MONASH University

Multimodal Graph Learning for Compatibility-oriented Fashion Recommendation

Weili Guan

Doctor of Philosophy

A Thesis Submitted for the Degree of Doctor of Philosophy at
Monash University in 2022
Faculty of Information Technology

Copyright notice

Abstract

With the unprecedented growth of online fashion products, it becomes difficult for people to find matching items to compose outfit. To alleviate this dilemma, it is highly desirable to investigate compatibility-oriented fashion recommendation schemes. Intuitively, it mainly involves two subtasks: 1) outfit compatibility modeling, aiming to assess the matching degree of a given set of complementary items; and 2) personalized outfit recommendation, targeting recommending the well-matched outfits for the user according to his/her personal preference. As fashion recommendation involves multiple types of fashion entities (*e.g.*, users, outfits, and items), and rich relations among them (*e.g.*, user-outfit interactions, outfit-item associations), graph learning has become the mainstream technique towards fashion recommendation. Nevertheless, although previous research efforts have gained compelling success, they still suffer from the following limitations: 1) overlook the latent correlation among different modalities; 2) fail to fully utilize the irregular attribute labels of items; 3) ignore the users' personal preference on outfit composition; and 4) focus on improving the outfit recommendation effectiveness, while overlooking the recommendation efficiency.

In this thesis, to address these limitations, we first present a correlation-oriented graph learning method towards outfit compatibility modeling, which explicitly models the consistent and complimentary relations between the visual and textual modalities. We next devise a partially supervised disentangled graph learning method, where the fine-grained compatibility is explored. To incorporate the personal tastes, we propose a metapath-guided heterogeneous graph learning scheme for personalized outfit compatibility modeling. To improve the model efficiency, we then present an efficient personalized outfit recommendation scheme with bi-directional heterogeneous graph hash learning.

Declaration

This thesis is an original work of my research and contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Signature: _____

Print Name: Weili Guan

Date: Sep. 28, 2022

Thesis including published works declaration

I hereby declare that this thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

This thesis includes 4 original papers published in peer reviewed journals and conferences. The core theme of the thesis is 4. The ideas, development and writing up of all the papers in the thesis were the principal responsibility of myself, the student, working within the department of data science and artificial intelligence under the supervision of Prof. Chung-Hsing Yeh and Xiaojun Chang.

The inclusion of co-authors reflects the fact that the work came from active collaboration between researchers and acknowledges input into team-based research.

In the case of chapter 3, 4, 5 and 6, my contribution to the work involved the following:

I played a central role in the idea of these chapters, the experiments, and the writing. I established the idea of these chapters after a lot of literature reviewing, designed the experimental model (*i.e.*, the data preprocessing, model implementation, and parameter tuning), and wrote the paper. These papers were written by me, and they were revised and polished by my advisors. Please refer to below published works declaration.

I have renumbered sections of submitted or published papers in order to generate a consistent presentation within the thesis.

Student name: Weili Guan

Student signature: Date: Sep. 28, 2022

I hereby certify that the above declaration correctly reflects the nature and extent of the student's and co-authors' contributions to this work. In instances where I am not the responsible author I have consulted with the responsible author to agree on the respective contributions of the authors.

Main Supervisor name: Chung-Hsing Yeh

Main Supervisor signature:

Date: Sep. 28, 2022

Thesis Chapter	Publication Title	Status published, in press, accepted or returned for revision, submitted	Nature and % of student contribution	Co-author name(s) Nature and % of Co-author's contribution*	Co-author(s), Monash student Y/N*
3	*Multimodal Compatibility Modeling via Exploring the Consistent and Complementary Correlations*	*Accepted*	*60%. Concept and collecting data and writing first draft*	*1) Haokun Wen, perform the baselines experiments 20%* *2) Xuemeng Song, Data analysis, input into manuscript 10%* *3)Liqiang Nie, supervision, provide the experiment equipment 10%*	*No*
4	*Partially Supervised Compatibility Modeling*	*Accepted*	*60%. Concept and experimentation and writing first draft*	*1) Haokun Wen and Chun Wang, perform the baselines experiments 20%* *2) Xuemeng Song, Data analysis, input into manuscript 10%* *3)Liqiang Nie, supervision, provide the experiment equipment 10%*	*No*
5	*Personalized Fashion Compatibility Modeling via Metapath-guided Heterogeneous Graph Learning*	*Accepted*	*60%. Concept and experimentation and writing first draft*	*1) Fangkai Jiao and Haokun Wen, collect the data and perform the baselines experiments 20%* *2) Xuemeng Song, Data analysis, input into manuscript 20%*	*No*
6	*Bi-directional Heterogeneous Graph Hashing towards Efficient Outfit Recommendation*	*Accepted*	*60%. Concept and experimentation and writing first draft*	*1) Haoyu Zhang, collect the data and perform the baselines experiments 20%* *2) Xuemeng Song and Meng Liu, Data analysis, input into manuscript 20%*	*No*

**Published works declaration*

Publications during enrolment

Journal Papers

J-1. **Weili Guan**, Haokun Wen, Xuemeng Song, Chun Wang, Chung-Hsing Yeh, Xiaojun Chang, Liqiang Nie. "Partially Supervised Compatibility Modeling." in IEEE Transaction on Image Processing. 2021.

J-2. **Weili Guan**, Xuemeng Song, Tian Gan, Junyu Lin, Xiaojun Chang, Liqiang Nie. "Cooperation Learning from Multiple Social Networks: Consistent and Complementary Perspectives." in IEEE Transactions on Cybernetics. 2021.

J-3. Caixia Yan, Xiaojun Chang, Zhihui Li, **Weili Guan**, Zongyuan Ge, Lei Zhu, Qinghua Zheng. "Zeronas: Differentiable generative adversarial networks search for zero-shot learning." in IEEE Transactions on Pattern Analysis and Machine Intelligence. 2021.

J-4. Yinwei Wei, Xiang Wang, **Weili Guan**, Liqiang Nie, Zhouchen Lin, Baoquan Chen. "Neural Multimodal Cooperative Learning Toward Micro-Video Understanding." in IEEE Transaction on Image Processing. 2020.

Conference Papers

C-1. **Weili Guan**, Haokun Wen, Xuemeng Song, Chung-Hsing Yeh, Xiaojun Chang, Liqiang Nie. "Multimodal Compatibility Modeling via Exploring the Consistent and Complementary Correlations." in Proceedings of the International ACM Conference on Multimedia. ACM, 2021.

C-2. **Weili Guan**, Fangkai Jiao, Xuemeng Song, Haokun Wen, Chung-Hsing Yeh, Xiaojun Chang. "Personalized Fashion Compatibility Modeling via Metapath-guided Heterogeneous Graph Learning." in Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2022.

C-3. **Weili Guan**, Xuemeng Song, Haoyu Zhang, Meng Liu, Chung-Hsing Yeh, Xiaojun Chang. "Bi-directional Heterogeneous Graph Hashing towards Efficient Outfit Recommendation." in Proceedings of the International ACM Conference on Multimedia. ACM, 2022.

C-4. Tianyu Su, Xuemeng Song, Na Zheng, **Weili Guan**, Yan Li, Liqiang Nie. "Complementary Factorization towards Outfit Compatibility Modeling." in Proceedings of the International ACM Conference on Multimedia. ACM, 2021.

C-5. Xiaocong Chen, Lina Yao, Julian Mcauley, **Weili Guan**, Xianzhi Wang, Xiaojun Chang. "Locality-Sensitive State-Guided Experience Replay Optimization for Sparse

Rewards in Online Recommendation." in Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2022.

Book

B-1. **Weili Guan**, Xuemeng Song, Xiaojun Chang , Liqiang Nie. "Graph Learning for Fashion Compatibility Modeling." in Morgan and Claypool Lecture. 1-102 pages, book, 2022.

Acknowledgements

This thesis would not have been completed, or at least not be what it looks like now, without the support of many colleagues, especially those from the GORSE Lab at Monash University and the iLearn Center at Shandong University. It is my pleasure to take this opportunity to express my appreciation for their contributions to this thesis.

First and foremost, I would like to thank my supervisors Professor Chung-Hsing Yeh and Xiaojun Chang. I am extremely grateful for their patience and support. Thanks to Prof Xuemeng Song, my advisor from Shandong University, from whom I learned how to do research for real-world applications and how to perform research in a systematic way.

Second, my sincere thanks undoubtedly go to our colleagues who contributed significantly to some chapters of this thesis: Dr. Yuanfang Li, Dr. Guanliang Chen from Monash University; Mr. Haokun Wen, Mr. Fangkai Jiao, and Mr. Haoyu Zhang from Shandong University. Thanks for their participation in the technical discussion of this research and their constructive feedback and comments that significantly benefit the shaping of this thesis.

Third, I would like to express our heartfelt gratitude to Dr. Julie Holden from Monash University, who spared no effort to polish my academic reports.

Fourth, I am very grateful to the anonymous reviewers, who read the thesis very carefully and gave me many insightful and constructive suggestions. Their assistance also largely improved the quality of this thesis.

Finally, my thanks would be reserved to our beloved families for their selfless consideration, endless love, and unconditional support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

According to Statista Research Department[1], as shown in Figure 1.1, in 2021, the global e-commerce fashion industry has reached an overall market value of 668 billion U.S. dollars, and the online clothing and apparel industry is expected to reach a value of 1.2 trillion U.S. dollars by 2025. This shows the great economic value of the online fashion industry.

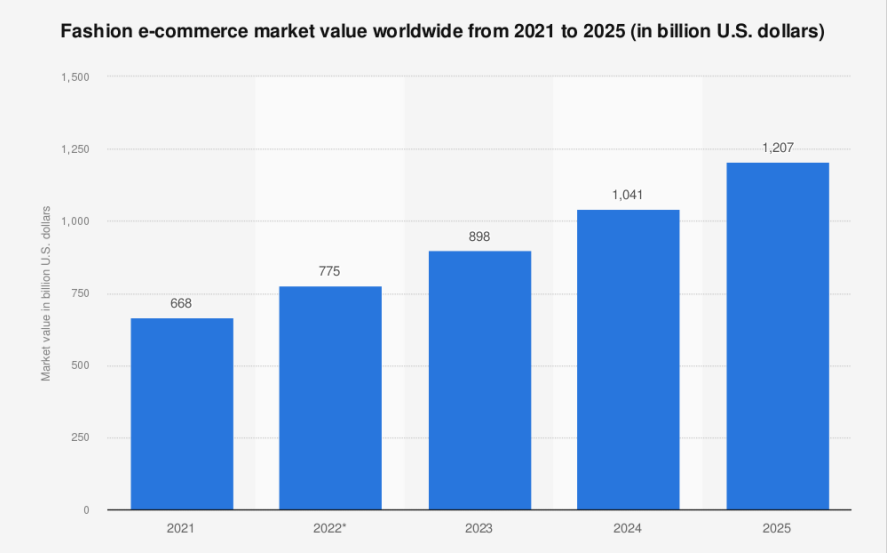


FIGURE 1.1: Statistics regarding the fashion e-commerce market value worldwide.

In fact, in modern society, clothing plays an increasingly important role in people's social lives, as a compatible outfit can largely improve one's appearance. Nevertheless, not all people grow a keen sense of aesthetics, and hence often find it difficult to find matching

[1] https://www.statista.com/statistics/1298198/market-value-fashion-ecommerce-global/.

(a) Composition 1 (b) Composition 2 (c) Composition 3

FIGURE 1.2: Examples of outfit compositions shared on fashion-oriented websites.

items to compose compatible outfits. The large volume of garments in e-commerce platforms aggravate the difficulty. Therefore, it is highly desirable to develop automatic compatibility-oriented fashion recommendation schemes. Typically, it involves two key subtasks: 1) outfit compatibility modeling (OCM), which aims to assess the matching degree of a given set of complementary items; and 2) personalized outfit recommendation (POR), which targets at recommending the well-matched outfits for the user according to his/her personal preference.

As for the task of outfit compatibility modeling, existing methods can be classified into three groups: pairwise, listwise, and graphwise modeling. The pairwise modeling [1, 2] focus on justifying the compatibility between two items. One key limitation of the pairwise methods is that they lack a global view of the outfit that involves multiple items. The listwise methods conceive the outfit as a list of items in a predefined order and evaluate the outfit compatibility with neural networks, like bidirectional long short-term memory (Bi-LSTM) [3, 4]. Apparently, one limitation of this branch is that there is no explicit order among composing outfit items. As the pairwise methods lack a global view of the outfit that involves multiple items (See Figure 1.2) and cannot flexibly cope with the outfit with arbitrary number of items, and the listwise methods ignore the fact that there is no explicit order among composing items, several graphwise methods emerged recently. They regard each outfit as an item graph and employ the graph learning technique to fulfill the compatibility modeling task. Significant prog

Regarding the task of personalized outfit recommendation, as this task involves multiple types of fashion entities (*e.g.*, users, outfits, and items), and various relations among them (*e.g.*, user-outfit interactions, outfit-item associations), recent studies [5, 6] also resort to graph learning due to its superior performance in entity representation learning [7].

1.2 Limitation of Previous Methods

Although existing graph learning-based methods have gained great success, they still suffer from the following limitations.

The first limitation is that they ignore the multiple correlated modalities. Online fashion items are usually associated with multiple modalities, such as images, textual descriptions, and semantic attribute labels. Moreover, as the multiple modalities actually (*e.g.*, visual and textual modalities) serve to characterize the same item, there should be a certain latent consistency shared by different modalities. Beyond that, each modality may express some unique aspects of the given items, and the multiple modalities hence supplement each other. Therefore, it is promising to explicitly model the consistent and complementary correlations among different modalities with graph learning to promote the utilization of the multiple modalities of items.

The second limitation is that they ignore the potential of the irregular attribute labels of items in supervising the hidden factor mining. In fact, there are multiple hidden factors (such as color, texture, and style) affecting the outfit compatibility evaluation. Moreover, the attribute labels usually convey rich feature information of items and should be considered to supervise the multiple hidden factors learning and hence improve the model performance. Nevertheless, the attribute labels are not unified or aligned, *i.e.*, different items could have different numbers of attribute labels. Therefore, how to fully employ the nonunified attribute labels to supervise the hidden factors learning merits our attention.

The third limitation is that they ignore the user's personal preferences. As a matter of fact, people have their preferences in making their personal ideal outfits, which may be caused by their diverse growing circumstances or educational backgrounds. For instance, given the same pink shirt, women who prefer a classic style prefer to match the shirt with a homochromatic skirt and high-heeled shoes, whereas women who prefer a sporty style like to coordinate the shirt with casual jeans and white sneakers. Accordingly, how to achieve the graph-based personalized outfit compatibility modeling needs to be addressed.

The last limitation is that they seldom consider the recommendation efficiency. With numerous outfit candidates have been available on fashion platforms, the efficiency of the outfit recommendation system merits our special attention. As this task involves multiple types of fashion entities (*e.g.,* users, outfits, items, and attributes) and relations, it is of importance to devise an efficient heterogeneous graph learning-based outfit recommendation scheme.

1.3 Contributions

To address the aforementioned limitations, we propose a series graph learning methods for compatibility-oriented fashion recommendation as follows.

- To model the correlation among different modalities, we propose a correlation-oriented graph learning method. It first nonlinearly projects each modality (visual image and textual description) into separable consistent and complementary spaces via multilayer perceptron and then models the consistent and complementary correlations between two modalities by parallel and orthogonal regularizations. Thereafter, we strengthen the visual and textual item representations with complementary information, and further induct both the text-oriented and vision-oriented outfit compatibility modeling with graph convolutional networks (GCNs). We ultimately employ the mutual learning strategy to reinforce the final compatibility modeling performance.

- Towards fine-grained compatibility modeling, we devise a partially supervised disentangled graph learning method, which can capture the hidden factors that affect the outfit compatibility and the irregular attribute labels of fashion items are utilized to guide the fine-grained compatibility modeling. In particular, we first devise a partially supervised attribute-level embedding learning component to disentangle the fine-grained attribute embeddings from the entire visual feature of each image. We then introduce a disentangled completeness regularizer to prevent information loss during disentanglement. Thereafter, we design a hierarchical GCN, which seamlessly integrates the attribute- and item-level compatibility modeling, to enhance the outfit compatibility modeling.

- To incorporate the user's personal tastes and make our compatibility modeling more practical, we propose a personalized compatibility modeling scheme. In particular, we creatively build a heterogeneous graph to unify the three types of entities (*i.e.*, users, items, and attributes) and their relations (*i.e.*, user-item interactions, item-item matching relations, and item-attribute association relations). We also define the user-oriented and item-oriented metapaths and propose performing metapath-guided heterogeneous graph learning to enhance the user and item embeddings. Moreover, we introduce contrastive regularization to improve the model performance.

- Towards efficient outfit recommendation, we devise a novel bi-directional heterogeneous graph hashing scheme. In particular, we first unify four types of entities (*i.e.*, users, outfits, items, and attributes) and their relations via a heterogeneous four-partite graph. To perform graph learning, we then creatively devise a bi-directional

graph convolution algorithm to sequentially transfer knowledge via repeating upwards and downwards convolution, whereby we divide the four-partite graph into three subgraphs and each subgraph only involves two adjacent entity types. We ultimately adopt the bayesian personalized ranking loss for the user preference learning and design the dual similarity preserving regularization to prevent the information loss during hash learning.

1.4 Thesis Structure

The remainder of this thesis consists of six chapters.Chapter 2 presents the related works in the fashion compatibility modeling and fashion recommendation tasks. Chapters 3,4,5, 6 target at addressing the identified limitations of previous work, respectively. Specifically, Chapters 3,4,5 focus on solving limitations (i.e., ignore the multiple correlated modalities, irregular attribute labels, and the user's personal preference) of the task of outfit compatibility modeling, while the Chapter 6 targets solving that (i.e., recommendation efficiency) of the task of personalized outfit recommendation. In particular, Chapter 3 details the proposed correlation-oriented graph learning method, which jointly models the consistent and complementary relations between the visual and textual modalities of fashion items. In Chapter 4, we develop a partially supervised disentangled graph learning to boost the outfit compatibility modeling performance and interpretability. In Chapter 5, we introduce a novel metapath-guided personalized outfit compatibility modeling scheme to deal with the user's personal preference for fashion items. In Chapter 6, we present a novel bi-directional heterogeneous graph hashing scheme, towards efficient outfit recommendation. Ultimately, we conclude this thesis and identify the future research directions in Chapter 7.

1.5 Thesis including Published Works

1) **Weili Guan**, Haokun Wen, Xuemeng Song, Chung-Hsing Yeh, Xiaojun Chang, Liqiang Nie. "Multimodal Compatibility Modeling via Exploring the Consistent and Complementary Correlations." In Proceedings of the International ACM Conference on Multimedia. ACM, 2021.

2) **Weili Guan**, Haokun Wen, Xuemeng Song, Chun Wang, Chung-Hsing Yeh, Xiaojun Chang, Liqiang Nie. "Partially Supervised Compatibility Modeling." In IEEE Transaction on Image Processing. IEEE, 2021.

3) **Weili Guan**, Fangkai Jiao, Xuemeng Song, Haokun Wen, Chung-Hsing Yeh, Xiaojun Chang. "Personalized Fashion Compatibility Modeling via Metapath-guided Heterogeneous Graph Learning." In Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2022.

4) **Weili Guan**, Xuemeng Song, Haoyu Zhang, Meng Liu, Chung-Hsing Yeh, Xiaojun Chang. "Bi-directional Heterogeneous Graph Hashing towards Efficient Outfit Recommendation." In Proceedings of the International ACM Conference on Multimedia. ACM, 2022.

Chapter 2

Literature Review

2.1 Fashion Compatibility Modeling

The recent flourish in the fashion industry has promoted researchers to address many fashion analysis tasks, such as clothing retrieval [8], compatibility modeling [9, 10], fashion trend prediction [11] and clothing recommendation [12, 13]. Specifically, as the key to many fashion-oriented applications, such as complementary item retrieval [14] and personal capsule wardrobe creation [15], fashion compatibility modeling has drawn great research attention. Existing fashion compatibility modeling methods can be grouped into three categories: pairwise methods [1, 9, 16, 17], listwise methods [3], and graphwise methods [18, 19].

The pairwise methods mainly focus on the compatibility between two given items. For example, McAuley *et al.* [16] used linear transformation to map items into a latent space, where the compatibility relation between items can be measured. Following that, Song *et al.* [9] proposed a multimodal compatibility modeling scheme, where neural networks are used to model the compatibility between fashion items with the Bayesian personalized ranking (BPR) [20] optimization. Later, Vasileva *et al.* [21] studied the compatibility for outfits with multiple fashion items based on pairwise modeling, where the item category information was additionally considered. Moreover, Yang *et al.* [22] utilized category complementary relations to model category-respected compatibility between fashion items in a translation-based embedding space. Thereafter, Liu *et al.* [23] introduced an auxiliary complementary template generation network equipped with the pixel-wise consistency and compatible template regularization to improve the compatibility modeling performance. The limitation of the pairwise methods is that it is cumbersome and time-consuming to directly apply them to analyze the real-world outfit

that usually comprises more than two items. Moreover, it maybe inappropriate to capture the complex compatibility relation among multiple items by separating the outfit into a set of independent item pairs.

By contrast, the listwise methods regard the outfit as an ordered list of items and utilize sequential neural networks to uncover the complex compatibility relationship among them. For instance, Han *et al.* [3] employed a Bi-LSTM network to sequentially model the compatibility relationships among the fashion items in a given outfit. Later, Dong *et al.* [4] presented a multi-modal try-on-guided compatibility modeling framework to jointly characterize the discrete interaction and try-on appearance of the outfit, where Bi-LSTM is used for discrete interaction modeling. One key limitation of the sequence-based methods is that there is no explicit and fixed order of items in an outfit. Moreover, the sequential neighborhood dependency in a given outfit is less stronger as compared to the tokens in a sentence. Taking this case as an intuitive example, in the sequence of $<top, bottom, shoes>$, the top may be tightly correlated with the shoes instead of the bottom.

Moving one step forward, the graphwise methods treat each outfit as an item graph, whereby each node represents an item and each edge bridges two items. Based upon the constructed graph, graph neural networks and their variants are designed to calculate the outfit compatibility. For example, Cui *et al.* [18] proposed the node-wise graph neural network (NGNN) towards fashion compatibility modeling. This method constructs a category-oriented fashion graph, where each node represents a category, and accordingly, each outfit can be abstracted as a subgraph consisted with the corresponding category nodes of its composing items. The outfit compatibility score is calculated based on the learned item representations with the attention mechanism. In addition, Cucurull *et al.* [19] utilized a graph neural network to learn the items' embeddings conditioned on their context, and cast the task of FCM as an edge prediction problem. Moreover, Li et al. [5] developed a hierarchical fashion graph network (HFGN) for personalized outfit recommendation, which models the relationship among users, items, and outfits simultaneously.

Although these studies have achieved significant success, they focus on either simply exploring the visual modality of the outfit, or considering both the visual and textual modalities while overlooking the sophisticated multimodal correlations.

2.2 Personalized Fashion Compatibility Modeling

Despite the significant progress made by previous outfit compatibility modeling efforts, they purely focus on the general item-item compatibility and overlook users' preferences in the fashion compatibility estimation. In fact, for the same fashion outfit, different users may have different evaluation results. Inspired by this, some studies have resorted to personalized outfit compatibility modeling. For example, a personalized compatibility modeling scheme for personalized clothing matching, named GP-BPR, is presented in [12], which jointly considers the general (item-item) compatibility and personal (user-item) preference for personalized clothing matching. Both the image and context description of items are utilized in the comprehensive modeling. Moving a step forward, Sagar *et al.* [24] introduced an attributewise interpretable personal preference modeling scheme to strengthen the model interpretability whereby the images and textual descriptions of items are explored. Additionally, Li *et al.* [5] developed a hierarchical fashion graph network to simultaneously model the rich relationships among users, items, and outfits.

Although these efforts have achieved compelling success, they overlook the item attributes when estimating compatibility. Attributes express the key item semantics items and reflect the specific user preferences. As a complementary effort, in this thesis, we incorporate the attribute entities and their semantic contents to comprehensively study the personalized outfit compatibility modeling problem.

2.3 Fashion Recommendation

Existing studies on fashion recommendation [25–28] can be roughly divided into two groups: item and outfit recommendation.

Item recommendation focuses on recommending a single item to the given user. For example, He et al. [29] developed a fashion item recommendation model based on the matrix factorization framework [30], which incorporates visual information of products to enhance the item representation. Later, Yu et al. [31] introduced a brain-inspired deep network to promote user preference modeling from the aesthetic perspective for fashion item recommendation. Towards explainable recommendation, Hou et al. [32] proposed a semantic attribute region guided approach, which extracts attributes from the item image, and employs the attention mechanism to explain the item recommendation result. Differently, Song et al. [12] noticed the necessity of considering the items that the user already has when recommending items, and hence investigated the task of recommending a complementary and compatible item (*e.g.*, bottom) for a given user with a given

item (*e.g.*, top). In particular, they proposed a personalized compatibility modeling scheme for clothing matching, whereby both the user-item preference and the item-item compatibility are jointly modeled for personalized complementary item recommendation.

By contrast, the outfit recommendation aims to recommend the whole outfits, *i.e.*, the sets of complementary items, to match the preference of a given user. For instance, Chen et al. [33] proposed a personalized outfit generation model that is able to generate outfits for the particular user by jointly considering the images and textual descriptions of items. Instead of generating outfits, Li et al. [5] proposed a hierarchical fashion graph network for personalized outfit recommendation, which models the relationships among users, items, and outfits, simultaneously. Meanwhile, Lin et al. [34] developed a bi-stage outfit recommendation system. It learns the compatibility between fashion items in the first stage, and then predicts the users' preference to outfits based on the visual information of items in the second one.

Although these studies have achieved great progress, they focus on exploring the user, outfit, and item entities. In other words, they overlook the semantic attributes of items, which actually well characterize the key features of items and reflect the user's specific preferences to items from the semantic perspective. In addition, existing methods generally focus on improving the recommendation effectiveness, while ignoring the recommendation efficiency. In light of this, we incorporated the attribute entities to promote the personalized outfit recommendation accuracy, and explored the hashing methods to improve the model efficiency.

Chapter 3

Correlation-oriented Graph Learning for OCM

3.1 Introduction

Existing graph-based methods for OCM focus on exploring the visual modality of fashion items, and seldom investigate an item's textual aspect, *i.e.*, the textual description. In fact, textual descriptions of fashion items usually contain key features, which benefit item representation learning. Notably, although some studies have attempted to incorporate the textual modality, they simply adopt early/late fusion or consistency regularization to boost performance. Nevertheless, the correlations among multimodalities are complex and sophisticated and are not yet clearly separated and explicitly modeled. Therefore, in this chapter, we study the graph-based outfit compatibility modeling via exploiting the multimodal correlations.

However, it is nontrivial considering the following facts. 1) The visual and textual modalities characterize the same item and thus should share certain consistency. As shown in Figure 3.1 (a), both visual and textual modalities deliver the item's features of "color" and "category". Additionally, the user-generated text may provide complementary information to the visual image, such as the item brand "New Ace" and material "Leather" in Figure 3.1 (b), yet certain features are difficult to describe using textual sentences but easy to visualize using the image, such as the stripe position in the item in Figure 3.1 (b). Consistent and complementary contents are often mixed in each modality and may be nonlinearly separable. Therefore, how to explicitly separate and model them poses one challenge. 2) How to leverage the correlation modeling results to strengthen the text and vision-oriented representation of the given item forms another challenge. And 3) outfit compatibility modeling can be derived separately from vision

FIGURE 3.1: Illustration of the consistent and complementary correlations between the visual and textual modalities. In (a), both the text and image reflect the color (dark blue) and category (shorts) of the item. In (b), the text reveals the item's material (leather) and brand (New Ace) that is rarely derived visually, but fails to describe the pattern (stripe) position.

or text-oriented representations, which characterizes the item from different angles. We argue that these two kinds of modeling share certain common knowledge on the outfit compatibility evaluation and can reinforce each other. How to mutually enhance the two kinds of modeling and thus boost the final compatibility modeling result constitutes the last challenge.

To address the aforementioned challenges, we devise a comprehensive **M**ulti**M**odal **O**utfit **C**ompatibility **M**odeling scheme, MM-OCM. As shown in Figure 3.2, MM-OCM consists of four components: a) multimodal feature extraction, b) multimodal correlation modeling, c) compatibility modeling, and d) mutual learning. The first component extracts the textual and visual features of the given item via two separate convolution neural networks (CNNs) [35] and long short-term memory (LSTM) networks [36], respectively. The reason for introducing two separate feature extractors is to facilitate later mutual learning. The second component aims to separate and model the consistent and complementary correlations. Considering the fact that these two kinds of correlations may not be separable in the original visual and textual feature spaces, we, therefore, employ the multilayer perceptrons to nonlinearly project the image/text feature into the consistent and complementary space, where the modal-modal consistency and complementarity, respectively, can be captured. In the third component, we incorporate the disengaged complementary content in the textual (visual) modality to complement the visual (textual) feature embedding and obtain the text (vision)-oriented representation. Thereafter, we build two independent graph convolutional networks to model outfit compatibility, namely text-oriented compatibility modeling (T-OCM) and vision-oriented compatibility modeling (V-OCM). Ultimately, the fourth component targets mutually transferring knowledge from one compatibility modeling to guide the other.

Once MM-OCM converges, we average the compatibility scores predicted by T-OCM and V-OCM as the final result. Extensive experiments on the real-world dataset demonstrate the superiority of our MM-OCM scheme as compared to several state-of-the-art baselines. As a byproduct, we released the codes to benefit other researchers[1].

The research work in this chapter has been published in ACM MM 2021.

Weili Guan, Haokun Wen, Xuemeng Song, Chung-Hsing Yeh, Xiaojun Chang, Liqiang Nie. "Multimodal Compatibility Modeling via Exploring the Consistent and Complementary Correlations." in Proceedings of the International ACM Conference on Multimedia. ACM, 2021.

3.2 Related Work

This work is related to the deep mutual learning.

Deep Mutual Learning. The idea of deep mutual learning is developed from the knowledge distillation, which was first introduced by Hinton *et al.* [37] for transferring the knowledge from a large cumbersome model to a small model to improve the model portability. Specifically, Hu *et al.* [38] designed an iterative teacher-student knowledge distillation approach, where the teacher network understands certain knowledge, while the student network iteratively mimics the teacher's solution to a certain problem to improve its performance. After that, the teacher-student knowledge distillation scheme attracted considerable attention [39, 40]. However, in many cases, it might be too difficult to obtain a teacher network with clear domain knowledge. Accordingly, Zhang *et al.* [41] proposed a deep mutual learning method for the classification task, where there is no explicit static teacher but an ensemble of students learning collaboratively throughout the training process. Thereafter, many researchers have investigated the deep mutual learning in various domains, such as person reidentification [42–44], domain-adapted sentiment classification [45], and deep metric learning [46]. Despite the value of mutual learning in these fields, its potential in outfit compatibility modeling has been largely unexplored, which is the major concern of this work.

3.3 Methodology

In this section, we first formulate the research problem and then detail the proposed MM-OCM scheme.

[1] https://site2750.wixsite.com/mmocm.

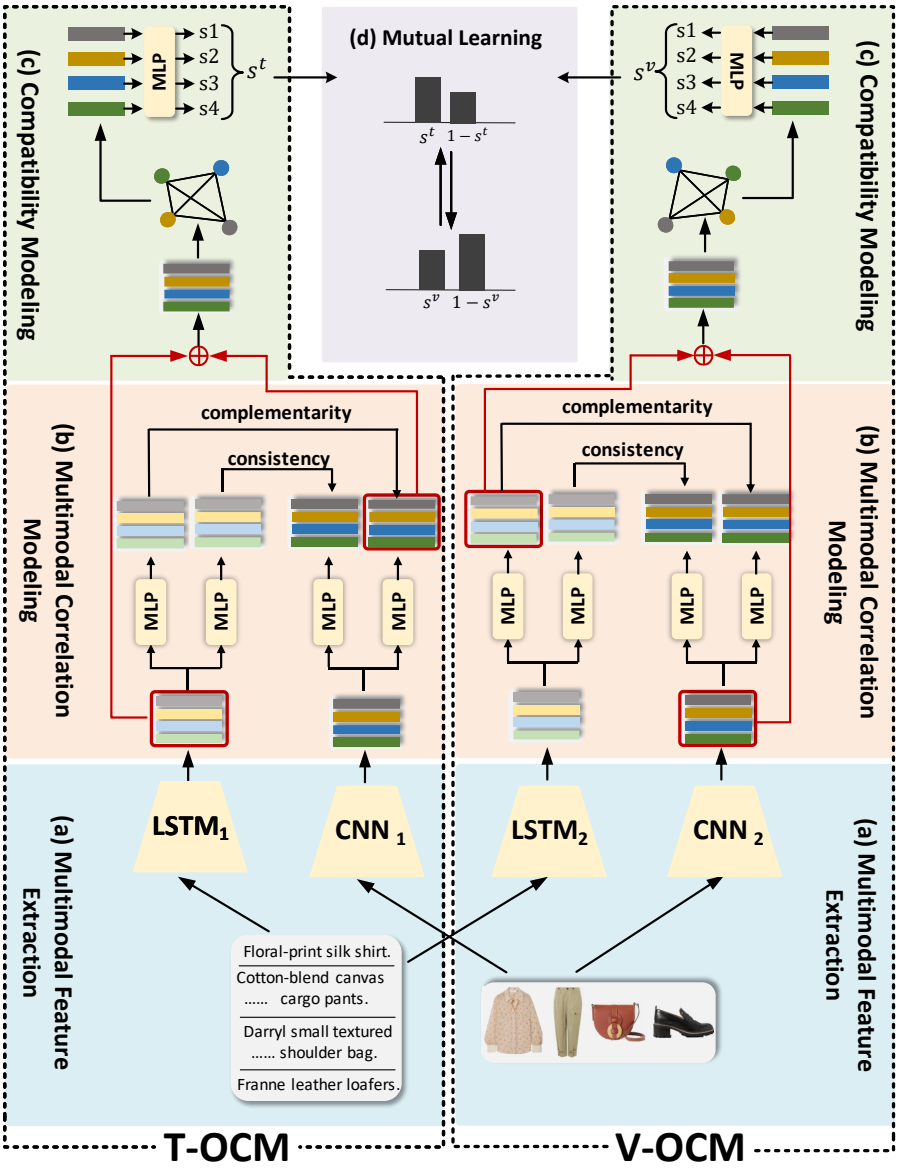


FIGURE 3.2: Illustration of the proposed MM-OCM scheme. It consists of four key components: (a) multimodal feature extraction, (b) multimodal correlation modeling, (c) compatibility modeling, and (d) mutual learning.

