined to generate the representations of nodes. In [7] the authors proposed a graph auto-encoder framework based on message passing on the bipartite interaction graph. However, these methods cannot model the multi-modal data including cases where data correlation modeling is not straightforward [33].

4.5 Conclusion

In this chapter, we model temporal user preferences and multi-modal item attributes to enhance the accuracy of CTR prediction. We design a novel HGCN-based framework, named HyperCTR, to leverage information interaction between users and micro-videos

by considering different modalities. We also refine user presentation from two aspects: time-aware and group-aware. With the stacking of hypergraph convolution networks, a self-attention and the fusion layer, our proposed model provides more accurate modeling of user preferences, leading to improved performance.

CHAPTER 5

SIMPLIFYING GRAPH-BASED COLLABORATIVE FILTERING FOR RECOMMENDATION

5.1 Introduction

Recommendation systems conduct personalized information to assist users in finding information of their interests and alleviate information overload. Collaborative filtering (CF) represents the techniques that learn user/item embeddings from their historical interactions and has been widely applied in various domains, such as online shopping and social media.

Since the interactions can naturally be modeled as graphs, recent studies have leveraged Graph Convolutional Networks (GCNs) to learn node representations. GCN-based models can exploit higher-order connectivity between users and items and have achieved impressive recommendation performance. PinSage and M2GRL are examples of successful applications of GCNs in industrial applications.

Despite the promising performance, existing GCN-based CF models are becoming more sophisticated than ever, aiming to capture higher-order collaborative signals. Such complicated models are difficult to train with large graphs and bring efficiency and scalability challenges, which hinder their adoption in broad applications. Moreover, it can be time-consuming for CF to train GCN-based models through message passing (i.e., neighborhood aggregation) on large graphs; and simplifications done by LightGCN [49] and SGC [144] do not help much. Until now, how to improve the efficiency of GCN models

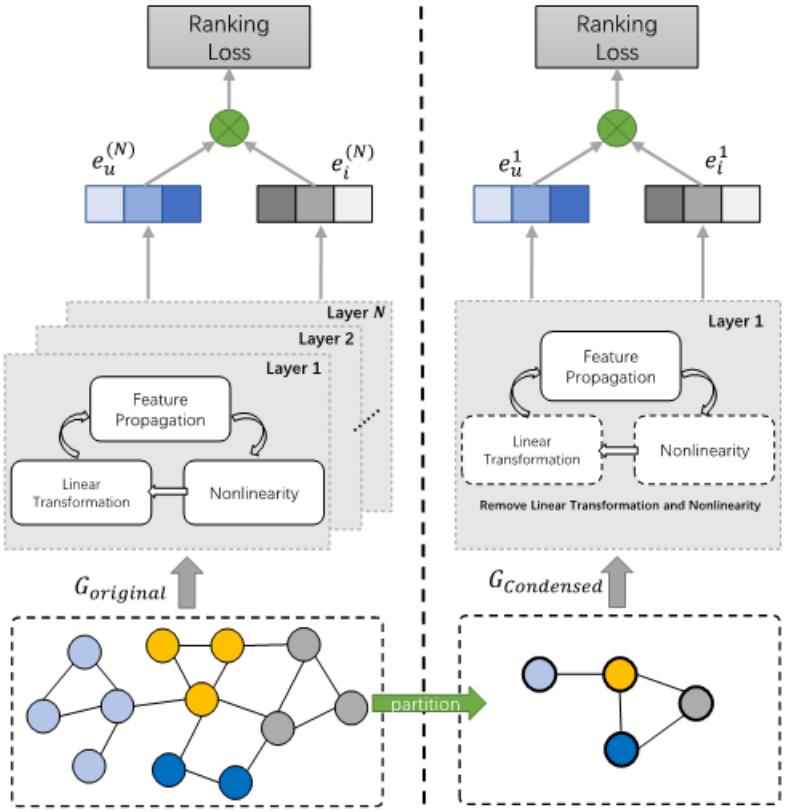

Figure 5.1: Illustrations of training of standard GCN (left) and Simplifying Graph-based Collaborative Filtering(SGCF) (right). Standard GCN needs to recurrently perform N-layers message passing to get the final embeddings for training with a large-scale graph structure $G_{original}$. At the same time, SGCF only has one layer with a condensed graph $G_{condensed}$ and removes other operations like self-connection, feature transformation, and nonlinear activation, largely improving training efficiency and helping real deployment.

while retaining their effectiveness on recommendation is still an open problem.

We address the necessity of feature transformation and nonlinear activation in GCN-based recommendation, aiming to accelerate GCNs in propagation on large-scale datasets. Given that GCN-based CF models are burdensome with many operations unjustified, we derive the simplest linear model that could precede GCNs. To this end, we reduce the excess complexity of GCNs by repeatedly removing the non-linearities between GCN layers and collapsing the resulting function into a single linear transformation. Specifically, we devise a graph partition-based algorithm to generate a model that is easy to implement, train and aggregate the multi-layer node information efficiently on large graphs. We empirically show that the final linear model exhibits comparable or superior performance to GCNs on various tasks while being more computationally efficient and

fitting significantly fewer parameters. We illustrate the above idea using a toy example in Figure 6.1.

We make the following contributions in this paper:

- We empirically reduce the excessive complexity of GCNs by repeatedly removing the nonlinearities between GCN layers and collapsing the resulting function into a single linear transformation.
- We propose Simplifying Graph-based Collaborative Filtering(SGCF), which largely simplifies the model design by including only the most essential components in GCN for more efficient recommendations. We offer an effective partition technique for reducing the scale of input graph structure to avoid infinite layers of explicit message passing for efficient recommendations.
- Our extensive experiments on four benchmark datasets show that SGCF achieves significant improvements over state-of-the-art GCN-based CF model. Notably, SGCF attains up to 10% improvement in NDCG@20 and more than 10x speed-up in training over our baselines on the Amazon-Books dataset. To allow for reproduciblility, we will release the source code and benchmark settings of SGCF at Github.

5.2 Preliminaries

Following SemiGCN [64], We define a graph as $\mathcal{G}$=$< \mathcal{V}, \mathbf{E} >$, where $\mathcal{V}$ denotes the set of nodes and $\mathbf{E}$ denotes the edge e_{ij}between node i and node j. We use $\mathbf{A}$ to denote the adjacency matrix—$a_{ij} = 1$ if an edge exists from node i to node j; and $a_{ij} = 0$ otherwise. To ease Illustration, we use $\mathbf{A} = [i|a_{ij} = 1]$ to denote the one-hop set of nodes, $\tilde{\mathbf{A}} = \tilde{D}^{-\frac{1}{2}}\mathbf{A}\tilde{D}^{-\frac{1}{2}}$ the normalized adjacency matrix with added self loops, where $\tilde{D}$ is the degree matrix of $\tilde{\mathbf{A}}$. $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$ is the adjacency matrix of the graph with added self-connections, where $\mathbf{I}$ is the identity matrix.

5.2.1 Graph Convolutional Networks

For each node $v \in \mathcal{V}$, we use e_i^0 to denote the node initial embedding, which is usually the feature vector x_i of node i, in which $e^0 = x_i$. In a graph $\mathcal{G}$, the main idea of GCNs is to stack L steps in a recursive message passing or feature propagation operation to learn node embedding [61]. Specifically, for each node i at the step, it is computed recursively

with following three steps: feature propagation, feature transformation and non-linear transition.

Feature propagation For each node i, the feature aggregation operation aggregate the embeddings from graph neighbors $\mathcal{N}_i$ and its own embedding e_i^k at previous layer l. As the focus of this work is not to design more sophisticated feature aggregation function, we follow the widely used feature aggregation function proposed in Kipf et al. [64], which is empirically effective and has been adopted by many GCN variants:

$$\overline{\mathbf{H}}^{(k+1)} = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}\mathbf{H}^k \tag{5.1}$$

where the features $\mathbf{H}$ at k-th layer, feature propagation output $\mathbf{H}$ layer can be regarded as the Laplacian smoothing on the features at previous layer.

Feature transformation and nonlinear transition After the local smoothing, a GCN layer is identical to a standard multi-layer perceptron (MLP). Each layer is associated with learned weight $\mathbf{W}^{(k)}$, and the smoothed hidden feature representations are transformed linearly. Finally, a nonlinear activation function such as $ReLU(.) = max(0,\cdot)$ is applied pointwisely before outputting feature representation $\mathbf{H}(k)$. In totally, the representation updating rule of the k-th layer is:

$$\mathbf{H}^{(k)} \leftarrow \mathrm{ReLU}\left(\overline{\mathbf{H}}^{(k)}\mathbf{W}^{(k)}\right) \tag{5.2}$$

The pointwise nonlinear transformation of the k-th layer is followed by the feature propagation of the $(k+1)$-th layer.

5.2.2 Graph Convolutional based Recommendation

In a recommender system, there are two sets of entities: a user set $\mathbf{U}$ with M users and an item set $\mathbf{I}$ with N items. As implicit feedback is the most common form in many recommender systems, we focus on implicit feedback based CF in this work, and it is easy to extend the proposed model for rating prediction in CF. Users show ratings to the items with a rating matrix $\mathbf{R} \in \mathbb{R}^{M\times N}$, with r_{ui}=1 denotes user u likes item i, otherwise it equals 0. The rating matrix is a key to the success of recommendation performance. With the huge success of GCNs, researchers attempted to formulate recommendation as a user-item bipartite graph, and adapted GCNs for recommendation. NGCF [138] are specifically designed under the CF settings. Given ratings of users to items, the user-item bipartite graph is denoted as $\mathcal{G}$=<$\mathbf{U} \cup \mathbf{I}$,$\mathbf{A}$>, with $\mathbf{A}$ is constructed from the rating matrix

R as:

$$\mathbf{A} = \begin{bmatrix} \mathbf{R} & \mathbf{0}^{N\times M} \\ \mathbf{0}^{M\times N} & \mathbf{R}^{\mathbf{T}} \end{bmatrix} \tag{5.3}$$

Let $\mathbf{E} \in \mathbb{R}^{(M+N)\times D}$ denote the free embedding matrix of users and items. By feeding the free embedding matrix $\mathbf{E}$ into GCNs with bipartite graph $\mathcal{G}$, i.e., $\forall i \in \mathcal{U} \cup \mathcal{I}$, $h_i^0 = e_i$. Then, GCNs iteratrively perform with embedding propagation step in Eq.(1) and nonlinear transformation with Eq.(2), each user's or item's embeddings can be updated in the iterative process. Therefore, the final embedding $\mathbf{H}^k$ explicitly injects the up to K-th order collective connections between users and items. All the parameters can be learned in an end-to-end framework.

5.2.3 Graph Partition Technique

A naive approach for the initialization of network embedding is by random, which assigns random numbers in $\mathbb{R}$ for the initial embedding of each node in the graph. However, this approach disregards the structure of the input graph, rendering it unsuitable for network embedding. Inspired by the graph partition base algorithm, we aim to describe the sketch of the input graph $\mathcal{G} = <\mathcal{V}, \mathbf{E}>$ using the partitioning of $\mathcal{G}$, which are then processed as the initial embedding of each node in $\mathcal{V}$. A partitioning $\mathcal{P}$ of $\mathcal{G}$ divides $\mathcal{V}$ into k disjoint subsets, denoted by $\mathcal{P}$=$\mathcal{V}_1$, $\mathcal{V}_2$, ..., $\mathcal{V}_k$, where k is a user defined number. Given a node $v \in \mathcal{V}$, let $\mathcal{V}' \in \mathcal{P}$ be the partition where v resides, denoted by $p(v) = \mathcal{V}'$. We call the neighbors in the same partition are internal nodes, while the others are external nodes. Moreover, a node $v \in \mathcal{V}$ is a border node of $\mathcal{G}$, if v has at least one neighbor $n \in N(v)$ whose partition is different from the one of v, namely $p(v) \neq p(n)$. Let $\mathcal{V}_b$ be the set of border nodes of $\mathcal{G}$. The border sub-graph $\mathcal{G}_b$ with respect to $\mathcal{P}$ is the induced sub-graph of $\mathcal{G}$ constructed on $\mathcal{V}_b$.

5.3 Method

5.3.1 Overall Structure of Our Model

In this part, we propose Simple Graph Convolutional Collaborative Filtering with graph partition techniques which is a general GCN-based CF model for recommendation. The overall architecture of SGCF is shown in Figure 6.2. SGCF advances current GCN-based model with two characteristics: (a) At each layer of the feature propagation step, we

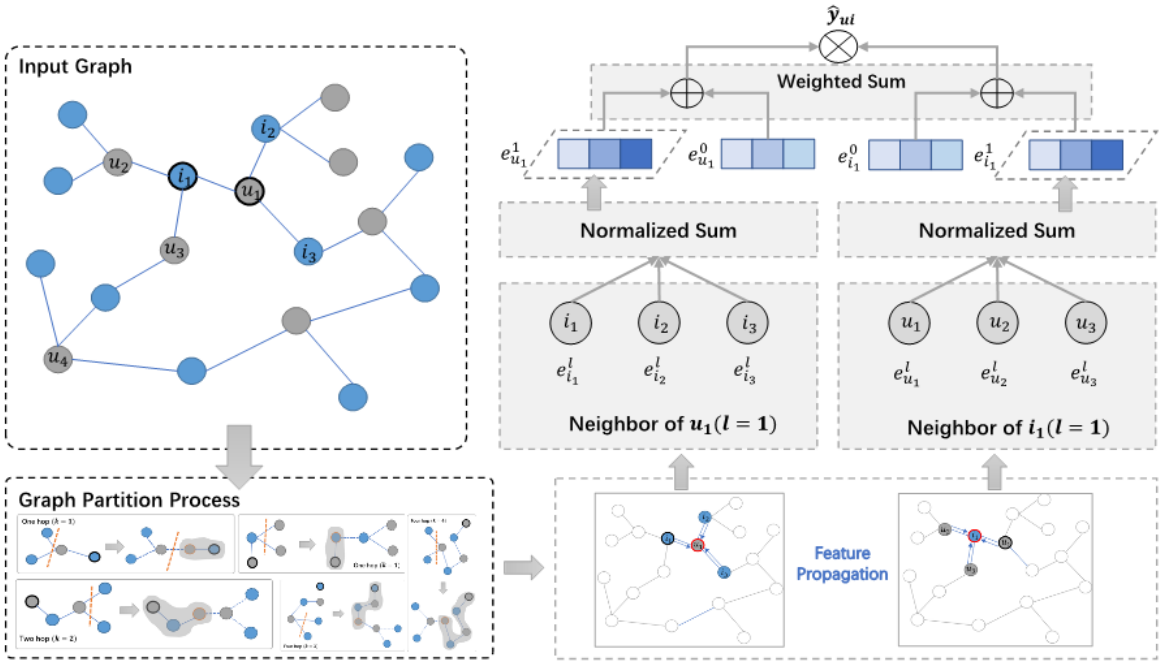

Figure 5.2: The overall architecture of our proposed mode. The graph process illustrates the procedure of embedding propagation with different hop. The partition algorithm works in several iterations with different hops k (left bottom). In each iteration the updating of the embedding of each node can be achieved in a k-layer computing framework. The final condensed graph feed into our simplified GCF model.

use a simplified linear embedding propagation without any nonlinear activation and linear transformations; (b) for accelerating the network embedding and improve the performance of the algorithms on both effectiveness and efficiency, we propose a graph resizing technique to recursively partition a graph into several small-sized sub-graphs to capture the internal and external structural information of nodes, and then compute the network embedding with low-order propagation process in a condensed graph.

5.3.2 Simplified Embedding Propagation

In traditional MLP's, deeper layers allow for more expressive features because they allow for feature hierarchies, such as features in the second layer building on top of features in the first layer. In GCNs, the layers have another important function: in each layer the hidden representations are averaged among neighbors that are one hop away. This implies that after k-layers a node obtains feature information from all nodes that are k-hops away in the graph. This effect is similar to convolutional neural networks, where

depth increases the receptive field of internal features [41]. Although convolutional networks can benefit substantially from the increased depth [54], typically MLPs obtain little benefit beyond 3 or 4 layers.

We hypothesize that the non-linearity between GCN layers is not critical - but that the majority of the benefit arises from the local averaging. We therefore remove the nonlinear transition functions between each layer.

Given the user-item bipartite graph as formulated in Eq.(3), let $\mathbf{E} \in \mathbb{R}^{(M+N)\times D}$ denote the free embeddings of users and items, with the first M rows of the matrix, i.e., $\mathbf{E}_{1:M}$ is the user embedding sub-matrix, and $\mathbf{E}_{M+1:M+N}$ is the item embedding sub-matrix. Then, our model takes the embedding matrix as input:

$$\mathbf{E}^0 = \mathbf{E} \tag{5.4}$$

which resembles the embedding based models in CF. Notably, different from GCN based tasks with node features as fixed input data, the embedding matrix is unknown and needs to be trained our model. Following the theoretical elegance with graph spectral connections and empirical competing results of SGC, at each iteration step $k+1$, we assume the embedding $\mathbf{E}^{k+1}$ is a nonlinear aggregation of the embedding matrix $\mathbf{E}^k$ at the previous layer k as:

$$\mathbf{E}^{k+1} = \mathbf{S}\mathbf{E}^k\mathbf{W}^k \tag{5.5}$$

where $\mathbf{S} = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}$ denotes the normalized adjacency matrix with added self loop, $\mathbf{W}^k$ is the nonlinear transformation. Further, Eq.(5) with matrix form is equivalent to modeling each user u's and each item i's update embedding as:

$$\left[\mathbf{E}^{k+1}\right]_u = \mathbf{e}_u^{k+1} = \left[\frac{1}{d_u}\mathbf{e}_u^k + \sum_{j\in R_u}\frac{1}{d_j \times d_u}\mathbf{e}_j^k\right]\mathbf{W}^k \tag{5.6}$$

$$\left[\mathbf{E}^{k+1}\right]_i = \mathbf{e}_i^{k+1} = \left[\frac{1}{d_i}\mathbf{e}_i^k + \sum_{u\in R_i}\frac{1}{d_i \times d_u}\mathbf{e}_u^k\right]\mathbf{W}^k \tag{5.7}$$

which $d_i(d_u)$ is the diagonal degree of item i (user u) in the user-item bipartite graph $\mathcal{G}$. R_u (and R_i) is neighbors of node user or item in graph $\mathcal{G}$.

5.3.3 Model Prediction with Condensed Graph

With a predefined depth K, the nonlinear embedding propagation would stop at the K-th layer with output of the embedding matrix $\mathbf{E}^K$. For each user (item), $e_u^K(e_i^K)$ captures the

up to K-th order bipartite graph similarity. Then, many embedding based recommendation models would predict the preference $\hat{y}_{ui}$ as the inner product between user and item latent vectors as:

$$\hat{y}_{ui} = < \mathbf{e}_u^K, \mathbf{e}_i^K > \tag{5.8}$$

which <,> denotes the vector inner product operation.

Most existing GCN based variants, as well as GCN based recommendation models, achieve the best performance with K=2 [43]. The overall trend for these GCN variants is that: (1) the performance increases as K increases from 0 to 1, (2) and drops quickly as K continues to increase. In fact, most recommended scenarios have large-scale input networks and the user-item graph will become more complicated. It will cause each node e_u or e_i has multiple neighbor hops ($K >= 2$). However, as k increases from 0 to K, the node embeddings at deeper layers tend to be over smoothed, i.e., they are more similar with less distinctive information. Meanwhile, stacking multiple layers of message passing likely introduces uninformative, noisy, or ambiguous relationships, which could largely affect the training efficiency and effectiveness. This problem not only exists in GCNs, but is much more severe in CF with very sparse user behavior data for model learning. To alleviate the problem, we utilize the graph partition techniques to reduce the scale of the input network and construct the condensed graph.

To construct the condensed graph $\mathscr{G}_c$=($\mathscr{V}_c$, $\mathscr{E}_c$), we first obtain a partitioning $\mathscr{P}$ of $\mathscr{G}$, denoted by $\mathscr{P} = \{\mathscr{V}_1, \mathscr{V}_2, \ldots, \mathscr{V}_k\}$ where k is a user-defined number. The goal of graph partition is (k, σ)-balanced where $0 < \sigma < 1$, and it satisfies the constraint:

$$\max_{1 \le i \le k} |\mathscr{V}_i| \le (1+\sigma) \left\lceil \frac{|\mathscr{V}|}{k} \right\rceil \tag{5.9}$$

and minimizes the size of edge-cut as:

$$\bigcup_{1 \le i,j \le k} \{(v,u) \in \mathbf{E} \mid v \in \mathscr{V}_i, u \in \mathscr{V}_j\} \tag{5.10}$$

However, the (k, σ)-balanced graph partition is a NP-hard problem [11]. To deal with this issue, we are motivated by the GPA algorithm [76] for graph partitioning, which has adopted in practice and costs a running time complexity $O(|\mathscr{V} + |\mathbf{E}| + klogk|)$. Based on $\mathscr{P}$, we construct the condensed graph $\mathscr{G}_c$ of $\mathscr{G}$ by creating an condensed node v_a for each sub-graph $\mathscr{V}' \in \mathscr{P}$ and connecting two condensed nodes v_a and u_a with an condensed edge (v_a, u_a) of a weight $w(v_a, u_a)$. Then, the number of condensed nodes in $\mathscr{G}_c$ is k, i.e., the number of partitions of $\mathscr{G}$. Besides, the number of condensed edges of $\mathscr{G}_c$ is bounded by the size of edge cut.

Algorithm 2 Condensed Graph Propagate

Input graph: $\mathcal{G} =< \mathcal{V}, \mathbf{E} >$, the embeddings e_c of $\mathcal{G}_c$ condensed graph, and the threshold δ

Result: The set e_i of initial embedding of each node $v \in \mathcal{V}$

initialization: Let $e_i(v) = e_c((c_v))$ for each node $v \in \mathcal{V}$

While $\triangle > \delta$
for each node $v \in \mathcal{V}$ **do**
    Let $e_{avg}(v)$= $\frac{1}{|N(v)|} \sum_{u \in N(v)} e_i(u)$
    Compute $e'_i(v) = \frac{1}{2}\left(e_i(v) + e_{avg}(v)\right)$
end for
Let $\triangle = \frac{1}{|V|} \sum_{v \in V} \left\| e'_i(v) - e_i(v) \right\|$;
for e **do**ach node $v \in \mathcal{V}$
    let $e_i(v) = e'_i(v)$
end for
return e_i

One crucial issue remaining is how to decide k. On one hand, if k is small, then one condensed node would be pertinent to a lot of nodes in the input graph $\mathcal{G}$. As such, the initial embedding of each node in $\mathcal{G}$ inherited from the corresponding abstract node would lose the power of effectiveness. On the other hand, if k is large, then the condensed graph $\mathcal{G}_c$ would be large too. Therefore, it would be highly expensive to compute the network embedding on $\mathcal{G}_c$, which increase the overall cost of the initialization phase. To reach a good balance, we set $k = \lceil \sqrt{|\mathcal{V}|} \rceil$, which is a sufficiently large number but much smaller than $|\mathcal{V}|$, that works well in practise.

In addition, to compute the condensed graph embedding of $\mathcal{G}_c$, a naive approach is to let the initial embedding of each node v equal the embedding of the corresponding condensed node $c(v)$. However, this approach would suffer from the issue where the nodes pertinent to the same condensed node have the same initial embeddings, rendering this method ineffective. For addressing this issue, we utilize a iterative approach where each node update its own embedding based on the embeddings of its neighbors until the convergence is reached. This means specifically, in each iteration, each node $v \in \mathcal{V}$ first aggregates the embeddings of v's neighbors, which results in the average embedding $e_{avg}(v)$. Then, we update v's embeddings as the aggregation of e_{avg} and its own embedding $e_i v$. The reason is that the embedding of each node should be close to its neighbors in the graph. Moreover, Algorithm 2 shows the procedure of embedding propagation. Consider a graph $\mathcal{G}$=$< \mathcal{V}, \mathbf{E} >$, the condensed graph $\mathcal{G}_c$ of $\mathcal{G}$, and the network embedding e_c of $\mathcal{G}_c$.

Based on the above condensed input graph, we argue that: instead of directly utiliz-

ing the original user-item bipartite network, we perform the preference learning with condensed graph as: $\hat{y}_{ui} =< \mathbf{e}_u^k, \mathbf{e}_i^k >$. We hypothesize that it is easier to optimize the condensed rating, and the condensed graph learning could help to alleviate the over smoothing effect with deeper layers. Based on the condensed preference prediction in above, we have:

$$\begin{aligned}\hat{y}_{ui} &= \hat{y}_{ui}^{k-1} + < \mathbf{e}_u^k, \mathbf{e}_i^k > \\ &= \hat{y}_{ui}^{k-2} + < \mathbf{e}_u^{k-1}, \mathbf{e}_i^{k-1} > + < \mathbf{e}_u^k, \mathbf{e}_i^k > \\ &= \hat{y}_{ui}^{0} + < \mathbf{e}_u^1, \mathbf{e}_i^1 > + \ldots + < \mathbf{e}_u^k, \mathbf{e}_i^k > \\ &=< \mathbf{e}_u^0 \left\| \mathbf{e}_u^1 \right\| \ldots \left\| \mathbf{e}_u^k, \quad \mathbf{e}_i^0 \right\| \mathbf{e}_i^1 \| \ldots \| \mathbf{e}_i^k > .\end{aligned} \tag{5.11}$$

The above equation is equivalent to concatenate embedding of each layer to form the final embedding of each node. This is quite reasonable as each node,Äôs sub-graph varies, and recording each layer,Äôs representation to form the final embedding of each node is more informative.

5.3.4 Model Learning

The trainable parameters of our model are only the embeddings of the first-order layer, such as $\mathbf{W} = \mathbf{E}^{(0)}$. In other words, the model complexity is same as the standard matrix factorization (MF). We adopt the ranking based loss function in Bayesian Personalized Ranking (BPR) [111], which a pairwise loss that encourages the prediction of an observed entry to be higher than its unobserved counterparts:

$$\min_{\mathbf{W}} L(\mathbf{R}, \hat{\mathbf{R}}) = \sum_{a=1}^{M} \sum_{(i,j) \in D_a} -\ln(s(\hat{r}_{ai} - \hat{r}_{aj})) + \lambda \|\mathbf{W}\|^2 \tag{5.12}$$

where λ controls the L_2 regularization strength. We employ the Adam SGD [63] optimizer and use it in a mini-batch manner. We are aware of other advanced negative sampling strategies which might improve the SGCF training, such as the hard negative sampling [110] and adversarial sampling [29]. We leave this extension in the future since it is not the focus of this work. Note that we do not introduce dropout mechanisms, which are commonly used in GCNs and NGCF. The reason is that we do not have feature transformation weight matrices in SGCF, thus enforcing L_2 regularization on the embedding layer is sufficient to prevent over fitting. This showcases SGCF,Äôs advantages of being simple ,Äî it is easier to train and tune than NGCF which additionally requires to tune two dropout ratios, such as node dropout and message dropout, and normalize the embedding of each layer to unit length. Moreover, there is one crucial issue remaining in

the network embedding learning on the condensed graph $\mathcal{G}_c$ which is the configuration of hyperparameters in the random walk based algorithm, i.e., the number of random walks and the length of a random walk. To cope with this issue, we utilize a pre-processing phase which trains a regression model that takes into account both the hyperparameters and the statistics of the condensed graphs. As such, given an condensed graph $\mathcal{G}_c$, we are able to infer from the model the suitable hyperparameters for $\mathcal{G}_c$ with a slight cost, as explained shortly.

5.3.5 Model Analysis

Detailed Analysis of Model Based on the prediction function in Eq.(11), we observe that SGCF is not a deep neural network but a wide linear model. The linearization has several advantages: First, as SGCF is built on the recent progress of SGC, it is theoretically connected as a low pass filter of graph on the spectral domain [144]. Second, with the linear embedding propagation and partition graph learning, SGCF is much easier to train compared to nonlinear GCN based models. Last but not least, we obtain the initialization of the embedding for each node in the graph by computing the network embedding on the condensed graph, which is much smaller than the input graph, and then propagating the embedding among the nodes in the input graph. Instead, we could resort to stochastic gradient descent for model learning. Therefore, SGCF is much more time efficient compared to classical GCN based models.

Connections with Existing Work We compare the key characteristics of our proposed model with three closely related GCN based recommendation models: PinSage [95], NGCF, and LightGCN. NGCF and LightGCN are both the first few attempts that also use a residual prediction function by taking each user (item)'s embedding as a concatenation of all layers' embeddings. However, the authors simply use this "trick" without any detailed explanation. We empirically show the reason why taking the output of the last layer embedding fails for CF, and show using residual prediction is equivalent to concatenate all the layer's embeddings as the final embedding of each node in the user-item bipartite graph. For PinSage, it has a lower time complexity compared to its deep learning based counterparts (e.g., NGCF) as this model designed a sampling technique in feature aggregation process.

5.4 Experiments

We first compare SGCF with various state-of-the-art CF methods to demonstrate its effectiveness and high efficiency. We also perform detailed parameter studies to justify the rationality and effectiveness of the design choice of SGCF.

5.4.1 Experimental Setup

5.4.1.1 Datasets

We utilize four publicly available datasets, including Yelp2018, Amazon-Books, Gowalla [1], and MovieLens to conduct our experiments, as many recent GCN-based CF models [16, 46, 47, 49, 120, 138, 139, 145] are evaluated on these four datasets. We closely follow these GCN-based CF studies and use the same data split as them. Table 5.1 shows the statistics of the used datasets.

Yelp2018: This dataset is adopted from the 2018 edition of the Yelp challenge. Where in, the local businesses like restaurants and bars are viewed as the items. We use the same 10-core setting in order to ensure data quality. **Amazon-Books**: Amazon-books is a widely used dataset for product recommendation [48]. We select Amazon-Books from the collection. Similarly, we use the 10-core setting to ensure that each user and item have at least ten interactions. **Gowalla**: is a location-based social networking website where users share their locations by checking-in. The friendship network is undirected and was collected using their public API, and consists of 196,591 nodes and 950,327 edges. **MovieLens**: The MovieLens dataset is obtained from the MovieLens 10M Data [2]. We assume a user has an interaction with a movie if the user gives it a rating of 4 or 5.

5.4.1.2 Baselines

In total, we compare SGCF with three types of the stat-of-the-art models, covering MF-based methods, metric learning-based approaches and GCN-based models.