3.3.1 Problem Formulation

Formally, we first clarify the notations. We use bold uppercase letters (*e.g.*, $\mathbf{W}$) and bold lowercase letters (*e.g.*, $\mathbf{b}$) to represent matrices and vectors, respectively. All vectors are in column forms. Additionally, we employ nonbold letters (*e.g.*, W and W) to denote scalars and Greek letters (*e.g.*, α) to represent regularization parameters.

We deem the outfit compatibility modeling task as a binary classification problem. Assume that we have a training set Ω composed of N outfits, i.e, $\Omega = \{(O_i, y_i) \,|i = 1, \cdots, N\}$, where O_i is the i-th outfit, and y_i denotes the ground truth label. We set $y_i = 1$ if the outfit O_i is compatible, and $y_i = 0$ otherwise. Given an arbitrary outfit O, it can be

TABLE 3.1: Summary of the Main Notations.

Notation	Explanation
O_i	The i-th outfit.
y_i	The i-th outfit compatibility label.
v_i	a visual image of item i.
t_i	a textual description of item i.
$\mathbf{\Theta}_1$	The to-be-learned parameters in the T-OCM.
$\mathbf{\Theta}_2$	The to-be-learned parameters in the v-OCM.
$\hat{\mathbf{v}}_i$	The extracted visual feature for T-OCM of the i-th item.
$\tilde{\mathbf{v}}_i$	The extracted visual feature for V-OCM of the i-th item.
$\hat{\mathbf{t}}_i$	The extracted textual feature for T-OCM of the i-th item.
$\tilde{\mathbf{t}}_i$	The extracted textual feature for V-OCM of the i-th item.
$\hat{\mathbf{v}}_i^s$	The consistent representation of the visual modality for the i-th item.
$\hat{\mathbf{v}}_i^p$	The complementary representation of the visual modality for the i-th item.
$\hat{\mathbf{t}}_i^s$	The consistent representation of the textual modality for the i-th item.
$\hat{\mathbf{t}}_i^p$	The complementary representation of the textual modality for the i-th item.
$\hat{\mathbf{o}}_i$	The final representation for the i-th item based on the text-oriented multimodal fusion.
$\tilde{\mathbf{o}}_i$	The final representation for the i-th item based on the vision-oriented multimodal fusion.

represented as a set of fashion items, *i.e.*, $O = \{o_1, o_2, \cdots, o_m\}$, where o_i is the i-th item, associated with a visual image v_i and a textual description t_i. The symbol m is a variable for different outfits, considering that the number of items in an outfit is not fixed. Based on these training samples, we target learning an outfit compatibility model $\mathcal{F}$ that can judge whether the given outfit O is compatible,

$$s = \mathcal{F}\big(\{(v_i, t_i)\}_{i=1}^{m} | \mathbf{\Theta}\big), \tag{3.1}$$

where $\mathbf{\Theta}$ is a set of to-be-learned parameters of our model, and s denotes the probability the given outfit is compatible. Table 3.1 summarizes the main notations.

3.3.2 Multimodal Outfit Compatibility Modeling

Based upon the defined research problem and notations, we present the comprehensive **M**ulti**M**odal **O**utfit **C**ompatibility **M**odeling scheme, MM-OCM. As shown in Figure 3.2, it consists of four key components: (a) multimodal feature extraction, (b) multimodal correlation modeling, (c) compatibility modeling, and (d) mutual learning.

3.3.2.1 Multimodal Feature Extraction

We first introduce the visual and textual feature extraction.

Visual Feature Extraction. To extract visual features, we utilize the CNNs, which have shown compelling success in many computer vision tasks [47–49]. To facilitate the mutual enhancement between the T-OCM and the V-OCM, which are alternatively optimized, we employ two separate CNNs to extract the visual features. Specifically, given the outfit O, the visual feature of the i-th item in the outfit can be obtained as follows,

$$\begin{cases} \hat{\mathbf{v}}_i = \mathrm{CNN}_1\left(v_i\right), \\ \tilde{\mathbf{v}}_i = \mathrm{CNN}_2\left(v_i\right), \end{cases} \tag{3.2}$$

where $\hat{\mathbf{v}}_i \in \mathbb{R}^{d_v}$ and $\tilde{\mathbf{v}}_i \in \mathbb{R}^{d_v}$ refer to the visual features to be processed by the following T-OCM and V-OCM, respectively. The symbol d_v is the dimension of the extracted visual feature embedding. CNN_1 and CNN_2 denotes the corresponding CNNs for the T-OCM and V-OCM, respectively.

Textual Feature Extraction. Due to its prominent performance in textual representation learning [50, 51], we adopt LSTM to extract the textual feature of the given item[2]. Similar to the visual feature extraction, we also use two separate LSTMs, *i.e.*, LSTM_1 and LSTM_2, to obtain the textual features for T-OCM and V-OCM, respectively. Formally, we have

$$\begin{cases} \hat{\mathbf{t}}_i = \mathrm{LSTM}_1\left(t_i\right), \\ \tilde{\mathbf{t}}_i = \mathrm{LSTM}_2\left(t_i\right), \end{cases} \tag{3.3}$$

where $\hat{\mathbf{t}}_i \in \mathbb{R}^{d_t}$ and $\tilde{\mathbf{t}}_i \in \mathbb{R}^{d_t}$ refer to the text features for the following T-OCM and V-OCM, respectively. d_t is the feature dimension. To facilitate the multimodal fusion, we set $d_t = d_v = d$ in this work.

3.3.2.2 Multimodal Correlation Modeling

As illustrated in Figure 3.1, we argue that the visual image and textual description may possess certain consistency and complementarity information. Inspired by this, instead of unreasonably fusing the general multimodal features, we propose clearly separating and explicitly modeling the consistent and complementary contents of each modality, whereby we expect the consistent content of a modality can capture the alignment information between two modalities, and the complementary one of a modality can encode the supplement information to the other modality.

[2]Before feeding into the LSTM, the text is first tokenized into standard vocabularies.

In particular, we first introduce two MLPs to separate the consistent and complementary parts of each modality, respectively. Mathematically, we have

$$\begin{cases} \hat{\mathbf{v}}_i^s = \mathrm{MLP}_v^s\left(\hat{\mathbf{v}}_i\right), \hat{\mathbf{t}}_i^s = \mathrm{MLP}_t^s\left(\hat{\mathbf{t}}_i\right), \\ \hat{\mathbf{v}}_i^p = \mathrm{MLP}_v^p\left(\hat{\mathbf{v}}_i\right), \hat{\mathbf{t}}_i^p = \mathrm{MLP}_t^p\left(\hat{\mathbf{t}}_i\right), \end{cases} \tag{3.4}$$

where $\hat{\mathbf{v}}_i^s$ and $\hat{\mathbf{v}}_i^p$ respectively denote the consistent and complementary representation of the visual modality, and $\hat{\mathbf{t}}_i^s$ and $\hat{\mathbf{t}}_i^p$ denote that of the textual modality. It is worth mentioning that the consistent and complementary parts are probably inseparable within the original low-dimensional space. After nonlinear mapping via MLPs, we can project them into a high-dimensional space, whereby the consistent and complementary parts are distinguishable.

We then argue that the consistent representations of the two modalities are parallel, and the complementary representations are orthogonal. Accordingly, to regulate the consistent and complementary representations, we use the following objective functions:

$$\begin{cases} \mathcal{L}_s = \sum_{i=1}^{m} \{\cos(\hat{\mathbf{v}}_i^s, \hat{\mathbf{t}}_i^s)^2 + \cos(\tilde{\mathbf{v}}_i^s, \tilde{\mathbf{t}}_i^s)^2\}, \\ \mathcal{L}_p = \sum_{i=1}^{m} \{[\cos(\hat{\mathbf{v}}_i^p, \hat{\mathbf{t}}_i^p) - 1]^2 + [\cos(\tilde{\mathbf{v}}_i^p, \tilde{\mathbf{t}}_i^p) - 1]^2\}. \end{cases} \tag{3.5}$$

where $\mathcal{L}_s$ and $\mathcal{L}_p$ refer to the consistent and complementary regularizations, respectively.

3.3.2.3 Compatibility Modeling

We here first introduce the text/vision-oriented representation learning for each item, and we then present the text/vision-oriented compatibility modeling.

Text/Vision-oriented Representation Learning. Based upon the component of multi-modal correlation modeling, we can derive the complementary cues of the textual (visual) modality from the visual (textual) modality. Distinguished from the consistent parts that are shared between modalities, complementarity means exclusive and supplement information. Inspired by this, to learn comprehensive item representations and hence boost the outfit compatibility modeling performance, we introduce two multimodal fusion strategies: text-oriented multimodal fusion and vision-oriented multimodal fusion. As to the first strategy, we take the textual feature extracted by LSTM as the basis and additionally incorporate the complementary representation of the visual modality. By contrast, in the latter fusion strategy, we strengthen the visual feature extracted by CNN with the complementary representation of the textual modality. Specifically, based upon the consistent and complementary representation of each modality, we can derive

the final item representations from different fusion schemes as follows,

$$\begin{cases} \hat{\mathbf{o}}_i = \hat{\mathbf{t}}_i + \hat{\mathbf{v}}_i^p, \\ \tilde{\mathbf{o}}_i = \tilde{\mathbf{v}}_i + \tilde{\mathbf{t}}_i^p, \end{cases} \tag{3.6}$$

where $\hat{\mathbf{o}}_i$ and $\tilde{\mathbf{o}}_i$ denote the final item representation based on the text-oriented multimodal fusion and vision-oriented multimodal fusion, respectively.

Text/Vision-oriented Compatibility Modeling. Similar to previous studies, we employ a graph convolutional network (GCN) to flexibly model the compatibility of the outfit with a variable number of items. In particular, we adopt two GCNs, one for the T-OCM, and the other for the V-OCM. Regarding the limited space, we take the T-OCM as an example, since the V-OCM can be derived in the same way. In particular, for each outfit O composed of m fashion items, we first construct an indirected graph $G = (\mathcal{E}, \mathcal{R})$. $\mathcal{E} = \{o_i\}_{i=1}^m$ is the set of nodes, corresponding to the items of the given outfit O. Additionally, $\mathcal{R} = \{(o_i, o_j) \,|i, j \in [1, \cdots, m]\}$ denotes the set of edges. In this work, for each pair of items o_i and o_j in the outfit, we introduce an edge. During learning, each node o_i is associated with a hidden state vector $\mathbf{h}_i$, which keeps dynamically updated to fulfill the information propagation over the graph. For T-OCM, we initialize the hidden state vector for the i-th node based on the text-oriented representation of the i-th item, namely, $\mathbf{h}_i = \hat{\mathbf{o}}_i$.

The information propagation from item o_j to item o_i is defined as follows:

$$\mathbf{m}_{j \to i} = \phi[\mathbf{W}_{pp}(\mathbf{h}_i \odot \mathbf{h}_j) + \boldsymbol{b}_{pp}], \tag{3.7}$$

where $\mathbf{W}_{pp} \in \mathbb{R}^{d \times d}$ and $\mathbf{b}_{pp} \in \mathbb{R}^d$ denote the weight matrix and bias vector to be learned; $\phi(\cdot)$ is a nonlinear activation function, which is set as LeakyReLU; $\mathbf{h}_i \odot \mathbf{h}_j$ accounts for the interaction between the fashion item o_i and o_j; $\odot$ is the elementwise product operation. By summarizing the information propagated from all neighbors, the hidden state vector corresponding to the item o_i can be updated as follows,

$$\mathbf{h}_i^* = \phi\left(\mathbf{W}_0 \mathbf{h}_i + \mathbf{b}_0\right) + \sum_{o_j \in \mathcal{N}_i} \mathbf{m}_{j \to i}, \tag{3.8}$$

where $\mathbf{W}_0 \in \mathbb{R}^{d \times d}$ and $\mathbf{b}_0 \in \mathbb{R}^d$ denote the weight matrix and bias vector to be learned; $\mathcal{N}_i$ denotes the set of neighbor nodes of node o_i and $\mathbf{h}_i^* \in \mathbb{R}^d$ is the updated hidden representation of the item o_i.

We ultimately feed the updated item representation to an MLP, consisting of two fully-connected layers, to derive its probability of being a compatible outfit as follows,

$$\begin{cases} s_t^i = \mathbf{W}_2 \left[\psi \left(\mathbf{W}_1 \mathbf{h}_i^* + \mathbf{b}_1 \right) \right] + \mathbf{b}_2, \\ s_t = \sigma \Big(\dfrac{1}{m} \sum_{i=1}^{m} s_t^i \Big), \end{cases} \tag{3.9}$$

where $\mathbf{W}_1$, $\mathbf{b}_1$, $\mathbf{W}_2$, and $\mathbf{b}_2$ are the to-be-learned layer parameters. $\psi(\cdot)$ refers to the Relu active function, and $\sigma(\cdot)$ denotes the Sigmoid function to ensure the compatibility probability falling in the range of $[0, 1]$. Notably, in the same way, we can derive the compatible probability of the outfit by V-OCM, which is termed as s_v.

3.3.2.4 Mutual Learning

In a sense, regardless of the text-oriented item representation or the vision-oriented representation, *i.e.*, $\hat{\mathbf{o}}_i$ and $\tilde{\mathbf{o}}_i$, both of them fuse the multimodal data of an item. Therefore, the information encoded by these two representations is largely aligned, and hence the corresponding outfit compatibility modeling yields similar outputs. Additionally, they emphasize the different aspects of the item and hence may complement each other from a global view. Therefore, the knowledge learned by one compatibility model can guide the other model. Inspired by this, we turn to the deep mutual learning knowledge distillation scheme to regularize these two compatibility modeling results, mutually reinforcing them.

Unlike the traditional teacher-student knowledge distillation network, mutual learning replaces the one-way knowledge transfer from the static pretrained teacher to the student with the mutual knowledge distillation. In particular, an ensemble of student networks is employed to learn collaboratively. In our context, the T-OCM and the V-OTM can be treated as two student networks, and optimized alternatively. Namely, in each iteration, we only train one student network, while keeping the other fixed, which temporarily acts as the teacher.

We cast the compatibility modeling as a binary classification task, and adopt the widely-used cross-entropy loss for both T-OCM and V-OCM. Accordingly, we have the objective functions,

$$\begin{cases} \mathcal{L}_{ce}^t = -ylog(s_t) - (1-y)log(1-s_t), \\ \mathcal{L}_{ce}^v = -ylog(s_v) - (1-y)log(1-s_v), \end{cases} \tag{3.10}$$

where y refers to the ground truth label of the outfit O. $\mathcal{L}_{ce}^t$ and $\mathcal{L}_{ce}^v$ are the objective functions for the T-OCM and V-OCM, respectively.

To encourage the two student networks to learn from each other, we adopt the Kullback Leibler (KL) divergence loss function to penalize the distance between the evaluation results of the T-OCM and V-OCM as follows,

$$\begin{cases} \mathcal{L}^{v->t} = s_v log\frac{s_v}{s_t} + (1-s_v)log\frac{(1-s_v)}{(1-s_t)}, \\ \mathcal{L}^{t->v} = s_t log\frac{s_t}{s_v} + (1-s_t)log\frac{(1-s_t)}{(1-s_v)}. \end{cases} \tag{3.11}$$

Notably, we use $\mathcal{L}_{v->t}$ for training T-OCM, and $\mathcal{L}_{t->v}$ for training V-OCM. Finally, we have

$$\begin{cases} \mathcal{L}_t = \mathcal{L}_{ce}^t + \lambda\mathcal{L}^{v->t} + \eta\mathcal{L}_s + \mu\mathcal{L}_p, \\ \mathcal{L}_v = \mathcal{L}_{ce}^v + \lambda\mathcal{L}^{t->v} + \eta\mathcal{L}_s + \mu\mathcal{L}_p, \end{cases} \tag{3.12}$$

where λ, η, and μ are tradeoff hyperparameters. $\mathcal{L}_t$ and $\mathcal{L}_v$ are the final loss functions for the T-OCM and V-OCM, respectively. Each compatibility modeling component (*i.e.*, T-COM or V-OCM) not only learns to correctly predict the true label of the training instances, but also learns to mimic the output of the other compatibility modeling component, where the consistent and complementary regularizations are also jointly satisfied. Notably, although both $\mathcal{L}_t$ and $\mathcal{L}_v$ have consistent and complementary regularizations, *i.e.*, $\mathcal{L}_s$ and $\mathcal{L}_p$, the parameters to be optimized are distinguished, where the regularizations in $\mathcal{L}_t$ optimize the T-OCM, while that in $\mathcal{L}_s$ aim to learn parameters of V-OCM. Once our MM-OCM is well-trained, we take the average of the predicted compatibility probabilities of the V-OCM and T-OCM as the final compatibility probability of the outfit.

3.4 Experiment

In this section, we conducted experiments over two real-world datasets by answering the following research questions.

- **RQ1**: Does MM-OCM outperform state-of-the-art baselines?
- **RQ2**: How does each module affect MM-OCM?
- **RQ3**: How is the qualitative performance of MM-OCM?

3.4.1 Experimental Settings

In this part, we present the datasets, evaluation tasks, and implementation details.

Algorithm 1 The Training Procedure of Our MM-OCM.

Require: Training set Ω, hyper-parameters λ, η, and μ.
Ensure: Parameters $\mathbf{\Theta_1}$ in the T-OCM, and parameters $\mathbf{\Theta_2}$ in the V-OCM.
1: Initialize neural network parameters $\mathbf{\Theta_1}$ and $\mathbf{\Theta_2}$.
2: **repeat**
3: Sample minibatch from Ω.
4: Update the parameters $\mathbf{\Theta_1}$ according to $\mathcal{L}_t$ in Eqn.(3.12).
5: Update the parameters $\mathbf{\Theta_2}$ according to $\mathcal{L}_v$ in Eqn.(3.12).
6: **until** Convergence

3.4.1.1 Datasets

To evaluate the proposed method, we adopted the Polyvore Outfits dataset [21], which is widely utilized by several fashion analysis works [5, 52]. This dataset is collected from the Polyvore fashion website. Considering whether fashion items overlap in the training, validation and testing dataset, this dataset provides two dataset versions: the nondisjoint and disjoint versions, termed Polyvore Outfits and Polyvore Outfits-D. There are a total of $68,306$ outfits in Polyvore Outfits, divided into three sets: training set ($53,306$ outfits), validation set ($5,000$ outfits), and testing set ($10,000$ outfits). The disjoint version, Polyvore Outfits-D, contains a total of $32,140$ outfits, where $16,995$ outfits are for training, $3,000$ outfits are for validation, and $15,145$ outfits are for testing. Each outfit in the Polyvore Outfits has at least 2 items and up to 19 items, while that in the Polyvore Outfits-D has at least 2 items and up to 16 items. The average number of items in an outfit in Polyvore Outfits and Polyvore Outfits-D is 5.3 and 5.1, respectively.

3.4.1.2 Evaluation tasks

To evaluate the proposed model, we conducted experiments on two tasks: outfit compatibility estimation and fill-in-the-blank (FITB) fashion recommendation, which are illustrated in Figure 3.3.

Outfit compatibility estimation: This task is to estimate a compatibility score for a given outfit. Different from the previous study [3] that generates negative outfits randomly without any restriction, we replaced each item in the positive compatible outfit with another randomly selected item in the same category, which makes the task more challenging and practical. The ratio of positive and negative samples is set to $1:1$. The positive compatible outfits are labeled as 1, while the negative outfits are labeled as 0. Similar to previous studies[3, 5], we selected the area under the receiver operating characteristic curve (AUC) as the evaluation metric.

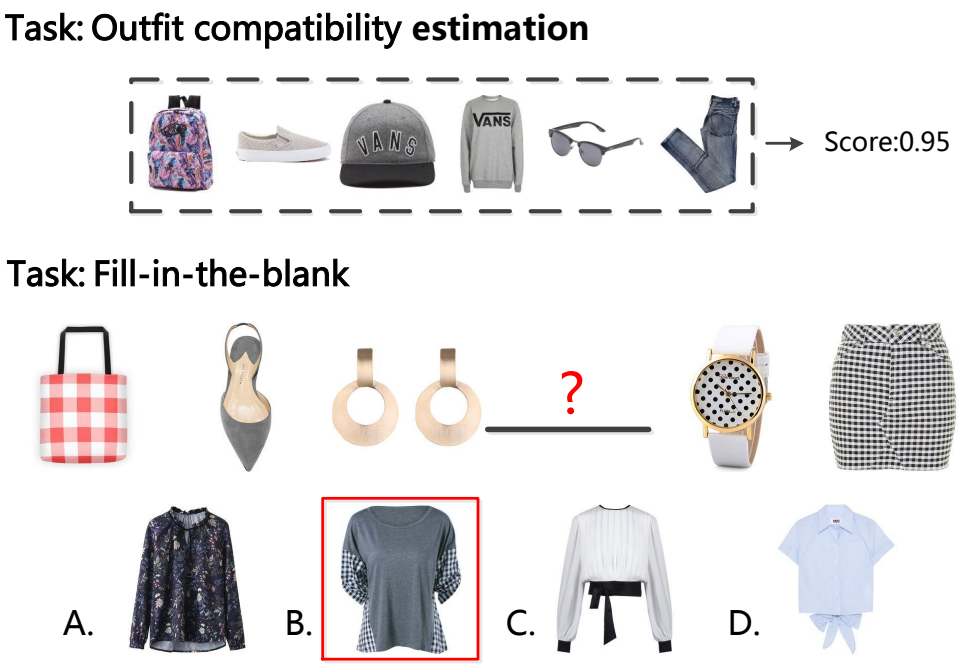


FIGURE 3.3: Illustration of outfit compatibility estimation and fill-in-the-blank tasks.

FITB fashion recommendation: Given an incomplete outfit and a target item annotated with the question mark, this task aims to select the most compatible fashion item from a candidate item set to fill in the blank and transform the given incomplete outfit into a compatible and complete outfit. This task is practical since people need to buy garments to match the garments they already have. Specifically, we constructed the FITB question by randomly selecting an item from a positive/compatible outfit as the target item and replacing it with a blank. We then randomly selected 3 items in the same category along with the target item to form the candidate set. The performance on this task was evaluated by the accuracy (ACC) of choosing the correct item from the candidate items.

3.4.1.3 Implementation Details

For the image encoder, we selected the ImageNet [53] pretrained ResNet18 [48] as the backbone, and modified the last layer to make the output feature dimension as 256. Regarding the text encoder, we set the word embedding size to 512, and the dimension of the hidden layer in LSTM to 256. We alternatively trained the T-OCM and V-OCM by the Adam optimizer [54] with a fixed learning rate of 0.0001, and batch size of 16. The tradeoff hyperparameters in Eqn.(3.12) are set as $\lambda = \eta = \mu = 1$. In particular, we launched 10-fold cross validation for each experiment and reported the average results.

All the experiments were implemented by PyTorch on a server equipped with 4 NVIDIA TITAN Xp GPUs, and the random seeds were fixed for reproducibility.

3.4.2 Model Comparison

To validate the effectiveness of our proposed scheme, we chose the following baselines for comparison.

- **Bi-LSTM** [3] takes the items in an outfit as a sequence ordered by the item category and fulfills the fashion compatibility modeling with Bi-LSTM. For a fair comparison, we only utilized the visual information.
- **Type-aware** [21] designs type-specific embedding spaces according to the item category, where the textual item descriptions are adopted via the visual-semantic loss.
- **SCE-NET** [52] is a pairwise method, which utilizes multiple similarity condition masks to embed the item features into different semantic subspaces. This method also considers textual information.
- **NGNN** [18] employs a GNN to address the compatibility modeling task, where the node is updated by a gate mechanism. For multimodal features, NGNN designs two graph channels, and the final compatibility score is derived as a weighted average.
- **Context-aware** [19] regards fashion compatibility modeling as an edge prediction problem, where a graph autoencoder framework is introduced. Notably, only the visual features are employed.
- **HFGN** [5] shares the same spirit with NGNN and builds a category-oriented graph, where an R-view attention map and an R-view score map are introduced to compute the compatibility score. This baseline only uses visual features.

Table 3.2 shows the performance comparison among different methods on two datasets under two tasks. From this table, we make the following observations.

- Among all the baselines, Bi-LSTM performs the worst, which suggests that modeling the outfit as an ordered list of items is not reasonable.
- The methods that use multimodal features gain more promising results (*e.g.*, Type-aware on Polyvore Outfits and SCE-NET on Polyvore Outfits-D) compared to those that only utilize the visual features (*i.e.*, HFGN and Context-aware). This

TABLE 3.2: Performance comparison between our proposed MM-OCM scheme and other baselines over two datasets. The baselines were re-trained by their released codes. The best results are in boldface, and the second best are underlined.

Method	Polyvore Outfits		Polyvore Outfits-D	
	Compat. AUC	FITB Accuracy	Compat. AUC	FITB Accuracy
Bi-LSTM	0.68	42.20%	0.65	40.10%
Type-aware	<u>0.87</u>	<u>56.60%</u>	0.78	47.30%
SCE-NET	0.83	52.80%	<u>0.82</u>	<u>52.10%</u>
NGNN	0.75	53.02%	0.68	42.49%
Context-aware	0.81	55.63%	0.77	50.34%
HFGN	0.84	49.90%	0.70	39.03%
MM-OCM	**0.93**	**63.40%**	**0.88**	**58.02%**

implies that considering both visual and textual modalities is rewarding in the outfit compatibility modeling task.

- MM-OCM consistently surpasses all baseline methods on the two datasets under both tasks. This indicates the advantage of our scheme that utilizes multimodal correlation modeling and mutual learning in the context of outfit compatibility modeling. Notably, we performed the ten-fold t-test between our proposed scheme and each of the baselines. We observed that all the p-values are much smaller than 0.05, and we concluded that the MM-OCM is significantly better than the baselines.

3.4.3 Ablation Study

To verify the importance of each component in our model, we also compared MM-OCM with the following derivatives.

- **w/o Correlation**: To explore the effect of multimodal correlation modeling, we removed this component by setting $\hat{\mathbf{o}}_i = \hat{\mathbf{t}}_i$ and $\tilde{\mathbf{o}}_i = \tilde{\mathbf{v}}_i$ in Eqn.(3.6).
- **w/o Mutual**: To study the effect of the mutual learning component, we removed the knowledge distillation between the T-OCM and V-OCM by setting $\lambda = 0$.
- **Image_Only** and **Text_Only**: The two derivatives were set to verify the importance of visual and textual information. Specifically, for the Image_Only, we removed T-OCM by setting $\tilde{\mathbf{o}}_i = \tilde{\mathbf{v}}_i$, while for the Text_Only, V-OCM was removed by setting $\hat{\mathbf{o}}_i = \hat{\mathbf{t}}_i$.
- **Concat_Directly**: To gain more insights into our utilization of visual and textual information, we directly concatenated the visual and textual features of each item

TABLE 3.3: Ablation study of our proposed MM-OCM scheme on two datasets. The best results are in boldface.

Method	Polyvore Outfits		Polyvore Outfits-D	
	Compat. AUC	FITB Accuracy	Compat. AUC	FITB Accuracy
w/o Correlation	0.91	52.91%	0.87	54.47%
w/o Mutual	0.92	60.80%	0.86	55.62%
Image_Only	0.90	57.80%	0.85	52.85%
Text_Only	0.79	42.28%	0.74	35.45%
Concat_Directly	0.91	58.67%	0.79	49.73%
MM-OCM	**0.93**	**63.40%**	**0.88**	**58.02%**

and fed them to an MLP to obtain $\hat{\mathbf{o}}_i$. Accordingly, the correlation modeling and mutual learning were simultaneously removed.

Table 3.3 shows the ablation results of our MM-OCM. From this table, we make the following observations.

- w/o Correlation performs worse than our MM-OCM, which proves the effectiveness of the proposed multimodal consistency and complementarity modeling.
- MM-OCM surpasses w/o Mutual, indicating that the mutual learning component is helpful for integrating the T-OCM and V-OCM by transferring knowledge between the two modules.
- Both Image_Only and Text_Only are inferior to MM-OCM, which suggests that it is essential to consider both visual and textual information to gain better outfit compatibility modeling effects. In addition, Image_Only outperforms Text_Only remarkably, which reflects that the image contains more useful information than the text, which corresponds with the saying that "a picture is worth a thousand words".
- Compared to our MM-OCM, Concat_Directly also delivers worse performance, implying that simply fusing visual and textual features is insufficient to explore the intrinsic correlation of the two modalities. This further verifies the superiority of our strategy that models the multimodal correlation and devises two schemes of multimodal fusion. Furthermore, it can be observed that on the more challenging Polyvore Outfits-D dataset, the results of Concat_Directly are better than those of Text_Only but worse than those of Image_Only. This phenomenon indicates that an inappropriate multimodal fusion method is less effective than only utilizing the more informative modality.

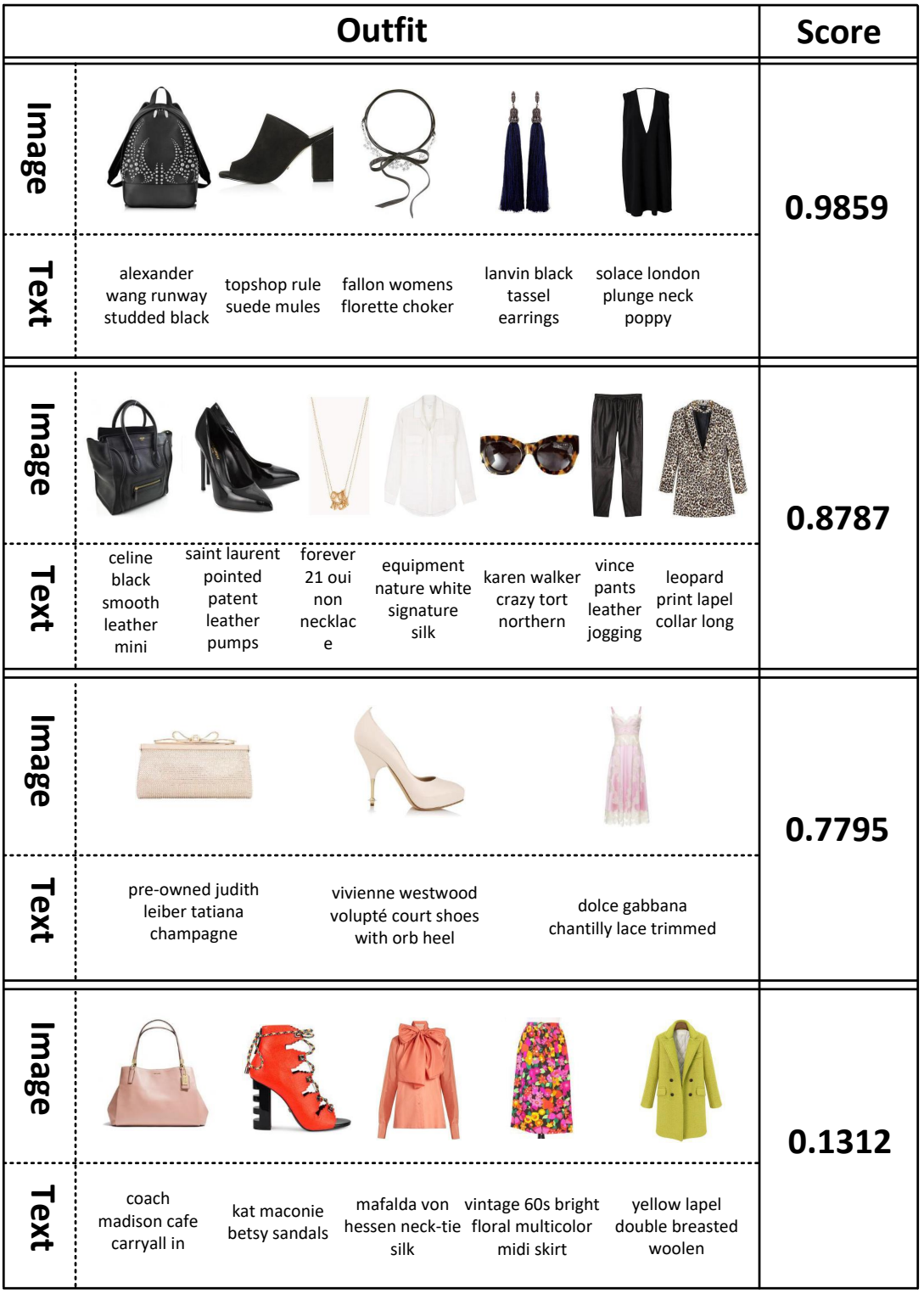


FIGURE 3.4: Qualitative results of MM-OCM on the task of outfit compatibility estimation.