MF-based methods: MF-BPR [85] a pairwise method that exploits different types of feedback with an extended sampling method. ENMF [87]) an Efficient Adaptive Transfer Neural Network (EATNN) for social-aware recommendation. Metric learning-based method - CML [52]. **Networking embedding methods**: DeepWalk [98] learns embeddings via the prediction of the local neighborhood of nodes, sampled from random walks

[1] http://www.gowalla.com/

[2] http://files.grouplens.org/datasets/movielens/

on the graph. LINE [129] is suitable for arbitrary types of information networks: undirected, directed, and/or weighted. Node2Vec [38] is a state of art graph representation learning method. It utilizes random walk to capture the proximity in the network and maps all the nodes into a low-dimensional representation space which preserves the proximity. **GCN-based methods**: NGCF achieves the target by leveraging high-order connectivities in the user-item integration graph. NIA-GCN [124] can explicitly model the relational information between neighbor nodes and exploit the heterogeneous nature of the user-item bipartite graph. LR-GCCF [16] is a general GCN based CF model for recommendation. LightGCN learns user and item embeddings by linearly propagating them on the user-item interaction graph, and uses the weighted sum of the embeddings learned at all layers as the final embedding and DGCF [153] considers user-item relationships at the finer granularity of user intents and generates disentangled user and item representations to get better recommendation performance.

5.4.1.3 Evaluation Metrics

Given a user, a top-K item list recommendation algorithm provides a list of ranked item lists according to the predicted preference of them. To assess the ranked lists with respect to the ground-truth lists set of what users actually interacted with, we adopt three evaluation metrics: Normalized Discounted Cumulative Gain (NDCG) [56] at 20 (NDCG@20), Hit Ratio at 20 (HR@20) and recall at 20 (Recall@20).

5.4.1.4 Parameter Settings

We implement our SGCF model in Tensorflow[3]. There are two important parameters in our model: 1) the dimension D of the user and item embedding matrix $\mathbf{E}$, and 2) the regularization parameter λ in the objective function (Eq.12). The embedding size is fixed to 64 for all models. In our proposed SGCF model, we try the regularization parameter λ in the range $[0.0001, 0.001, 0.01, 0.1]$ and find $\lambda = 0.01$ reaches the best performance. We adopt Gaussian distribution with 0 mean 10^{-4} standard deviation to initialize embeddings. There are several parameters in the baselines, for fail comparison, all the parameters in the baselines are also tuned to achieve the best performance.

[3]https://www.tensorflow.org/

Table 5.1: Statistics of the datasets.

Dataset	#Users	#Items	#Interactions	Density
Amazon-Books	52,643	91,599	2,984,108	0.062 %
MovieLens-10M	71,567	10,681	10,000,054	0.371 %
Gowalla	29,858	40,981	1,027,370	0.084 %
Yelp2018	31,668	38,048	1,561,406	0.130 %

Table 5.2: Overall performance comparison. Improv. denotes the relative improvements over the best GNN-based baselines.

Model	Amazon-Books		Yelp2018		Gowalla		MovieLens-10M	
	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
ENMF	0.0334	0.0279	0.0614	0.0525	0.1532	0.1351	0.2345	0.2098
CML	0.0413	0.0313	0.0621	0.0536	0.1639	0.1298	0.1725	0.1536
MF-BPR	0.0324	0.0259	0.0539	0.0432	0.1623	0.1346	0.2134	0.2135
DeepWalk	0.0347	0.0266	0.0478	0.0382	0.1042	0.0741	0.1351	0.1047
Node2Vec	0.0412	0.0307	0.0448	0.0361	0.1020	0.0711	0.1476	0.1190
LINE	0.0412	0.0321	0.0547	0.0445	0.1336	0.1057	0.2338	0.2232
NGCF	0.0345	0.0261	0.0580	0.0478	0.1571	0.1337	0.2515	0.2513
LR-GCCF	0.0336	0.0264	0.0560	0.0345	0.1521	0.1286	0.2230	0.2131
LightGCN	0.0412	0.0314	0.0651	0.0529	0.1824	0.1548	0.2573	0.2423
NIA-GCN	0.0371	0.0289	0.0589	0.0492	0.1361	0.1116	0.2361	0.2243
DGCF	0.0423	0.0325	0.0654	0.0534	0.1843	0.1563	0.2640	0.2504
SGCF	0.0466	0.0358	0.0683	0.0561	0.1862	0.1580	0.2787	0.2642
Improv.	10.16 %	10.15 %	4.43 %	4.66 %	1.03 %	1.08 %	5.57 %	5.51 %

Table 5.3: Efficiency comparison with full training time

Model	Epoch Count	Time per Epoch	Totally Time
MF-BPR	**25**	**33s**	**13.75 m**
LR-GCCF	170	70s	3h 30m
ENMF	85	135s	3h 11m
LightGCN	55	850s	12h 58m
SGCF	64	**36s**	**38.4 m**

5.4.2 Quantitative Performance Comparison

Our experimental results are reported in Table 6.2. We have several observations: 1) SGCF consistently outperforms all baseline approaches across all four datasets. In particular, SGCF hugely improves over the strongest GCN-based baseline on Amazon-Books by 10.16% and 10.15% by using Recall@20 and NDCG@20 respectively. The results of significance testing indicates that our improvements over the current strongest GCN-based baseline are statistically significant. In particular, SGCF show the effectiveness of modeling the information passing of a graph. NGCF is the baseline that captures higher-order user-item bipartite graph structure. It performs better than most baselines. Our proposed SGCF model consistently outperforms NGCF, thus showing the effectiveness

of modeling the user preference by the residual preference prediction and the linear embedding propagation. Compared with other baselines, SGCF can leverage powerful graph convolution to exploit useful and deeper collaborative information in graphs. These advantages jointly lead to the superiority of SGCF than compared state-of-the-art models. 2) In total, network embedding models perform worse than GCN-based models, especially on Gowalla. The reason might be that the powerful graph convolution is more effective than traditional random walk in many network embedding methods, to capture collaborative information for recommendation. 3) Since SGCF is a special fixed filter on the graph spectral domain, its architecture is orthogonal to some stat-of-the-art models (e.g., SGC). Therefore, similar to low-pass-type filters, SGCF can be deemed as an effective and efficient CF framework which is possible to be incorporated with other methods. such as enabling disentangled representation for users and items as DGCF, to achieve better performance.

5.4.3 Efficiency Comparison

As highlighted in Section 3.5, SGCF is endowed with high training efficiency for CF due to its concise and unified designs. In this section, we further empirically demonstrate the superiority of SGCF on training efficiency compared with other CF models, especially GCN-based models. To be specific, we select MF-BPR, ENMF, LightGCN, and LR-GCCF as the competitors, which are relatively efficient models in their respective categories. To be more convincing, we compare their training efficiency from two aspects: 1) The total training time and epochs for achieving their best performance. 2) Training them with the same epochs to see what performance they can achieve. Note that Table 5.3 shows that the training speed (i.e., Time per Epoch) of SGCF is close to MF-BPR, which empirically justifies our analysis that the time complexities of SGCF and MF are on the same level. SGCF needs 64 epochs to converge which is much less than LR-GCCF and LightGCN, leading to only 38.4 minutes for total training. Finally, SGCF has around 20x, 5x, 5x speedup compared with LightGCN, LR-GCCF, and ENMF respectively, demonstrating the big efficiency superiority of SGCF.

Moreover, Table 5.4 shows that when SGCF converges (i.e., train the fixed 64 epochs), the performances of all the other compared models are much worse than SGCF. That is to say, SGCF can achieve much better performance with less time, which further demonstrates the higher efficiency of SGCF that the other GCN-based CF models.

Table 5.4: Efficiency comparison with same epochs. All models are trained with the fixed 64 epochs except MF-BPR. Since MF-BPR needs less than 64 epochs to converge, we report its actual training time.

Model	Training Time	Recall@20	NDCG@20
MF-BPR	**16m**	0.0342	0.0264
ENMF	2h45m	0.0357	0.0281
LR-GCCF	1h25m	0.0314	0.0191
LightGCN	1h41m	0.0345	0.0264
SGCF	**43m**	**0.0682**	**0.0561**

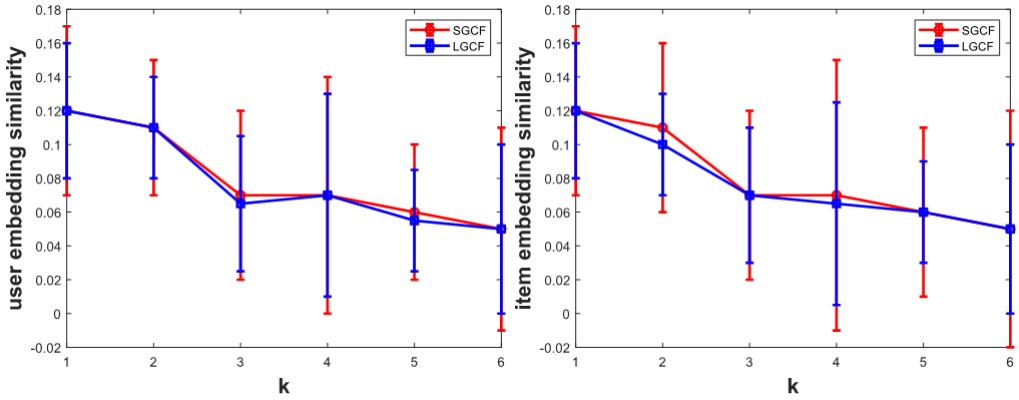

Figure 5.3: **(left):** Error-bar of user embedding similarity. **(right):** Error-bar of item embedding similarity. Comparisons with and without graph partition process structure under different layers depth k on Amazon-Books dataset.

5.4.4 SGCF Model Component Analysis

To explore the effect of different components in SGCF model, we design a simplified version that removes the graph partition module in our framework. We call the simplified version model as LGCF. For LGCF and SGCF, with each predefined depth k, we calculate the cosine similarity of each pair of nodes (i.e., users and items) between their k-layer output embedding, i.e., e^k for each node of the graph. The statistics of the mean and variance of user-user (item-item) embedding similarities are shown in Figure 5.3. It obviously shows our proposed model has larger variance of the user-user cosine similarity compared to its counterparts LGCF that does not perform condensed graph learning. This empirically validates that the condensed graph learning could partially alleviate the over smoothing issue, and achieves better performance. Please note that, the overall trend on the other three dataset is similar, and we do not illustrate it due to page limit.

Table 5.5: Performance of HR@20 and NDCG@20 with different depth k

Model	Amazon-Books		Gowalla	
	HR@20	NDCG@20	HR@20	NDCG@20
k=0	0.0284	0.0219	0.1379	0.1126
k=1	0.0317	0.0242	0.1506	0.1245
k=2	0.0327	0.0248	0.1504	0.1246
k=3	0.0337	0.0255	**0.1518**	**0.1561**
k=4	**0.0341**	**0.0324**	0.1496	0.1241
k=5	0.0340	0.0356	0.1504	0.1249

5.4.5 SGCF Model Parameter Study

5.4.5.1 Parameter Analysis

We would analyze the influence of the recursive label propagation depth k, and a detailed analysis of the learned embeddings of the preference prediction with condensed input graph in SGCF. Table 5.5 shows the results on SGCF with different k values. Specially, the layer-wise propagation part disappears when k=0, i.e., our proposed model degenerates to BPR. As can be observed from Table 5.5, when k increase from 0 to 1, the performance increase quickly on both datasets. For Amazon-Books, the best performance reaches with four propagation depth. Meanwhile, our model reaches the best performance when k=3 on Gowalla.

5.4.5.2 Scalability Analysis

As GCN-based networks are complex and contain such a large number of nodes in the real world application scenario, it is necessary for a model being feasible to be applied in the large-scale datasets. We investigate the scalability of SGCF model optimized by gradient descent, which deploys multiple threads for parallel model optimization. Our experiments are conducted in a computer server with 12 cores and 128GiB memory. We run experiments with different threads from 1 to 20. We depict in Figure 5.4 the speedup ratio vs. the number of threads. The speedup ratio is very close to linear, which indicates that the optimization algorithm of the SGCF is reasonably scalable.

5.5 Related Work

In this section, we briefly review some representative GCN-based methods and their efforts for model simplification toward recommendation tasks. With the development and

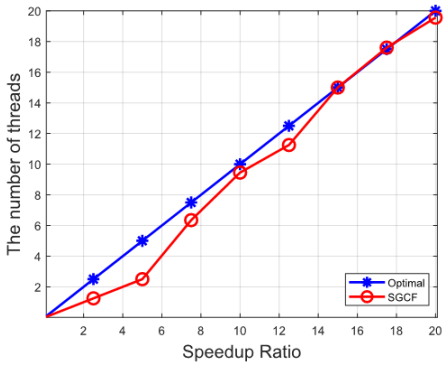

Figure 5.4: Scalability of SGCF

success of GCN in various machine learning areas, there appears a lot of users and items could be naturally formed to a user-item bipartite graph and adapted GCNs for recommendation [43, 45, 49, 79, 138, 165]. Earlier works on GCN based models relied on the spectral theories of graphs, and are computationally costly when applying in real-world recommendation. Some of recent works on GCN based recommendation models focused on the spatial domain [64]. PinSage was designed for similar item recommendation under the content based model, with the item feature x_v and the item-tiem correlation graph as the inputs. GC-MC [7] and NGCF are specifically designed under the CF setting. Although NGCF achieves good performance compared with previous non-GNN based methods, its heavy designs limit its efficiency and full exertion of GCN. To model the diversity of user intents on items, Wang et al. [153] devise Disentangled Graph Collaborative Filtering (DGCF) [139], which considers user-item relationships at the finer granularity of user intents and generates disentangled user and item representations to get better recommendation performance.

Although GCN-based recommendation models have achieved impressive performance, their efficiencies are still unsatisfactory when facing large-scale recommendation scenarios. How to improve the efficiency of GCNs and reserve their high performance for recommendation becomes a urgency research problem. Recently, Dai et al. [26] and Gu et al. [39] extend fixed-point theory on GNN for better representation learning. Liu et al. [80] propose UCMF that simplifies GCN for the node classification task. Wu et al. [144] find the non-necessity of nonlinear activation and feature transformation in GCN, proposing a simplified GCN (SGCN) model by removing these two parts. Inspired by SGC, He et al. [49] devise LightGCN for recommendation by removing nonlinear activation and feature transformation too. However, its efficiency is still limited by the time-consuming message passing. Qiu et al. [102] demonstrate that many network em-

bedding algorithms with negative sampling can be unified into the MF framework which may be efficient, however,their performances still have a gap between that of GCNs. We are inspired by these instructive studies, and propose SGCF for both efficient and effective recommendation.

5.6 Conclusion

In this chapter, we revisited the current GCN-based recommendation models and proposed an SGCF model for CF-based recommendation. SGCF consists of two main parts: First, with the recent progress of simple GCNs, we empirically removed the non-linear transformations in GCNs, and replaced it with linear embedding propagation. Second, to reduce the over smoothing effect introduced by higher layers of graph convolutions, we designed a condensed graph learning process for the input network. Extensive experimental results clearly showed the effectiveness and efficiency of our proposed model. In the future, we will explore better integration of different layers' representations with well-defined deep neural architectures to further enhance CF-based recommendation.

CHAPTER 6

A TOPIC-CONTROLLABLE KEYWORDS-TO-TEXT GENERATOR WITH KNOWLEDGE BASE NETWORK

6.1 Introduction

A keyword-to-text generation (K2T) problem seeks to create sentence-level texts that look like humans with only a few given keywords. Numerous application scenarios, including the generation of stories, reports, dialogue responses, second language, and other uses, have relied heavily on it [66, 118, 119]. A great deal of interest has been drawn to K2T because of its enormous potential in practical use and scientific research. Despite this, two problems remain to be solved in K2T: 1) the neglect of controllable text generation from unorder keywords and 2) the underutilization of topic-aware information.