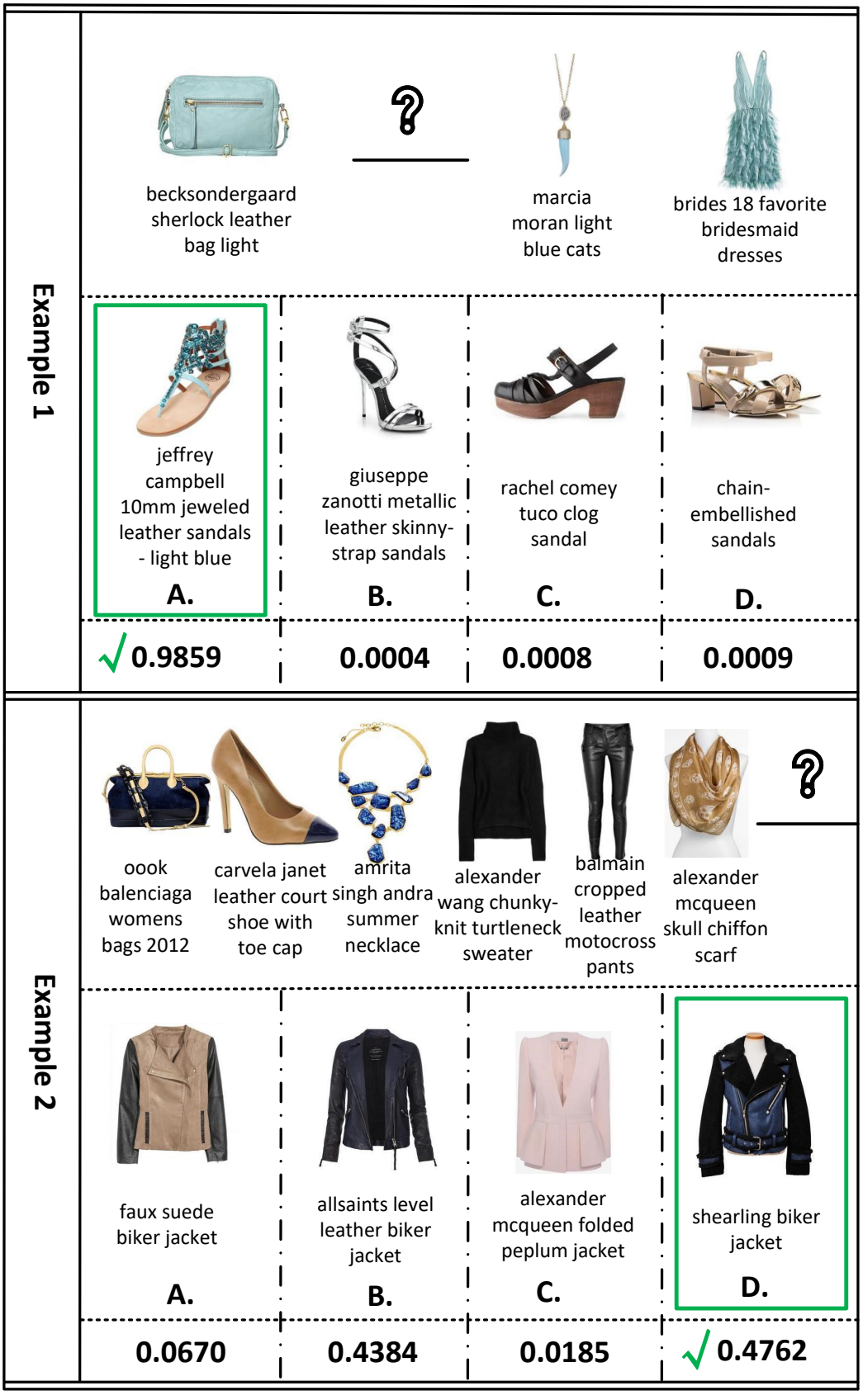


FIGURE 3.5: Qualitative results of MM-OCM on the task of fill-in-the-blank.

3.4.4 Case Study

To gain a thorough understanding of our model, we also conducted a qualitative evaluation of our method. Figure 3.4 and Figure 3.5 intuitively show several testing examples on the outfit compatibility estimation and fill-in-the-blank tasks. From Figure 3.4, we observed that for the example in the first row, which contains items with consistently black color and elegant style, our MM-OCM can assign it with a high compatible probability. As for the outfit in the last row with obviously incompatible colors, *e.g.*, green does not go well with red, our MM-OCM gives a low compatibility score. In Figure 3.5, we can see that our method can choose the most suitable item from the candidate set to form a compatible outfit. For the example in the first row, the outfit lacks a pair of shoes and our MM-OCM correctly selects the first item by attributing a high compatibility score. As can be seen, the selected item matches well with other items in the query. As to the example in the second row, although our method chooses the correct answer (item D), it also gives a high compatibility score to the item B, since these two items are both dark jackets of the same style. This reconfirms the compatibility modeling capabilities of our model.

3.5 Summary

In this chapter, we solved the outfit compatibility modeling problem with graph convolutional networks by exploring the multimodal correlations. In particular, we clearly separated and explicitly modeled the consistent and complementary relations between the visual and textual modalities. This was accomplished by nonlinearly projecting the consistent and complementary contents into the separable spaces, whereby they were respectively formulated by parallel and orthogonal regularizers. We then applied the complementary information to strengthen the vision- and text-oriented representations. Based upon these two kinds of representations, two compatibility modeling brunches were derived and reinforced by mutual learning via knowledge transfer. Extensive experiments over two benchmark datasets verified the effectiveness of our proposed MM-OCM scheme compared with several state-of-the-art baselines.

Chapter 4

Partially Supervised Disentangled Graph Learning for OCM

4.1 Introduction

In Chapter 3, we studied the correlation-oriented outfit compatibility modeling. Despite its effectiveness, it still suffers from two key limitations. 1) It evaluate the outfit compatibility based on the single latent compatibility space. The outfit compatibility is essentially affected by multiple complementary hidden factors, such as the color, style, shape, and material. Therefore, we argue that previous methods can only achieve the suboptimal solution, as it entangles all the factors in a single latent space. 2) It only investigates the visual content of fashion items while overlooking the items' semantic attributes. The item attribute labels usually contain rich information that characterizes the key item parts, which can be adopted to supervise the attribute-level representation learning, and hence promote the model's performance as well as interpretability. Thus, in this chapter, we aim to fulfill the fine-grained outfit compatibility modeling by incorporating the semantic attributes of fashion items.

However, fulfilling this goal is nontrivial due to the following challenges. 1) The fashion item attribute labels are not unified or aligned. In other words, each item may have different attribute labels. For instance, as shown in Figure 4.1, one T-shirt is labeled with attributes of price, sleeve length, design, and brand; while the other has color, material, brand, and price. Thereby, how to fully take advantage of these irregular attribute labels to partially supervise the attribute-level representation learning of fashion items poses a considerable challenge. 2) When disentangling the entire visual embedding into multiple attribute-level representations, how to ensure information intactness during the disentanglement is another challenge. 3) To comprehensively capture the compatibility

among fashion items, we incorporate both the coarse-grained item-level and fine-grained attribute-level information into the compatibility modeling. Accordingly, how to seamlessly combine multiple granularities to strengthen the learning performance constitutes another tough challenge.

To address the aforementioned challenges, we present a partially supervised compatibility modeling scheme, called PS-OCM. As shown in Figure 4.2, it consists of three key components: 1) partially supervised attribute-level embedding learning, 2) disentangled completeness regularization, and 3) hierarchical outfit compatibility modeling. Specifically, the first component extracts visual features from each composing item of the given outfit via a pretrained model. It then turns to disentangle the visual feature vector into a set of fine-grained attribute embeddings, which is partially supervised by the irregular attribute labels of each fashion item. The second component works toward an intact disentanglement. This is accomplished by adopting two strategies: orthogonal residual embedding and visual representation reconstruction. An orthogonal residual embedding is introduced to compensate for the information loss, and regularize the orthogonal relationship between the residual embedding and each attribute-level embedding. Additionally, it leverages the deconvolution neural network to ensure that the original image can be reconstructed from the disentangled attribute-level and residual embeddings. The last component contains a hierarchical graph convolutional network, which models the outfit compatibility by jointly integrating the fine-grained attribute-level and coarse-grained item-level information. Ultimately, it fuses the attribute-level compatibility scores and the item-level ones via a multilayer perceptron (MLP) to derive the final compatibility score of the given outfit.

The research work in this chapter has been published in IEEE TIP 2021.

Weili Guan, Haokun Wen, Xuemeng Song, Chun Wang, Chung-Hsing Yeh, Xiaojun Chang, Liqiang Nie. "Partially Supervised Compatibility Modeling." In IEEE Transaction on Image Processing. IEEE, 2021.

4.2 Related Work

Disentangled representation learning. Disentangled representation learning [55] targets learning multiple factorized representations to capture the latent explanatory factors residing in the observed data, which has drawn increasing research attention from various domains, such as the recommendation domain [56, 57] and computer vision domain [58–60]. For example, in the recommendation domain, Hu *et al.* [61] proposed a

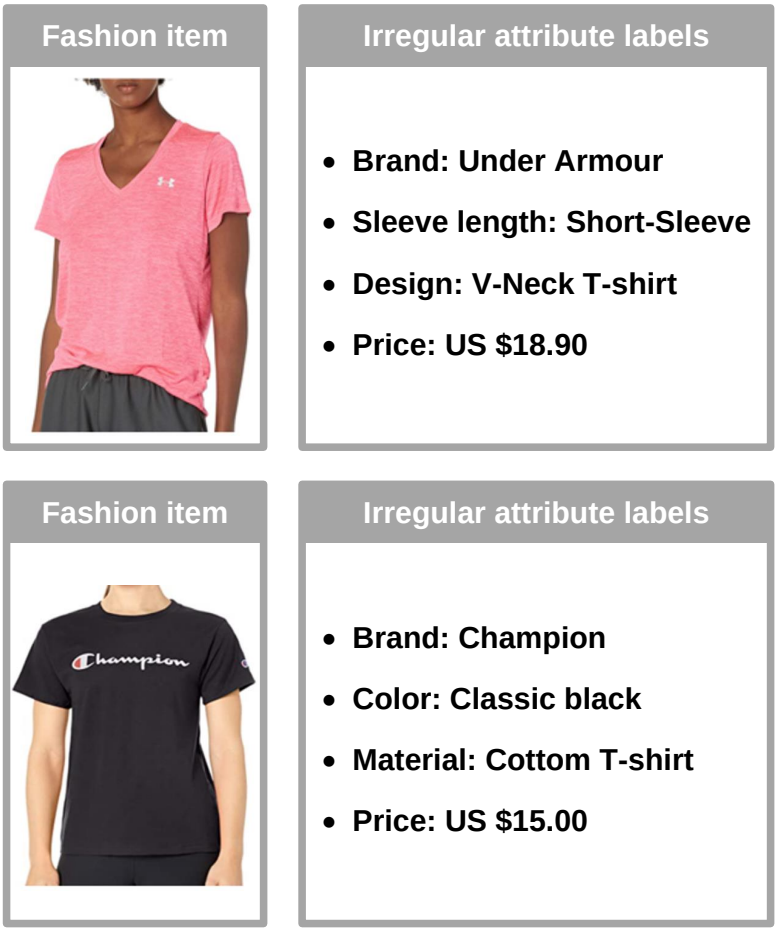


FIGURE 4.1: Illustration of two fashion items and their associated irregular attribute labels.

novel graph neural news recommendation model with unsupervised preference disentanglement, where a neighborhood routing mechanism is introduced to dynamically identify the latent preference factors affecting the click between a user and a piece of news. In addition, Wang *et al.* [62] presented a disentangled graph collaborative filtering model to mine the fine-grained user-item relationships. In the computer vision field, Ma *et al.* [58] strengthened the natural person image generation by disentangling the input image into three intermediate embedding features, corresponding to three main factors: foreground, background, and pose.

As the compatibility relationship among fashion items can be influenced by multiple latent factors, such as color, texture, and style, some researchers also incorporated the disentangled representation to address the task of fashion compatibility modeling. For example, Zheng *et al.* [63] devised a disentangled graph learning scheme, where the collocation compatibility is disentangled into multiple fine-grained compatibilities among fashion items. Similarly, Guan *et al.* [64] presented a comprehensive multimodal outfit

compatibility modeling scheme, which not only explores the fine-grained outfit compatibility with disentangled item representations but also explicitly models the consistent and complementary correlations between the visual and textual modalities of items. Despite their significant value, the existing efforts mostly overlook the potential of the semantic labels in supervising the disentangled representation learning. Therefore, in this work, we propose utilizing the irregular attributes as partial supervision to guide the disentangled representation learning of items and introducing the completeness regularizer to prevent information loss during disentanglement.

Graph Convolutional Network. Graph Neural Network (GNN) is devised to learn effective graph representations by updating the node embedding via information aggregation from the node's neighbors. Initially, Gori et al. [65] utilized the graph neural network to model the relationship among a set of items. To remedy the long-term message propagation problem, Li et al. [66] introduced the Gate Recurrent Units (GRU) in the propagation process. Although GNNs can be applied to most types of graphs, it is hard to train for a fixed point. Inspired by this, Kipf et al. [67] introduced the Graph Convolutional Network (GCN), which applies the convolutional operations directly on graphs by updating each node's representation via the information aggregation from its neighbor nodes. In order to improve the model generalization ability, Hamilton et al. [68] presented a general inductive framework to learn a function that generates embeddings by sampling and aggregating features from a node's local neighborhood. Thus far, GCNs have been widely explored in various tasks, including but not limited to the tasks of visual comprehension [69], natural language processing [70], recommendation [71], and image recognition [72]. By virtue of its powerful modeling capabilities for unstructured data, we elaborate a hierarchical GCN-based outfit compatibility modeling scheme, where the attribute-level and item-level compatibility modeling is jointly investigated.

4.3 Methodology

In this section, we first formulate the research problem and then detail the proposed partially supervised outfit compatibility modeling scheme (PS-OCM).

4.3.1 Problem Formulation

Formally, we first clarify the notations. We use bold uppercase letters (*e.g.*, $\mathbf{W}$) and bold lowercase letters (*e.g.*, $\mathbf{b}$) to represent matrices and vectors, respectively. All vectors are

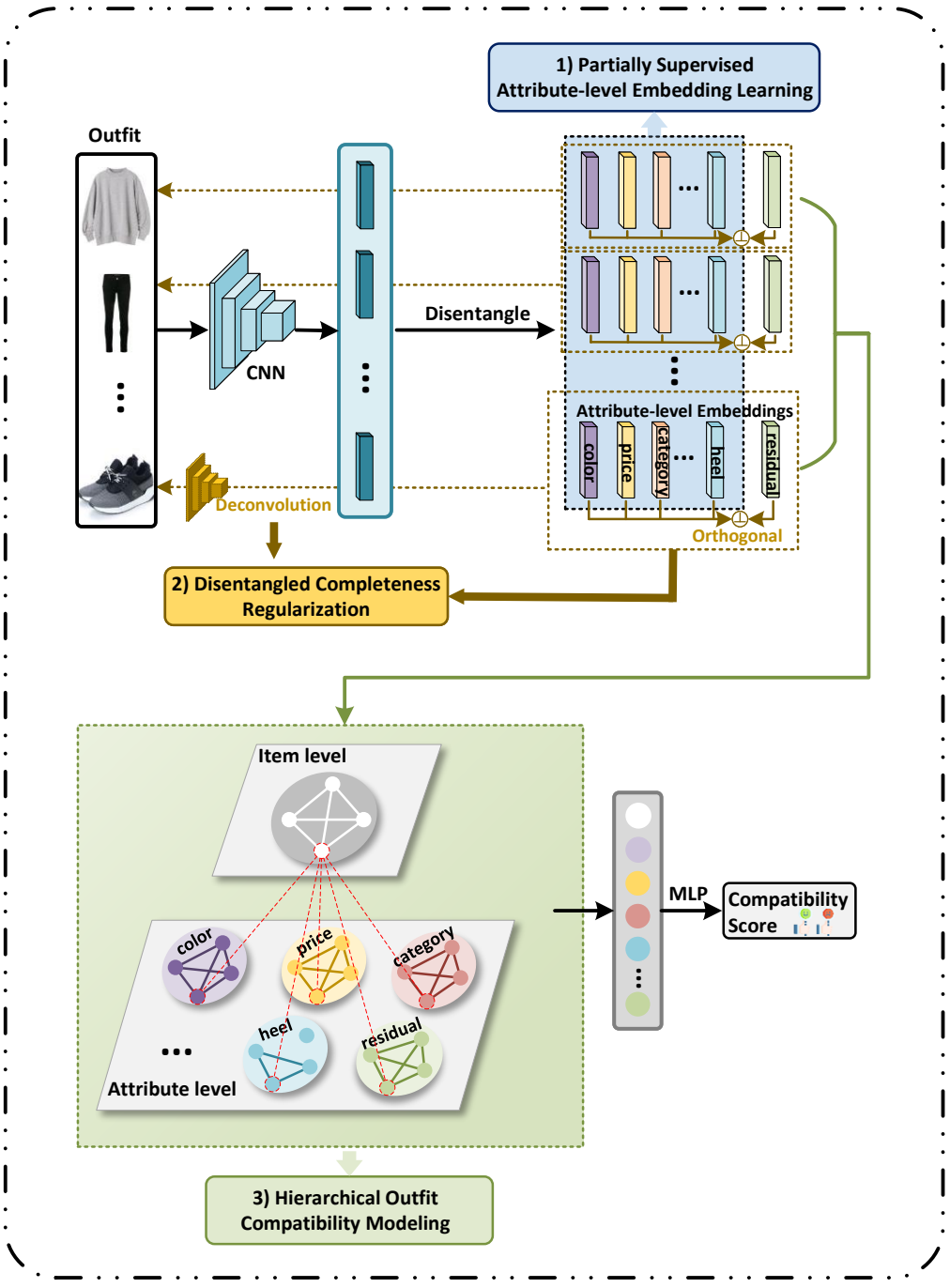


FIGURE 4.2: Illustration of our proposed PS-OCM scheme. It consists of three components: partially supervised attribute-level embedding learning, disentangled completeness regularization, and hierarchical outfit compatibility modeling.

in column forms. Additionally, we employ nonbold letters (*e.g.*, W and W) to denote scalars and Greek letters (*e.g.*, α) to represent regularization parameters.

In this work, we cast the outfit compatibility modeling task as a binary classification problem, *i.e.*, *whether the given outfit is compatible*. Assume we have a set of N outfits, denoted as $\Omega = \{(O_i, y_i)\}_{i=1}^{N}$, where O_i is the i-th outfit, and y_i denotes its corresponding

compatibility label. Specifically, $y_i = 1$ if the outfit O_i is compatible, and $y_i = 0$, otherwise. In addition, we have a set of fashion items $\mathcal{I}$ distributed over T categories. For simplicity, we temporally omit the subscript i of each outfit. An outfit O comprises K fashion items, *i.e.*, $\{I_1, I_2, \cdots, I_K\}$, where $I_i \in \mathcal{I}$ is the i-th composing item of the outfit. Considering that the number of items in an outfit is not fixed, K is a variable. Each item I_i is associated with a visual image V_i and a set of attribute labels $\mathcal{L}_i$. We heuristically predefined a set of attributes (*e.g.*, the *color* and *material*) $\mathcal{A} = \{a_m\}_{m=1}^{M}$ that can be applied to characterize all the fashion items, where a_m is the m-th attribute, and M is the total number of attributes. Moreover, each attribute has a set of corresponding attribute values, *e.g.*, *red* and *blue* are two possible values for the attribute *color*. We then formally use $\mathcal{V}_m = \{v_m^n\}_{n=1}^{N_m}$ to denote all the possible values for the attribute a_m, and N_m is the corresponding total number of values. Therefore, the set of attribute labels of the i-th item can be written as $\mathcal{L}_i = \{l_i^1, l_i^2, \cdots, l_i^M\}$, where $l_i^m \in \mathcal{V}_m$ if the item I_i has m-th attribute; otherwise, $l_i^m = none$. Usually there are two possible reasons leading to $l_i^m = none$: one reason is the intrinsic flaws of the dataset due to loose user-generated annotation, and the other is that items of certain categories essentially cannot present certain attributes (*e.g.*, the *trousers* do not have the attribute of *sleeve length*).

In this work, we target at learning an outfit compatibility model $\mathcal{F}$ to judge whether a given outfit O is compatible. It is formulated as follows,

$$s = \mathcal{F}\big(\{(V_i, \mathcal{L}_i)\}_{i=1}^{K} | \mathbf{\Theta}\big), \tag{4.1}$$

where $\mathbf{\Theta}$ refers to the to-be-learned parameters of our model, and s denotes the compatible probability of the given outfit. Table 4.1 summarizes the main notations.

4.3.2 Partially Supervised Compatibility Modeling

As illustrated in Figure 4.2, PS-OCM consists of three key components: 1) partially supervised attribute-level embedding learning, 2) disentangled completeness regularization, and 3) hierarchical outfit compatibility modeling. We explain them as follows.

4.3.2.1 Partially Supervised Attribute-Level Embedding Learning

This component aims to derive the fine-grained attribute-level representation of the fashion item, which is the basis for the following hierarchical outfit compatibility modeling. Given an outfit, we first extract the visual feature of each composing item via the convolutional neural networks, which have obtained remarkable success in many computer vision tasks [73, 74]. Specifically, we obtain the overall visual feature embedding of the

TABLE 4.1: Summary of the Main Notations.

Notation	Explanation
Ω	The set of outfits.
O_i	The i-th outfit.
y_i	The i-th outfit compatibility label.
I_i	The i-th item of an outfit.
V_i	The visual image of the i-th item of an outfit.
$\mathcal{L}_i$	The set of attribute labels of the i-th item of an outfit.
$\mathcal{A}$	The set of predefined attributes.
a_m	The m-th attribute.
$\mathcal{V}_m$	The set of possible values for the attribute a_m.
v_m^n	The n-th attribute value of the m-th attribute.
l_i^m	The i-th item's m-th attribute label.
$\mathbf{v}_i$	The extracted visual feature of the i-th item.
$\mathbf{e}_i^j$	The j-th disentangled attribute-level embedding of I_i.
$\mathcal{L}_i$	The set of attribute labels of the i-th item.
p_i^m	The binarized mask denoting whether item I_i has a_m.
q_t^m	The binarized mask denoting whether the attribute a_m is meaningful to items of the t-th category.
$\mathbf{\Theta}$	The to-be-learned parameters of the proposed model.
$\mathbf{e}_i^{M+1}$	The residual attribute embedding.
$\hat{\mathbf{V}}_i$	The reconstructed visual representation of the i-th item.
$\mathbf{q}_{t_i^*}^m$	The binary masks.
$\mathcal{G}_a^m$	The m-th parallel compatibility modeling graph.
α_{ij}	The importance of node n_j's hidden state to node n_i.
$\tilde{\mathbf{h}}_i$	The updated hidden representation of node n_i.

i-th item in the outfit O as follows,

$$\mathbf{v}_i = \mathrm{CNN}\left(\mathbf{V}_i\right), \tag{4.2}$$

where $\mathbf{V}_i$ refers to the i-th item image in its raw RGB pixels, $\mathbf{v}_i \in \mathbb{R}^{D_v}$ denotes the extracted visual feature of the i-th item, and D_v is the dimension of the visual feature. In this work, the function CNN refers to ResNet18 [48] pretrained on ImageNet.

As previously mentioned, we predefined a set of M attributes to characterize all the items. Accordingly, we disentangle the visual feature of each item I_i, *i.e.*, $\mathbf{v}_i$, into M attribute-level embeddings. We argue that the attributes are not linearly separable, and hence accomplish this task by the nonlinear MLP mapping. Mathematically, we have

$$\begin{cases} \mathbf{e}_i^1 = \mathrm{MLP}_1\left(\mathbf{v}_i\right), \\ \mathbf{e}_i^2 = \mathrm{MLP}_2\left(\mathbf{v}_i\right), \\ \qquad \vdots \\ \mathbf{e}_i^M = \mathrm{MLP}_M\left(\mathbf{v}_i\right), \end{cases} \tag{4.3}$$

where $\mathbf{e}_i^j \in \mathbb{R}^{D_e}$ $(j = 1, \cdots, M)$ denotes the j-th disentangled attribute-level embedding of the i-th item, and D_e is the dimension.

Different from existing studies that focus on the unsupervised disentangled representation learning, we argue that even the irregular attribute labels of fashion items contain rich cues. Therefore, they can be used to supervise the attribute-level embedding learning and hence strengthen the final compatibility modeling performance. Thereby, we further utilize M MLPs as the attribute classifiers to explore the attribute labels. As aforementioned, the fashion item attribute labels are irregular. We thus introduce a binary mask $\mathbf{p}_i$ for each item I_i in the outfit to select the available attribute labels of the i-th item. In particular, we define the mask as $\mathbf{p}_i = [p_i^1, p_i^2, \cdots, p_i^M]$, where $p_i^m = \phi(l_i^m)$, and $\phi(\cdot)$ is an indicator function defined as follows,

$$\phi(x) = \begin{cases} 0 & x \textit{ is none}, \\ 1 & \textit{else}. \end{cases} \tag{4.4}$$

By utilizing the binarized mask, if and only if the item has the corresponding attribute label, we enforce the supervision over the embedding for that attribute. In particular, we adopt the cross-entropy loss to achieve the partial supervision. Formally, for a given outfit O consisting of K items, the partial supervision loss function is formulated as follows,

$$\mathcal{L}_{ps} = \sum_{i=1}^{K} \sum_{m=1}^{M} -\log\left(p\left(l_i^m \mid \mathcal{C}^m\left(\mathbf{e}_i^m\right)\right)\right) p_i^m, \tag{4.5}$$

where $\mathcal{C}^m(\cdot)$ is the label classifier for the m-th attribute, $\mathbf{e}_i^m$ is the disentangled embedding of the m-th attribute, and l_i^m is the ground-truth attribute label. We illustrate the procedure of partially supervised disentangled attribute-level embedding in Figure 4.3.

4.3.2.2 Disentangled Completeness Regularization

To prevent information loss during the disentangling process which may degrade the model performance, we devise a disentangled completeness regularizer, as illustrated in Figure 4.2. In particular, we rely on two strategies to regulate the disentangling process: orthogonal residual embedding and visual representation reconstruction.

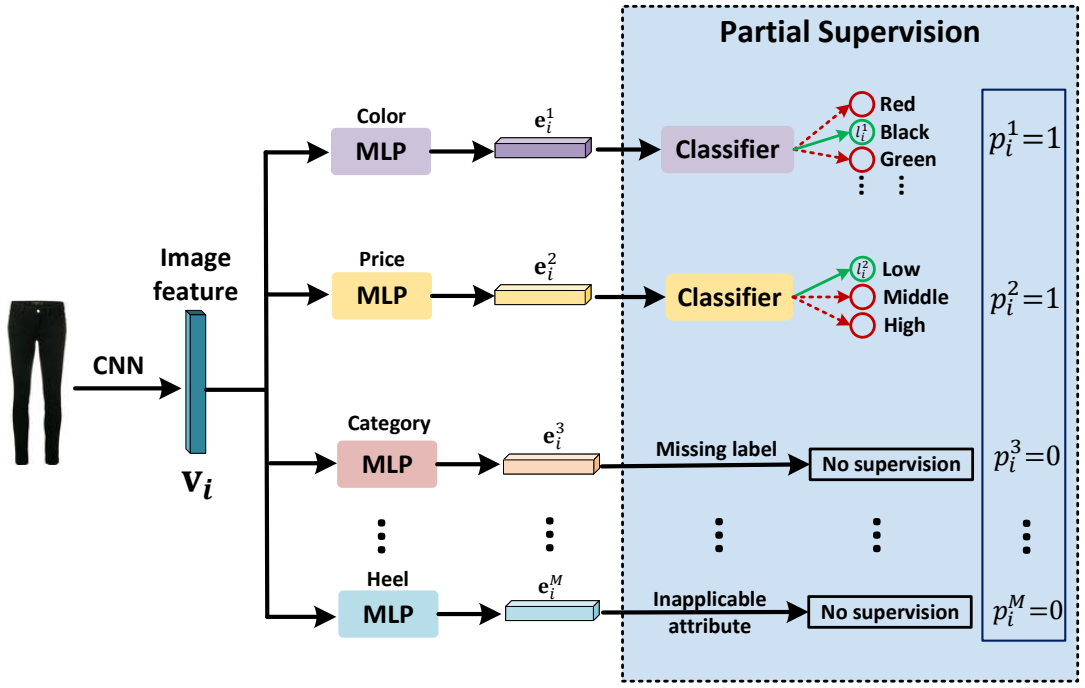


FIGURE 4.3: Illustration of the partially supervised attribute-level embedding learning module.

Orthogonal Residual Embedding. There may be some implicit visual properties of the item that cannot be represented by the predefined set of attributes. We thus introduce another special attribute *residual* to compensate for the information loss during the disentangled representation learning. Specifically, similar to the M attribute-level embeddings, we adopt another MLP to derive the residual attribute embedding via,

$$\mathbf{e}_i^{M+1} = \mathrm{MLP}_{M+1}(\mathbf{v}_i), \tag{4.6}$$

where $\mathbf{e}_i^{M+1} \in \mathbb{R}^{D_e}$ denotes the residual attribute embedding.

Since the residual attribute embedding acts as compensation for fully representing the item, we argue that it should be complementary to other M attribute-level embeddings that have clear semantics. In other words, the residual embedding should be orthogonal to every other attribute-level embedding. It is worth noting that although we disentangle the visual feature of each fashion item into M attribute-level embeddings, certain embeddings of the given item may be meaningless since some attributes are not universal and cannot be applied to certain items. For example, we can discuss the attribute *sleeve length* for a *T-shirt* but not *trousers*, and the attribute *heel* for a pair of *shoes* rather than a *T-shirt*. Therefore, for each item category, we define a set of meaningful attributes to guarantee effective orthogonal regularization. Thus, we first build the *category-attribute* associations. For the t-th category, we take the union set of attributes

used to label items in the t-th category as the whole set of applicable attributes, denoted as $\mathcal{T}_t$. We then introduce a mask $\mathbf{q}_t = [q_t^1, q_t^2, \cdots, q_t^M]$ to select the meaningful attributes for the t-th item category, where $q_t^m = 1$ if the predefined m-th attribute belongs to the applicable attribute set $\mathcal{T}_t$; otherwise, $q_t^m = 0$. It is worth noting that in the aforementioned partial supervision module, only the attribute-level embeddings that have corresponding labels are triggered. Whereas in this orthogonal regularization, we further utilize the attribute-level embedding that even has no corresponding label, as long as it can be possibly presented by this item.

Ultimately, we have the following orthogonal regularization,

$$\begin{aligned}\mathcal{L}_{or} &= \sum_{i=1}^{K}\sum_{m=1}^{M}\left[\cos\left(\hat{\mathbf{e}}_i^m, \mathbf{e}_i^{M+1}\right)\right]^2 \\ &= \sum_{i=1}^{K}\sum_{m=1}^{M}\left[\cos\left(q_{t_i^*}^m\mathbf{e}_i^m, \mathbf{e}_i^{M+1}\right)\right]^2,\end{aligned}$$

where $\cos(\cdot,\cdot)$ is the cosine similarity function, and $t_i^* \in \{1, 2, \cdots, T\}$ refers to the category of the i-th item. It is worth noting that once the m-th attribute cannot be applied to the item I_i, *i.e.*, $q_{t_i^*} = 0$, we ignore the orthogonal regularization between that attribute-level embedding and the residual embedding.

Visual Representation Reconstruction. To avoid information loss during disentangled representation learning, we regulate the disentangled embeddings to be able to reconstruct the original item visual representation. Thus, we feed the concatenation of the meaningful disentangled attribute-level embeddings of the item I_i and the residual one into the deconvolutional neural network [75]. It is formulated as,

$$\hat{\mathbf{V}}_i = \mathcal{D}\left(\left[q_{t_i^*}^1\mathbf{e}_i^1 \| q_{t_i^*}^2\mathbf{e}_i^2 \| \cdots, q_{t_i^*}^M\mathbf{e}_i^M \| \mathbf{e}_i^{M+1}\right]\right), \tag{4.7}$$

where the binary masks $\mathbf{q}_{t_i^*}^m$ are used to select the meaningful attribute embeddings of the item I_i, $[\cdot\|\cdot]$ refers to the concatenation operation, $\mathcal{D}(\cdot)$ denotes the deconvolutional neural network, and $\hat{\mathbf{V}}_i$ denotes the reconstructed visual representation of the i-th item. We hereafter utilize l_2 loss to regulate the distance between the reconstructed visual representation and the origin one via,

$$\mathcal{L}_{rec} = \sum_{i=1}^{K}\left\|\hat{\mathbf{V}}_i - \mathbf{V}_i\right\|_F^2. \tag{4.8}$$

Combining the losses of both the orthogonal residual embedding and the visual representation reconstruction constraints, we reach the final loss for regularizing the disentangled

completeness as follows,

$$\mathcal{L}_{dc} = \mathcal{L}_{or} + \mathcal{L}_{rec}. \tag{4.9}$$

4.3.2.3 Hierarchical Outfit Compatibility Modeling

Inspired by previous studies [18, 19], we leverage GCNs to model outfit compatibility. Beyond existing work, we design a novel hierarchical graph convolutional network, which can model the complex compatibility relations among items in an outfit from both attribute and item levels. In particular, the attribute-level compatibility modeling aims to investigate the fine-grained compatibility among fashion items, while the item-level model summarizes the coarse-grained outfit compatibility from the item level.

Attribute-level Compatibility Modeling. Regarding the attribute-level compatibility modeling, given an outfit, we first construct $M+1$ parallel compatibility modeling graphs $\mathcal{G}_a^m = (\mathcal{N}_a^m, \mathcal{E}_a^m), (m = 1, 2, \cdots, M+1)$, with each devised to model the outfit compatibility from an attribute aspect[1]. In particular, $\mathcal{N}_a^m$ and $\mathcal{E}_a^m$ refer to the set of nodes and edges, respectively, of the graph $\mathcal{G}_a^m$. In $\mathcal{G}_a^m$ graph, each node refers to a composing item of the outfit that has the corresponding attribute, *i.e.*, a_m. Notably, as previously mentioned, not every attribute can be applied to all the items, *e.g.*, the attribute *sleeve length* cannot be used to characterize a pair of *trousers*. Therefore, for different attributes, different numbers of items are applicable for the attribute-level compatibility modeling. In other words, graphs corresponding to different attributes may have different numbers of nodes. Therefore, for the ease of presentation, we still deploy K item nodes for all these graphs, *i.e.*, $\mathcal{N}_a^m = \{\hat{n}_i^m\}_{i=1}^K$, where $\hat{n}_i^m$ is the i-th node in the $\mathcal{G}_a^m$ graph. However, some nodes in these graphs are defined as the virtual isolated nodes and are inactive during the attribute-level compatibility propagation.

During the learning process, each node $\hat{n}_i^m$ is associated with a hidden state vector $\mathbf{h}_i^m$, which is updated to fulfill the compatibility information propagation over the graph. We initialize the hidden vector of node $\hat{n}_i^m$ by,

$$\mathbf{h}_i^m = \begin{cases} q_{t_i^*}^m \mathbf{e}_i^m, & m \in \{1, 2, \cdots, M\}, \\ \mathbf{e}_i^{M+1}, & m = M+1. \end{cases} \tag{4.10}$$

Therefore, if the m-th attribute can be applied to the item of the i-th node, we initialize the node with the item's corresponding attribute feature. Otherwise, the node is initialized with an all-zero vector, making it an isolated node in the graph, and it will not join the subsequent compatibility information propagation. Regarding the edge construction

[1] As previously stated, the residual attribute is also incorporated as a special implicit attribute.

for each graph, we introduce an edge between each pair of nonisolated nodes, *i.e.*, each pair of meaningful items in the corresponding attribute-level compatibility modeling.

To simplify the notation, considering that the parallel attribute-level compatibility modeling for different attributes follow the same learning process, we temporally remove all the superscripts m from the above notations and present the general attribute-level compatibility modeling scheme as an example. Inspired by graph attention networks (GAT) [76], we employ the attention mechanism to make each node adaptively absorb compatibility information from the neighbors. Formally, we have

$$\alpha_{ij} = \frac{\exp\left(\mathbf{W}_a \left[\mathbf{h}_i \| \mathbf{h}_j\right]\right)}{\sum_{n_k \in \mathcal{N}_i} \exp\left(\mathbf{W}_a \left[\mathbf{h}_i \| \mathbf{h}_k\right]\right)}, \tag{4.11}$$

where α_{ij} indicates the importance of node n_j's hidden state to node n_i, $\mathbf{W}_a$ is a weight matrix to perform the linear transformation, $[\cdot\|\cdot]$ refers to the concatenation operation, and $\mathcal{N}_i$ denotes the neighborhood of node n_i. Once the attention weights α_{ij} are obtained, they are then used to propagate information from the neighbors of node n_i to the node by,

$$\mathbf{h}_i^{'} = \omega\left\{\mathbf{W}_u\left[\sum_{n_j \in \mathcal{N}_i} \alpha_{ij}\left(\mathbf{h}_i \odot \mathbf{h}_j\right)\right] + \mathbf{b}_u\right\}, \tag{4.12}$$

where $\odot$ denotes the elementwise multiplication, $\mathbf{W}_u$ and $\mathbf{b}_u$ are the parameters of the fully-connected layer, and ω refers to the nonlinear activation function LeakyReLU. The elementwise multiplication $\mathbf{h}_i \odot \mathbf{h}_j$ indicates the compatibility information between the items I_i and I_j. More generally, instead of propagating the features of node n_i's neighbors, we propagate the compatibility information between node n_i and its neighbors, which has proven to be effective in addressing the outfit compatibility modeling task [64].

Based upon the above inference and computation, the updated hidden representation of node n_i is written as,

$$\tilde{\mathbf{h}}_i = \omega\left(\mathbf{W}_o \mathbf{h}_i + \mathbf{b}_o\right) + \mathbf{h}_i^{'}, \tag{4.13}$$

where $\mathbf{W}_o$ and $\mathbf{b}_o$ denote the weight matrix and bias to be learned, respectively. The symbol ω denotes the LeakyReLU function. We ultimately feed the updated hidden node embeddings into an MLP to derive the attribute-specific compatibility score of the given outfit via,

$$\begin{cases} c_i = \mathbf{W}_2\left[\psi\left(\mathbf{W}_1 \tilde{\mathbf{h}}_i + \mathbf{b}_1\right)\right] + \mathbf{b}_2, \\ c = \frac{1}{K}\sum_{i=1}^{K} c_i, \end{cases} \tag{4.14}$$

where $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{b}_1$, and $\mathbf{b}_2$ are the parameters of the MLPs, the symbol ψ denotes the ReLU active function, and c is the compatibility score. Following the above general scheme, we can obtain all the attribute-level compatibility scores, denoted as $\mathbf{c}_a = \left[c^1, c^2, \cdots, c^M, c^{M+1}\right]$, as well as the updated hidden attribute-level embeddings of each node/item, *i.e.*, $\tilde{\mathbf{h}}_i = [\tilde{\mathbf{h}}_i^1, \tilde{\mathbf{h}}_i^2, \cdots, \tilde{\mathbf{h}}_i^M, \tilde{\mathbf{h}}_i^{M+1}]$.

Item-level Compatibility Modeling. Similar to attribute-level compatibility modeling, we also construct a compatibility modeling graph $\mathcal{G}_o = (\mathcal{N}_o, \mathcal{E}_o)$ at the overview item level, where $\mathcal{N}_o$ and $\mathcal{E}_o$ refer to the node set and the edge set, respectively. The difference is that we initialize the hidden vector of the i-th node in the graph $\mathcal{G}_o$ from two aspects: the item's original visual feature $\mathbf{v}_i$, and the updated attribute-level item embedding $\tilde{\mathbf{h}}_i^m$'s from the attribute-level compatibility modeling scheme. In this way, a more comprehensive overview representation of the item is derived. Specifically, for the i-th node in the graph $\mathcal{G}_o$, we initialize its hidden vector as follows,

$$\mathbf{g}_i = \left[\mathbf{v}_i \| \mathbf{W}_h \left(\left[\tilde{\mathbf{h}}_i^1 \| \tilde{\mathbf{h}}_i^2 \| \cdots \| \tilde{\mathbf{h}}_i^{M+1}\right]\right)\right], \tag{4.15}$$

where $[\cdot\|\cdot]$ denotes the concatenation operation, and $\mathbf{W}_h \in \mathbb{R}^{D_v \times D_e(M+1)}$ is the to-be-learned weight matrix, which projects the attribute-level embeddings to the same space of the entire visual embeddings. Following the same information propagation scheme as the attribute-level compatibility modeling, we can obtain the item-level compatibility score c_o.

Considering both the attribute- and item-level compatibility modeling results, we feed the concatenation of the attribute- and item-level compatibility scores, *i.e.*, $\mathbf{c} = [c_a \| \mathbf{c}_o]$, into the MLP to obtain the final compatibility probability score as follows,

$$s = \sigma\left\{\mathbf{W}_4\left[\psi\left(\mathbf{W}_3\mathbf{c} + \mathbf{b}_3\right)\right] + \mathbf{b}_4\right\}, \tag{4.16}$$

where $\mathbf{W}_3$, $\mathbf{W}_4$, $\mathbf{b}_3$, and $\mathbf{b}_4$ are the parameters of the MLP, the symbol ψ denotes the ReLU active function, and σ refers to the sigmoid active function. We finally adopt the cross-entropy loss to optimize our proposed PS-OCM, and reach the following formulation,

$$\mathcal{L}_{hc} = -ylog(s) - (1-y)log(1-s), \tag{4.17}$$

where y is the ground-truth compatibility label for the outfit O. Accordingly, the total loss for our PS-OCM can be written as follows,

$$L = \mathcal{L}_{hc} + \lambda\mathcal{L}_{ps} + \mu\mathcal{L}_{dc}, \tag{4.18}$$

where λ and μ are tradeoff hyperparameters.

Interpretability. The semantic attributes have explicit meaning and can be used naturally to interpret the compatibility evaluation result. In particular, we can identify the prominent attributes that contribute to the final compatibility evaluation most, according to the absolute values of these attribute-specific compatibility scores, *i.e.*, c^ms.

4.4 Experiment

In this section, we first introduce the dataset and experimental settings, and then detail the extensive experiments that we conducted on a real-world dataset by answering the following research questions:

- **RQ1**: Does the proposed PS-OCM outperform the state-of-the-art methods?
- **RQ2**: How does each component affect PS-OCM?
- **RQ3**: What is the intuitive evaluation result of PS-OCM?

4.4.1 Experimental Settings

In this part, we presenst the dataset, evaluation metrics, and implementation details.

4.4.1.1 Dataset and Evaluation Metrics

To justify our model, we resorted to the public dataset IQON3000 [12], due to the fact that each item in IQON3000 has not only the visual image but also several semantic attributes, such as color and category. In particular, IQON3000 consists of $308,747$ outfits, composed by $672,335$ items. In total, there are 11 attributes provided by this dataset. Table 4.2 shows the possible value examples and the corresponding number for each attribute. To ensure the dataset quality, we empirically sampled $20,000$ compatible outfits, each of which consisted of at least 2 but no more than 10 items. Since the dataset only provided the compatible outfits, incompatible outfits were composed for training. Specifically, for each compatible outfit, we replaced each of its composing items with a randomly sampled item from the same category to construct the incompatible outfit. In this manner, we end up with a set of $40,000$ compatible/incompatible outfits. We then divided it into the training set, validation set, and test set according to the ratio of $8:1:1$.

TABLE 4.2: Attributes and the possible value.

Attribute	Possible Value	Total Number
Color	Grey, Black, Green, $\cdots$	12
Price	Low, Middle, High.	3
Brand	ABISTE, FURLA, BEIGE, $\cdots$	5,180
Category	Trousers, Belt, Handbag, $\cdots$	61
Variety	Coat, Bag, Cosmetics, $\cdots$	20
Material	Fur, Leather, Denim, $\cdots$	37
Pattern	Stripe, Embroidery, Animal, $\cdots$	15
Design	Turtleneck, Frill, Ribbons, $\cdots$	23
Heel	Chunky, Pin, High, $\cdots$	6
Dress Length	Short, Middle, Long.	3
Sleeve Length	Sleeveless, Long, Short, $\cdots$	4

Similar to previous studies [3, 18, 19, 21, 64], we justified our proposed PS-OCM scheme with two specific tasks: outfit compatibility estimation and fill-in-the-blank. For these two tasks, we also used the AUC and the accuracy (ACC) as the evaluation metrics, respectively.

4.4.1.2 Implementation Details

For the image encoder, we employed the ResNet18 [48] pretrained on ImageNet [53] as the backbone, and modified the last layer to make the output feature dimension 256. Pertaining to the MLPs that obtain the disentangled attribute-level embeddings, we set the output dimension to 64. We selected Adam [54] as the training optimizer, with a fixed learning rate of 0.0001. We empirically set the batch size as 32, and both tradeoff hyperparameters, *i.e.*, λ and μ in Eqn.(4.18), as 1. All the experiments were implemented by PyTorch over a server equipped with 4 GeForce RTX 2080 Ti GPUs, and the random seeds for model initialization were fixed for the reproducibility.

4.4.2 Model Comparison

To validate the effectiveness of our proposed scheme, we chose the following baselines for comparison, including the pairwise, sequencewise, and graphwise models.

- **Type-aware** [21] devises the type-specific embedding spaces according to the item types, to facilitate the outfit compatibility measurement. The visual-semantic loss is utilized to incorporate the visual and textual information.

TABLE 4.3: Performance comparison between our proposed PS-OCM and other baseline methods on two tasks over the IQON3000 dataset. Notably, the baseline methods were re-trained by the released codes. The best results are in bold, while the second best results are underlined.

Method	Compatibility AUC	FITB Accuracy
Type-aware [21]	0.6688	0.3901
SCE-NET [52]	0.6792	0.3783
Bi-LSTM [3]	0.7739	0.3813
NGNN [18]	0.7591	0.4002
HFGN [5]	0.8243	0.4511
MM-OCM [64]	0.8444	0.4661
OCM-CF [77]	0.8402	0.4825
MOCM-MGL [78]	<u>0.8929</u>	<u>0.5160</u>
PS-OCM	0.9009	0.5412
PS-OCM-ResNet50	0.9029	0.5746
PS-OCM-SwinTransformer	**0.9295**	**0.5853**

- **SCE-NET** [52] embeds the item visual features into multiple semantic subspaces by multiple condition masks, and uses the multimodal features to derive the importance weights for different subspace features to obtain the final item representations.

- **Bi-LSTM** [3] takes the items in an outfit as a sequence ordered by the item categories, and exploits the latent item interaction by a bi-directional LSTM. Notably, the textual information is also adopted to regularize the outfit compatibility modeling by the visual-semantic consistency loss.

- **NGNN** [18] is the first research attempt to employ GNN to model the outfit compatibility, where each outfit is represented as a subgraph, and an attention mechanism is utilized to calculate the outfit compatibility score. For the multimodal features, NGNN designs two graph channels, and the final compatibility score is derived in a late fusion manner.

- **HFGN** [5] develops a hierarchical fashion graph network to jointly fulfill the fashion compatibility modeling and personalized outfit recommendation, where a category-oriented fashion graph is built for each outfit. It only uses the visual features.

- **MM-OCM** [64] explicitly models the consistent and complimentary relations between the visual and textual modalities of fashion items by the parallel and orthogonal regularizations. Moreover, MM-OCM jointly unifies the text-oriented and vision-oriented outfit compatibility modeling with the mutual learning strategy.

- **OCM-CF** [77] directly learns the context-aware global outfit representation by GCNs and the multihead attention mechanism, and employs multiple network

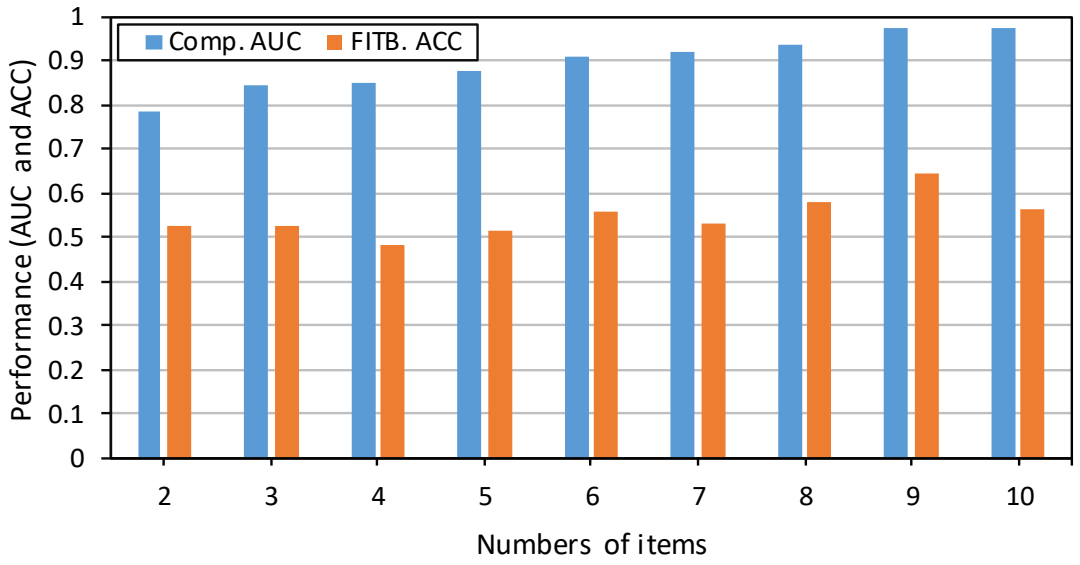


FIGURE 4.4: Performance of our PS-OCM on two tasks for outfits with different numbers of items.

TABLE 4.4: Ablation study of our proposed PS-OCM on IQON3000 dataset. The best results are in bold.

Method	Compat. AUC	FITB ACC
w/o Partial_Supervision	0.8433	0.4866
w/o Orthogonal	0.8938	0.5293
w/o Deconvolution	0.8909	0.5293
w/o Hierarchical_Graph	0.8197	0.4459
Attribute-level_Only	0.8848	0.5337
Item-level_Only	0.8720	0.5292
PS-OCM	**0.9009**	**0.5412**

branches to explore the hidden complementary factors that affect the outfit compatibility.

- **MOCM-MGL** [78] proposes a multi-modal outfit compatibility modeling with modality-oriented graph learning. It takes both visual, textual, and category modalities as input and jointly propagates the intra-modal and inter-modal compatibilities among fashion items in the outfit[2].

- **PS-OCM-Resnet50/PS-OCM-SwinTransformer**: To study the effect of utilizing different backbones to extract the image features, we replaced the Resnet18 backbone to Resnet50 and SwinTransformer [79], respectively.

Table 4.3 shows the performance of different methods on the outfit compatibility estimation task and fill-in-the-blank task. Notably, the baseline methods were retrained by

[2]For fair comparison, the attribute information is utilized as the pure text in MOCM-MGL.

the released corresponding codes over the IQON3000 dataset. From this table, we make the following observations:

1. The pairwise methods, *i.e.*, Type-aware and SCE-NET, achieved the worst performance on both two tasks. This may be due to the fact that the pairwise methods justify the local compatibility between two items, lacking the global view of the whole outfit.

2. The sequencewise method, *i.e.*, Bi-LSTM, performs better than the pairwise methods, but worse than the graphwise methods, *i.e.*, HFGN and MM-OCM. On the one hand, this confirms the advantage of treating the outfit as a unified sequence rather than the item pairs. On the other hand, this implies that treating the outfit as an ordered sequence of fashion items is still suboptimal. This may be attributed to the sequencewise method being able to suffer from the cumulative error propagation problem since it computes the outfit compatibility score by continually predicting the next item with the previous items.

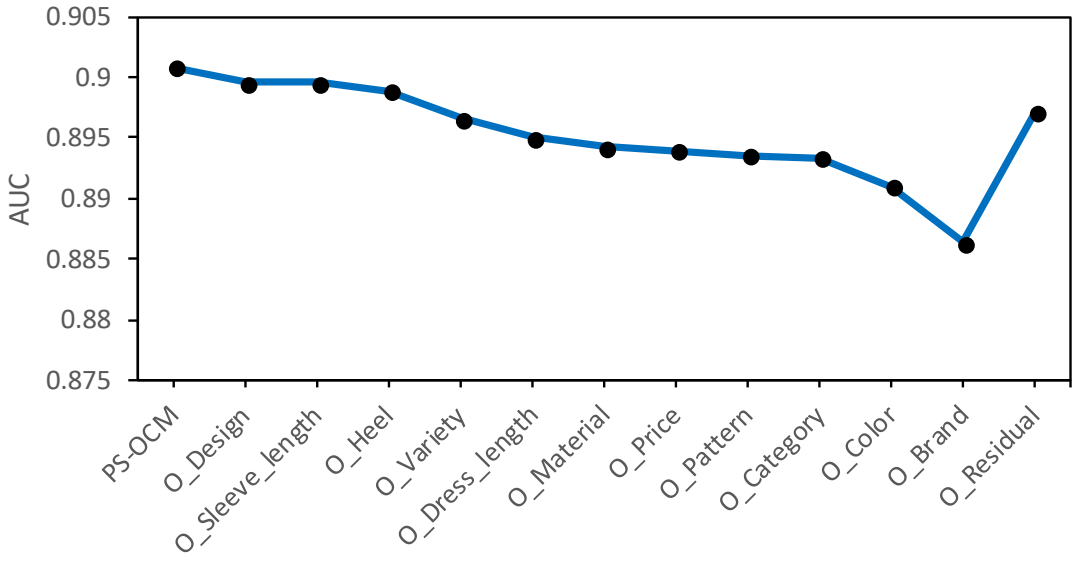
3. Our proposed PS-OCM consistently surpasses all the baseline methods on both tasks. This confirms the advantage of our scheme that utilizes the irregular attribute labels to provide partial supervision to strengthen the item representation learning and employs the hierarchical graph convolutional network to integrate the attribute-level and item-level outfit compatibility learning.

4. PS-OCM-SwinTransformer performs better than both PS-OCM and PS-OCM-ResNet50, indicating the superiority of swin transformer in image feature extraction and hence boost the final performance.

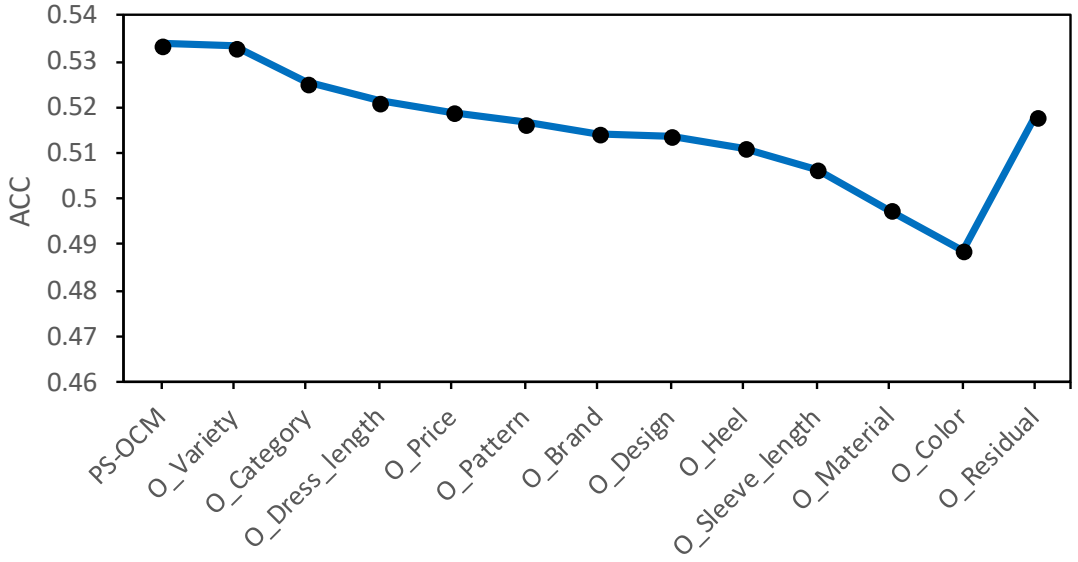
To gain deep insights into our proposed PS-OCM, we further checked the performance of our PS-OCM for outfits with different numbers of composing items on the two tasks. In particular, we reported the performance of our model for outfits with the number of composing items ranging from 2 to 10. As can be seen in Figure 4.4, our PS-OCM is generally not sensitive to the composing numbers, which indicates that our model PS-OCM can handle the compatibility modeling for outfits with various numbers of items.

4.4.3 Ablation Study

To verify the importance of each component in our model, we conducted ablation experiments on the following derivatives.



(a) Comp.AUC



(b) FITB ACC

FIGURE 4.5: Comparison of the effect of removing a single attribute from our PS-OCM on two tasks.

- **w/o Partial_Supervision**: To explore the effect of the partially supervised attribute embedding learning component, we removed the partial supervision loss by setting $\lambda = 0$ in Eqn.(4.18).

- **w/o Orthogonal**: To study the effect of the orthogonal regularization during the visual attributes disentanglement, we removed the orthogonal regularization $\mathcal{L}_{or}$

in Eqn.(4.9).

- **w/o Reconstruction**: To validate the necessity of visual representation reconstruction learning, we removed the visual representation reconstruction constraint $\mathcal{L}_{rec}$ in Eqn.(4.9).
- **w/o Hierarchical_Graph**: To validate the function of the hierarchical graph compatibility modeling component in our model, we removed this part by directly concatenating the attribute-level embeddings of each outfit to obtain the overall outfit representation and passing it to an MLP to obtain the outfit's compatibility score.
- **Attribute-level_Only**: To verify the importance of introducing the coarse-grained item-level information, this derivative only utilizes the fine-grained attribute-level compatibility modeling part in the hierarchical graph compatibility modeling component.
- **Item-level_Only**: Similarly, to justify the necessity of introducing the fine-grained attribute-level compatibility modeling, we removed it from the hierarchical outfit compatibility modeling network.

Based on the ablation experiment illustrated in Table 4.4, we found that our model consistently outperforms all the above derivatives on both tasks, which demonstrates the effectiveness of each component in our proposed PS-OCM. Specifically, we make the following detailed observations.

1. The performance of w/o Partial_Supervision significantly drops, as compared to PS-OCM, indicating that the partially supervised attribute embedding learning component is indeed helpful to strengthen the visual representation learning performance.
2. Both w/o Orthogonal and w/o Reconstruction are inferior to PS-OCM, which suggests that it is essential to consider the orthogonal regularization and visual feature reconstruction to prevent the visual information loss during the visual feature disentanglement and guarantee the completeness of the disentanglement.
3. w/o Hierarchical_Graph delivers the worst performance, reflecting the effectiveness of our proposed hierarchical outfit compatibility modeling component. Moreover, both Attribute-level_Only and Item-level_Only perform better than w/o Hierarchical_Graph, which confirms the necessity of jointly incorporating the attribute-level and item-level compatibility modeling modules. This also reflects that the fine-grained attribute-level features and the overview item-level features complement each other to a certain level toward the outfit compatibility modeling.

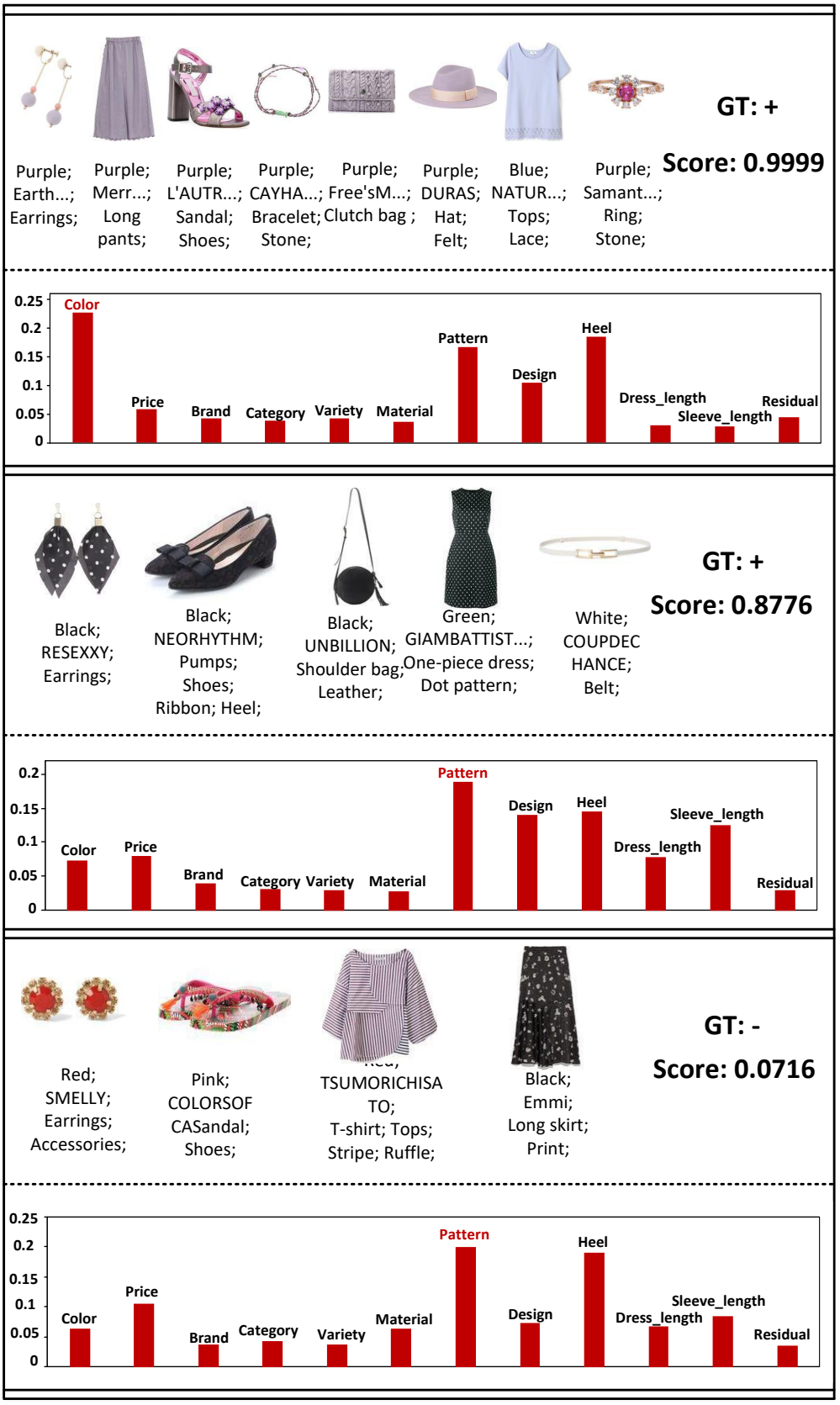


FIGURE 4.6: Case study of PS-OCM on the outfit compatibility estimation task.

As the partial supervised attribute-level embedding learning contributes the key novelty of our work, we further studied the effect of removing each attribute embedding from the training phase of our PS-OCM. As aforementioned, we had 12 attributes, including 11 concrete attributes in the original dataset and one "residual" attribute we newly defined. Accordingly, we omitted each of the 12 attributes from our model, and hence obtained 12 derivatives of our model, with each named as O_{each_attribute}. Figure 4.5 shows the performance of our PS-OCM and its derivatives on the two tasks. As can be seen, removing any concrete attribute (e.g., the design or color) hurts our model's performance, which verifies that each concrete attribute contributes to the outfit compatibility modeling. In particular, we noticed that the color attribute greatly affects our model's performance on both tasks, which is reasonable, as the color attribute is the most straightforward influential factor on the outfit compatibility modeling. Meanwhile, we found that O_residual underperforms our PS_OCM. This reflects the importance of the residual attribute, and indicates its capability of compensating the information loss during the attribute representation disentanglement.

4.4.4 Case Study

To obtain an intuitive understanding of our model, we also conducted a case study of our method in the two tasks: outfit compatibility estimation and fill-in-the-blank.