An appropriately executed K2T generator should be able to generate a variety of vivid and varied sentences when keywords are used. However, existing work tends to produce generic and uncontrollable texts [116, 134, 151]. They ignore the information produced by text that is subject to topic control, which is one of the reasons. Our ability to produce text that is far more varied and fascinating is enhanced by modeling and controlling the subject matter that can be controlled by the generated text. As shown in Figure 6.1, given the keywords“Basketball”, “Exercise” and “Game”, the “without topic-controllable” models generate flat sentences. Meanwhile, the topic-controllable model generates controllable statements such as “The NBA is a game loved by basketball fans all over the world, and basketball is also a great exercise.” when given the topic

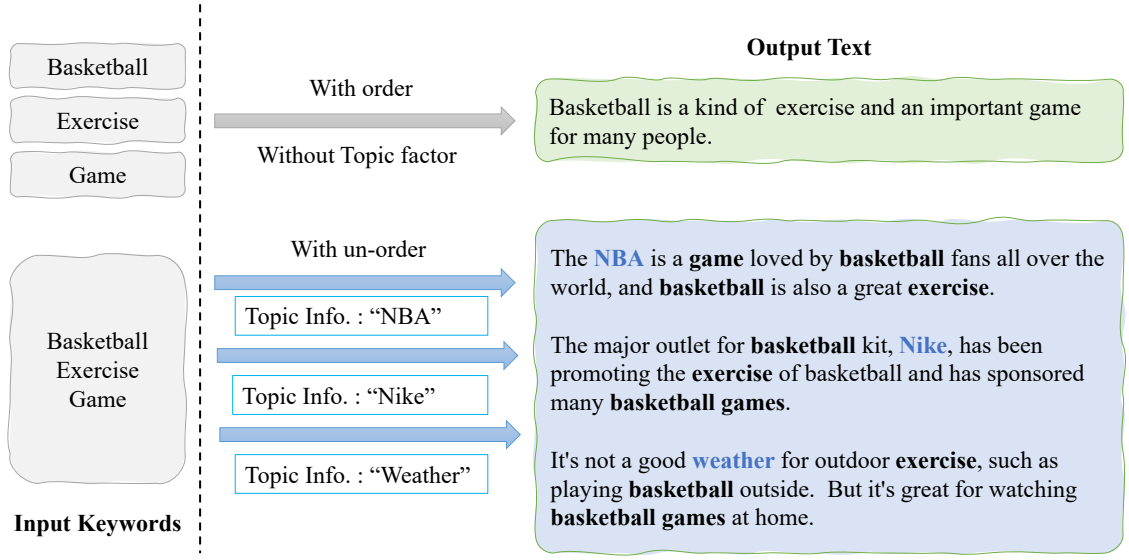

Figure 6.1: Examples of comparison between the generated text from ordered keywords with topic control and from un-ordered keywords without topic control. We show the first three sentences for each generated text and denote topic words in blue and keywords in black bold. Sentences without topic factor are showed in green text box.

of "NBA", and generates phrases such as "The major outlet for basketball kit, Nike, has been promoting the exercise of basketball and has sponsored many basketball games." when given the topic of "Nike". Additionally, topic control is critical to the K2T generation process, which aims to generate a variety of sentences. The search space for the generation model multiplies exponentially when the topic polarity for each sentence is controlled as the number of words increases. For the task of K2T generation, therefore, the ability to manage topics is essential to enhance diversity at the discourse level.

The fact that we humans rely heavily on our common sense knowledge when asked to write sentences with some keywords and related topics should also be taken into account. Because of this, K2T generation relies heavily on the proper utilization of knowledge. Early cutting-edge approaches based on fixed templates used a set of keywords and partial speech as input. [89]. Nevertheless, they disregard the network structure of the knowledge base, which only makes reference to concepts in the knowledge network [18] and neglects to consider their correlations, as well as the topic-controllable information. Due to this restriction, conceptions become dissociated from each other. As an illustration, given two keyword corpus inputs, delight, antonym, sadness, and delight, part of, emotion] about the topic word "delight", simply use the neighboring concepts sadness and emotion as a complement to the input information. While "emotion", which is a hypernym

to “delight”, is a hypernym that can be learned from their correlations(edges) in the knowledge network, their approach fails to recognize that “sadness” has the opposite meaning from “delight”. Intelligently, the lack of information about the relationships between concepts in a knowledge network makes it difficult for a model to construct useful and informative texts.

This article explores a novel topic-controllable keywords-to-text generator with a topic information network decoder called TC-K2T that is based on a proposed conditional language encoder framework in order to address the aforementioned issues. As part of our model's encoder and decoder, we inject topic-controllable information to control text content from unordered keywords in order to control the subject from two perspectives: word-level and sentence-level. The label of each topic is provided by a topic classifier during training process. Based on ConceptNet [122], a large-scale common sense knowledge base, the model recovers a topic knowledge network in order to fully utilize the information. Instead of preserving the network structure of the knowledge base [150], we provide a novel Topic Attention (TA) mechanism that is distinct from many existing methods. In order for future generations to benefit from structured, topic-controllable, connected data from networks, the TA conducts an in-depth review of knowledge networks. As a result, we employ adversarial training based on multi-label discriminator to make the generated text more closely related to the topic-controllable information and to include all input keywords. Depending on how much the output covers the given keywords, the discriminator rewards the generator.

In conclusion, we make the following significant contributions:

- We propose a novel topic-controllable keywords-to-text generator using the conditional language framework that is capable of producing high-quality text and controlling the subject. According to our knowledge, we are the first to apply topic-controllable information to the task of keyword-to-text generation and demonstrate the potential of our model to generate diverse text by controlling the topic at the sentence level.

- We propose an innovative Topic Attention (TA) mechanism and use a topic knowledge network to enhance our decoder. TAs make the most of the structured, aggregated subject data from the subject knowledge network, they are able to produce text that is more pertinent and informative.

- With the aid of extensive experiments, we validate that our model accurately controls the topic for text generation and outperforms cutting-edge methods in

both automatic and human annotation.

6.2 Related Work

A growing portion of research is being done on text generation as people rely more and more on automatic text generation in their daily lives. A fundamental model for generating text is the Seq2Seq model, which is based on attention. Among the tasks for text generation that the attention-based Seq2Seq model is effective at, are neural machine translation, abstract text summarization, dialogue generation, etc. In general, the Seq2Seq model has developed into one of the most well-known text generation frameworks.

RNNs are the foundation of the majority of Seq2Seq models, but recent research has led to frameworks based on CNNs and attention systems. In neural machine translation, the transformer has produced cutting-edge results and is rapidly becoming a popular framework for sequence-to-sequence learning as a result of its excellent performance and high efficiency. A transformer-based pre-trained language model called BERT is proposed to perform natural language processing tasks with the most sophisticated performance.

6.2.1 Controllable Text Generation

A challenging task in the development of natural languages is the automatic generation of text. For the first time, K.Uchimoto et al. [131] came up with a framework for generating sentences based on n-grams and dependency trees. They created the framework solely for the Japanese people. For generating the context before and after a single keyword input in Chinese, a recurrent neural network RNN-based model [127] was recently employed.

Methods for managing style for tasks involving text generation have been the subject of some research. Artificial inputs that can be used to generate controlled text have received considerable attention in the text-to-text domain [59]. Another recently proposed approach is the controllable plug-n-play language model developed by Dathathri et al. [28]. Although their generator is able to generate fluent output based on the control specification, the generation process is still open-ended and may not adhere to any user-desired syntax. An alternative framework for variational auto-encoder (VAE) [53], which offers minimal control options, including sentiment, has been created. According to Ghosh et al. [36], a method can be employed to determine the degree of emotional content in generated sentences. Moreover, since the system relies heavily on actual

textual data annotated with these categories and has a fixed set of emotion categories, it is unable to accept certain approaches. The linguistic properties of a text are controlled by a language model that is influenced by a particular style in a similar effort by Ficler and Goldberg [34]. As a result, there are several possible styles, including theme, sentiment, professional, and descriptive, that they use in the movie-review industry. Although the system lacks the ability to transform data, these styles may require only a small number of values to which the generated text ought to adhere. In the context of modern English texts, Jhamtani et al. [57] investigate an approach to applying Shakespearean English style. In order to replace words by copying the style, the model employs an external dictionary of stylistic words; this might not always maintain the intended meaning.

6.2.2 Topic-controllable Generation

A wide variety of research initiatives on topic controlled generation employ templates to control the direction of the sentences generated. Using templates provided in the form of "sparse" trees that are frequently used in a language, Iyyer et al. [55] propose syntactically controlled paraphrase network (SCPN). By relying on well-formed sentences and the accompanying complete parse trees, the system is unable to transform the input into data (shown by keywords). The fact that the system can accept input templates is noteworthy because they are both syntactically rigid and difficult to interpret. Chen et al. [17] proposes an approach that uses a sentence as a syntactic example rather than requiring an external parser. Although this system can take up keywords in any order, it is not intended to accept data/keywords for input (different from our system). Recently, a method [137] inspired by the data-to-text generation dataset generated sentences given a structured record, and a reference sentence. For fidelity to the structured material, manipulating the reference text (by rewriting, adding, or deleting portions of the text) is a different task. Our analysis reveals that keywords are not organized, even ordered, and may require morphological, syntactic, and numerical transformations (such as number, tense, and aspect change); as a result, it is not feasible to modify, add to, or delete portions of text Similar to the aforementioned, Laha et al. [65] proposes a modular system that converts input from structured data (tables) into canonical form, develops straightforward sentences from canonical data, and ultimately combines sentences to produce a coherent and fluent paragraph statement. This approach, which involves table row representations as a collection of binary relationships (or triples), differs from ours in terms of task size.

Table 6.1: Notations

Notation	Description
$\mathcal{N}$	the number of input keywords
K	keywords array $K = [k_1, k_2, \cdots, k_N]$
S	output sentences array $[s_1, s_2, \cdots, s_M]$
T	topic sequence array $T = [T_1, T_2, \cdots, T_M]$
$S_{1:i-1}$	the previous sentences as context
$KL(q\|\|p)$	the KL annealing technique
$softmax(.)$	the softmax operation
c	control code
s	the context S encoded as $[s_1, s_2, \cdots, s_{i-1}]$
$\mathcal{N}$	the prior network
$\mathbf{q}$	query vector
$\mathbf{W}_\alpha$	the weight matrices

As far as we are aware, there are still numerous challenges in translating order-invariant keywords into natural language text without subject-aware information.

6.2.3 Problem Formulation

In this section, we first introduce some fundamental concepts that are necessary to understand our model. The notations used in this paper are summarized in Table 6.1.

Moreover, we formulate the problem of automatic controllable language generation. The objective of the task is to build a system that can generate topic-aware sentences automatically based on the input keywords. Given the keywords represented as an input sequence of words $k = [k_1, k_2, \cdots, k_N] \in K$, the objective of the system is to generate the

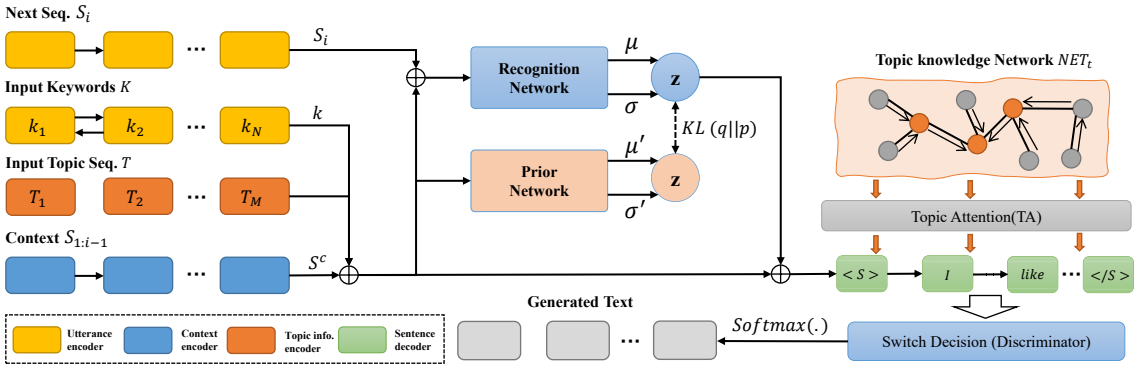

Figure 6.2: Our topic-controllable keywords-to-text generation framework.

topic-aware sentences $s = [s_1, s_2, \cdots, s_M] \in S$, a sequence of words describing the topic.

6.3 Framework Architecture

For generating topic-aware text from keywords, the framework takes as input a set of $\mathcal{N}$ keywords denoted $K = [k_1, k_2, \cdots, k_N]$, and aims to generate text with $\mathcal{M}$ sentences $[s_1, s_2, \cdots, s_M]$ corresponding to keyword sets K. In addition, in this research, we provide a topic sequence $T = [T_1, T_2, \cdots, T_M]$, each of which corresponds to a specific sentence in text. Entities or virtualities can be used for each topic.

The sentence-by-sentence process is used to create a continuous paragraph of text. After generating the first sentence s_1 based solely on the keyword sets K, the model continues to generate the following sentence using all the previously generated sentences and keyword sets until the entire text is finished. In this paper, the preceding phrases are represented as $S_{1:i-1}$.

The overall architecture is given in Figure 6.2, where $\oplus$ represents the vector concatenation operation. The $KL(q||p)$ represents the KL annealing technique. Topic sequence T denotes subject control. The orange solid arrows represent the TA process at each decoding step. The text generated by the TC-K2T generator is fed into the topic switching decision modules. The output gray blocks representing the given text are generated after a $softmax(.)$ operation.

Based on a Knowledge-Guided CVAE (kgCVAE [161] strategy consisting of an encoder and an enhanced topic knowledge network decoder), we have developed our TC-K2T generator. Keywords, topic sequences, and context are encoded by the encoder and are viewed as conditional variables c. A latent variable is then calculated from c using a recognition network (during training) or a previous network (during inference). A topic knowledge network and topic related information are connected by the decoder to create texts. Through effective use of the topic knowledge network, TA is utilized at each decoding step to enhance input topic information.

As part of the training process, we take the following two steps: (1) Train the TC-K2T generator with the kgCVAE loss; and (2) Next, we give a topic-controllable information discriminator to evaluate the performance of the TC-K2T generator. In order to further enhance the TC-K2T generator's effectiveness, we utilize adversarial training to train both the generator and the discriminator occasionally.

6.4 Topic-controllable Keywords-to-text model

6.4.1 Encoder Part

In a vector of some size, the keyword encoder aims to capture contextual representations of each keyword. In addition, it should be ensured that the encoding process is insensitive to the order of the input keywords. Using the last hidden states of the forward and backward Gate Recurring Unit(GRU) [24] as our utterance encoder, we utilize a bidirectional GRU to produce input sets in a vector of fixed size. We use the utterance encoder to encode the keyword sets K into $u^k = [\overrightarrow{h^k}, \overleftarrow{h^k}], h^k \in \mathbb{R}^d$, which d is the vector dimension. The next sequence S_i is also encoded by utterance encoded as $u^i = [\overrightarrow{h^i}, \overleftarrow{h^i}]$, $h^i \in \mathbb{R}^d$. According to the context encoder, inspired by [161], we utilize a strategy of multi-layer encoding. For each sentence, firstly, in context $S_{1:i-1}$ is encoded by utterance encoder to get a fixed-size vector. By doing so, the context $S_{1:i-1}$ is encoded as $s_{text} = [s_1, s_2, \cdots, s_{i-1}]$. Once this has been accomplished, a 1-layer forward GRU is used to encode sentence representations s_{text} into a final state vector $s^c \in \mathbb{R}$.