Figure 4.6 shows several testing examples of our model on the outfit compatibility estimation task, where the importance distribution of attributes, i.e., the normalization of the absolute values of the attribute-level compatibility scores, is also given to intuitively demonstrate the interpretability of our model. As can be seen from the first example, our model yields the correct compatibility estimation, and captures the color attribute as the most important influential factor. This is reasonable as the color presented by the outfit is harmonious. In second example, our model also gives a high compatible probability score, and identifies that the pattern attribute is the most important factor. As we can see, the earrings and the dress in the given outfit do consistently present the dotted pattern. Accordingly, the result makes sense. As for the last incompatible example, our PS-OCM gives a low compatibility score, and the pattern attribute is also captured as the most important factor contributing to the incompatible estimation result. From this example we found that the striped pattern of the T-shirt, spotted pattern of the skirt, and floral pattern of the sandal indeed form no compatible look.

Figure 4.7 shows several testing results of our PS-OCM, compared with its several derivatives and MOCM-MGL which gains the best performance among baselines. In particular, the first row of each example refers to the questions of the fill-in-the-blank task, and the

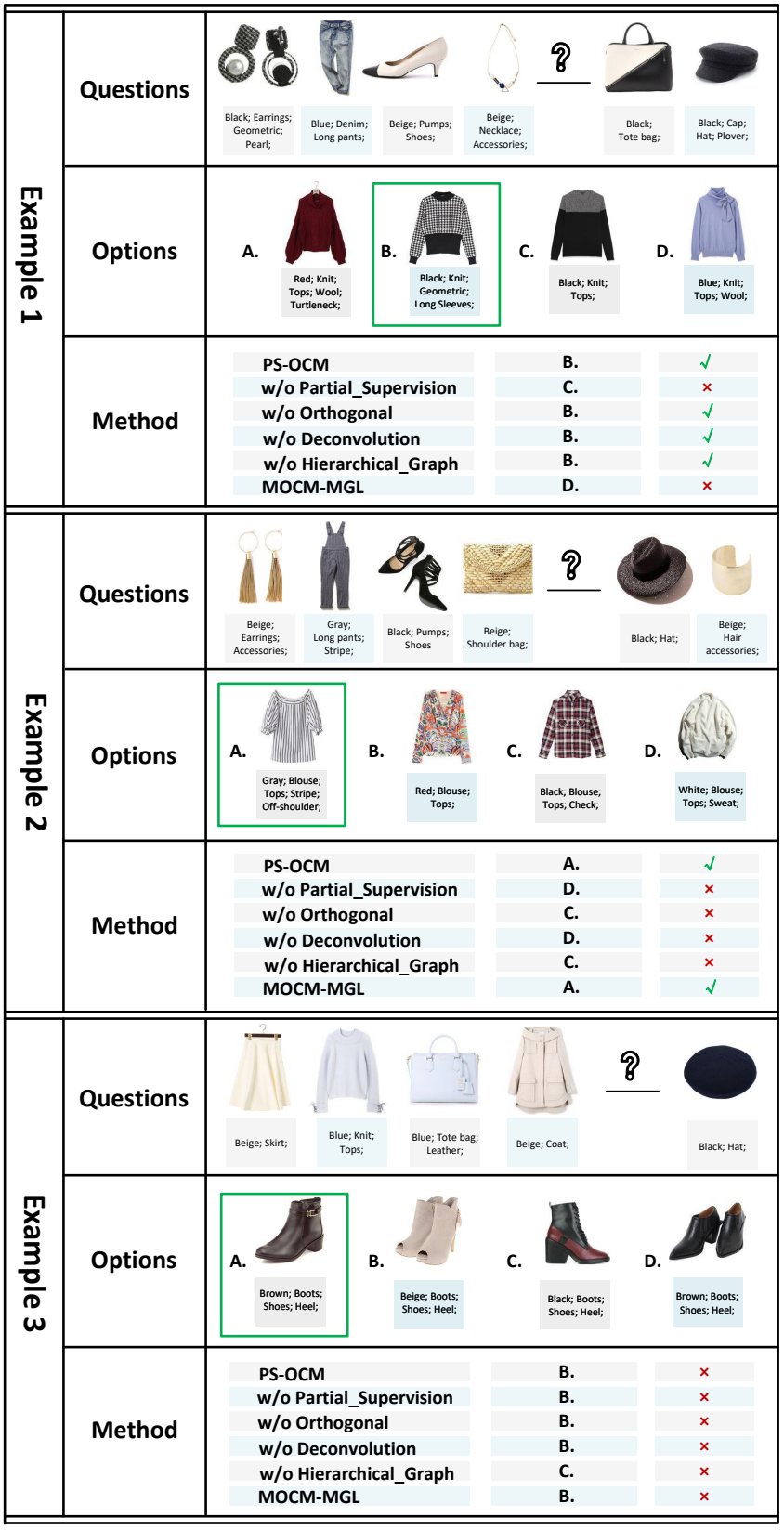


FIGURE 4.7: Case study of PS-OCM and its several derivatives as well as the best baseline MOCM-MGL on the FITB task.

second row lists the corresponding four options, where the ground truth item is denoted with a green box. The last row shows the choice yielded by each method and indicates whether the choice is true or not by a green tick and red cross. As can be seen from the first example in Figure 4.7, only w/o Partial_Supervision and MOCM-MGL fail to give the correct choice, i.e, the second item that has the same geometric pattern with the given earrings. This suggests the effectiveness of incorporating irregular attribute information as the partial supervision. In the second example, all the derivatives chose the false item, which further demonstrates the importance of each designed component in PS-OCM. Regarding the last example, although all our methods fail to give the correct answer, we noticed that their chosen items also go well with the given question items, especially from the color perspective. This also implies the effectiveness of our model in practice.

4.5 Summary

In this chapter, for outfit compatibility modeling, we presented a novel partially supervised compatibility modeling, named PS-OCM, which consists of three key components: 1) partially supervised attribute embedding learning, 2) disentangled completeness regularization, and 3) hierarchical outfit compatibility modeling. In particular, we first presented a partially supervised disentangled learning method to disentangle the visual representation of each item into several attribute-level embeddings, where the irregular attribute labels of fashion items are used as the supervision to strengthen the visual representation learning of items. In addition, we devised the disentangled completeness regularization, including the orthogonal residual embedding and visual representation reconstruction, to prevent information loss during disentanglement. Finally, we designed a hierarchical graph convolutional network that jointly performs attribute- and item-level compatibility modeling. Extensive experiments were conducted on a real-world dataset with two popular tasks: outfit compatibility prediction and fill-in-the-blank. The encouraging experiment results validate the superiority of our proposed model and the importance of each component. In addition, we found that our PS-OCM is not sensitive to the number of items in the outfit, and removing each attribute, including the introduced residual attribute, from the embedding disentanglement will hurt the model's performance. This shows that each attribute affects the outfit compatibility modeling to some extent.

Chapter 5

Heterogeneous Graph Learning for POCM

5.1 Introduction

One common limitation of our previous works presented in Chapters 3 and 4 is that they all evaluate the outfit compatibility from the general standard. In fact, there may be some subjective factors influencing the outfit compatibility evaluation, namely, for the same garment, different users may have different evaluations. In other words, different people usually have different preferences to make their personal ideal outfits, which may be caused by their diverse growing circumstances or educational backgrounds. For example, as shown in Figure 5.1, given the same pink shirt, user A prefers to match it with a homochromatic skirt and high-heeled shoes; whereas user B likes to coordinate it with casual jeans and white sneakers. Therefore, personalized outfit compatibility modeling, called POCM, considering users' preferences when measuring the compatibility among fashion items, merit our special attention. A few pioneer researchers have noticed this phenomenon and dedicated their efforts to POCM [5, 12, 24]. These efforts study the user and item entities, as well as their relations. They, however, overlook another important entity type in POCM, namely, attributes. Conveying rich semantics, attributes play a pivotal role in characterizing items and delivering users' preferences to items. For instance, we may express "I would like to buy a black coat with a fur collar", whereby the key information is conveyed via the semantic attributes. To alleviate such a problem, we incorporate attributes associated with fashion items and work toward fully exploring all the related entities (*i.e.,* users, items, and attributes) and their various relations (*i.e.,* user-item interactions, item-item matching relations, and item-attribute association relations) to promote the POCM performance. Without loss of generality,

FIGURE 5.1: Examples of users' outfit compositions shared on the online fashion-oriented website.

we specifically study the research problem of "which bottom (top) is compatible to the given top (bottom) for a specific user".

Addressing the aforementioned research is, however, nontrivial due to the following challenges.

- **C1**: POCM involves three kinds of entities with heterogeneous contents: users, items, and attributes. Specifically, users are pure IDs, items are composed of images and textual descriptions, while attributes are in the form of textual phrases. Thereby, how to effectively organize these heterogeneous data seamlessly poses the first research challenge.
- **C2**: Different from the item and attribute entities, we do not have the specific content information of user entities. The conventional user embedding paradigm usually assigns a fixed one-hot embedding or learnable embedding to represent each user. This is actually not applicable to new users arriving during the testing phase, even for the case in which we have the historical interactions of these new users. Accordingly, how to derive the user embedding is another challenge.
- **C3**: In fact, apart from the direct relations, like the user-item interaction relation, item-item matching relation, and item-attribute association relation, there are also high-order relations among the three types of entities. For example, similar bottoms matching the same top may share some common attributes. Another example

is that users with similar tastes tend to like items with similar attributes. Therefore, how to explore the high-order relations among these entities to strengthen the model's performance constitutes the third challenge.

To address the challenge **C1**, we organize the users, items, and attributes in the context of POCM into a unified heterogeneous graph. Specifically, these three kinds of entities are nodes of this graph. The nodes are linked by three kinds of edges, which are user-item interactions, item-item matching relations, and item-attribute association relations. It is worth mentioning that in this graph there is no direct edge linking the user and attribute entities. We then devise a novel metapath-guided personalized compatibility modeling scheme to address **C2** and **C3**, named as MG-POCM, as shown in Figure 5.2. This scheme consists of three key components: heterogeneous graph node embedding, metapath-guided heterogeneous graph learning, and personalized outfit compatibility modeling. The first component works on embedding each type of entity in our heterogeneous graph. To represent users, we devise a multimodal content-oriented user embedding module, which derives the user embedding based on the multimodal contents of his/her interacted items, a straightforward cue indicating the user's preference. As to the second component, we first define multiple user-oriented and item-oriented metapaths (*e.g.,* User $\rightarrow$ Item $\rightarrow$ User and Item $\rightarrow$ Attribute $\rightarrow$ Item) to capture the high-order relations among entities, which naturally resolves the third challenge **C3**. Thereafter, we conduct the multiple metapath-guided heterogeneous graph learning to obtain the multiple semantic-enhanced user/item embedding of each user/item, whereby each metapath corresponds to a specific semantic. A transformer [80] is used to adaptively fuse the semantic-enhanced user/item embeddings under different metapaths for each user/item. Ultimately, in the last component, in addition to the typical cross-entropy loss, we also introduce the contrastive regularization to enhance embedding learning.

The research work in this chapter has been published in ACM SIGIR 2022.

Weili Guan, Fangkai Jiao, Xuemeng Song, Haokun Wen, Chung-Hsing Yeh, Xiaojun Chang. "Personalized Fashion Compatibility Modeling via Metapath-guided Heterogeneous Graph Learning." In Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2022.

5.2 Related Work

This work is related to heterogeneous graph learning.

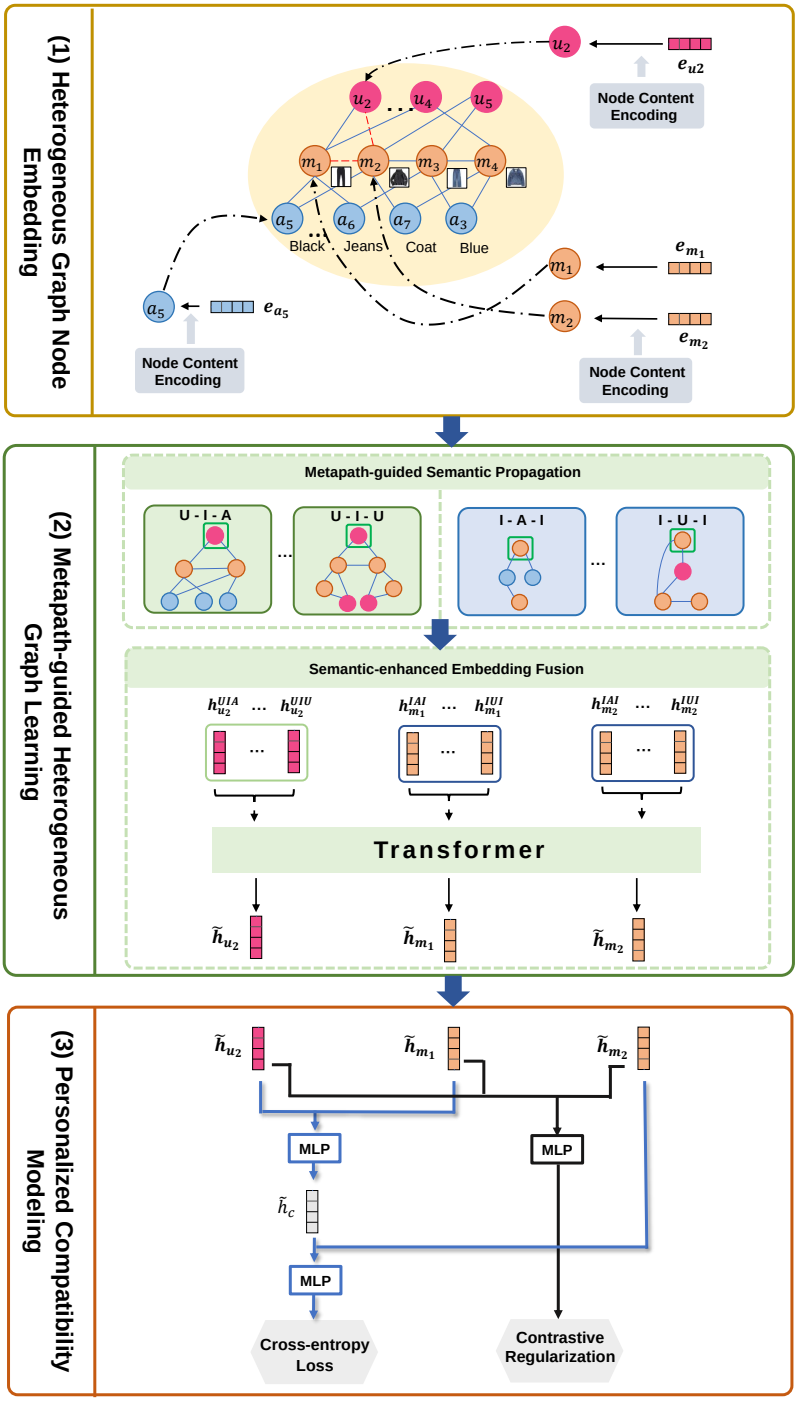


FIGURE 5.2: Illustration of the proposed MG-POCM scheme. It consists of three key components: (1) heterogeneous graph node embedding, (2) metapath-guided heterogeneous graph learning, and (3) personalized outfit compatibility modeling.

Heterogeneous Graph Learning. Due to the ubiquity of heterogeneous graphs in the real-world setting, containing multiple types of nodes and relations among these nodes [81, 82], increasing research efforts have been dedicated to heterogeneous graph learning. In a sense, existing methods focus on the heterogeneous graph embedding via learning a powerful low-dimensional vector representation for each node to benefit the

potential downstream applications, such as node classification [83, 84] and personalized recommendation [85, 86]. To accomplish this task, previous methods mostly rely on the metapath [87], *i.e.*, a sequence of node and edge types, delivering certain semantic information of the graph. For example, Dong *et al.* [88] developed metapath-based random walks to construct the heterogeneous neighborhood of a node and then utilized a skip-gram model [89] to perform node embeddings. One key limitation of this method is that it only utilizes a single metapath, which may be insufficient to cover all useful information. To address this issue, Shi *et al.* [90] designed a novel strategy to generate the meaningful node sequences and utilized fusion functions to learn node representation. In addition, Zhang *et al.* [91] introduced a heterogeneous graph neural network model, named HetGNN, to jointly explore the heterogeneous structures and contents of each node. To obtain superior node representation, several researchers [92–94] have utilized the attention mechanism to select the most useful metapath. For example, Wang *et al.* [93] proposed a heterogeneous graph attention network, which incorporates both node- and semantic-level attention to learn the importance of nodes and metapaths in the node embedding. Subsequently, Zhang *et al.* [94] proposed an attentive heterogeneous graph neural network for heterogeneous graph embedding, where the node-level attention is considered, and a semantic-level neural network is utilized rather than the semantic-level attention for capturing the feature interaction among node embeddings under different metapaths. Differently, Xing *et al.* [95] regarded each metapath as a specific view, and borrowed the idea of multiview learning to comprehensively encode the node representations of different views into a latent representation. To address the practical issue of missing attributes, Jin *et al.* [96] proposed a general framework for heterogeneous graph neural network via attribute completion, comprising two key components: prelearning topological embedding and attribute completion with attention mechanism.

Inspired by the great success of these methods on heterogeneous graph learning, we seamlessly organize the various entities and relations in the context of POCM into a unified heterogeneous graph. It is worth emphasizing that we design task-specific metapaths and creatively incorporate the transformer to fuse the semantic-enhanced user/item embeddings.

5.3 Methodology

In this section, we first formulate the research problem and then detail the three components of our proposed MG-POCM scheme.

TABLE 5.1: Summary of the Main Notations.

Notation	Explanation
$\mathcal{U}$	The set of users.
$\mathcal{M}$	The set of items.
N_u	The number of users in $\mathcal{U}$.
N_m	The number of items in $\mathcal{M}$.
$\mathcal{M}^t$	The sets of tops.
$\mathcal{M}^b$	The sets of bottoms.
m_i	The i-th item.
$\mathbf{e}_{m_i}$	The embedding of the item m_i.
v_i	The image of item m_i.
t_i	The textual description of item m_i.
$\mathcal{A}_i$	The set of attributes of item m_i.
N_a	The number of attributes in $\mathcal{A}_i$.
$\mathcal{A}$	The entire attribute set in the form of semantic phrases.
$\mathcal{X}^u$	The set of top-bottom pairs associated with user u.
$\mathcal{G}$	The heterogeneous graph.
$\mathcal{E}$	The set of entity nodes in $\mathcal{X}$, consisting of user, item, and attribute entities.
$\mathcal{R}$	The set of edges linking nodes to characterize relations among entities.
p_{ij}^k	The compatibility degree of a bottom (top) m_k to the given top (bottom) m_j for the user u_i.
$\mathbf{e}_{a_l}$	The embedding of the attribute entity a_l.
$\mathbf{e}_{u_i}$	The embedding of the user u_i.
$\mathbf{e}_{m_i}^{UIA}$	The semantic-enhanced embedding of the item entity u_i.
$\mathbf{h}_{u_i}^{UIA}$	The semantic-enhanced embedding of the user entity u_i.
$\mathbf{e}_{u_i}^{UIA}$	The semantic-enhanced embedding of the item entity u_i.
$\mathbf{h}_{u_i}^{UIA}$	The semantic-enhanced embedding of the user entity u_i.
$\tilde{\mathbf{h}}_{u_i}$	The final representation of the user entity u_i.
$\tilde{\mathbf{h}}_{m_i}$	The final representation of the item u_i.

5.3.1 Problem Formulation

Formally, we first clarify the notations. We use bold uppercase letters (*e.g.*, $\mathbf{W}$) and bold lowercase letters (*e.g.*, $\mathbf{b}$) to represent matrices and vectors, respectively. All vectors are in column forms. Additionally, we employ nonbold letters (*e.g.*, W and W) to denote scalars and Greek letters (*e.g.*, α) to represent regularization parameters. Table 5.1 summarizes the main notations.

In this work, we focus on fulfilling the task of POCM. Without loss of generality, we study the particular problem of "whether the given bottom (top) matches the given top (bottom) and together compose a favorable outfit for the given user". Assume that we have a set of N_u users $\mathcal{U} = \{u_1, u_2, \cdots, u_{N_u}\}$, and a set of N_m items $\mathcal{M} = \{m_1, m_2, \cdots, m_{N_m}\}$. For an arbitrary item $m_i, i = 1, 2, \cdots, N_m$, it is composed of an image v_i, a textual

description t_i, and a set of attributes $\mathcal{A}_i \subseteq \mathcal{A}$, where $\mathcal{A} = \bigcup_{i=1}^{N_m} \mathcal{A}_i = \{a_1, a_2, \cdots, a_{N_a}\}$ represents the entire attribute set in the form of semantic phrases, like *red color*, *wool material*, and *V-neck design*. The symbol N_a denotes the total number of attributes in our dataset. To simplify the formulation, in this work, we only consider the tops and bottoms. Therefore, the set of items can be rewritten as $\mathcal{M} = \mathcal{M}^t \cup \mathcal{M}^b$, where $\mathcal{M}^t$ and $\mathcal{M}^b$ refer to the sets of tops and bottoms, respectively. Each user u is historically associated with a set of top-bottom pairs $\mathcal{X}^u = \{(m^u_{t_1}, m^u_{b_1}), (m^u_{t_2}, m^u_{b_2}), \cdots, (m^u_{t_{M_u}}, m^u_{b_{M_u}})\}$, where $m^u_{t_*} \in \mathcal{M}^t$, $m^u_{b_*} \in \mathcal{M}^b$, and M_u denotes the total number of interacted top-bottom pairs by the user u. We resort to a heterogeneous graph to organize the complicated entities and relations within a unified structure. In particular, we denote the graph as $\mathcal{G} = (\mathcal{E}, \mathcal{R})$, where $\mathcal{E} = \mathcal{U} \cup \mathcal{M} \cup \mathcal{A}$ denotes the set of entity nodes, consisting of user entities, item entities, and attribute entities, while $\mathcal{R}$ denotes the set of edges linking nodes to characterize various relations among entities, *i.e.*, user-item historical interactions, item-attribute association relations, and item-item matching relations. Ultimately, we work in learning the following compatibility estimation function,

$$p^k_{ij} = \mathcal{F}(m_k \in \mathcal{M}^{b(t)} | u_i, m_j \in \mathcal{M}^{t(b)}), \tag{5.1}$$

where p^k_{ij} denotes the compatibility degree of a bottom (top) m_k to the given top (bottom) m_j for the user u_i.

5.3.2 Metapath-Guided Personalized Compatibility Modeling

As illustrated in Figure 5.2, MG-POCM consists of three components: 1) heterogeneous graph node embedding, 2) metapath-guided heterogeneous graph learning, and 3) personalized outfit compatibility modeling. In this subsection, we explain each of them.

5.3.2.1 Heterogeneous Graph Node Embedding.

This component aims to derive the initial node-level representations in the heterogeneous graph. The heterogeneous graph has three types of entities and the node contents differ remarkably. Therefore, we learn their embeddings separately as shown in Figure 5.3.

Item Entity Embedding. Each item entity is composed of an image and a textual description. The multimodal cues of each item mutually complement each other. For the arbitrary item m_i, regardless of its category (*i.e.*, top or bottom), we utilize the ResNet, which has shown compelling success in many computer vision tasks [48], to extract its visual feature. We adopt the pretrained BERT to obtain its textual feature[1]

[1]Before feeding a text into the BERT, the text is first tokenized into standard vocabularies.

due to its prominent performance in textual representation learning [97]. Specifically, we employ the averaged hidden states corresponding to the special token attached at the beginning of the input sequence, *i.e.*, [CLS], of the last two layers of BERT as the textual description representation. Finally, we concatenate the visual and textual features of each item to derive the final item embedding, and use a learnable fully-connected layer to project the item embedding into a lower dimensional space. Mathematically, we have

$$\begin{cases} \mathbf{e}_{v_i} = \text{ResNet}\left(v_i\right), \\ \mathbf{e}_{t_i} = \text{BERT}\left(t_i\right)_{[CLS]}, \\ \mathbf{e}_{m_i} = f_t\left(\left[\mathbf{e}_{v_i}, \mathbf{e}_{t_i}\right]\right), \end{cases} \tag{5.2}$$

where $\mathbf{e}_{v_i} \in \mathbb{R}^{D_v}$ and $\mathbf{e}_{t_i} \in \mathbb{R}^{D_t}$ refer to the visual and textual embedding of the item m_i, respectively. Accordingly, the symbols D_v and D_t are the dimensions of the visual and textual embeddings, respectively. ResNet and BERT denote the corresponding neural networks. $[,]$ refers to the concatenation operation, f_t denotes the learnable fully-connected layer, and $\mathbf{e}_{m_i} \in \mathbb{R}^D$ is the final embedding of the item m_i.

Attribute Entity Embedding. To fully utilize the semantic content of each attribute entity, we also resort to the pretrained BERT with a learnable fully-connected layer to derive its embedding instead of using the one-hot vector or treating it as the learnable parameter. Notably, for attribute embedding, we only adopt the representation of the special token [CLS] from the last layer of BERT due to its shorter length, as compared with the textual description. Formally, for each attribute entity a_l, we obtain its embedding as follows,

$$\mathbf{e}_{a_l} = f_a(\text{BERT}\left(a_l\right)_{[CLS]}), \tag{5.3}$$

where $\mathbf{e}_{a_l} \in \mathbb{R}^D$ denotes the initial embedding of the attribute entity a_l, and f_a denotes the fully-connected layer for embedding fine-tuning.

User Entity Embedding. Instead of using the one-hot embeddings, we resort to aggregating all the embeddings of the user's one-hop neighbor nodes (*i.e.,* all the items interacted by the user before) to derive the initial embedding of each user entity. The underlying philosophy is two-fold: 1) the items that are historically interacted by users signal users' preferences and tastes, and 2) the embedding of a cold-start user can also be derived as long as his/her interacted items appeared before. Specifically, we determine the user embedding below,

$$\mathbf{e}_{u_i} = \frac{1}{|\mathcal{N}^{u_i}|} \sum_{m_i \in \mathcal{N}^{u_i}} \mathbf{e}_{m_i}, \tag{5.4}$$

where $\mathbf{e}_{u_i} \in \mathbb{R}^D$ denotes the user embedding u_i, and $\mathcal{N}^{u_i}$ refers to the set of one-hop neighbors of the user entity u_i.

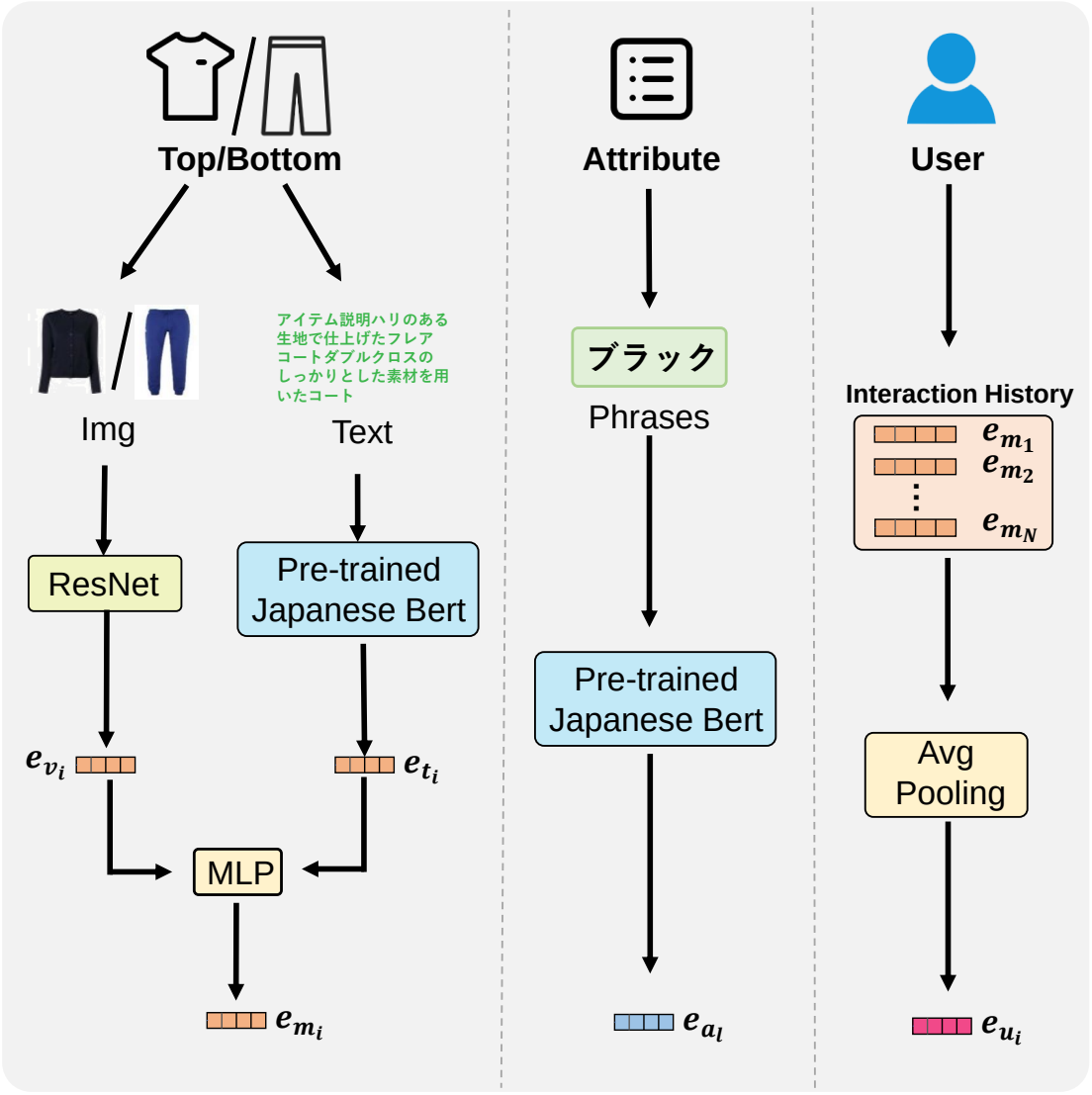


FIGURE 5.3: Illustration of the heterogeneous graph node embedding component.

5.3.2.2 Metapath-guided Heterogeneous Graph Learning

In this component, we conduct the metapath-guided heterogeneous graph representation learning to refine each entity's embedding with their context information. In particular, we first define a few user-/item-oriented metapaths to capture the high-order relations among entities, and then perform the metapath-guided semantic propagation to derive multiple semantic-enhanced embeddings for each user/item entity. Therein, each applicable metapath corresponds to a specific semantic-enhanced embedding. Ultimately, we fuse all the semantic-enhanced embeddings via a transformer to obtain the final user/item representation.

User-/Item-oriented Metapath Definition. According to [87], a metapath is defined as a path in the form of $X_1 \xrightarrow{R_1} X_2 \xrightarrow{R_2} \cdots \xrightarrow{R_n} X_{n+1}$, which describes a composite

relation between entities. In our work, as illustrated in Figure 5.4, there are actually various metapaths residing in our constructed heterogeneous graph, whereby three entities and rich relations exist. Intuitively, different metapaths reflect different semantics. For example, the metapath UIA[2] implies that a user historically prefers an item and that item possesses an attribute, while UIU indicates that these two end users like the same fashion item. Analogously, the metapath IAI refers to that the two end items share the same attribute, while IUI conveys that the two end items are interacted with by the same user. Pertaining to the POCM context, we only adopt metapaths that start from user entities and item entities. Formally, let $\mathcal{P}_{user} = \{r_1, \cdots, r_Y\}$ and $\mathcal{P}_{item} = \{s_1, \cdots, s_Z\}$ denote the set of predefined user-oriented and item-oriented metapaths, respectively. Y and Z denote the total number of user-oriented and item-oriented metapaths, respectively.

Metapath-guided Semantic Propagation. Based on the predefined user- and item-oriented metapaths, we can derive the corresponding metapath-guided user-oriented subgraphs for each user entity and the item-oriented subgraphs for each item entity via the breadth-first search strategy. Thereafter, based on the different information encoded by different metapaths, we can learn the user/item entity's embeddings with different semantics. To intuitively clarify how to refine users' or items' embeddings, we take the metapath UIA as an example. Other metapath-guided learning repeats the same procedure.

Assume that the metapath UIA is applicable to the user entity u_i. We then build a subgraph $\mathcal{G}_{u_i}^{\text{UIA}}$ for the user entity u_i. Since the length of the metapath UIA is three, we denote the one-hop neighbors of user entity u_i as $\mathcal{N}_{u_i}^{\text{UIA}(1)}$ consisting of all the items the user once interacted with. In the same way, we denote the two-hop neighbors of the user entity u_i as $\mathcal{N}_{u_i}^{\text{UIA}(2)}$, comprising all the attributes associated with items in $\mathcal{N}_{u_i}^{\text{UIA}(1)}$. Following that, we first aggregate the information from the two-hop neighbors to enhance the one-hop neighbors' embeddings, and then based on that learn the user's semantic-enhanced embedding as follows:

$$\begin{cases} \mathbf{e}_{m_i}^{\text{UIA}} = \mathcal{H}\Big(\mathbf{e}_{a_l} | a_l \in \mathcal{N}_{u_i}^{\text{UIA}(2)}\Big), m_i \in \mathcal{N}_{u_i}^{\text{UIA}(1)}, \\ \mathbf{h}_{u_i}^{\text{UIA}} = \mathcal{H}\Big(\mathbf{e}_{m_i}^{\text{UIA}} | m_i \in \mathcal{N}_{u_i}^{\text{UIA}(1)}\Big), \end{cases} \tag{5.5}$$

where $\mathcal{H}$ is the aggregation function, and $\mathbf{e}_{m_i}^{\text{UIA}}$ denotes the semantic-enhanced embedding of the item entity m_i, which is an one-hop neighbor of the user entity u_i. $\mathbf{h}_{u_i}^{\text{UIA}}$ represents the semantic-enhanced embedding of the user entity u_i. It is worth noting that during each hop aggregation, different neighbors may play different roles in characterizing the center entity. Specifically, some attributes may be more important in

[2]Due to the limited space, we omit the relation types between entities.

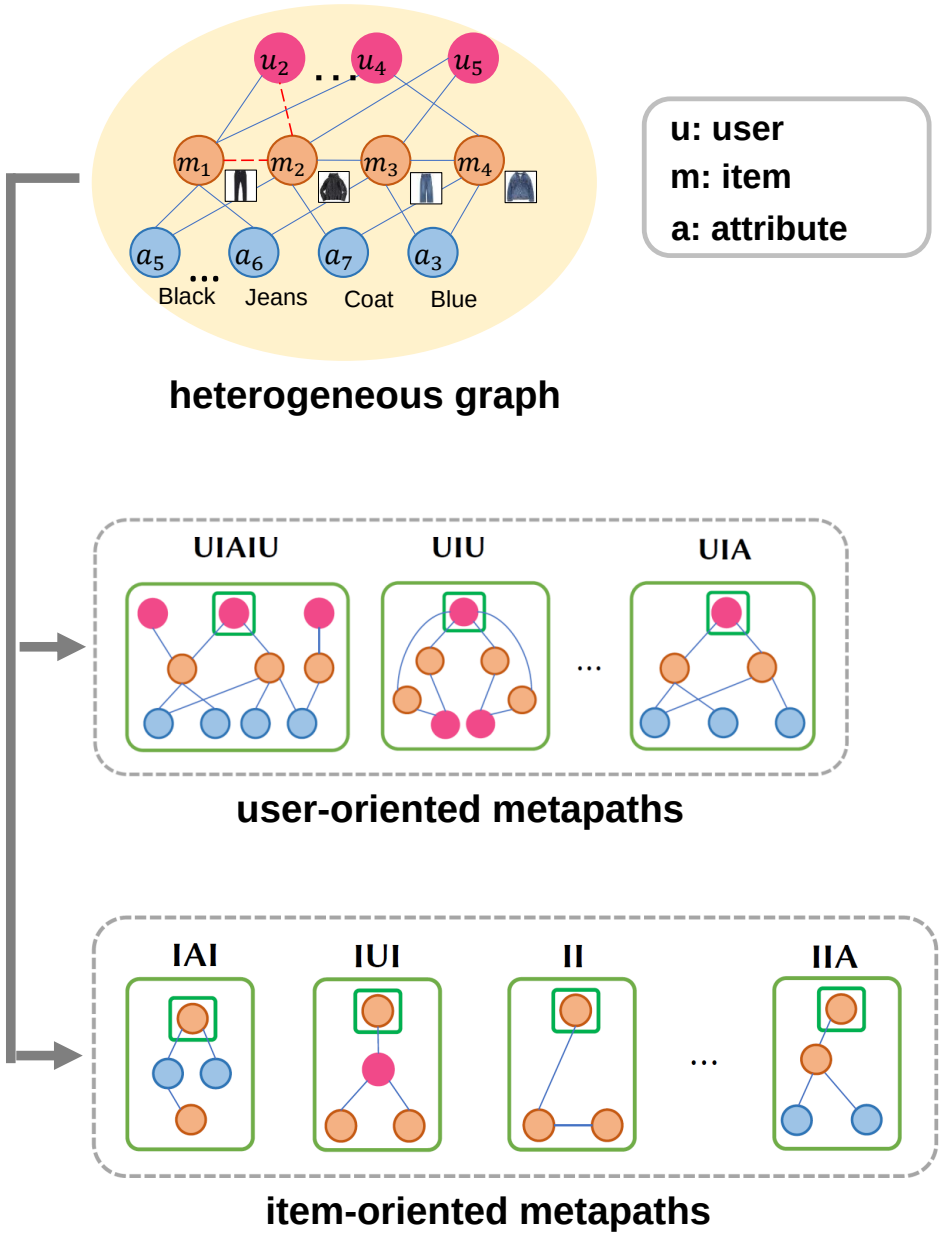


FIGURE 5.4: Illustration of user-oriented and item-oriented metapaths via heterogeneous graph.

conveying the item's properties, while some items may contribute more in reflecting the user's preference. Therefore, we adopt the graph attention mechanism of GAT [76] as the aggregation function, to highlight the informative and meaningful neighbor nodes. For simplicity, we take the aggregation operation over the one-hop neighbors of the user entity u_i as an example, and that over the two-hop neighbors can be defined similarly. Specifically, the aggregation operation $\mathcal{H}$ over the one-hop neighbors of the user entity

u_i can be written as follows,

$$\begin{cases} \mathbf{h}_{u_i}^{\text{UIA}} = \mathbf{e}_{u_i} + \sigma\left(\sum_{m_j \in \mathcal{N}_{u_i}^{\text{UIA}}} \alpha_{ij} \mathbf{e}_{m_j}\right), \\ \alpha_{ij} = \frac{\exp\left(\sigma\left(\mathbf{W}^{\text{UIA}}[e_{u_i}, e_{m_j}]\right)\right)}{\sum_{m_j \in \mathcal{N}_{u_i}^{\text{UIA}}} \exp\left(\sigma\left(\mathbf{W}^{\text{UIA}}[e_{u_i}, e_{m_j}]\right)\right)}, \end{cases} \tag{5.6}$$

where $\sigma(\cdot)$ denotes the activation function, $[,]$ refers to the concatenation operation, and $\mathbf{W}^{\text{UIA}} \in \mathbb{R}^{2D*1}$ is the node-level attention vector for the information aggregation under the metapath UIA.

Theoretically, repeating the above process for semantic propagation for all the other user-oriented metapaths, we can derive Y semantic-enhanced user embeddings for user entity u_i. However, in practice, not every user-oriented (item-oriented) metapath can be applied to a given user (item) entity. For example, once a user shares no preferred item with other users, we cannot derive the subgraph according to the metapath UIU. Accordingly, we use $\mathcal{P}^{u_i} = \{r_{i_1}, \cdots, r_{i_{Y_i}}\}$ to denote the set of metapaths that can be applied to the user entity u_i, where Y_i refers to the total number of metapaths applicable to the user u_i, and $r_{i_n} \in \mathcal{P}_{user}, n = 1, \cdots, Y_i$. Based upon $\mathcal{P}^{u_i}$, we can derive the corresponding semantic-enhanced embeddings for the user entity u_i, termed as $\{\mathbf{h}_{u_i}^p | p \in \mathcal{P}^{u_i}\}$, following the above metapath-guided semantic propagation process. Similarly, we use $\mathcal{P}^{m_i} = \{s_{i_1}, \cdots, s_{i_{Z_i}}\}$ to denote the set of metapaths that can be applied to the item entity m_i, where Z_i is the total number of metapaths applicable to the item m_i, and $s_{i_z} \in \mathcal{P}_{item}, z = 1, \cdots, Z_i$. In the same manner, we reach the semantic-enhanced embeddings for the item entity m_i as $\{\mathbf{h}_{m_i}^p | p \in \mathcal{P}^{m_i}\}$.

Semantic-enhanced Embedding Fusion. Thus far, we have achieved multiple semantic-enhanced embeddings for each user and item entity under different metapaths, and each embedding characterizes one aspect. To comprehensively represent each user or item, we propose fusing the multiple embeddings of each user or item.

In particular, we leverage the transformer [80] without the positional coding to perform the multisemantic embedding fusion due to the following two concerns: 1) the number of semantic-enhanced embeddings for different users can be different, and 2) there is no explicit order among these semantic-enhanced embeddings of each user or item entity. To ensure that the fused embeddings of the users and items are in the same space, we adopt a single transformer to fulfill both user and item entities' embedding fusion as follows,

$$\begin{cases} \tilde{\mathbf{h}}_{u_i} = \text{Transformer}(\mathbf{h}_{u_i}^p | p \in \mathcal{P}^{u_i}) \\ \tilde{\mathbf{h}}_{m_i} = \text{Transformer}(\mathbf{h}_{m_i}^p | p \in \mathcal{P}^{m_i}) \end{cases} \tag{5.7}$$

where $\tilde{\mathbf{h}}_{u_i}$ and $\tilde{\mathbf{h}}_{m_i}$ are the final representation for the user u_i and item m_i, respectively.

5.3.2.3 Personalized Outfit Compatibility Modeling

To accomplish the POCM task, we first build the training set $\Omega = \{(u_i, m_j, m_{k+}, m_{k-}) | m_j \in \mathcal{M}^{t(b)}, m_{q+}, m_{q-} \in \mathcal{M}^{b(t)}, y_{ij}^{k+} = 1, y_{ij}^{k-} = 0\}$, where $y_{ij}^{k+} = 1$ denotes the triplet (u_i, m_j, m_{k+}) is compatible, *i.e.,* the item m_{k+} goes well with the given item m_j according to user u_i's preference. $y_{ij}^{k-} = 0$ indicates that the triplet (u_i, m_j, m_{k-}) is incompatible. Following that, for each triplet, we obtain each entity's representation according to Eqn. (6.7), namely, $\tilde{\mathbf{h}}_{u_i}$, $\tilde{\mathbf{h}}_{m_j}$, and $\tilde{\mathbf{h}}_{m_{k+}}$/ $\tilde{\mathbf{h}}_{m_{k-}}$. We then resort to the MLP to derive the compatibility score for each triplet as follows,

$$\hat{p}_{ij}^{k+(-)} = \text{MLP}_0([\tilde{\mathbf{h}}_{u_i}, \tilde{\mathbf{h}}_{m_j}, \tilde{\mathbf{h}}_{m_{k+(-)}}]), \tag{5.8}$$

where $\hat{p}_{ij}^{k+(-)}$ is the predicted compatibility score for the given triplet.We then adopt the cross-entropy loss as follows,

$$\mathcal{L}^{(i,j,k+,k-)} = -log\Big(\frac{exp(\hat{p}_{ij}^{k+})}{exp(\hat{p}_{ij}^{k+}) + exp(\hat{p}_{ij}^{k-})}\Big). \tag{5.9}$$

Intuitively, the compatible and incompatible triplets should follow some compatible and incompatible patterns, respectively. In light of this, given a compatible triplet $(u_{i^+}, m_{j^+}, m_{k^+})$, we argue that its latent representation should be more similar to that of a compatible triplet as compared to that of an incompatible one $(u_{i^-}, m_{j^-}, m_{k^-})$. Accordingly, we further introduce contrastive regularization to regulate the similarity between latent representations of different triplet pairs. Assume that $p_1^+ = (u_{i_1^+}, m_{j_1^+}, m_{k_1^+})$ and $p_2^+ = (u_{i_2^+}, m_{j_2^+}, m_{k_2^+})$ are two compatible triplets, while $n^- = (u_{i^-}, m_{j^-}, m_{k^-})$ is an incompatible triplet. We utilize two MLPs to obtain the latent representations for these three triplets as follows,

$$\begin{cases} \tilde{\mathbf{h}}_{p_1^+} = \text{MLP}_1([\tilde{\mathbf{h}}_{u_{i_1^+}}, \tilde{\mathbf{h}}_{m_{p_1^+}}, \tilde{\mathbf{h}}_{m_{q_1^+}}]), \\ \tilde{\mathbf{h}}_{p_2^+} = \text{MLP}_2([\tilde{\mathbf{h}}_{u_{i_2^+}}, \tilde{\mathbf{h}}_{m_{p_2^+}}, \tilde{\mathbf{h}}_{m_{q_2^+}}]), \\ \tilde{\mathbf{h}}_{n^-} = \text{MLP}_2([\tilde{\mathbf{h}}_{u_i^-}, \tilde{\mathbf{h}}_{m_{p^-}}, \tilde{\mathbf{h}}_{m_{q^-}}]), \end{cases} \tag{5.10}$$

where $\tilde{\mathbf{h}}_{p_1^+}$ and $\tilde{\mathbf{h}}_{p_2^+}$ are the latent representations of the two compatible/positive triplets, while $\tilde{\mathbf{h}}_{n^-}$ is the latent representation of the incompatible/negative triplet. We then use the following contrastive regularization as follows,

$$\mathcal{L}_{cons}^{(p_1^+, p_2^+, n^-)} = -log\frac{exp(sim(\tilde{\mathbf{h}}_{p_1^+}, \tilde{\mathbf{h}}_{p_2^+}))}{exp(sim(\tilde{\mathbf{h}}_{p_1^+}, \tilde{\mathbf{h}}_{p_2^+})) + exp(sim(\tilde{\mathbf{h}}_{p_1^+}, \tilde{\mathbf{h}}_{n^-}))}, \tag{5.11}$$

TABLE 5.2: Statistics over our newly constructed dataset.

Table of Content	Statistical Results
User	$1,769$
Top	$53,092$
Bottom	$41,157$
Attribute	98
Outfit (top-bottom)	$81,937$
Triplet (user-top-bottom)	$82,079$
User historical interacted outfits-min	10
User historical interacted outfits-max	200
User historical interacted outfits-avg	46

where $sim(,)$ refers to the dot product operation. Finally, our objective function can be written as follows,

$$\mathcal{L} = \sum_{(u_i, m_j, m_{k+}, m_{k-})} \mathcal{L}^{(i,j,k+,k-)} + \lambda \sum_{(p_1^+, p_2^+, n^-)} \mathcal{L}_{cons}^{(p_1^+, p_2^+, n^-)}, \tag{5.12}$$

where λ is the nonnegative hyperparameter balancing the importance of the cross-entropy loss and contrastive regularization.

5.4 Experiment

In this section, we conducted experiments over real-world datasets by answering the following research questions.

- **RQ1**: Does MG-POCM outperform state-of-the-art baselines?
- **RQ2**: How does each module affect MG-POCM?
- **RQ3**: Is our model sensitive to the number of the transformer and GAT layers?
- **RQ4**: What is the intuitive performance of MG-POCM?

5.4.1 Experimental Settings

In this part, we present the dataset, evaluation tasks, metrics, and the implementation details.

5.4.1.1 Dataset.

To justify our model, similar to existing methods [12, 24], we also resorted to the public benchmark dataset IQON3000 [12], due to the fact that each item in IQON3000 has not only the visual image and textual description but also the semantic attributes, such as the color and category. In particular, IQON3000 consists of 308, 747 outfits, composed by 672, 335 items. To fit our task and ensure the quality of the dataset, we did not completely follow up the experimental setting in [12, 24] considering the following two concerns. 1) As to a given user, they only focus on matching bottoms for a given top. By contrast, in our work, the top and bottom are arbitrarily switchable for a given user. That is, we aim to match either tops for a given bottom or bottoms for a given top. And 2) they did not set the criterion for filtering out users with limited interacted items. Accordingly, we derived our dataset from IQON3000. In particular, we only retained the outfits that contained a top and a bottom, and users who had interacted with no less than 10 and no more than 200 outfits to keep the dataset relatively balanced. Finally, there were 82, 079 user-top-bottom triplets involved 1, 769 users. The detailed statistics are summarized in Table 5.2. The attributes and their corresponding value examples of the derived dataset are shown in Table 5.3.

Notably, all these retained triplets are positive, namely, compatible triplets. We then randomly split these user-top-bottom triplets into four chunks: graph construction set, training set, validation set, and testing set, by the ratio of $6 : 2 : 1 : 1$, resulting in a 49, 297 triplet for constructing the heterogeneous graph, 16, 416 triplets for training, 8, 208 triplets for validation, and 8, 208 triplets for testing. Thereafter, as to each positive triplet in the training set, validation set, or testing set, we randomly selected an item (either the top or the bottom) from this triplet as the given item, leaving the other item as the target (positive). Following that, we replaced the target (positive) item with a randomly sampled item sharing the same category as the target item, to derive a negative triplet. It is worth noting that to ensure fairness, considering the baseline methods do not need the specific graph construction set, we trained them with both the graph construction set and training set, where the negative triplets in the graph construction set were derived in the same manner.

5.4.1.2 Evaluation Tasks and Metrics.

Similar to previous studies [3, 12, 18, 24, 64], we justified our proposed MG-POCM scheme via the compatibility estimation task. This task is to evaluate the compatibility score of an arbitrary top-bottom pair for a specific user, where we adopted the AUC (area under the ROC curve) [98] as the evaluation metric. In addition, we evaluated

TABLE 5.3: Attributes and their possible value examples in the derived dataset.

Attribute	Possible Value Examples	Total Number
Color	Gray, Black, Red, ···	12
Price	Low, Middle, High.	3
Category	Coat, Skirt, Jacket, ···	12
Variety	Tops, Dress, Trousers, ···	5
Material	Fur, Leather, Denim, ···	31
Pattern	Stripe, Print, Dot, ···	15
Design	Frill, V-neck, Ribbons, ···	13
Dress Length	Short, Middle, Long.	3
Sleeve Length	Sleeveless, Long, Short, ···	4

TABLE 5.4: Performance comparison between our proposed MG-POCM and other baseline methods in terms of AUC and MRR over IQON3000. The best results are in bold, while the second best results are underlined.

Approaches	AUC	MRR
PAI-BPR [24]	0.6096	0.5456
HFGN [5]	0.6783	0.6173
GP-BPR [12]	<u>0.7146</u>	<u>0.6346</u>
MG-POCM	**0.7730**	**0.6427**

the performance of our model with the complementary item retrieval task [12]. For each positive triplet, we derived a corresponding negative triplet by randomly replacing one item (either a top or a bottom). We merged the original replaced item in the positive triplet and the newly added item in the negative triplet as the set of item candidates. These item candidates were ranked according to their compatibility scores to the given user and item, *i.e.*, the unchanged item in the original positive triplet, calculated by Eqn.(5.8). To measure the complementary item retrieval task, we utilized the mean reciprocal ranking (MRR) [99] as the metric.

5.4.1.3 Implementation Details

Pertaining to the visual embedding of items, we utilized ResNet18 and converted each item image into a 512-D vector. Notably, ResNet18 was also fine-tuned with the whole model. Regarding the textual feature extraction of items, we implemented BERT[3] for Japanese text considering our dataset is in Japanese and embedded each item's textual description into a 768-D vector. The dimension of the final item embedding was $D = 512$. Similarly, using this BERT implementation, each semantic attribute was also embedded into a 768-D vector. We set the number of layers of all MLPs used in our scheme as 2 and employed Gaussian error linear units (GELU) as the activation function. In practice, we

[3] https://huggingface.co/cl-tohoku/bert-base-japanese-char/tree/main.

adopted the following set of user-oriented metapaths $\mathcal{P}_{user} = \{\text{UIAIU}, \text{UIU}, \text{UIA}\}$, and item-oriented metapaths $\mathcal{P}_{item} = \{\text{IAI}, \text{IUI}, \text{II}, \text{IIA}\}$. During the subgraph construction for each user/item entity, for efficiency, we set the maximum neighbor size of each node as 5. As to the optimization, we adopted the adaptive moment estimation method (Adam [54]). The learning rate was prepared in the 6% steps to the peak value, which were set to 1e-4, and then linearly decayed to 0. The hyperparameter λ was set to 1, and the batch size was set to 24. All the experiments were implemented by PyTorch over a server equipped with 4 A100-PCIE-40GB GPUs.

5.4.2 Model Comparison

To validate the effectiveness of our proposed scheme, we chose the following state-of-the-art baselines for comparison.

- **GP-BPR** [12] is a comprehensive personal preference modeling scheme, where the multimodal data (*e.g.*, the image and text description) of fashion items are jointly explored.
- **PAI-BPR** [24] is an attributewise interpretable compatibility modeling scheme, which solves the problem of interpretability in clothing matching by locating the discordant and harmonious attributes between fashion items.
- **HFGN** [5] refers to a hierarchical fashion graph network, which simultaneously models the relationships among users, items, and outfits.

Table 5.4 shows the performance comparison among different methods on IQON3000 dataset in terms of AUC and MRR. From this table, we make the following observations.

- Our proposed MG-POCM scheme consistently outperforms all baseline methods over different metrics. In particular, MG-POCM performs better than PAI-BPR and GP-BPR, which indicates the advantage of our scheme that organizes the various entities and relations in the context of POCM into a unified heterogeneous graph and utilizes the metapath-guided heterogeneous graph learning towards personalized outfit compatibility modeling.
- Our method surpasses the heterogeneous graph-based method HFGN remarkably over both metrics, implying the necessity of considering the items' attributes.
- GP-BPR outperforms HFGN, which may be because HFGN only utilizes the visual information of fashion items, while GP-BPR jointly considers the images and textual descriptions of items.

TABLE 5.5: Ablation study of our proposed MG-POCM on IQON3000 dataset. The best results are in bold.

Method	AUC	MRR
w/o text	0.7630	0.6392
w/o image	0.7655	0.6382
w/o attribute	0.7339	0.5717
w/o (II,UIA)	0.7639	0.6392
w/o contrastive	0.7647	0.6338
w mean pooling	0.7627	0.6392
MG-POCM	**0.7730**	**0.6427**

5.4.3 Ablation Study

To verify the importance of each component in our model, we conducted ablation experiments on the following derivatives.

- **w/o text**: To study the impact of the textual description of fashion items in POCM, we removed the textual embeddings of items, and kept other parts unchanged.
- **w/o image**: Similarly, to justify the necessity of incorporating the item images in the context of POCM, we omitted the items' visual embeddings, and kept other parts unchanged.
- **w/o attribute**: To verify the importance of the item attributes, we discarded the attribute entities as well as the attribute-related metapaths. The rest of our MG-POCM was unchanged.
- **w/o (II,UIA)**: To validate the necessity of incorporating the metapaths II and UIA, which can be treated as the subpaths of metapaths *i.e.,* IIA and UIAIU, respectively, we omitted them during our heterogeneous graph learning.
- **w/o contrastive**: To explore the effect of the contrastive regularization component, which is used to enhance the latent representation of each entity, we removed the contrastive regularization by setting $\lambda = 0$ in Eqn.(5.12).
- **w/ mean pooling**: To evaluate the function of the transformer component in the semantic embedding fusion, we replaced the transformer component with the mean pooling function.

Table 5.5 summarizes the ablation study results. From this table, we observed that our model consistently outperforms all the above derivatives, which demonstrates the effectiveness of each component in our proposed MG-POCM. Specifically, we make the following detailed observations.

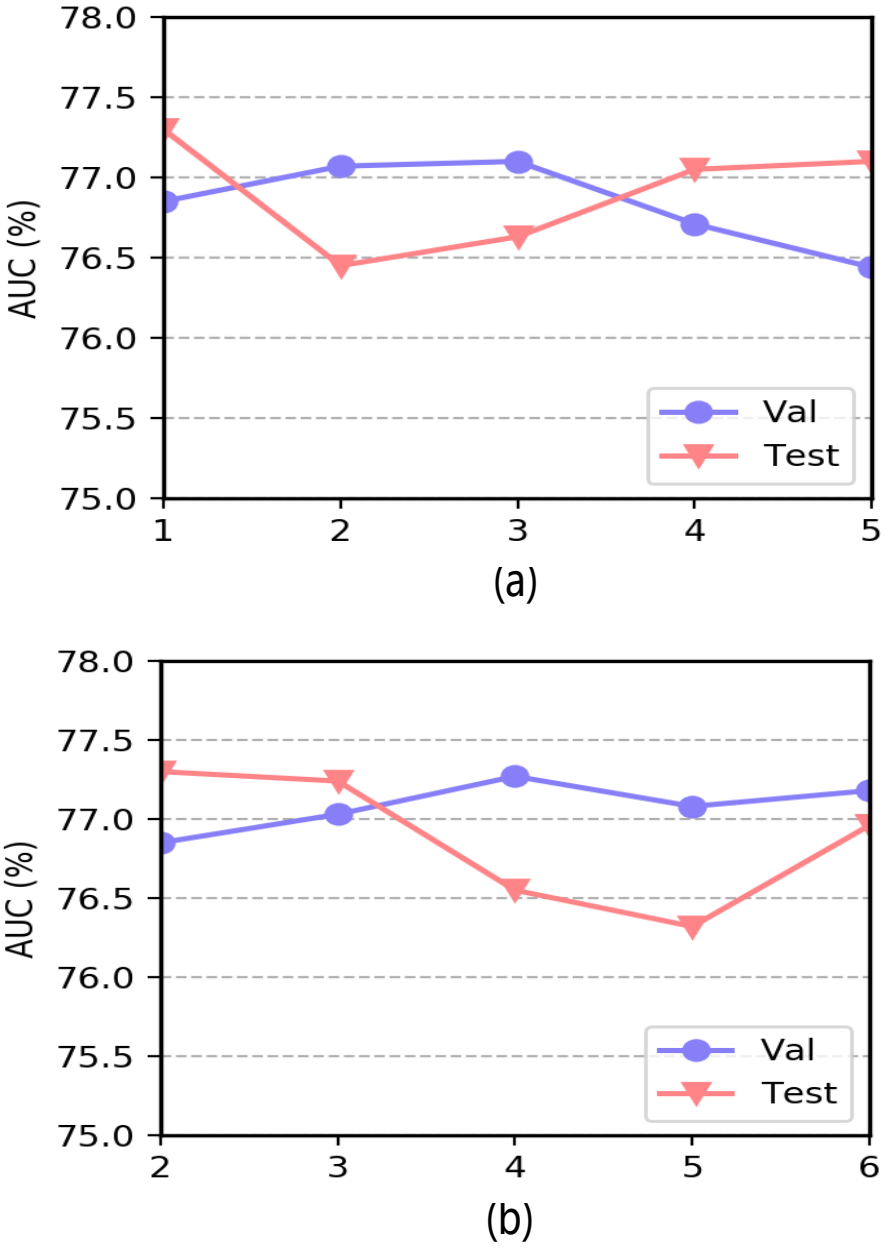


FIGURE 5.5: Sensitivity analysis of our model performance in terms of AUC with respect to (a) the number of transformer layers, and (b) the number of GAT layers.

- Both w/o text and w/o image perform inferior to MG-POCM, which suggests that it is essential to consider both modalities of fashion items to boost the item representation learning.
- w/o attribute presents the worst performance, reflecting the benefit of incorporating the attribute entities as well as their semantic contents into personalized outfit compatibility modeling.
- w/o (II,UIA) performs worse than our MG-POCM, which suggests subpaths may emphasize the high-order correlations without adding noisy information.

- The performance of w/o contrastive drops by a large margin, as compared to MG-POCM, indicating that the contrastive regularization is indeed helpful to strengthen the fashion entity representation learning.
- w/ mean pooling also performs worse than our MG-POCM, reflecting the effectiveness of the transformer in fusing the unfixed number of semantic-enhanced embeddings of users/items.

5.4.4 Sensitivity Analysis

In this part, we evaluated the sensitivity of our model in terms of the number of transformer and GAT layers. In particular, we varied the number of transformer layers from 1 to 5 with the step size of 1. Considering that most of our metapaths involve more than 2 entities, we changed the number of GAT layers from 2 to 6 with the step of 1. Figure 5.5 (a) and (b) illustrate the performance of our model on the validation set and testing set with different numbers of transformer layers and GAT layers, respectively. As can be seen, our model achieves relatively stable performance with different numbers of transformer and GAT layers, which implies that our model is not sensitive to these two hyperparameters. Accordingly, in practice, to improve the model efficiency, we set the number of transformer and GAT layers as 1 and 2, respectively.

5.4.5 Case Study

To gain more intuitive insights into our model, we also conducted a case study of our method and the w/o attribute derivative. Figure 5.6 shows three testing samples, where the users' historical preferred top-bottom pairs and items' attributes are also listed to facilitate the experimental result analysis[4]. As can be seen, for the case of the first user with the given brown skirt, although both our MG-POCM and its derivative w/o attribute gives the correct prediction, our MG-POCM assigns a much higher score to the positive item than the negative item. By contrast, w/o attribute gives the former a slightly higher score than the latter item. Namely, our model has a high confidence than its derivative. This may be because incorporating the attribute entities in FPCM enables our model MG-POCM to learn the "floral pattern" that the user prefers for tops, and accordingly gives the positive item with the "floral pattern" a higher score. Similarly, the same phenomenon can be observed in the second case. As can be seen, the second user tends to prefer bottoms of the category "long pants" to match tops, which can be more easily captured by our MG-POCM rather than the w/o attribute derivative.

[4]Due to the limited space, we did not provide the text description of the items.

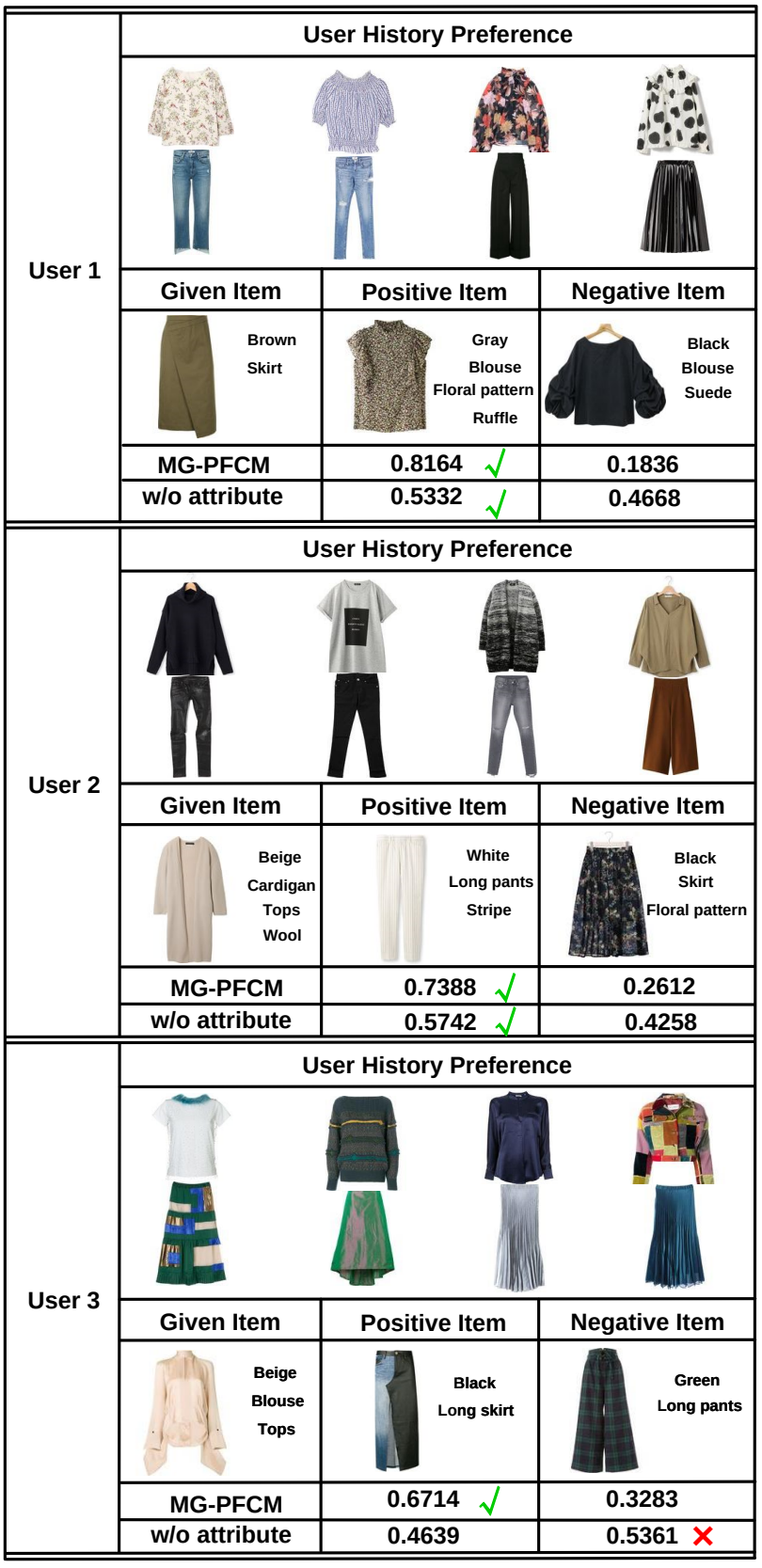


FIGURE 5.6: Illustration of several POCM results obtained by our MG-POCM and w/o attribute derivative.

Additionally, as the negative item is a black skirt, which does not go well with the beige cardigan, our MG-POCM assigns a much higher score to the positive item, while w/o attribute only rates a slightly higher score to it, as compared with the negative item. In the last case, we can see that the third user prefers "long skirts" with blouses. The positive item is a black long skirt, looking like long pants, while the negative item is long pants looking like a long skirt. Then with the help of their category attributes, our MG-POCM correctly selects the compatible item for the given top, while the w/o attribute method gives the incorrect judgment. Overall, based on these case studies, we can confirm the effectiveness of our method in POCM, and the benefit of incorporating the attribute information in the context of POCM.