We then concatenate s^c, u^k and $e(t)$ (the embedding of topic information label), and define the conditional vector as $c = [e(t)|s^c, u^k]$. Since we assume z follows isotropic Gaussian distribution, the recognition network $q_\phi(z|s_i, c)$ and the prior network $p_\theta(z|c)$ follow $\mathcal{N}(\mu, \sigma^2\mathbf{I})$ and $\mathcal{N}(\mu', \sigma'^2\mathbf{I})$, respectively. $\mathbf{I}$ is identity matrix and then we have:

$$\begin{aligned} [\mu, \sigma^2] &= \mathrm{MLP}_{\text{recognition}}\ (s_i, c) \\ [\mu', \sigma'^2] &= \mathrm{MLP}_{\text{prior}}\ (c). \end{aligned} \tag{6.1}$$

We then use the reparametrization trick [161] to obtain samples of z either from $\mathcal{N}(z|\mu, \sigma^2\mathbf{I})$ predicted by the training recognition network or $\mathcal{N}(z|\mu', \sigma'^2\mathbf{I})$ predicted by the testing prior network.

6.4.2 Topic-controllable Decoder

In general, Seq2Seq models can produce sentences that are generic and meaningless. An enhanced topic knowledge network decoder is proposed to produce more meaningful text. The decoder is based on a 1-layer GRU network with initial state $s_0 = W_d[z, c, e(t)] + b_d$. W_d and b_d are trainable decoder parameters and $e(t)$ is the embedding topic as mentioned above. As shown in Figure 6.2, we built the decoder with a topic knowledge network to incorporate commonsense knowledge from ConceptNet [1] [123]. A semantic network

[1] https://conceptnet.io

called ConceptNet is intended to assist computers understand the meanings of words that people utilize. As triples of the start, connection label, and end nodes, this type of network is represented. Relationship exists between the end node and the start node. We use word vectors to represent start and end concepts and learn a trainable vector $\mathbf{v}^{rel}$ for the relation, which is randomly initialized. Our approach consists of learning trainable vectors for relation that are randomly initialized and using word vectors to represent start and end concepts. Using each word in the keyword sets as a query, ConceptNet is used to locate a subnetwork that forms the topic knowledge network. After that, we use the Topic Attention (TA) mechanism to read from the topic knowledge network at each generation stage.

It is essential for the success of our work that external expertise is used properly, as already stated. TA takes as input the retrieved topic knowledge network and query vector $\mathbf{q}$ to produce a network vector NET_t. We set $\mathbf{q} = [d_{t-1}, c, z]$, where d_{t-1} represents the hidden state of the decoder for step $t-1$, c is the conditional vector and sample z from the recognition network.

Our algorithm calculates the correlation score between each of the triples in the network and $\mathbf{q}$ during decoding, at each stage t. After that, the weighted sum of all the neighboring concepts to the topic terms is calculated using the correlation score to create the final network vector NET_t.

According to reports, neighboring things are those that are directly connected to topic terms. We denote the embedding of n^{th} neighboring concept as o_n, then NET_t can be defined as:

$$NET_t = \sum_{n=1}^{N} \alpha_n \boldsymbol{o}_n \tag{6.2}$$

In order to capture important information, we put an attention on our decoding process, the attention weights on query are computed by:

$$\frac{\exp(f_n)}{\sum_{j=1}^{N} \exp\left(f_j\right)} \tag{6.3}$$

where

$$f_\alpha = (\mathbf{W}_\alpha \mathbf{q})^T \tanh\left(\mathbf{W}_\alpha \mathbf{v}_n^{rel} + o_n\right) \tag{6.4}$$

A weight matrix for queries, relationships, start entities, and end entities is represented by $\mathbf{W}_\alpha$. Additionally, adjacent concepts, which are the start/end ideas in their triples, fall under the category o_n. The correlation between the query q and the neighboring concept

o_n is represented by the matching score f_α. It can measure the topic relationship between the i-th word in the source and the j-th target word to be predicted. In essence, it makes up a network vector NET_t by combining adjacent concepts of topic words. Be aware that different weight matrices are utilized to distinguish between the neighbouring concepts in various positions (in start or in end). This distinction is necessary, for instance, in the light of two triples of knowledge (Opera House, part of, Sydney) and (Sydney , part of, Australia). The concepts Opera House and Australia have different meanings for Sydney despite the fact that they are both adjacent concepts to Sydney with the same connection component. We need to model this difference in the weight matrices set $\mathbf{W}_\alpha$.

In order to calculate the final chance of generating a word, the following steps must be taken:

$$\mathscr{P}_t = \text{softmax}(\mathbf{W}_o[d_t; e(t); NET_t] + b_o), \tag{6.5}$$

where d_t is the decoder state at t step and $\mathbf{W}_o \in \mathbb{R}^{d_{all} \times |V|}, b_0 \in \mathbb{R}^{|V|}$ are trainable decoder parameters, d_{all} is the dimension of $[d_t; e(t); NET_t]$ and $|V|$ is vocabulary size.

A strong correlation needs to be established between the generated text and keywords and topic terms. We use a soft switcher to figure out if a word should be generated as the target word by using $\lambda_j \in [0, 1]$:

$$\lambda_j = sigmoid(\mathbf{W}_\lambda[e(t)]) \tag{6.6}$$

with $\mathbf{W}_\lambda$ being learnable parameter. This section also contains information that can be controlled by topics $e(t)$ to guide the switch selection. Further, the sigmoid probability distribution over $(m+1)$ classes [143]. According to the $(m+1)^{th}$ index, the probability that the sample is the generated text is represented by the score. The likelihood of it being a real text with the j^{th} topic is represented by the score on the j^{th} index.

6.4.3 Model Training

Throughout this section, we discuss the two-step training method. The first one is similar to a conventional kgCVAE model. The loss of our TC-K2T generator $-logp(Y|c)$ can be defined as:

$$\begin{aligned} -\mathscr{L}(\theta; \phi; c; Y)_{kgcvae} &= \mathscr{L}_{\text{KL}} + \mathscr{L}_{\text{decoder}} \\ &= \text{KL}\left(q_\phi(z \mid Y, c) \| p_\theta(z \mid c)\right) \\ &- \mathbb{E}_{q_\phi(z|Y,c)}(\log p_{\text{D}}(Y \mid z, c)) \end{aligned} \tag{6.7}$$

This parameter list includes θ and ϕ for the recognition network and the prior network, respectively. Intuitively, $\mathscr{L}_{\text{decoder}}$ maximizes the sentence generation probability after sampling from the recognition network, while $\mathscr{L}_{\text{KL}}$ minimizes the distance between the prior and the recognition network. Our adversarial training between the generator and the topic label discriminator described above begins after training the TC-K2T generator with equation 6.7, inspired by SeqGan [155]. Additional information is provided to the reader using the SeqGan approach due to page restrictions. Besides, we use the annealing trick and BOW-loss equation to alleviate the vanishing latent variable problem in VAE training.

6.5 Experiments

Throughout this section, we discuss the dataset, evaluation metrics, all baselines, and settings in more detail.

6.5.1 Datasets

We conducted experiments on the Quora corpus [2]. Experiments were done on the datasets, which is between 30 and 120 words in length. In light of the frequency of each keyword, we choose words from the NOUN, VERB, ADJECTIVE, and ADVERB categories as input keywords and remove uncommon keywords. There are 30,000 and 3,000 test sets for training and testing, respectively. As the validation setting, we utilized 12% of the training samples to tune the hyperparameters.

Our method of introducing topic labels involved manually annotating items with 100 categories, such as sports, beauty, and so on, using 3000 sentences. To fine-tune our manually labeled training set, we utilize the topic model proposed by Zandie et al. [157], which achieves an accuracy of 0.87 on the test set. By using an automatic topic extractor during training, the target topic label s is calculated. The direction of each text that is generated is controlled by user input of any topic labels throughout inference.

6.5.2 Settings

In order to implement topic embeddings, we utilize 120-dim pre-trained word embeddings with 32 dimensions. The vocabulary size is 50,000 and the batch size is 64. Selecting the hyperparameter values uses a manually tuned procedure, and the criteria chosen

[2] https://www.kaggle.com/competitions/quora-question-pairs/data

Table 6.2: Automatic and human annotations result. In human annotation, four level (L4 - L1) to quantify : topic-consistency, novelty, text-diversity, fluency and informative. The best performance is highlighted with underline.

	Automatic evaluation							Human annotation			
Methods	BLEU	Dist-1	Dist-2	Consis.	PPL(T)	PPL(D)	Novelty	L4	L3	L2	L1
S2SA-MMI [148]	6.07	4.81	21.64	9.20	147.04	143.11	67.98	0.13	0.23	0.38	0.26
TRANS [89]	6.32	5.01	22.03	26.51	134.23	144.47	71.23	0.19	0.34	0.21	0.26
CONCAT [89]	7.01	5.12	22.54	35.54	141.45	156.91	72.45	0.27	0.23	0.25	0.25
TAV [31]	6.52	5.32	22.43	16.57	131.67	148.52	69.45	0.23	0.18	0.29	0.30
NMT [2]	7.12	5.31	22.67	32.67	128.63	149.34	72.34	0.28	0.21	0.24	0.27
CTEG [150]	9.72	5.92	23.07	39.32	127.27	142.37	73.39	0.31	0.18	0.31	0.20
TC-K2T(w/o-Topic)	10.01	5.64	23.21	44.12	119.55	140.45	78.26	0.36	0.22	0.21	0.21
TC-K2T(R-Topic)	9.98	5.86	23.11	42.01	122.81	133.81	80.12	0.42	0.27	0.17	0.14
TC-K2T(Pro-Topic)	12.01	6.01	23.08	42.63	123.14	134.63	78.87	0.45	0.24	0.19	0.12

is BLEU. We employ GRUs with a hidden size of 512 and a hidden size of 300 for both encoders and decoders. We develop the model with Tensorflow framework [3]. With a total parameter count of 72 MiB, our model parameters were randomly selected over a uniform distribution [0.1, 0.1]. We pre-train our model for 65 epochs with the Maximum Likelihood estimation model [157] and adversarial training [29] for 20 epochs. Our model is pre-trained with the Maximum Likelihood estimation model for 65 epochs and with adversarial training for 20 epochs. The average runtime for our model is 35 hours on an Intel(R) i7 CPU @ 2.50GHz, 512GB RAM and 2 NVIDIA 1060Ti-16GB GPUs. The optimizer is Adam [63] with 10^{-3} learning rate for pre-training and 10^{-5} for adversarial training. In addition, to prevent overfitting [86] with the dropout rate of 0.2 and gradients to the maximum norm of ten, we use dropout on the output layer. The average length of the generated text is 56.2, and greedy search is used in our model's decoding strategy.

6.5.3 Evaluation metrics

Both human annotations and automatic evaluation are employed by us in order to thoroughly assess the generated text.

BLEU [96]: By measuring word overlap between ground truth and generated sentences, the BLEU score is frequently used in machine translation, conversation, and other text generation tasks.

Distinct-1 & distinct-2: Several distinct bigrams and unigrams were taken into account in the responses generated. Furthermore, we split the numbers by the total number of unigrams and largerams using [69]. We define metrics as distinct-1 and distinct-2, respectively, for both numbers and ratios. Using both metrics, one can assess

[3]https://github.com/tensorflow/tensorflow

how informative and varied the text produced is. In addition, high numbers indicate that the produced text is lengthy, and high numbers and high ratios indicate that there is a lot of content in it.

Consistency [150]: Using all the keystrokes entered, a suitable text should surround a particular topic closely. For the purpose of evaluating the topic consistency of the output, we use a multi-label classifier pretrain. Higher scores on "Consistency" indicate that the generated texts are more closely related to the given topics. Taking into account the input topics T, the topic consistency of the generated text $\hat{\boldsymbol{y}}$ is defined as:

$$\text{Consistency}(\hat{\boldsymbol{y}} \mid \boldsymbol{x}) = \varphi(\boldsymbol{x}, \hat{\boldsymbol{x}}) \tag{6.8}$$

where φ is Jaccard similarity function and $\hat{\boldsymbol{x}}$ is topics predicted by a pre-trained multi-label classifier.

Perplexity: following [73], we employ perplexity as an evaluation metric. Perplexity is defined by Eqa.(6.9). Higher generation performance is associated with a lower perplexity rating. The purpose of this work is to determine when training will end using PPL(D). For five validation scenarios, if the perplexity stops decreasing and the difference is less than 2.0, we think the algorithm has reached an agreement and the training is terminated. In the test data, we use PPL(T) to evaluate the ability of various models to be produced.

$$PPL = \exp\left\{-\frac{1}{N}\sum_{i=1}^{N}\log(p(\mathbf{Y_i}))\right\} \tag{6.9}$$

Novelty [150]:Our analysis of novelty took into account the difference between output and text with similar subjects in the training corpus. Increased "Novelty" scores indicate that the text in the output corpus is more distinct from that in the training corpus.

Precision, Recall & F1: For the purpose of determining the accuracy of theme control, these metrics are used. A valid result can be obtained if the topic labels on the generated sentences are consistent with the ground truth. We predict topic labels using the topic classifier.

Human annotation:For the purpose of evaluating the quality of the created text of various models, we also employ human annotators. A total of three annotators with extensive Quora knowledge are invited to participate in the evaluation. Random shuffling and pooling of text created by different models is performed for each annotator. Test messages are reviewed by annotators who evaluate the quality of the text according to the following criteria:

Level-4 (L4): The generated text shows the topic consistency and novelty. Meanwhile, the text represents not only the text-diversity, but also the natural and fluent degree.

Level-3 (L3): The output text has a clear topic-aware direction and the sentence pattern is natural and smooth.

Level-2 (L2): The output text is just natural and informative.

Level-1 (L1): The text is difficult to understand, and is either semantically irrelevant or disfluent.

6.5.4 Baseline

Based on previous state-of-the-art text generation approaches, we make the following baseline measurements:

S2SA-MMI: With an attention mechanism in a standard Seq2Seq model, it performs at its best in [70]. We utilized the bidirectional-MMI decoder and, in accordance with the paper's suggestions, set the hyperparameter λ to 0.5.

TRANS: keywords can only be entered without requiring any controllable operations [89].

NMT [2]: The purpose of this paper is to describe a neural machine translation that uses an encoder-decoder framework based on LSTM and only accepts keywords as input without any topic control mechanisms.

CONCAT [89]: Words and templates are concatenated as input by the transformer-based framework.

TAV [31]: All topic embeddings' average topic semantics is used to create each word using an LSTM. The semantic correlation between each topic word and output of the generator is modeled using an attention mechanism that extends LSTM.

CTEG [150]: To improve the generation process, a combination of common sense knowledge and adversarial training was suggested. This work achieves state-of-the-art performance on the topic-to-essay generation task.

6.5.5 Quantitative Performance Comparison

As shown in Table 6.2, we list automatic and human evaluation results. We provide three different versions of our model for a comprehensive comparison. (a) "TC-K2T(w/o-Topic)" means that we do not attach any topic information to the model. (b) "TC-K2T(R-Topic)" means that we randomly set the topic information for each generated text. (c) "TC-

Table 6.3: Text quality analysis results, "w/o AT" represents without adversarial training. "w/o TGA" represents without TA. T-Con.(topic-consistency), Nov.(novelty), T-div.(text-diversity) and Flu.(Fluency) represents different text quality. Full model shows TC-K2T(Pro-Topic) in this table.