5.5 Summary

In this chapter, we solved the personalized outfit compatibility modeling problem by organizing the various fashion entities and relations into a unified heterogeneous graph and presented a novel metapath-guided personalized compatibility modeling (MG-POCM) scheme to learn entity embeddings. Extensive experiments were conducted on the public dataset IQON3000, which demonstrates the superiority of our model over existing methods. The ablation study verifies the importance of each key module, such as jointly considering the text, image, and attribute information of items towards POCM, the contrastive regularization and using a transformer to fulfill the semantic-enhanced embedding fusion. Moreover, experimental results show that our model is insensitive to the numbers of transformer and GAT layers, which enables the model to perform well with fewer parameters.

Chapter 6

Heterogeneous Graph Hashing for Efficient Outfit Recommendation

6.1 Introduction

Recent years have witnessed the unprecedented growth of fashion-oriented products in E-commerce platforms. Nevertheless, as the online fashion products surge, it becomes increasingly difficult and expressive for users to seek their desired ones from the numerous candidates. In the light of this, fashion recommendation has gained increasing attention from both industry and academia, due to its great economic value and benefit in improving the users' consumption experience. In previous chapters we focus on the outfit compatibility modeling task. In this chapter, we shift to the task of personalized outfit recommendation (POR), which aims to directly recommend a whole well-matched outfit to the user. In fact, a few efforts have been dedicated to the task of POR. For example, Li et al. [5] addressed this task by adopting Graph Neural Networks (GNNs) [65] to learn outfit and user representations. Lin et al. [34] proposed the OutfitNet model via leveraging attention-based multiple instance learning. Despite their promising performance, they merely focus on improving the recommendation accuracy, while ignoring the efficiency. In fact, the number of available outfit candidates on fashion platforms has kept increasing exponentially, which implies the necessity of deploying an efficient POR system. Towards this end, Lu et al. [100] proposed to learn a hash binary code for each fashion entity (*i.e.,* the user and item). Nevertheless, they overlooked the influence of attributes for POR. As a matter of fact, attributes usually characterize key semantic information of an item, which can be hardly expressed by the visual image, such as the material and price. Therefore, in this work, we bring in attributes associated with fashion items, and work towards fully exploring all entities (*i.e.,* users, outfits, items and

User	Outfit	
 User A	  	
Item (Image)	**Item (Text)**	**Item (Attribute)**
	SweatyRocks Women's Round Neck Slim Fit Short Sleeve T-shirt	Fabric, Short Sleeve, Round Neck, $11.99
	Levi's Women's 501 Original Shorts	Casual Jean Shorts High Waisted, $21.99
	Adidas Women's Grand Court Sneaker	Soft Leather Adidas Tennis $35.99
	Champion Classic Twill Hat	Adjustable Cap Champion $15.99
User	**Outfit**	
 User B	  	
Item (Image)	**Item (Text)**	**Item (Attribute)**
	Women's Vintage 1950s style Wrap V Neck Tie Waist Formal Cocktail Dress	Sweetheart Neckline Wrap Ruched Tie Waist $10.99
	DREAM PAIRS Women's Low-Chunk Low Heel Pump Sandals	Leather-like material Two Strap Heeled $28.99
	14k Gold Long Tassels Chain Dangle Drop Earrings for Women Girls	Gold Punk Sleek Drop Earrings $10.99

FIGURE 6.1: Examples of users' outfit compositions.

attributes) and their various relations (*i.e.*, user-outfit, outfit-item, and item-attribute interactions) to further promote the performance of the efficient POR.

Inspired by the remarkable success of graph learning in entity representation learning [7] and outfit recommendation [5], we resort to the graph learning technique to fulfill the efficient POR. However, this is non-trivial due to the following three key challenges.

- **C1: Heterogeneous fashion entities.**

  As shown in Figure 6.1, our work involves four kinds of entities, namely, users, outfits, items and attributes, where some entities (*e.g.,* items, and attributes) have explicit contents, while others (*e.g.,* users and outfits) have not. Therefore, how to seamlessly unify these heterogeneous entities within a graph is the tough challenge we face.

- **C2: Efficient graph convolution.** Traditional graph convolution methods focus on propagating message among graph nodes and updating entity representations simultaneously at the end of the whole graph convolution. Such graph convolution schemes suffer from two key limitations. First, they do not explicitly distinguish the types of entities during information propagation. Second, their convolution is not totally efficient, especially for our heterogeneous graph that involves multiple

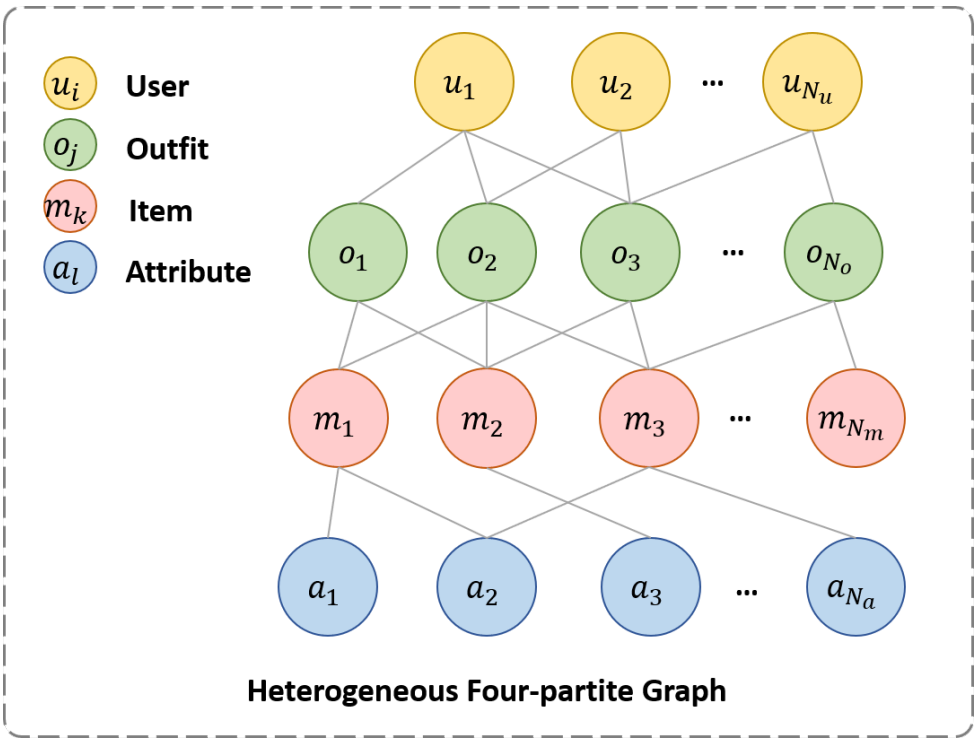


FIGURE 6.2: Illustration of the heterogeneous four-partite graph.

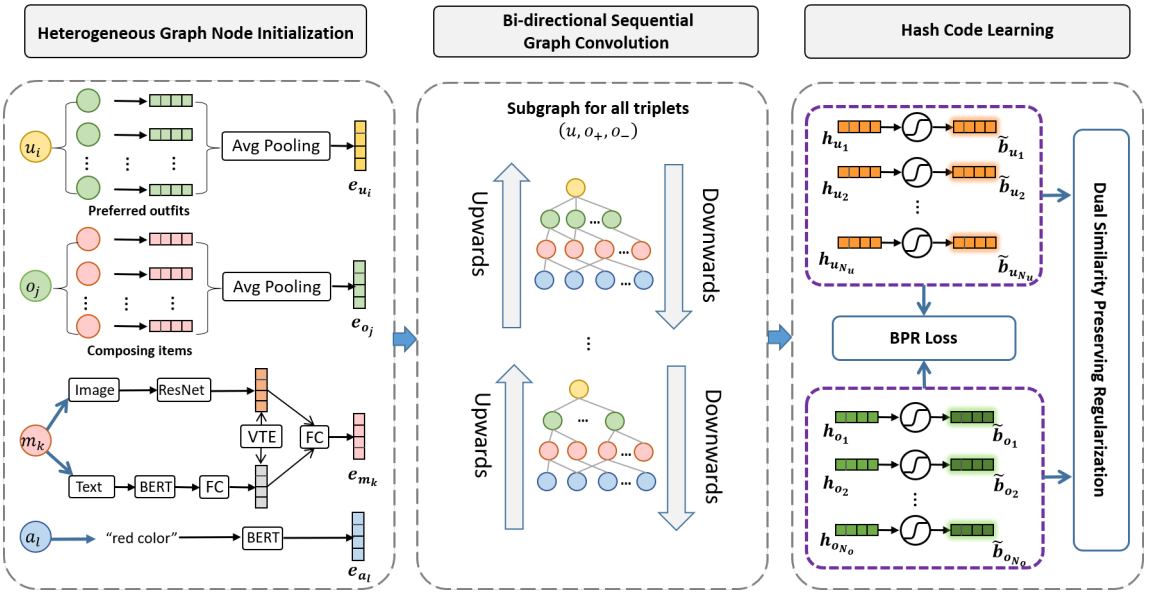


FIGURE 6.3: Illustration of the proposed bi-directional heterogeneous graph hashing scheme. It consists of three key components: 1) heterogeneous graph node initialization, 2) bi-directional sequential graph convolution, and 3) hash code learning.

types of entities and relations. This is because once a type of entity representations get updated, they can be used timely for updating other types of entities. Therefore, how to fulfill the efficient graph convolution is another challenge.

- **C3: Information preservation during hashing.** Converting the continuous hash representation of each entity to the binary hash code inevitably loses certain information. Therefore, how to retain the original information to the greatest extent during the hash code learning constitutes the third challenge.

To address these challenges, we first organize the user, outfit, item and attribute entities in the context of POR into a unified heterogeneous four-partite graph, as shown in Figure 6.2. Specifically, the nodes of this graph correspond the four types of entities, and are linked by three types of edges: user-outfit interactions, outfit-item relations, and item-attribute association relations. We then devise a novel bi-directional heterogeneous graph hashing scheme, called BiHGH, as shown in Figure 6.3. This scheme consists of three key components: heterogeneous graph node initialization, bi-directional sequential graph convolution, and hash code learning. The first component works on embedding entities in our heterogeneous graph from the content level. It is worth mentioning that instead of using the conventional fixed one-hot or learn-from-scratch embedding, we initialize each outfit entity based upon the contents of its composing item entities, and each user entity by its historically interacted outfit entity embeddings. Moreover, we incorporate the contrastive loss [101] to encourage the semantic consistency between visual and textual modalities. In the second component, we first divide the four-partite graph into

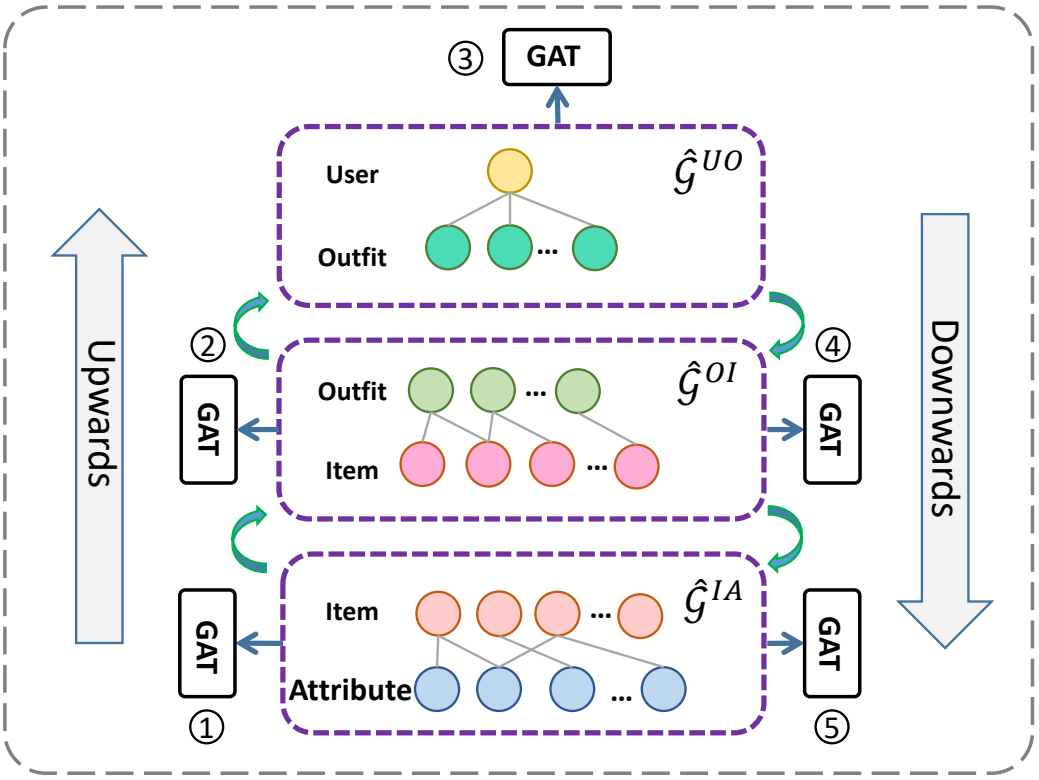


FIGURE 6.4: Illustration of the proposed bi-directional sequential graph convolution algorithm.

three subsequent subgraphs, namely user-outfit subgraph, outfit-item subgraph, and item-attribute subgraph, as illustrated in Figure 6.4. We then conduct bi-directional graph convolution over the subgraphs sequentially and iteratively upwards and downwards. In this manner, we can separately deal with different types of neighbor entities and reduce the computational cost of the big graph convolution. As to the third one, it aims to obtain the binary hash code of each entity. To accomplish this, we adopt the BPR [20] loss for optimization, which has been proven to be effective in the user preference modeling. Meanwhile, to enhance the hash code learning of users and outfits, we design a dual similarity preserving regularization to remain information during hashing. The underlying philosophy is that if two users prefer the same outfit or two outfits share a certain item, their corresponding hash codes should be similar. We conduct extensive experiments on the benchmark dataset, and the results demonstrate the superiority of our model to several cutting-edge baselines. We have released our code and parameters[1].

Our main contributions can be highlighted in threefold:

- We present a novel heterogeneous graph learning-based outfit recommendation scheme, where the four types of entities (*i.e.*, users, outfits, items, and attributes) and their relations are seamlessly integrated.

[1] https://outfitrec.wixsite.com/bihgh.

- We creatively devise a bi-directional graph convolution algorithm to fulfill the efficient optimization, which works on sequentially transferring knowledge via repeating upwards and downwards optimization. It greatly alleviates the computational cost of big graph convolution.
- We design a dual (*i.e.*, outfit-level and user-level) similarity preserving regularizations to prevent the information loss during the hash learning.

The research work in this chapter has been published in ACM MM 2022.

Weili Guan, Xuemeng Song, Haoyu Zhang, Meng Liu, Chung-Hsing Yeh, Xiaojun Chang. "Bi-directional Heterogeneous Graph Hashing towards Efficient Outfit Recommendation." In Proceedings of the International ACM Conference on Multimedia. ACM, 2022.

6.2 Related Work

This work is related to the learning to hash.

Learning to Hash. Learning to hash, which aims to learn a compact binary hash code for the given instance, has attracted a large amount of research interest [102, 103], due to its prominent advantages in saving the time and storage costs. One key feature of hashing methods [100, 104, 105] is that the *sign* function is usually used to convert the continuous representations of instances into binary hash codes. This hinders the direct optimization of the discrete hash codes. Therefore, most hashing methods would allow some relaxations over the optimization, that is, optimizing the model over the continuous item hash representation, which can be seen as the surrogates of the binary hash codes. To reduce the quantization error, HashNet [106] leverages the scaled *tanh* function to approximate the *sign* function.

Inspired by its astonishing success in efficiency improvement, hash learning has been adopted in several recommendation tasks [100, 107]. For example, Zhou et al. [107] investigated the efficient item recommendation by learning binary codes for collaborative filtering. The hamming distance between the binary codes of the user and the item is used for measuring the user's preference over items. Lu et al. [100] proposed to learn binary codes for fashion entities towards efficient personalized outfit recommendation, where the BPR loss for user preference modeling and contrastive loss for vision-text embedding regularization are used for optimization.

Despite their significance in efficient recommendation, they cannot effectively prevent the information loss during hash code learning. Towards this end, in this work, we designed a dual similarity preserving regularization.

6.3 Methodology

In this section, we first formulate the research problem, and then detail the three components of our proposed BiHGH scheme.

6.3.1 Problem Formulation

Formally, we first declare some notations. We use boldface uppercase letters (e.g., X) and boldface lowercase letters (e.g., x) to denote matrices and vectors, respectively. We employ non-bold letters (e.g., i and N) to represent scalars and Greek letters (e.g.,α) to denote hyperparameters. Let $\|\mathbf{A}\|_F$ denotes the Frobenius norm of matrix $\mathbf{A}$. If not clarified, all vectors are in the column forms.

In this work, we focus on the task of POR, which targets at recommending an outfit rather than a single fashion item to a given user. Suppose that we have a set of N_u users $\mathcal{U} = \{u_1, u_2, \cdots, u_{N_u}\}$, a set of N_o outfits $\mathcal{O} = \{o_1, o_2, \cdots, o_{N_o}\}$, a set of N_m items $\mathcal{M} = \{m_1, m_2, \cdots, m_{N_m}\}$, and a set of N_a attributes $\mathcal{A} = \{a_1, a_2, \cdots, a_{N_a}\}$. Each user u_i historically interacts with a set of preferred outfits $\mathcal{O}_i \subseteq \mathcal{O}$, each outfit o_j is composed by a set of compatible items, denoted as $\mathcal{M}_j \subseteq \mathcal{M}$, and each item m_k is associated with a set of attributes $\mathcal{A}_k \subseteq \mathcal{A}$. Moreover, as illustrated in Figure 6.1, each item m_j also involves an image v_j and a textual description t_j.

We resort to a heterogeneous four-partite graph to organize entities within a unified structure. In particular, we denote the graph as $\mathcal{G} = (\mathcal{E}, \mathcal{R})$, where $\mathcal{E} = \mathcal{U} \cup \mathcal{O} \cup \mathcal{M} \cup \mathcal{A}$ represents the set of entity nodes, and $\mathcal{R}$ denotes the set of edges linking these nodes. It is noteworthy that edges characterize various relations among entities, including user-outfit historical interactions, outfit-item subordinating relations, and item-attribute association relations. In addition, we have a set of triplet training samples $\mathcal{D} = \{(u_i, o_j, o_k)|u_i \in \mathcal{U}, o_j \in \mathcal{O}, o_k \in \mathcal{O}, (u_i, o_j) \notin \mathcal{E}\}$, where o_j and o_k denote the positive and negative outfit according to the user u_i's preference, respectively. In a sense, based on the heterogenerous four-partite graph and training set $\mathcal{D}$, we aim to optimize a model $\mathcal{F}$, which can accurately and efficiently estimate the preference degree of a given user towards an arbitrary outfit as follows,

$$s_{ij} = \mathcal{F}(u_i, o_j|\Theta_F, \mathcal{G}), \tag{6.1}$$

where s_{ij} denotes the preference of the user u_i to the outfit o_j, and Θ_F refers to the to-be-learned model parameters.

6.3.2 BiHGH

As illustrated in Figure 6.3, BiHGH consists of three components: heterogeneous graph node initialization, bi-directional heterogeneous graph convolution, and hash code learning. In this subsection, we elaborate each component of BiHGH.

6.3.2.1 Heterogeneous Graph Node Initialization

This component aims to initialize node-level embeddings in the heterogeneous graph. As the heterogeneous graph has four types of entities, we introduce their corresponding embedding methods one by one.

Item Embedding. Inspired by previous studies [3, 63, 64], we jointly exploit the image and textual information to learn the item embedding. In particular, we utilize the pre-trained ResNet [48], which will be fine-tuned during training, to obtain the item's visual feature. Meanwhile, we adopt the pre-trained BERT [97] as well as a fully-connected layer for fine-tuning, to obtain the item's textual feature. More concretely, we average the hidden states corresponding to the special token [CLS] of the last two layers as the textual feature. We ultimately concatenate the visual and textual features of each item, and feed it into a learnable fully-connected layer to derive the final item embedding. Formally, we have

$$\begin{cases} \mathbf{e}_{v_k} = \text{ResNet}\left(v_k\right), \\ \mathbf{e}_{t_k} = f_t(\text{Bert}\left(t_k\right)_{[CLS]}), \\ \mathbf{e}_{m_k} = f_m\left(\mathbf{e}_{v_k} \| \mathbf{e}_{t_k}\right), \end{cases} \tag{6.2}$$

where f_t and f_m denote the learnable fully-connected layers. $\mathbf{e}_{v_k} \in \mathbb{R}^d$ and $\mathbf{e}_{t_k} \in \mathbb{R}^d$ are the d-dimensional visual and textual embedding of the item m_k, respectively. $\|$ is the concatenation operation, and $\mathbf{e}_{m_k} \in \mathbb{R}^d$ is the final initial embedding of the item m_k.

To regularize the semantic consistency between visual and textual embeddings of the same item, we adopt the following contrastive loss,

$$\begin{aligned} \mathcal{L}_{VTE} = &\sum_{i \neq k}^{N_m} max\{0, c - d(\mathbf{e}_{v_i}, \mathbf{e}_{t_i}) + d(\mathbf{e}_{v_i}, \mathbf{e}_{t_k})\} \\ &+ \sum_{i \neq k}^{N_m} max\{0, c - d(\mathbf{e}_{v_i}, \mathbf{e}_{t_i}) + d(\mathbf{e}_{v_k}, \mathbf{e}_{t_i})\}, \end{aligned} \tag{6.3}$$

where c is a margin hyperparameter and $d(\mathbf{a}, \mathbf{b}) = \mathbf{a}^T\mathbf{b}$.

Attribute Embedding. To mine the semantic content of each attribute, we also resort to the pre-trained BERT model with a learnable fully-connected layer, to derive its embedding. For each attribute entity a_l, its embedding can be defined as follows,

$$\mathbf{e}_{a_l} = f_a(\text{Bert}\,(a_l)_{[CLS]}), \tag{6.4}$$

where $\mathbf{e}_{a_l} \in \mathbb{R}^d$ stands for the initial embedding of the attribute entity a_l, and f_a denotes the fully-connected layer to fine-tune the embedding. Notably, considering that attributes are very short compared with the textual descriptions, we only adopt the representation of the special token [CLS] from the last layer of BERT as the attribute embedding.

Outfit Embedding. Each outfit is a set of fashion items and hence has no concrete content information. We thus derive the initial embedding of each outfit by aggregating all the embeddings of its composing items as,

$$\mathbf{e}_{o_j} = \text{Avg}(m_k | m_k \in \mathcal{M}_j), \tag{6.5}$$

where $\mathbf{e}_{o_j} \in \mathbb{R}^d$ denotes the initial embedding of the outfit o_j. Avg refers to the average pooling operation.

User Embedding. Analogous to the outfit, each user entity also has no straightforward content information. We hence fuse all the embeddings of his/her historically interacted and preferred outfits, to derive the user embedding as follows,

$$\mathbf{e}_{u_i} = \text{Avg}(o_j | o_j \in \mathcal{O}_i), \tag{6.6}$$

where $\mathbf{e}_{u_i} \in \mathbb{R}^d$ denotes the initial embedding of the user u_i.

6.3.2.2 Bi-directional Sequential Graph Convolution

To perform the heterogeneous graph learning and refine the embedding of each entity, we build a specific graph $\hat{\mathcal{G}}$ for each training triplet sample (u, o_+, o_-) according to the whole graph $\mathcal{G}$. In particular, we here temporally treat the whole graph $\mathcal{G}$ as a tree structure, whereby the user, outfit, item, and attribute entities are positioned in the 1st, 2nd, 3rd, and 4th levels, respectively, as shown in Figure 6.4. Then, the entity set of $\hat{\mathcal{G}}$ includes all the children entity nodes of these three entity nodes, *i.e.*, u, o_+, and o_-. We thereafter gather all the edges among these entities in the graph $\mathcal{G}$ as the edge set of $\hat{\mathcal{G}}$. To perform graph learning, the most straightforward way is to conduct the

conventional graph convolution operations, like GAT, which refine entity embeddings by simultaneously propagating information among entity nodes. Although feasible, we argue that this manner suffers from two key limitations. First, it fails to distinguish the types of neighbor nodes during information propagation. Second, as this graph has four kinds of entities, the graph convolution requires extremely powerful servers and its computational cost is relatively high, according to our preliminary attempts.

To address these issues, we propose to conduct the bi-directional sequential graph convolution to refine each node embedding. In particular, we decompose the entire graph into three subgraphs: user-outfit graph, outfit-item graph, and item-attribute graph. They are respectively denoted as $\hat{\mathcal{G}}^{UO}$, $\hat{\mathcal{G}}^{OI}$, and $\hat{\mathcal{G}}^{IA}$. Following that, we conduct graph convolution in two directions: upwards (*i.e.*, in the order of $\hat{\mathcal{G}}^{IA}, \hat{\mathcal{G}}^{OI}, \hat{\mathcal{G}}^{UO}$) and then downwards (*i.e.*, in the order of $\hat{\mathcal{G}}^{UO}, \hat{\mathcal{G}}^{OI}, \hat{\mathcal{G}}^{IA}$), as illustrated in Figure 6.4. In other words, in each training iteration, we conduct five subgraph convolution sequentially over the subgraphs $\hat{\mathcal{G}}^{IA}, \hat{\mathcal{G}}^{OI}, \hat{\mathcal{G}}^{UO}, \hat{\mathcal{G}}^{OI}, \hat{\mathcal{G}}^{IA}$. It is worth noting that once the entity embedding is updated by one subgraph convolution, it is then directly used at the next one, rather than waiting for the end of all subgraph learning. By doing so, we are able to timely utilize the messages passed by various entities and promote the model convergence.

In this work, we adopt the same graph convolution module for all subgraphs and both directions. We here take the graph convolution of the subgraph $\hat{\mathcal{G}}^{OI}$ as an example, and the rest follows the same procedure. Suppose that our graph convolution module has L layers. According to GAT [16], the l-th layer of graph convolution towards the outfit entity o in the subgraph $\hat{\mathcal{G}}^{OI}$ can be written as,

$$\begin{cases} \mathbf{e}_o^{l+1} = \mathbf{e}_o^l + \|_{k=1}^{M} \Big(\phi \Big(\sum\limits_{m_s \in \mathcal{N}_o} \alpha_{m_s}^{lk} \mathbf{W}_k^l \mathbf{e}_{m_s}^l \Big) \Big), l = 0, \cdots, L-1, \\ \alpha_{m_s}^{lk} = softmax(\eta(\boldsymbol{\gamma}_l^T [\mathbf{W}_k^l \mathbf{e}_o^l || \mathbf{W}_k^l \mathbf{e}_{m_s}^l])), \end{cases} \tag{6.7}$$

where $\mathcal{N}_o$ denotes the set of one-hop neighbors of the outfit entity o. The symbols $\mathbf{e}_o^l$ and $\mathbf{e}_{m_s}^l$ refer to the embedding of the outfit o and the item m_s derived by the l-th layer, respectively. $\boldsymbol{\gamma}_l$ and $\mathbf{W}_k^l$ are the to-be-learned parameters. M is the number of heads in the multi-head attention. η and ϕ are the LeakyReLU and Exponential Linear Unit (ELU) activation functions, respectively. The graph convolution towards each item entity m_s in the subgraph $\hat{\mathcal{G}}^{OI}$ can be defined in a similar way. Notably, $\mathbf{e}_o^0$ and $\mathbf{e}_{m_s}^0$ are the initial representations of the outfit entity o and item entity m_s for the convolution over the subgraph $\hat{\mathcal{G}}^{OI}$, respectively, which are different for different directions' subgraph convolution. In the upwards direction, they are yielded by the graph convolution over

the subgraph $\hat{\mathcal{G}}^{IA}$. While in the downwards one, they are produced by that over the subgraph $\hat{\mathcal{G}}^{UO}$.

We denote the refined user, outfit, item, and attribute embeddings as $\tilde{\mathbf{e}}_u$, $\tilde{\mathbf{e}}_o$, $\tilde{\mathbf{e}}_m$, and $\tilde{\mathbf{e}}_a$, respectively. We ultimately adopt the MLP to derive the final hash representation for each entity as $\mathbf{h}_x = MLP(\tilde{\mathbf{e}}_x)$, where $x = \{u, o, m, a\}$, and $\mathbf{h}_x \in \mathbb{R}^D$.

6.3.2.3 Hash Code Learning

To improve the efficiency of POR, we introduce the hash layer to learn the binary codes for user and outfit entities. Due to the non-differentiable problem over discrete hash functions, we adopt the *tanh* function to approximate the *sign* operation and derive the continuous hash codes as,

$$\begin{cases} \tilde{\mathbf{b}}_{u_i} = tanh(\beta \mathbf{h}_{u_i}), \\ \tilde{\mathbf{b}}_{o_j} = tanh(\beta \mathbf{h}_{o_i}), \end{cases} \tag{6.8}$$

where β is a scale parameter to ensure the suitable interval before *tanh* function. $\tilde{\mathbf{b}}_{u_i}$ and $\tilde{\mathbf{b}}_{o_j}$ refer to the continuous hash codes of the user u_i and outfit o_j, respectively.

6.3.3 Optimization

Towards optimization, we adopt the conventional BPR loss for user preference learning and design the dual similarity preserving regularization to enhance the hash code learning.

BPR Loss for User Preference Learning. Based on our training set of triplets, *i.e.*, $\mathcal{D}$, the BPR loss can be formulated as follows,

$$\begin{cases} \mathcal{L}_{BPR} = - \sum_{(u_i, o_j, o_k) \in \mathcal{D}} log(1 + exp(s_{ij} - s_{ik})), \\ s_{ij} = \tilde{\mathbf{b}}_{u_i}^T \tilde{\mathbf{b}}_{o_j}, \end{cases} \tag{6.9}$$

where s_{ij} stands for the preference of the user u_i towards the outfit o_j, calculated by the inner product of the continuous hash codes of the user u_i and outfit o_j. Intuitively, we expect that the user's preference towards the positive outfit o_j should be larger than that to the negative one o_k.

Dual Similarity Preserving Regularization. Similarity preserving regularization has proven to be effective in supervising the hash code learning by remaining the original

pairwise semantic similarity [108]. As each outfit is composed by a set of complementary items, these composing items can be seen as labels of the outfit entity. Similarly, each user prefers a few outfits, and thus each user entity can be analogously labeled by outfits. Based on these labels, we can derive the ground truth similarity matrix for the user and outfit entities. Formally, let $\mathbf{S}^o \in \mathbb{R}^{N_o \times N_o}$ and $\mathbf{S}^u \in \mathbb{R}^{N_u \times N_u}$ be the ground truth similarity matrices for outfit and user entities, respectively. In particular, the (i, j)-th entry $S^o_{ij} = 1$, if the outfits o_i and o_j share at least one composing item, otherwise $S^o_{ij} = 0$. Similarly, the (i, j)-th entry $S^u_{ij} = 1$, if both users u_i and u_j prefer the same outfit, otherwise $S^u_{ij} = 0$.

Afterwards, we deploy the dual similarity preserving regularization, with the goal of maximizing the Hamming distance between two entities whose semantic similarity is 0, while minimizing those with semantic similarity as 1. To be specific, inspired by [109, 110], we define the following regularizations,

$$\begin{cases} \mathcal{L}_{S_o} = -\sum\limits_{i,j=1}^{N_o} \Big(S^o_{ij}\phi^o_{ij} - log(1 + e^{\phi^o_{ij}}) \Big), \\ \mathcal{L}_{S_u} = -\sum\limits_{i,j=1}^{N_u} \Big(S^u_{ij}\phi^u_{ij} - log(1 + e^{\phi^u_{ij}}) \Big), \\ \phi^o_{ij} = cos(\tilde{\mathbf{b}}_{o_i}, \tilde{\mathbf{b}}_{o_j}), \\ \phi^u_{ij} = cos(\tilde{\mathbf{b}}_{u_i}, \tilde{\mathbf{b}}_{u_j}), \end{cases} \tag{6.10}$$

where ϕ^o_{ij} refers to the hash code based semantic similarity between outfits o_i and o_j, ϕ^u_{ij} is the one between users u_i and u_j, and $cos(\cdot, \cdot)$ denotes the cosine function. In a sense, we expect that the hash representation based semantic similarities approximate the ground truth ones.

Ultimately, we reach the final objective function as follows,

$$\mathcal{L} = \mathcal{L}_{BPR} + \alpha_1 \mathcal{L}_{VTE} + \alpha_2(\gamma \mathcal{L}_{S_o} + \mathcal{L}_{S_u}), \tag{6.11}$$

where α_1, α_2, and γ are the non-negative hyperparamters controlling the importance of the corresponding regularizations.

6.4 Experiment

In this section, we conducted experiments over the real-world dataset by answering the following research questions.