Methods	BLEU	T-Con.	Nov.	T-div.	Flu.
Full model	**12.01**	**3.92**	**3.26**	**4.01**	**3.81**
w/o TA	10.32	3.61	3.18	3.89	3.39
w/o AT	9.72	3.31	3.51	3.92	3.54

Table 6.4: Topic control analysis. "w/o En-topic" represents to remove the topic embedding in the encoder process and "w/o De-topic" represents to remove from the decoder. Full model represents TC-K2T(R-Topic) in this table.

Methods	Pre.	Recall	F1
Full model	**0.71**	**0.69**	**0.68**
w/o En-topic	0.53	0.52	0.53
w/o De-topic	0.56	0.63	0.62
w/o TA	0.63	0.65	0.64

Table 6.5: Given keywords "Cabbage", "Vegetable", "Diet" and "Option", and set a topic word "Health". We generated an text according to the topic with keywords.

Input keywords: Cabbage, Option, Vegetable, Diet
Input Topics: Health
Output text: Cabbage is a leafy green vegetable that is often overlooked, but it is actually quite healthy. Cabbage is a good source of vitamins C and K, as well as fiber and antioxidants. Additionally, cabbage has been shown to have a number of health benefits, including reducing the risk of cancer, improving heart health, and aiding in Digestion. So if you're looking for a healthy vegetable to add to your diet, cabbage is a great option.

K2T(Pro-Topic)" represents how we set the high frequency topic information used for sentence generation. The results in Table 6.2 show the following conclusions:

- Whether it is with hot topics, random topics, or without topic, all variants of our models outperform the baselines in all evaluation metrics, demonstrating the proposed TC-K2T model's capacity to generate better texts than baseline models.
- The superiority of our model architecture can be seen in the comparison of TC-K2Ts (w/o-Topic) and baselines. Based on human annotation results, TC-K2T(Pro-Topic) performs best on level-4 metrics. Most significantly, there has been an

improvement in text diversity. We achieve this improvement with our kgCVAE architecture because sampling a continuous latent variable serves as our sentence representation. Compared to baselines, this sampling procedure introduces more randomness.

- The "L4" texts exhibit a 0.29 increase in TC-K2T(R-Theme) compared to S2SA-MMI, while the "L1" texts display a 0.12 decrease. Compared to TC-K2T(w/o-Topic), TC-K2T(R-Topic) performs better, demonstrating that topic content contributes more to text quality than bias probability in generation.

- Thermal topic information is used by TC-K2T(Pro-Theme), which performs well in BLEU with a score of 12.01. In other metrics, TC-K2T(Pro-Theme) does not significantly outperform other TC-K2T models, according to our analysis. These results demonstrate that our suggested model is more appropriate for the text set because of the hot topic information of the target texts, although there is no obvious improvement for other important indicators such as Distinct, Consistency, and Perplexity.

- When the topic information is removed, we find it intriguing that TC-K2T(w/o-Topic) achieves the best topic-consistency score. However, the effect of this interference is trivial because we believe that topic labels may somehow interfere with the subject information in the latent variable. For automated evaluation, we find that the topic consistency for TC-K2T(w/o-Topic) and TC-K2T(Pro-Topic) drops by only 1.49 (44.12 vs 42.63), which is completely acceptable for a model that can handle subjects.

6.5.6 Text Quality Analysis

Both ablated versions of our main model are trained to better comprehend how each component of our model contributes to the task: 1) removing adversarial training - "w/o AT", 2) removing TA - "w/o TA". Moreover, we employ a memory network [122] in the "w/o TA" experiment that incorporates ConceptNet concepts but does not take into account their correlation. All models use frequency topic words. According to the findings in Table 6.3, the human annotation and BLEU scores of the ablation study are presented. A score for the generated text is obtained using $[1:5]$ using four metrics (Novelty, Fluency, Topic-Consistency, and Text-Diversity) in order to assess the quality of the text. In order

to offer annotators a reference, we use the TF-IDF features of topic words to find the 20 training samples that are most similar for "novelty".

With the exception of the topic attention layer, we find that the full model and "w/o TA" both have lower model performance in all metrics. In instance, topic-consistency dropped by 0.31, demonstrating that concepts with stronger connections to subject words receive greater attention during generation when the relationship between those concepts and the topic words is explicitly learned. TA is an expansion of the information contained in the external knowledge network, resulting in a drop of 0.08 in novelty. As a result, the text output is more novel and informative. The TA provides our model with the benefit of selecting a concept that is more appropriate in the topic knowledge network in the current context, which results in a 0.42 decrease in fluency. In addition, the BLEU decreases to 1.69, demonstrating TA's contribution to our model's ability to better fit the dataset by modeling the connections between topic words and nearby concepts.

We find that adversarial training can improve BLEU, topic consistency, and fluency by comparing the complete model and the "w/o AT". As a result, the discriminative signal increases the topic consistency and authenticity of the texts that are generated.

6.5.7 Topic Control Analysis

Our focus in this section is on whether the model properly regulates the topic and how each component influences our topic control performance. Our model is trained in three ablated versions: 1) without topic information in the encoder, 2) without subject information in the decoder, and 3) without TA. We randomly sampled 120 texts in our test set with 510 sentences. Instead of using frequent topic words, all topic inputs are randomly given in this section. Predicting the topic is relatively straightforward because there are times when these types of terminology can be directly related to context consistency. To generate sentences based on arbitrary information about the topic, we employ a more difficult experimental setting.

According to Table 6.4, removing the topic embedding from the encoder or decoder has the greatest impact on control performance, and the topic embedding in the encoder is the most significant, since removing it results in the greatest decline. It demonstrates that learning correlations between concepts in the topic knowledge network enhances the model's ability to control topics even though TA does not directly impose topic information. For example, when giving information about a "sports" topic, concepts related to the relationship "basketball games" are more likely to gain attention, because concepts

with the relationship "basketball games" have a certain probability of matching "sports" meaning.

6.5.8 Case Study

In this section, we present a case study of the texts we have actually created. Table 6.5 presents an instance of our output text with a random topic sequence. Keywords are shown in blue and topic-controllable words are shown in red. As we learn, the output text is closely related to the topic words in addition to covering all the input keywords. The TA assists us in developing our model, which makes full use of common sense knowledge. For example, “fiber and antioxidants” and “reducing the risk of cancer” are correlation concepts related to the topic words "Health".

6.6 Conclusions and Future Work

This chapter proposes a novel topic-controllable text generator with an enhanced decoder for topic knowledge networks called TC-K2T model, which is the first attempt at the challenging task of keywords-to-text generation.

The chapter focuses on generating coherent and topic-relevant text from keywords, leveraging a knowledge base network to ensure the accuracy and contextual relevance of the generated content. This ability to control the topic of generated text has significant implications for user behavior analysis.

On the other hand, our thesis theme centers around analyzing user behavior through graph-structured representations. Graphs are effective in capturing the intricate relationships and patterns within user data, providing valuable insights into user preferences and behaviors. These insights can, in turn, inform the keywords and topics that are fed into the topic-controllable keywords-to-text generator.

By combining the topic-controllable text generation capabilities of the chapter’s approach with the user behavior analysis techniques of the thesis’s theme, we can create a more comprehensive understanding of user preferences and needs. This, in turn, can lead to more personalized and effective user experiences, as well as improved decision-making based on a deeper understanding of user behavior.

In conclusion, the chapter provides insights into how advancements in structured information utilization can enhance both text generation and user behavior analysis.

Considering the recent success of multimedia sharing platforms, the content posted on these online social media websites contains a wealth of multimedia information (e.g., text and images). A fascinating future course would involve exploiting these multi-modality features. In addition, extending our model to address multimodal generation by introducing topic interaction and label information can be considered a new research line in this field.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

User behavior analysis has always been a hot issue in scientific research. The automatic feature extract and powerful representative ability of graph-structured learning make it an important learning machine for handling various downstream tasks, such as prediction problems and information generation tasks. This thesis has proposed a series of deep learning methods based on graph-based representation techniques. It first focused on the spam review detection problem and introduced Graph-aware Deep Fusion Networks (GDFN) that utilizes information from relevant metadata (review text, features of users, and items) and relational data (network) to capture the semantic information from their complex heterogeneous interactions via graph convolutional networks. Then, it designed a temporal framework with user and item level hypergraphs to enhance CTR prediction. Further, it simplified the GCN-based CF models from two aspects: remove non-linearities operations and utilized the condensed graph. Finally, it explored the topic-controllable keywords-to-text generator with a user information network to help improve the generated language is more in line with user preferences.

The first and foremost contribution of this thesis is the combination of graph-based methods in user behaviour modeling. It will both structured and unstructured data within a predictive model to provide meaning insights to user behaviour. This novel approach would directly contribute to the current literature of in both user behaviour and graph-structured representation research fields. It will help e-commerce platforms

with better consumer understanding and user experiences.

Online spam review detection In this thesis, we propose a novel graph-based model, namely Graph-aware Deep Fusion Networks (GDFN) that utilizes information from all metadata (review text, features of users and items) as well as relational data (network) to capture the semantics from their complex heterogeneous interactions via graph convolutional networks. Besides, GDFN also uses a novel fusion technique to synthesize low and high-order interactions with propagated information across multiple review-related sub-graphs.

Click-through rate Prediction We propose a model to exploit the temporal user-item interactions to guide the representation learning with multi-modal features, and further predict the user click rate of the micro-video item. We design a Hypergraph Click-Through Rate prediction framework (HyperCTR) built upon the hyperedge notion of hypergraph neural networks, which can yield modal-specific representations of users and micro-videos to better capture user preferences. We construct a time-aware user-item bipartite network with multi-modal information and enrich the representation of each user and item with the generated interests-based user hypergraph and item hypergraph.

GCN Simplification We empirically reduce the excessive complexity of GCNs by repeatedly removing the nonlinearities between GCN layers and collapsing the resulting function into a single linear transformation. Further, the proposed model, SGCF, which largely simplifies the model design by including only the most essential components in GCN for more efficient recommendations. We offer an effective partition technique for reducing the scale of input graph structure to avoid infinite layers of explicit message passing for efficient recommendations.

Natural Language Generation We propose a novel topic-controllable keywords-to-text generator using the conditional language framework that is capable of producing high-quality text and controlling the subject. According to our knowledge, we are the first to apply topic-controllable information to the task of keyword-to-text generation and demonstrate the potential of our model to generate diverse text by controlling the topic at the sentence level. In addition, an innovative Topic Attention (TA) mechanism and a topic knowledge network was used to enhance our decoder. TAs make the most of the structured, aggregated subject data from the subject knowledge network, they are able to produce text that is more pertinent and informative.

7.2 Future Work

Although this study involves several contributions to the advancement of graph-based methods, some experts argue that graph learning is almost a block box model hence lacking interpretability and theoretical support. Therefore, there are still many improvements and concerns that need to be addressed in the graph learning research. The following research directions could serve as worthwhile future study:

- Improving interpretability: Developing methods and techniques to enhance the interpretability of graph learning models is crucial. This could involve devising visualization techniques, feature importance analysis, or model explanation methods specific to graph-based models. By gaining a better understanding of the inner workings of these models, researchers can increase the trust and transparency in the results they provide.

- Theoretical foundations: Building a strong theoretical foundation for graph learning is essential to ensure its validity and generalizability. This could involve exploring the mathematical principles and statistical properties behind graph-based methods. Establishing rigorous theoretical frameworks can provide a solid basis for understanding the behavior of these models and enable researchers to make more informed decisions about their applicability in different scenarios.

- Robustness and generalization: Investigating the robustness and generalization capabilities of graph learning models is an important aspect to consider. Understanding how these models perform under different conditions, such as varying graph structures, noise levels, or data distribution shifts, will help assess their reliability and enable the development of more robust algorithms.

- Incorporating domain knowledge: Integrating domain knowledge into graph learning models can greatly enhance their effectiveness and applicability. This could involve designing algorithms that combine graph-based techniques with expert knowledge or incorporating prior information about the underlying problem into the learning process. By incorporating domain knowledge, researchers can provide more meaningful and domain-specific insights from the graph data.

- Scalability and efficiency: Addressing the scalability and efficiency challenges associated with graph learning is another important research direction. Developing algorithms that can handle large-scale graphs or process data in a computationally

efficient manner is essential for real-world applications of graph-based methods. Exploring parallel processing techniques, distributed computing frameworks, or graph partitioning strategies can aid in tackling these challenges effectively.

By exploring these research directions, future studies can contribute to overcoming the limitations of graph learning methods, fostering their interpretability, refining their theoretical foundations, and enhancing their practical utility. Through these endeavors, graph learning can continue to evolve into a more mature and robust field, offering valuable insights and solutions in various domains.

BIBLIOGRAPHY

[1] S. ALI ALHOSSEINI, R. BIN TAREAF, P. NAJAFI, AND C. MEINEL, *Detect me if you can: Spam bot detection using inductive representation learning*, in WWW, 2019, pp. 148–153.

[2] D. BAHDANAU, K. CHO, AND Y. BENGIO, *Neural machine translation by jointly learning to align and translate*, arXiv preprint arXiv:1409.0473, (2014).

[3] A. R. BAIG AND H. JABEEN, *Big data analytics for behavior monitoring of students*, Procedia Computer Science, 82 (2016), pp. 43–48.

[4] T. BALTRUŠAITIS, C. AHUJA, AND L.-P. MORENCY, *Multimodal machine learning: A survey and taxonomy*, TPAMI, 41 (2018), pp. 423–443.

[5] T. BALTRUSAITIS, C. AHUJA, AND L. P. MORENCY, *Multimodal Machine Learning: A Survey and Taxonomy*, TPAMI, 41 (2019), pp. 423–443.

[6] J. BARCENILLA AND J.-M.-C. BASTIEN, *Acceptability of innovative technologies: Relationship between ergonomics, usability, and user experience*, Le travail humain, 72 (2009), pp. 311–331.

[7] R. V. D. BERG, T. N. KIPF, AND M. WELLING, *Graph convolutional matrix completion*, arXiv preprint arXiv:1706.02263, (2017).

[8] T. BIAN, X. XIAO, T. XU, P. ZHAO, W. HUANG, Y. RONG, AND J. HUANG, *Rumor detection on social media with bi-directional graph convolutional networks*, in AAAI, vol. 34, 2020, pp. 549–556.

[9] M. BLONDEL, A. FUJINO, N. UEDA, AND M. ISHIHATA, *Higher-order factorization machines*, arXiv preprint arXiv:1607.07195, (2016).

[10] A. BRETTO, *Hypergraph theory*, Springer, (2013).

[11] A. BULUÇ, H. MEYERHENKE, I. SAFRO, P. SANDERS, AND C. SCHULZ, *Recent advances in graph partitioning*, Algorithm engineering, (2016), pp. 117–158.

[12] M. CALLARA AND P. WIRA, *User behavior analysis with machine learning techniques in cloud computing architectures*, in 2018 International Conference on Applied Smart Systems (ICASS), IEEE, 2018, pp. 1–6.

[13] C. CASTILLO, M. MENDOZA, AND B. POBLETE, *Information credibility on twitter*, in WWW, 2011, pp. 675–684.

[14] H. CHEN, Y. LI, X. SUN, G. XU, AND H. YIN, *Temporal meta-path guided explainable recommendation*, in Proceedings of the 14th ACM International Conference on Web Search and Data Mining, WSDM '21, New York, NY, USA, 2021, Association for Computing Machinery, p. 1056,Äì1064.

[15] J. CHEN, T. MA, AND C. XIAO, *Fastgcn: fast learning with graph convolutional networks via importance sampling*, arXiv preprint arXiv:1801.10247, (2018).

[16] L. CHEN, L. WU, R. HONG, K. ZHANG, AND M. WANG, *Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach*, in Proceedings of the AAAI conference on artificial intelligence, vol. 34, 2020, pp. 27–34.

[17] M. CHEN, Q. TANG, S. WISEMAN, AND K. GIMPEL, *Controllable paraphrase generation with a syntactic exemplar*, arXiv preprint arXiv:1906.00565, (2019).

[18] Q. CHEN, J. LIN, Y. ZHANG, H. YANG, J. ZHOU, AND J. TANG, *Towards knowledge-based personalized product description generation in e-commerce*, in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 3040–3050.

[19] T. CHEN AND C. GUESTRIN, *Xgboost: A scalable tree boosting system*, in KDD, 2016, pp. 785–794.

[20] X. CHEN, D. LIU, Z. J. ZHA, W. ZHOU, AND Y. LI, *Temporal hierarchical attention at category- and item-level for micro-video click-through prediction*, 2018.

[21] Y. R. CHEN AND H. H. CHEN, *Opinion spammer detection in web forum*, 2015, pp. 759–762.

[22] Z. CHENG, X. CHANG, L. ZHU, R. C. KANJIRATHINKAL, AND M. KANKANHALLI, *Mmalfm: Explainable recommendation by leveraging reviews and images*, TOIS, 37 (2019), pp. 1–28.

[23] Z. CHENG, S. JIALIE, AND S. C. HOI, *On effective personalized music retrieval by exploring online user behaviors*, in SIGIR, 2016, pp. 125–134.

[24] J. CHUNG, C. GULCEHRE, K. CHO, AND Y. BENGIO, *Empirical evaluation of gated recurrent neural networks on sequence modeling*, arXiv preprint arXiv:1412.3555, (2014).

[25] Z. CUI, H. CHEN, L. CUI, S. LIU, X. LIU, G. XU, AND H. YIN, *Reinforced kgs reasoning for explainable sequential recommendation*, World Wide Web, (2021), pp. 1–24.

[26] H. DAI, Z. KOZAREVA, B. DAI, A. SMOLA, AND L. SONG, *Learning steady-states of iterative algorithms over graphs*, in International conference on machine learning, PMLR, 2018, pp. 1106–1114.

[27] T. DAO, K. DUONG, AND C. VRAIN, *A filtering algorithm for constrained clustering with within-cluster sum of dissimilarities criterion*, in ICTAI, 2013, pp. 1060–1067.

[28] S. DATHATHRI, A. MADOTTO, J. LAN, J. HUNG, E. FRANK, P. MOLINO, J. YOSINSKI, AND R. LIU, *Plug and play language models: A simple approach to controlled text generation*, arXiv preprint arXiv:1912.02164, (2019).

[29] J. DING, Y. QUAN, X. HE, Y. LI, AND D. JIN, *Reinforced negative sampling for recommendation with exposure data.*, in IJCAI, 2019, pp. 2230–2236.

[30] T. EBESU, B. SHEN, AND Y. FANG, *Collaborative memory network for recommendation systems*, in SIGIR, 2018, p. 515,Äì524.

[31] X. FENG, M. LIU, J. LIU, B. QIN, Y. SUN, AND T. LIU, *Topic-to-essay generation with neural networks.*, in IJCAI, 2018, pp. 4078–4084.

[32] Y. FENG, F. LV, W. SHEN, M. WANG, F. SUN, Y. ZHU, AND K. YANG, *Deep session interest network for click-through rate prediction*, IJCAI, 2019-August (2019), pp. 2301–2307.

[33] Y. FENG, H. YOU, Z. ZHANG, R. JI, AND Y. GAO, *Hypergraph neural networks*, AAAI, (2019), pp. 3558–3565.

[34] J. FICLER AND Y. GOLDBERG, *Controlling linguistic style aspects in neural language generation*, arXiv preprint arXiv:1707.02633, (2017).

[35] G. R. FOXALL, *Behavior analysis and consumer psychology*, Journal of Economic Psychology, 15 (1994), pp. 5–91.

[36] S. GHOSH, M. CHOLLET, E. LAKSANA, L.-P. MORENCY, AND S. SCHERER, *Affect-lm: A neural language model for customizable affective text generation*, arXiv preprint arXiv:1704.06851, (2017).

[37] B. GRANDHI, N. PATWA, AND K. SALEEM, *Data-driven marketing for growth and profitability*, EuroMed Journal of Business, 16 (2021), pp. 381–398.

[38] A. GROVER AND J. LESKOVEC, *node2vec: Scalable feature learning for networks*, in Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, 2016, pp. 855–864.

[39] F. GU, H. CHANG, W. ZHU, S. SOJOUDI, AND L. EL GHAOUI, *Implicit graph neural networks*, Advances in Neural Information Processing Systems, 33 (2020), pp. 11984–11995.

[40] W. HAMILTON, Z. YING, AND J. LESKOVEC, *Inductive representation learning on large graphs*, in NIPS, 2017, pp. 1024–1034.

[41] B. HARIHARAN, P. ARBELÁEZ, R. GIRSHICK, AND J. MALIK, *Hypercolumns for object segmentation and fine-grained localization*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 447–456.

[42] K. HE, X. ZHANG, S. REN, AND J. SUN, *Deep residual learning for image recognition*, (2015).

[43] L. HE, H. CHEN, D. WANG, S. JAMEEL, P. YU, AND G. XU, *Click-Through Rate Prediction with Multi-Modal Hypergraphs*, Association for Computing Machinery, New York, NY, USA, 2021, p. 690,Äì699.

[44] L. HE, D. WANG, H. WANG, H. CHEN, AND G. XU, *TagPick: A System for Bridging Micro-Video Hashtags and E-Commerce Categories*, Association for Computing Machinery, New York, NY, USA, 2021, p. 4721,Äì4724.

[45] L. HE, D. WANG, H. WANG, H. CHEN, AND G. XU, *Tagpick: A system for bridging micro-video hashtags and e-commerce categories*, in CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021, G. Demartini, G. Zuccon, J. S. Culpepper, Z. Huang, and H. Tong, eds., ACM, 2021, pp. 4721–4724.

[46] L. HE, X. WANG, H. CHEN, AND G. XU, *Online spam review detection: A survey of literature*, Human-Centric Intelligent Systems, (2022), pp. 1–17.

[47] L. HE, G. XU, S. JAMEEL, X. WANG, AND H. CHEN, *Graph-aware deep fusion networks for online spam review detection*, IEEE Transactions on Computational Social Systems, (2022).

[48] R. HE AND J. MCAULEY, *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, 2016, pp. 507–517.

[49] X. HE, K. DENG, X. WANG, Y. LI, Y. ZHANG, AND M. WANG, *Lightgcn: Simplifying and powering graph convolution network for recommendation*, in Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, 2020, pp. 639–648.

[50] S. HERSHEY, S. CHAUDHURI, D. P. W. ELLIS, J. F. GEMMEKE, A. JANSEN, R. C. MOORE, M. PLAKAL, D. PLATT, R. A. SAUROUS, B. SEYBOLD, M. SLANEY, R. J. WEISS, AND K. WILSON, *Cnn architectures for large-scale audio classification*, 2017.

[51] B. HIDASI, A. KARATZOGLOU, L. BALTRUNAS, AND D. TIKK, *Session-based recommendations with recurrent neural networks*, (2016).

[52] C.-K. HSIEH, L. YANG, Y. CUI, T.-Y. LIN, S. BELONGIE, AND D. ESTRIN, *Collaborative metric learning*, in Proceedings of the 26th international conference on world wide web, 2017, pp. 193–201.

[53] Z. HU, Z. YANG, X. LIANG, R. SALAKHUTDINOV, AND E. P. XING, *Toward controlled generation of text*, in International conference on machine learning, PMLR, 2017, pp. 1587–1596.

[54] G. HUANG, Y. SUN, Z. LIU, D. SEDRA, AND K. Q. WEINBERGER, *Deep networks with stochastic depth*, in European conference on computer vision, Springer, 2016, pp. 646–661.

[55] M. IYYER, J. WIETING, K. GIMPEL, AND L. ZETTLEMOYER, *Adversarial example generation with syntactically controlled paraphrase networks*, arXiv preprint arXiv:1804.06059, (2018).

[56] K. JÄRVELIN AND J. KEKÄLÄINEN, *Cumulated gain-based evaluation of ir techniques*, ACM Transactions on Information Systems (TOIS), 20 (2002), pp. 422–446.

[57] H. JHAMTANI, V. GANGAL, E. HOVY, AND E. NYBERG, *Shakespearizing modern language using copy-enriched sequence-to-sequence models*, arXiv preprint arXiv:1707.01161, (2017).

[58] N. JINDAL AND B. LIU, *Opinion spam and analysis*, in WSDM, 2008, pp. 219–230.

[59] J. KABBARA AND J. C. K. CHEUNG, *Stylistic transfer in natural language generation systems using recurrent neural networks*, in Proceedings of the Workshop on Uphill Battles in Language Processing: Scaling Early Achievements to Robust Methods, 2016, pp. 43–47.

[60] W. C. KANG AND J. MCAULEY, *Self-Attentive Sequential Recommendation*, ICDM, 2018-November (2018), pp. 197–206.

[61] G. KARYPIS AND V. KUMAR, *A fast and high quality multilevel scheme for partitioning irregular graphs*, SIAM Journal on scientific Computing, 20 (1998), pp. 359–392.

[62] Y. KIM, *Convolutional neural networks for sentence classification*, arXiv, (2014).

[63] D. P. KINGMA AND J. BA, *Adam: A method for stochastic optimization*, arXiv preprint arXiv:1412.6980, (2014).

[64] T. N. KIPF AND M. WELLING, *Semi-supervised classification with graph convolutional networks*, arXiv, (2016).

[65] A. LAHA, P. JAIN, A. MISHRA, AND K. SANKARANARAYANAN, *Scalable micro-planned generation of discourse from structured data*, Computational Linguistics, 45 (2020), pp. 737–763.

[66] M. LEWIS, Y. LIU, N. GOYAL, M. GHAZVININEJAD, A. MOHAMED, O. LEVY, V. STOYANOV, AND L. ZETTLEMOYER, *Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension*, arXiv preprint arXiv:1910.13461, (2019).

[67] A. LI, Z. QIN, R. LIU, Y. YANG, AND D. LI, *Spam review detection with graph convolutional networks*, in CIKM, 2019, pp. 2703–2711.

[68] F. H. LI, M. HUANG, Y. YANG, AND X. ZHU, *Learning to identify review spam*, in AAAI, 2011.

[69] J. LI, M. GALLEY, C. BROCKETT, J. GAO, AND B. DOLAN, *A diversity-promoting objective function for neural conversation models*, in Proc. of NAACL-HLT, March 2016.

[70] J. LI, W. MONROE, A. RITTER, M. GALLEY, J. GAO, AND D. JURAFSKY, *Deep reinforcement learning for dialogue generation*, arXiv preprint arXiv:1606.01541, (2016).

[71] J. LI, X. WANG, L. YANG, P. ZHANG, AND D. YANG, *Identifying ground truth in opinion spam: an empirical survey based on review psychology*, Applied Intelligence, (2020), pp. 1–16.

[72] X. LI, C. WANG, J. TAN, X. ZENG, D. OU, AND B. ZHENG, *Adversarial Multimodal Representation Learning for Click-Through Rate Prediction*, The Web Conference 2020 - Proceedings of the World Wide Web Conference, WWW 2020, (2020).

[73] X. L. LI, J. THICKSTUN, I. GULRAJANI, P. LIANG, AND T. B. HASHIMOTO, *Diffusion-lm improves controllable text generation*, arXiv preprint arXiv:2205.14217, (2022).

[74] Y. LI, H. CHEN, X. SUN, Z. SUN, L. LI, L. CUI, Y. PHILIP S., AND G. XU, *Hyperbolic hypergraphs for sequential recommendation*, arXiv preprint arXiv:2108.08134, (2021).

[75] Y. LI, C. CUI, M. LIU, X. S. XU, J. YIN, AND L. NIE, *Routing micro-videos via a temporal graph-guided recommendation system*, MM, (2019), pp. 1464–1472.

[76] W. LIN, F. HE, F. ZHANG, X. CHENG, AND H. CAI, *Initialization for network embedding: A graph partition approach*, in Proceedings of the 13th International Conference on Web Search and Data Mining, 2020, pp. 367–374.

[77] B. LIU, J. YU, R. TANG, H. GUO, Y. CHEN, AND Y. ZHANG, *Feature generation by convolutional neural network for click-through rate prediction*, WWW, (2019), pp. 1119–1129.

[78] B. LIU, C. ZHU, G. LI, W. ZHANG, J. LAI, R. TANG, X. HE, Z. LI, AND Y. YU, *AutoFIS: Automatic Feature Interaction Selection in Factorization Models for Click-Through Rate Prediction*, KDD, (2020), pp. 2636–2645.

[79] K. LIU, F. ZHAO, H. CHEN, Y. LI, G. XU, AND H. JIN, *Da-net: Distributed attention network for temporal knowledge graph reasoning*, in Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022, M. A. Hasan and L. Xiong, eds., ACM, 2022, pp. 1289–1298.

[80] Q. LIU, H. ZHANG, AND Z. LIU, *Simplification of graph convolutional networks: A matrix factorization-based perspective*, arXiv preprint arXiv:2007.09036, (2020).

[81] X. LIU, F. ZHANG, Z. HOU, Z. WANG, L. MIAN, J. ZHANG, AND J. TANG, *Self-supervised Learning: Generative or Contrastive*, (2020), pp. 1–23.

[82] Z. LIU, C. CHEN, X. YANG, J. ZHOU, X. LI, AND L. SONG, *Heterogeneous graph neural networks for malicious account detection*, in CIKM, 2018, pp. 2077–2085.

[83] Z. LIU, V. W. ZHENG, Z. ZHAO, H. YANG, AND J. YING, *Subgraph-augmented path embedding for semantic user search on heterogeneous social network*, (2018).

[84] L. LOGESWARAN AND H. LEE, *An efficient framework for learning sentence representations*, (2018).

[85] B. LONI, R. PAGANO, M. LARSON, AND A. HANJALIC, *Bayesian personalized ranking with multi-channel user feedback*, in Proceedings of the 10th ACM Conference on Recommender Systems, 2016, pp. 361–364.

[86] Y. LUO, M. LU, G. LIU, AND S. WANG, *Few-shot table-to-text generation with prefix-controlled generator*, arXiv preprint arXiv:2208.10709, (2022).

[87] X. MA AND D. DONG, *Evolutionary nonnegative matrix factorization algorithms for community detection in dynamic networks*, IEEE transactions on knowledge and data engineering, 29 (2017), pp. 1045–1058.

[88] S. MAI, H. HU, AND S. XING, *Modality to modality translation: An adversarial representation learning and graph fusion network for multimodal fusion*, in AAAI, vol. 34, 2020, pp. 164–172.

[89] A. MISHRA, M. F. M. CHOWDHURY, S. MANOHAR, D. GUTFREUND, AND K. SANKARANARAYANAN, *Template controllable keywords-to-text generation*, arXiv preprint arXiv:2011.03722, (2020).

[90] R. F. MOLANES, K. AMARASINGHE, J. RODRIGUEZ-ANDINA, AND M. MANIC, *Deep learning and reconfigurable platforms in the internet of things: Challenges and opportunities in algorithms and hardware*, IEEE industrial electronics magazine, 12 (2018), pp. 36–49.

[91] F. NIE, X. WANG, M. I. JORDAN, AND H. HUANG, *The constrained laplacian rank algorithm for graph-based clustering.*, in AAAI, 2016, pp. 1969–1976.

[92] A. OH AND A. RUDNICKY, *Stochastic language generation for spoken dialogue systems*, in ANLP-NAACL 2000 Workshop: Conversational Systems, 2000.

[93] M. OTT, C. CARDIE, AND J. T. HANCOCK, *Negative deceptive opinion spam*, in Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Atlanta, Georgia, June 2013, Association for Computational Linguistics, pp. 497–501.

[94] W. OUYANG, X. ZHANG, L. LI, H. ZOU, X. XING, Z. LIU, AND Y. DU, *Deep spatio-temporal neural networks for click-through rate prediction*, in SIGKDD, 2019, p. 2078,Äì2086.

[95] A. PAL, C. EKSOMBATCHAI, Y. ZHOU, B. ZHAO, C. ROSENBERG, AND J. LESKOVEC, *Pinnersage: Multi-modal user embedding framework for recommendations at pinterest*, in Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 2311–2320.

[96] K. PAPINENI, S. ROUKOS, T. WARD, AND W.-J. ZHU, *Bleu: A method for automatic evaluation of machine translation*, in Proceedings of the 40th Annual Meeting

on Association for Computational Linguistics, ACL '02, USA, 2002, Association for Computational Linguistics, p. 311,Äì318.

[97] J. PENNINGTON, R. SOCHER, AND C. MANNING, *Glove: Global vectors for word representation*, in EMNLP, 2014.

[98] B. PEROZZI, R. AL-RFOU, AND S. SKIENA, *Deepwalk: Online learning of social representations*, in KDD, 2014, pp. 701–710.

[99] Q. PI, W. BIAN, G. ZHOU, X. ZHU, AND K. GAI, *Practice on long sequential user behavior modeling for click-through rate prediction*, KDD, (2019), pp. 2671–2679.

[100] F. PUKELSHEIM, *The three sigma rule*, The American Statistician, 48 (1994), pp. 88–91.

[101] J. QIN, W. ZHANG, X. WU, J. JIN, Y. FANG, AND Y. YU, *User Behavior Retrieval for Click-Through Rate Prediction*, SIGIR, (2020), pp. 2347–2356.

[102] J. QIU, Y. DONG, H. MA, J. LI, K. WANG, AND J. TANG, *Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec*, in Proceedings of the eleventh ACM international conference on web search and data mining, 2018, pp. 459–467.

[103] A. RADFORD, J. WU, R. CHILD, D. LUAN, D. AMODEI, I. SUTSKEVER, ET AL., *Language models are unsupervised multitask learners*, OpenAI blog, 1 (2019), p. 9.

[104] J. W. RAE, L. O. N. G. ANGE, AND C. HILLIER, *COmpressive TRansformers*, (2019), pp. 1–19.

[105] S. RAYANA AND L. AKOGLU, *Collective opinion spam detection: Bridging review networks and metadata*, in KDD, 2015, pp. 985–994.

[106] K. REN, W. ZHANG, K. CHANG, Y. RONG, AND J. WANG, *Bidding machine: Learning to bid for directly optimizing profits in display advertising*, TKDE, 30 (2018), pp. 645–659.

[107] K. REN, W. ZHANG, K. CHANG, Y. RONG, Y. YU, AND J. WANG, *Bidding Machine: Learning to Bid for Directly Optimizing Profits in Display Advertising*, TKDE, 30 (2018), pp. 645–659.

[108] K. REN, W. ZHANG, Y. RONG, H. ZHANG, Y. YU, AND J. WANG, *User response learning for directly optimizing campaign performance in display advertising*, in CIKM, 2016, pp. 679–688.

[109] S. RENDLE, *Factorization machines*, in ICDM, IEEE, 2010, pp. 995–1000.

[110] S. RENDLE AND C. FREUDENTHALER, *Improving pairwise learning for item recommendation from implicit feedback*, in Proceedings of the 7th ACM international conference on Web search and data mining, 2014, pp. 273–282.

[111] S. RENDLE, C. FREUDENTHALER, Z. GANTNER, AND L. SCHMIDT-THIEME, *Bpr: Bayesian personalized ranking from implicit feedback*, arXiv preprint arXiv:1205.2618, (2012).

[112] I. RISH ET AL., *An empirical study of the naive bayes classifier*, 3 (2001), pp. 41–46.

[113] M. SALEHIE AND L. TAHVILDARI, *Self-adaptive software: Landscape and research challenges*, ACM transactions on autonomous and adaptive systems (TAAS), 4 (2009), pp. 1–42.

[114] F. SCARSELLI, M. GORI, A. C. TSOI, M. HAGENBUCHNER, AND G. MONFARDINI, *The graph neural network model*, IEEE TNN, 20 (2008), pp. 61–80.

[115] A. SEE, P. J. LIU, AND C. D. MANNING, *Get to the point: Summarization with pointer-generator networks*, arXiv preprint arXiv:1704.04368, (2017).

[116] L. SHANG, Z. LU, AND H. LI, *Neural responding machine for short-text conversation*, arXiv preprint arXiv:1503.02364, (2015).

[117] S. SHEHNEPOOR, M. SALEHI, R. FARAHBAKHSH, AND N. CRESPI, *Netspam: A network-based spam detection framework for reviews in online social media*, Trans on IFS, 12 (2017), pp. 1585–1595.

[118] K. SHI, H. LU, Y. ZHU, AND Z. NIU, *Automatic generation of meteorological briefing by event knowledge guided summarization model*, Knowledge-Based Systems, 192 (2020), p. 105379.

[119] K. SHI, Y. WANG, H. LU, Y. ZHU, AND Z. NIU, *Ekgtf: A knowledge-enhanced model for optimizing social network-based meteorological briefings*, Information Processing & Management, 58 (2021), p. 102564.

[120] J. SONG, C. CHANG, F. SUN, X. SONG, AND P. JIANG, *Ngat4rec: Neighbor-aware graph attention network for recommendation*, arXiv preprint arXiv:2010.12256, (2020).

[121] Q. SONG, D. CHENG, H. ZHOU, J. YANG, Y. TIAN, AND X. HU, *Towards Automated Neural Interaction Discovery for Click-Through Rate Prediction*, KDD, (2020), pp. 945–955.

[122] R. SPEER, J. CHIN, AND C. HAVASI, *Conceptnet 5.5: An open multilingual graph of general knowledge*, in Thirty-first AAAI conference on artificial intelligence, 2017.

[123] R. SPEER, J. CHIN, AND C. HAVASI, *Conceptnet 5.5: an open multilingual graph of general knowledge*, in National Conference on Artificial Intelligence, 2017.

[124] J. SUN, Y. ZHANG, W. GUO, H. GUO, R. TANG, X. HE, C. MA, AND M. COATES, *Neighbor interaction aware graph convolution networks for recommendation*, in Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 1289–1298.

[125] X. SUN, H. YIN, B. LIU, H. CHEN, J. CAO, Y. SHAO, AND N. Q. VIET HUNG, *Heterogeneous hypergraph embedding for graph classification*, in Proceedings of the 14th ACM International Conference on Web Search and Data Mining, 2021, pp. 725–733.

[126] X. SUN, H. YIN, B. LIU, H. CHEN, Q. MENG, W. HAN, AND J. CAO, *Multi-level hyperedge distillation for social linking prediction on sparsely observed networks*, in Proceedings of the Web Conference 2021, 2021, pp. 2934–2945.

[127] S. SURYA, A. MISHRA, A. LAHA, P. JAIN, AND K. SANKARANARAYANAN, *Unsupervised neural text simplification*, arXiv preprint arXiv:1810.07931, (2018).

[128] I. SUTSKEVER, O. VINYALS, AND Q. V. LE, *Sequence to sequence learning with neural networks*, Advances in neural information processing systems, 27 (2014).

[129] J. TANG, M. QU, M. WANG, M. ZHANG, J. YAN, AND Q. MEI, *Line: Large-scale information network embedding*, in Proceedings of the 24th international conference on world wide web, 2015, pp. 1067–1077.

[130] J. TANG AND K. WANG, *Personalized top-n sequential recommendation via convolutional sequence embedding*, in WSDM, 2018.

[131] K. UCHIMOTO, S. SEKINE, AND H. ISAHARA, *Text generation from keywords*, in COLING 2002: The 19th International Conference on Computational Linguistics, 2002.

[132] A. VASWANI, N. SHAZEER, N. PARMAR, J. USZKOREIT, L. JONES, A. N. GOMEZ, Ł. KAISER, AND I. POLOSUKHIN, *Attention is all you need*, Advances in neural information processing systems, 30 (2017).

[133] P. VELIČKOVIĆ, G. CUCURULL, A. CASANOVA, A. ROMERO, P. LIO, AND Y. BENGIO, *Graph attention networks*, arXiv preprint arXiv:1710.10903, (2017).

[134] O. VINYALS AND Q. LE, *A neural conversational model*, arXiv preprint arXiv:1506.05869, (2015).

[135] C.-C. WANG, M.-Y. DAY, C.-C. CHEN, AND J.-W. LIOU, *Detecting spamming reviews using long short-term memory recurrent neural network framework*, in E-commerce, E-Business and E-Government, 2018, pp. 16–20.

[136] G. WANG, S. XIE, B. LIU, AND S. Y. PHILIP, *Review graph based online store review spammer detection*, in ICDM, 2011, pp. 1242–1247.

[137] W. WANG, Z. HU, Z. YANG, H. SHI, F. XU, AND E. XING, *Toward unsupervised text content manipulation*, arXiv preprint arXiv:1901.09501, (2019), p. 91.

[138] X. WANG, X. HE, M. WANG, F. FENG, AND T.-S. CHUA, *Neural graph collaborative filtering*, in Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval, 2019, pp. 165–174.

[139] X. WANG, H. JIN, A. ZHANG, X. HE, T. XU, AND T.-S. CHUA, *Disentangled graph collaborative filtering*, in Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval, 2020, pp. 1001–1010.

[140] X. WANG, K. LIU, AND J. ZHAO, *Handling cold-start problem in review spam detection by jointly embedding texts and behaviors*, in ACL, 2017, pp. 366–376.

[141] Y. WEI, X. WANG, L. NIE, X. HE, R. HONG, AND T.-S. CHUA, *Mmgcn: Multi-modal graph convolution network for personalized recommendation of micro-video*, in ICM, 2019, pp. 1437–1445.

[142] H. WENG, S. JI, F. DUAN, Z. LI, J. CHEN, Q. HE, AND T. WANG, *Cats: Cross-platform e-commerce fraud detection*, in ICDE, 2019, pp. 1874–1885.

[143] S. WISEMAN, S. M. SHIEBER, AND A. M. RUSH, *Challenges in Data-to-Document Generation*, (2017).

[144] F. WU, A. SOUZA, T. ZHANG, C. FIFTY, T. YU, AND K. WEINBERGER, *Simplifying graph convolutional networks*, in International conference on machine learning, PMLR, 2019, pp. 6861–6871.

[145] J. WU, X. WANG, F. FENG, X. HE, L. CHEN, J. LIAN, AND X. XIE, *Self-supervised graph learning for recommendation*, in Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval, 2021, pp. 726–735.

[146] Y. WU, D. LIAN, Y. XU, L. WU, AND E. CHEN, *Graph convolutional networks with markov random field reasoning for social spammer detection*, in AAAI, vol. 34, 2020, pp. 1054–1061.

[147] Y. WU, E. W. NGAI, P. WU, AND C. WU, *Fake online reviews: Literature review, synthesis, and directions for future research*, Decision Support Systems, (2020), p. 113280.

[148] C. XING, W. WU, Y. WU, J. LIU, Y. HUANG, M. ZHOU, AND W.-Y. MA, *Topic aware neural response generation*, in Proceedings of the AAAI conference on artificial intelligence, vol. 31, 2017.

[149] J. XU, Z. ZHU, J. ZHAO, X. LIU, AND J. GUO, *Gemini: A novel and universal heterogeneous graph information fusing framework for online recommendations*, (2020).

[150] P. YANG, L. LI, F. LUO, T. LIU, AND X. SUN, *Enhancing topic-to-essay generation with external commonsense knowledge*, in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 2002–2012.

[151] L. YAO, N. PENG, R. WEISCHEDEL, K. KNIGHT, D. ZHAO, AND R. YAN, *Plan-and-write: Towards better automatic storytelling*, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 7378–7385.

[152] H. YIN, H. CHEN, X. SUN, H. WANG, Y. WANG, AND Q. V. H. NGUYEN, *Sptf: a scalable probabilistic tensor factorization model for semantic-aware behavior prediction*, in 2017 IEEE International Conference on Data Mining (ICDM), IEEE, 2017, pp. 585–594.

[153] Z. YIN, X. XU, K. FAN, R. LI, W. LI, W. LIU, AND B. NIU, *Dgcf: A distributed greedy clustering framework for large-scale genomic sequences*, in 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, 2019, pp. 2272–2279.

[154] R. YING, R. HE, K. CHEN, P. EKSOMBATCHAI, W. L. HAMILTON, AND J. LESKOVEC, *Graph convolutional neural networks for web-scale recommender systems*, in KDD, 2018, pp. 974–983.

[155] L. YU, W. ZHANG, J. WANG, AND Y. YU, *Seqgan: Sequence generative adversarial nets with policy gradient*, in Proceedings of the AAAI conference on artificial intelligence, vol. 31, 2017.

[156] C. YUAN, W. ZHOU, Q. MA, S. LV, J. HAN, AND S. HU, *Learning review representations from user and product level information for spam detection*, ICDM, 2019-Novem (2019), pp. 1444–1449.

[157] R. ZANDIE AND M. H. MAHOOR, *Topical language generation using transformers*, (2021).

[158] J. ZHANG, L. NIE, X. WANG, X. HE, X. HUANG, AND T. S. CHUA, *Shorter-is-better: Venue category estimation from micro-video*, in MM, 2016, p. 1415,Äì1424.

[159] S. ZHANG, H. CHEN, X. MING, L. CUI, H. YIN, AND G. XU, *Where are we in embedding spaces? a comprehensive analysis on network embedding approaches for recommender systems*, arXiv preprint arXiv:2105.08908, (2021).

[160] Z. ZHANG, L. LIAO, M. HUANG, X. ZHU, AND T.-S. CHUA, *Neural multimodal belief tracker with adaptive attention for dialogue systems*, in WWW, 2019, pp. 2401–2412.

[161] T. ZHAO, R. ZHAO, AND M. ESKENAZI, *Learning discourse-level diversity for neural dialog models using conditional variational autoencoders*, arXiv preprint arXiv:1703.10960, (2017).

[162] G. ZHOU, N. MOU, Y. FAN, Q. PI, W. BIAN, C. ZHOU, X. ZHU, AND K. GAI, *Deep interest evolution network for click-through rate prediction*, (2018).

[163] J. ZHOU, G. CUI, Z. ZHANG, C. YANG, Z. LIU, L. WANG, C. LI, AND M. SUN, *Graph neural networks: A review of methods and applications*, arXiv, (2018).

[164] K. ZHOU, H. WANG, W. X. ZHAO, Y. ZHU, S. WANG, F. ZHANG, Z. WANG, AND J.-R. WEN, *S^3-Rec: Self-Supervised Learning for Sequential Recommendation with Mutual Information Maximization*, (2020), pp. 1893–1902.

[165] Y. ZHOU, Y. SHANG, Y. CAO, Q. LI, C. ZHOU, AND G. XU, *API-GNN: attribute preserving oriented interactive graph neural network*, World Wide Web, 25 (2022), pp. 239–258.