- **RQ1**: Does BiHGH outperform state-of-the-art baselines?

- **RQ2**: How does each component affect BiHGH?
- **RQ3**: How sensitive is BiHGH?
- **RQ4**: What is the intuitive performance of BiHGH?

6.4.1 Experimental Settings

In this part, we present the datasets, evaluation tasks, metrics, and the implementation details.

6.4.1.1 Dataset.

In general, there are three popular datasets for personalized outfit recommendation: Polyvore-630 [100], Polyvore-519 [100], and IQON-550 [111]. However, the former two datasets lack the attribute information of items. Therefore, we adopted the dataset IQON-550 that has rich attribute labels to evaluate our method in the context of POR. Derived from the dataset IQON3000 [12], this dataset consists of 550 users with 57,750 positive outfits, where each user historically interacted with 120 positive outfits. For each user, the 120 positive outfits are divided into three parts: 85, 15, and 20 outfits for training, validation, and testing, respectively. It is worth noting that we further split the training set into 2 chunks: 50% for building the whole heterogeneous graph $\mathcal{G}$, and 50% for creating the training triplets. For each positive user-outfit pair, IQON-550 also provides 10 negative outfits, randomly sampled from other users' positive outfits. In this work, we randomly selected one from the 10 negative samples, and constituted a triplet (user, positive outfit, negative outfit). Ultimately, we obtained 23,650 training and 8,250 validation triplets. As for testing, we adopted the original testing set provided by IQON-550, where for each user, there are 20 positive outfits with 200 negative ones for ranking. Each outfit in IQON-550 involves 3∼8 items from different categories.

6.4.1.2 Baselines.

To validate the effectiveness of our proposed BiHGH scheme, we chose the following baselines for comparison.

- **BPR-MF** [20]. This is a matrix factorization model equipped with BPR loss, which projects the users and outfits into a latent space, and uses BPR loss to regulate the user's relative preference over the positive and negative outfits.

TABLE 6.1: Performance comparison between our BiHGH and other baselines on IQON-550. The best results are in boldface, and the second best are underlined.

Method	Non-binary			Binary		
	AUC	MRR	NDCG	AUC	MRR	NDCG
BPR-MF	0.6587	0.4735	0.5953	0.6259	0.4203	0.5546
VBPR	0.7491	0.5305	0.6424	0.7242	0.4999	0.6253
VBPR-T	<u>0.7625</u>	<u>0.5463</u>	<u>0.6547</u>	0.7354	<u>0.5136</u>	<u>0.6293</u>
LPAE	0.7204	0.5180	0.6316	0.7052	0.4899	0.6101
HFGN	0.7423	0.5091	0.6264	0.7264	0.5117	0.6281
FHN	0.7360	0.4950	0.6157	0.7297	0.4889	0.6108
FHN-T	0.7494	0.5109	0.6281	<u>0.7426</u>	0.5027	0.6217
BiHGH	**0.7974**	**0.5778**	**0.6799**	**0.7933**	**0.5742**	**0.6771**

- **VBPR** [29]. It is a personalized fashion recommendation model based upon the matrix factorization framework, where the visual features of products are incorporated.
- **VBPR-T** [29]: This is extended from VBPR, which further incorporates the textual modality of products to enhance the item representation.
- **LPAE** [111]. This is a learnable personalized anchor embedding approach towards POR, which encodes the outfit with the self-attention mechanism, and models the user's preference with multiple anchors.
- **HFGN** [5]. This is a hierarchical fashion graph network devised for POR, which models the relationships among users, items, and outfits within a hierarchical graph.
- **FHN** [100]. This baseline aims to learn the binary hash code for each user and outfit towards efficient POR. In FHN, both visual and textual modalities of items are considered, and both BPR and VTE losses are jointly used.
- **FHN-T** [100]. Similar to VBPR-T, this baseline is extended from FHN, where the textual modality is incorporated.

It is worth noting that among these baselines, only FHN and FHN-T focus on learning hash codes of fashion entities, like ours.

6.4.1.3 Evaluation Tasks and Metrics.

Towards comprehensive evaluation, we investigated two evaluation configurations: non-binary and binary. Specifically, in the former case, for hashing based methods (*i.e.,*

TABLE 6.2: Ablation study results.

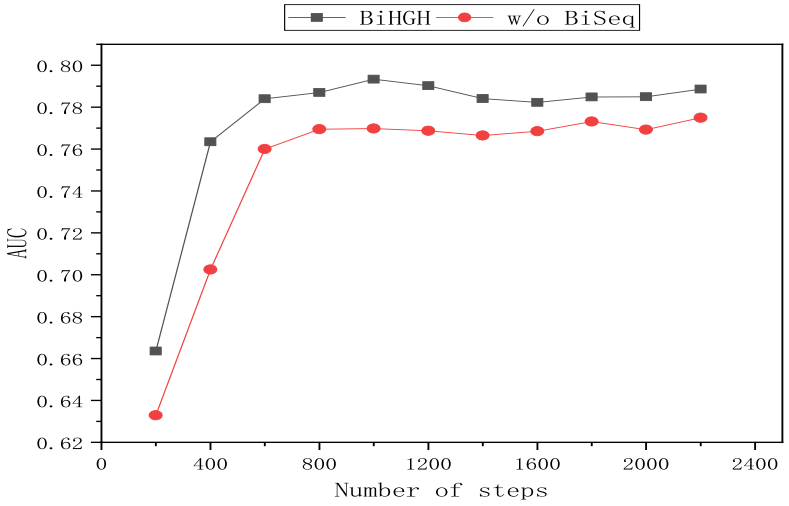
Method	Non-binary			Binary		
	AUC	MRR	NDCG	AUC	MRR	NDCG
w/o text	0.7911	0.5653	0.6704	0.7841	0.5578	0.6646
w/o image	0.7041	0.4842	0.6059	0.6978	0.4799	0.6025
w/o attribute	0.7911	0.5646	0.6698	0.7875	0.5675	0.6719
w/o DualSim	0.7902	0.5599	0.6658	0.7795	0.5335	0.6467
w/o VTE	0.7933	0.5686	0.6729	0.7850	0.5608	0.6669
w/o BiSeq	0.7825	0.5550	0.6622	0.7750	0.5537	0.6611
BiHGH	**0.7974**	**0.5778**	**0.6799**	**0.7933**	**0.5742**	**0.6771**

FHN, FHN-T, and ours), we used the continuous hash code of each user/outfit entity for evaluation. Regarding the non-hashing based methods, we judged them based upon their learned continuous entity representations. In the latter case, we directly imposed the *sign* operation to convert the continuous representations to binary hash codes for hashing based methods in the testing phase. As for the other non-hashing based baselines, we re-trained them by deploying the same *tanh* activation function over their fashion entity representations, and then tested them with the binary hash codes. For both configurations, we adopted two testing scenarios: binary preference prediction and personalized outfit recommendation. For the first testing scenario, we adopted the Area Under the ROC curve (AUC) [98] as the evaluation metric. Intuitively, AUC measures the probability that for a randomly selected pair of outfits for a give user, one positive and one negative, the predicted user preference score towards the positive outfit is higher than that towards the negative one. As for the personalized outfit recommendation, we adopted commonly used Mean Reciprocal Rank (MRR@K) [99] and Normalized Discounted Cumulative Gain (NDCG@K) [112, 113] as evaluation metrics. Specifically, MRR@K provides the insights into the ability of the model to return the positive outfit at top of the ranking list with K outfit candidates, and NDCG@K reflects the ranking positions of the positive outfits within the top K ranking list. In our work, we set $K = 10$.

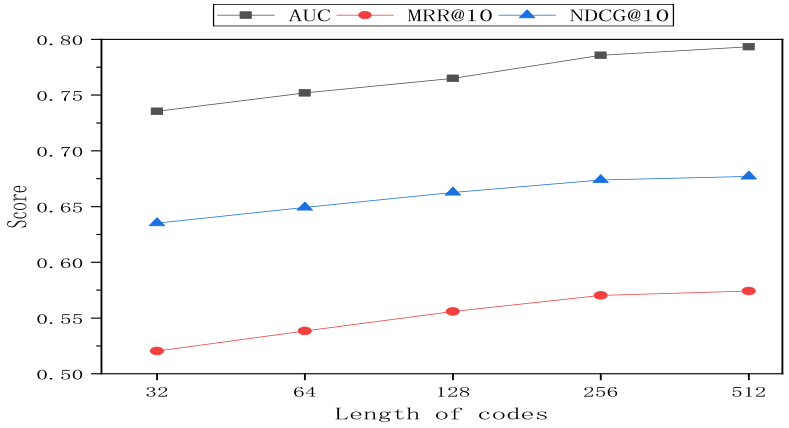
6.4.1.4 Implementation Details.

We utilized ResNet34 to derive the visual embedding of each item, whose final dimension is 512. Regarding the textual/attribute feature extraction, we resorted to BERT[2] for Japanese, since the dataset is in Japanese. The text/attribute feature outputted by BERT is a 768-dimensional vector. The GAT we adopted has 8 heads and 4 layers (*i.e.*, $M = 8$ and $L = 4$). The dimension of the hash code is set to 512. As to the

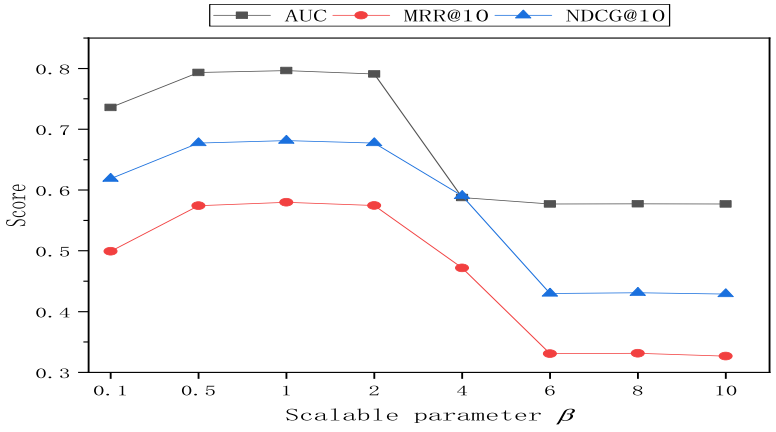
[2] https://huggingface.co/cl-tohoku/bert-base-japanese-char/tree/main.



(a) The number of steps



(b) The length of codes.



(c) The scale parameter β.

FIGURE 6.5: Sensitivity analysis of our model in terms of the (a) number of steps, (b) length of codes, and (c) scale parameter β.

optimization, we adopted the adaptive moment estimation method. The learning rate is warmed up to the peak value, which is set to 5e-5, in the first 6% steps, and then linearly decayed to 0. We used the grid search strategy to derive the optimal hyperparameters. Ultimately, the batch size is set to 4. The margin hyperparameter c in contrastive loss is set to 0.1 and the weight hyperparameters α_1, α_2, and γ are set to 1, 1, and 0.04, respectively. All the experiments are implemented by PyTorch over a server equipped with 4 A100-PCIE-40GB GPUs.

6.4.2 On Model Comparison

Table 6.1 shows the performance comparison among different approaches with respect to three metrics. From Table 6.1, we have the following observations.

- Our proposed BiHGH consistently outperforms all the baselines over all metrics across different evaluating configurations, which indicates the superiority of our scheme.
- As compared to the non-binary setting, the performance of all the methods drops slightly in the binary evaluation setting. This is reasonable as there should be inevitable information loss during the binarization. Nevertheless, our model decreases the least. This may be attributed to that the dual similarity preserving regularization, which is beneficial to prevent the information loss in the hash layer.
- Our method consistently surpasses all the baselines remarkably over all metrics under two settings, which implies the necessity of considering the attributes of items.

6.4.3 On Ablation Study

We conducted ablation study with the following derivatives.

- **w/o text and w/o image**: To study the impact of the textual description and image of each fashion item for POR, we removed the their embeddings, respectively.
- **w/o attribute**: To investigate the effect of the semantic attributes, we removed the attribute embeddings.
- **w/o DualSim**: To justify the effectiveness of dual similarity preserving loss during the hash code learning, we eliminated both $\mathcal{L}_{S_o}$ and $\mathcal{L}_{S_u}$ from Eq. 6.11.

- **w/o VTE**: To verify the VTE objective function, we removed the loss $\mathcal{L}_{VTE}$ from Eq. 6.11.
- **w/o BiSeq**: To explore the effect of the bi-directional sequential graph learning, we replaced it with the general GAT over the graph $\hat{\mathcal{G}}$ with four kinds of entities.

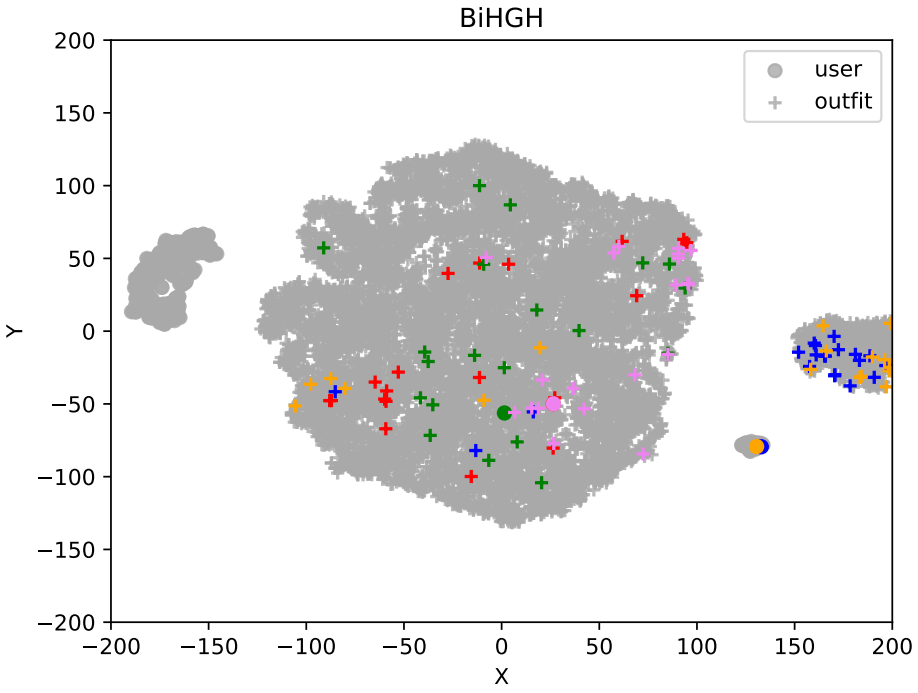
Table 6.2 summarizes the ablation study results. It can be seen that our model consistently outperforms all the above derivatives, which demonstrates the effectiveness of each component in our proposed BiHGH. Specifically, we have the following four detailed observations. 1) Both w/o text and w/o image perform inferior to BiHGH, which indicates that it is essential to consider both modalities of fashion items to boost the item representation learning. 2) w/o DualSim delivers the worse performance than our model, reflecting the benefit of the dual similarity preserving regularizations to prevent the information loss during the hash learning. 3) w/o VTE loss performs worse, which reveals that visual-semantic consistency regularizer largely benefits the hash code learning. 4) The performance of w/o BiSeq drops significantly, as compared to BiHGH, demonstrating that bi-directional sequential graph convolution is more powerful in transferring knowledge over the heterogeneous graph with various kinds of entities. Meanwhile, we compared the inference time of our model before and after applying the hash learning module, and the result was that adopting the hash module leads to a reduction in inference time of about 45%. This validates the effectiveness of the hash learning module in our framework.

6.4.4 On Sensitivity Analysis

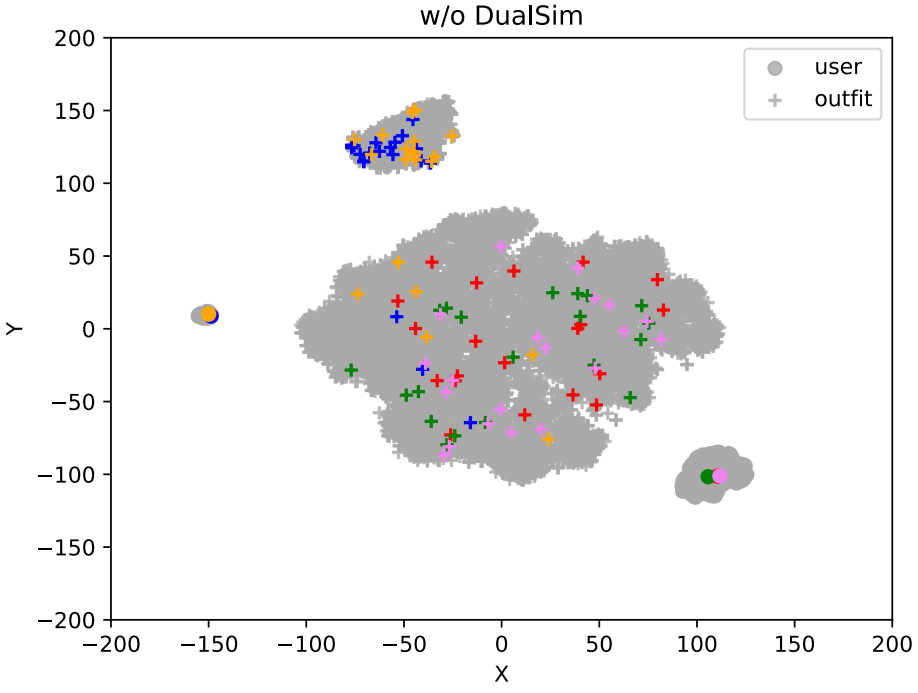
We also analyzed the sensitivity of our model regarding three key parameters: the number of training steps, the length D of hash codes, and the scale parameter β in the *sign* function approximation.

On the number of training steps. We comparatively plotted the detailed performance of our proposed BiHGH and the variant model w/o BiSeq on testing set at different training steps in Figure 6.5(a). We have two observations: 1) our model consistently outperforms w/o BiSeq at different training steps. And 2) our model converges faster as compared with w/o BiSeq. These observations validate the benefit of the bi-directional sequential graph convolution in promoting the entity representation learning and improving model convergence.

On the length of hash codes. Figure 6.5(b) shows our model's performance with variable code lengths, *i.e.*, {32, 64, 128, 256, 512} bits. As can be seen, with the increase



(a) BiHGH



(b) w/o DualSim

FIGURE 6.6: Visualization of the learned hash codes of our BiHGH and its variant w/o DualSim.

of the code length, the performance of our model becomes better. This is reasonable, because longer hash codes convey more information.

On the scale parameter β. We varied this scale parameter β among values of $\{0.1, 0.5, 1, 2, 4, 6, 8, 10\}$, and reported the corresponding performance in Figure 6.5(c). As can be seen, all the three metrics follow the same trend of first increasing until stable, and then sharply decreasing until stable again, with the increase of β. In particular, our model achieves the highest performance, when the scalar parameter β is set to 1. This implies that with the scale larger than 1, the elements of the continuous hash representations tend to be overly squeezed to 0 or 1, which leads to the severe information loss, and thus hurts the model performance.

6.4.5 On Case Study

To gain the intuitive understanding of our BiHGH, we conducted two case studies.

Firstly, we visualized the learned hash codes for users and outfits in the testing set by our model and its variant w/o DualSim with the tool of t-SNE [114]. The results are illustrated in Figure 6.6, whereby different colors indicate different users and their corresponding positive outfits. From Figure 6.6, we can see that our model is able to get the hash codes for the users and their preferred outfits closer than those obtained via w/o DualSim.

Secondly, we showed the top-4 ranking results of our method, and its derivatives w/o DualSim and w/o attribute for a given user in Figure 6.7, where the users' historical preferred outfits are also listed to facilitate the experimental result analysis[3]. As can be seen, our model outperforms its derivatives. Meanwhile, we noticed that as compared with w/o DualSim, our model is able to return more positive outfits that contain the same pair of shoes, highlighted in red boxes, which is preferred historically by the user. This confirms the effectiveness of our dual similarity preserving regularization. Furthermore, we observed that as compared with the w/o attribute, our model additionally returns a positive outfit at the fourth place. One possible explanation is that our model is able to uncover the user's preference over bottom items with the attribute "jeans", while the w/o attribute derivative fails. This also validates the benefit of incorporating the attribute information into the context of POR.

[3]Due to the limited space, we did not provide textual descriptions and attributes of items.

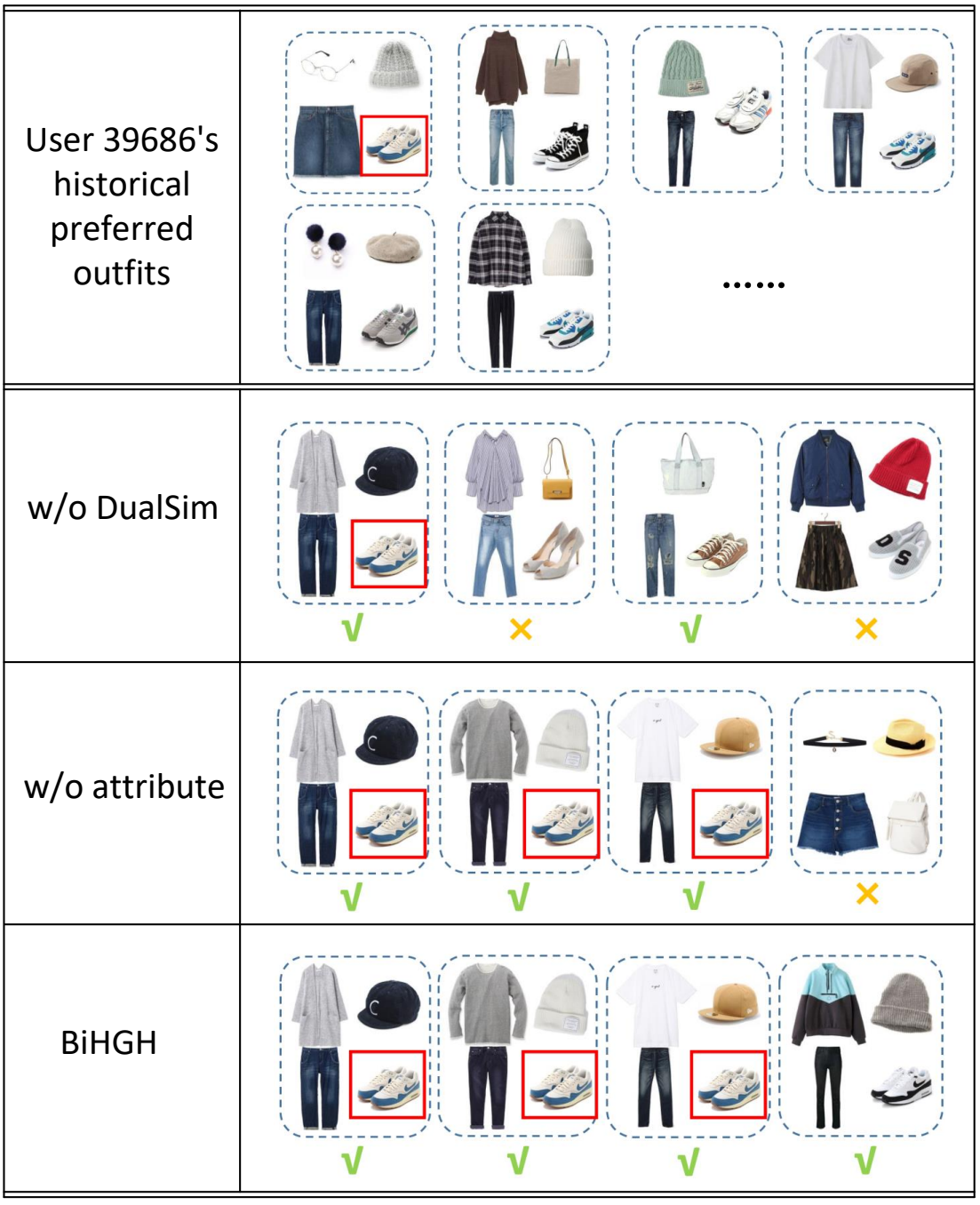


FIGURE 6.7: Illustration of the POR ranking results obtained by our BiHGH, w/o similarity loss, and w/o attribute derivative.

6.5 Summary

In this chapter, we present an efficient personalized outfit recommendation scheme with Bi-directional heterogeneous graph hash learning, named BiHCH. Different from previous studies, we comprehensively unify the four types of entities (*i.e.*, users, outfits, items and attributes) and their relations via a heterogeneous four-partite graph. Moveover, we creatively devise a bi-directional graph convolution algorithm to sequentially transfer

knowledge via repeating upwards and downwards convolution to enhance the representation of users and outfits. We ultimately design the BPR loss for the user preference learning and dual similarity preserving regularization to prevent the information loss during the hash learning, respectively. Extensive experiments on the benchmark dataset demonstrate the superior performance and efficiency of BiHCH over existing methods. The ablation study also verifies the advantages of our bi-directional graph convolution over conventional ones, as well as the benefit of incorporating the dual similarity preserving regularization.

Chapter 7

Conclusion and Future Work

In this thesis, we study the task of compatibility-oriented fashion recommendation, and drawn the following conclusions:

- We have presented a novel cooperation learning model from multiple social networks. It is capable of jointly characterizing the source consistency, complementarity, and confidence within a unified model. In particular, we enforce the predicted results from consistent parts among sources to be similar and filter out the task-unrelated information in the complementary parts via group lasso. Meanwhile, we learn the source confidence from the data instead of painstaking tuning. Besides, we have theoretically relaxed the nonsmooth objective function to a smooth one and derived its analytical solution. We verified our model on the application of user interest inference from multiple social networks. By conducting experiments on the real-world datasets, it is validated that our proposed model yields significant gains in user interest inference. In particular, we verified that the complementary factor, which was not explicitly studied in the previous work, plays a pivotal role in enhancing the learning performance, especially, when combined with source consistency. It should be emphasized that our model is also applicable to other applications such as occupation inference from multiple networks.

- We present a novel partially supervised compatibility modeling, named PS-OCM, which consists of three key components: 1) partially supervised attribute embedding learning; 2) disentangled completeness regularization; and 3) hierarchical outfit compatibility modeling. In particular, we first present a partially supervised disentangled learning method to disentangle the visual representation of each item into several attribute-level embeddings. In addition, we devise the disentangled completeness regularization to prevent the information loss during disentanglement. Finally, we design a hierarchical graph convolutional network that jointly

performs the attribute- and item-level compatibility modeling. Extensive experiments have been conducted on a real-world dataset with two popular tasks: the outfit compatibility prediction and fillin- the-blank. The encouraging experiment results validate the superiority of our proposed model and the importance of its each component. In addition, we found that our PS-OCM is not sensitive to the number of items in the outfit, and removing each attribute, including the introduced residual one, from the embedding disentanglement will hurt the model's performance. This shows that each attribute could affect the outfit compatibility modeling to some extent.

- We solve the personalized fashion compatibility modeling problem by organizing the various fashion entities and relations into a unified heterogeneous graph, and present a novel metapathguided personalized compatibility modeling scheme to learn entity embeddings. Extensive experiments have been conducted on the public dataset IQON3000, which demonstrates the superiority of our model over existing methods. The ablation study verifies the importance of each key module, like jointly considering the text, image, and attribute information of items towards PFCM, the constrastive regularization as well as using a non-position transformer to fulfil the semantic-enhanced embedding fusion. In addition, experimental results show that our model is insensitive to the numbers of transformer and GAT layers, which enables the model to perform well with fewer parameters.

- We present an efficient personalized outfit recommendation scheme with Bi-directional heterogeneous graph hash learning, named BiHGH. Different from previous studies, we comprehensively unify the four types of entities (i.e., users, outfits, items and attributes) and their relations via a heterogeneous four-partite graph. Moreover, we creatively devise a bi-directional graph convolution algorithm to sequentially transfer knowledge via repeating upwards and downwards convolution to enhance the representation of users and outfits. We ultimately design the BPR loss for the user preference learning and dual similarity preserving regularization to prevent the information loss during the hash learning, respectively. Extensive experiments on the benchmark dataset demonstrate the superior performance and efficiency of BiHGH over existing methods. The ablation study also verifies the advantages of our bi-directional graph convolution over conventional ones, as well as the benefit of incorporating the dual similarity preserving regularization.

In the future, I plan to move forward from the following two directions.

- **Unbiased Fashion Compatibility Modeling.** Current mainstream datasets have some bias regarding the item categories. For example, the item categories of

pants and jeans occur very frequently in the positive outfits, while categories of kimono and slippers appear few times. Therefore, models trained on such datasets may be misled to fit the spurious correlations between the item categories and the compatibility label of the outfit, ignoring the other factors (*e.g.,* color and pattern) that affect the outfit compatibility. In the future, I plan to use the causal inference technique and introduce the causal graph to inspect the causal relationships between multiple modalities (*i.e.,* visual, textual and category modality) and the outfit compatibility label, to fulfill the unbiased outfit compatibility modeling.

- **Try-on Enhanced Fashion Compatibility Modeling.** Existing methods on fashion compatibility focus on modeling the compatibility relationship among discrete items in an outfit but overlook the fact that people usually evaluate the matching degree of a given outfit based on the try-on effect. Therefore, it is promising to incorporate the try-on effect to strengthen the outfit compatibility evaluation. In the future, I plan to comprehensively analyze the outfit compatibility from both the discrete collocation and unified try-on angles. The key is how to derive the try-on effect of the given outfit based on its composing items' multimodal content information and how to fuse the two angles seamlessly.

- **Deployment on Resource-constrained IoT Devices.** The wide use of the Internet of Things (IoT) makes it possible to create smart life. Obviously, intelligent fashion recommendation is crucial to improve people's life quality. The IoT comprises networks of interconnected objects, computing systems, and physical or mechanical machines that are capable of sending and receiving data through a network without human intervention. It is feasible to deploy the proposed compatibility-oriented fashion recommendation methods on the existing IoT devices. Even though IoT devices are commonly resource-constrained, we can resort to advanced model lightweight solutions, such as knowledge distillation and network pruning, to fulfill the deployment of our proposed methods.

Bibliography

[1] Xianjing Han, Xuemeng Song, Jianhua Yin, Yinglong Wang, and Liqiang Nie. Prototype-guided attribute-wise interpretable scheme for clothing matching. In *Proceedings of the international ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pages 785–794. ACM, 2019.

[2] Jinhuan Liu, Xuemeng Song, Liqiang Nie, Tian Gan, and Jun Ma. An end-to-end attention-based neural model for complementary clothing matching. *ACM Transactions on Multimedia Computing, Communications and Applications,*, 15 (4):114:1–114:16, 2020.

[3] Xintong Han, Zuxuan Wu, Yu-Gang Jiang, and Larry S. Davis. Learning fashion compatibility with bidirectional lstms. In *Proceedings of the ACM International Conference on Multimedia*, pages 1078–1086. ACM, 2017.

[4] Xue Dong, Jianlong Wu, Xuemeng Song, Hongjun Dai, and Liqiang Nie. Fashion compatibility modeling through a multi-modal try-on-guided scheme. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 771–780. ACM, 2020.

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

[6] Huijing Zhan, Jie Lin, Kenan Emir Ak, Boxin Shi, Ling-Yu Duan, and Alex C. Kot. a^3-fkg: Attentive attribute-aware fashion knowledge graph for outfit preference prediction. *IEEE Trans. Multim.*, 24:819–831, 2022.

[7] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *Proceedings of the International Conference on Learning Representations*, pages 1–12. OpenReview.net, 2018.

[8] Xiaodan Liang, Liang Lin, Wei Yang, Ping Luo, Junshi Huang, and Shuicheng Yan. Clothes co-parsing via joint image segmentation and labeling with application to clothing retrieval. volume 18, pages 1175–1186. IEEE, 2016.

[9] Xuemeng Song, Fuli Feng, Jinhuan Liu, Zekun Li, Liqiang Nie, and Jun Ma. Neurostylist: Neural compatibility modeling for clothing matching. In *Proceedings of the ACM International Conference on Multimedia*, pages 753–761. ACM, 2017.

[10] Jinhuan Liu, Xuemeng Song, Zhumin Chen, and Jun Ma. Neural fashion experts: I know how to make the complementary clothing matching. *Neurocomputing*, 359 (24):249–263, 2019.

[11] Xiaoling Gu, Yongkang Wong, Pai Peng, Lidan Shou, Gang Chen, and Mohan S Kankanhalli. Understanding fashion trends from street photos via neighbor-constrained embedding learning. In *Proceedings of the ACM International Conference on Multimedia*, pages 190–198. ACM, 2017.

[12] Xuemeng Song, Xianjing Han, Yunkai Li, Jingyuan Chen, Xin-Shun Xu, and Liqiang Nie. GP-BPR: personalized compatibility modeling for clothing matching. In *Proceedings of the ACM International Conference on Multimedia*, pages 320–328. ACM, 2019.

[13] Wen Chen, Pipei Huang, Jiaming Xu, Xin Guo, Cheng Guo, Fei Sun, Chao Li, Andreas Pfadler, Huan Zhao, and Binqiang Zhao. Pog: personalized outfit generation for fashion recommendation at alibaba ifashion. In *Proceedings of the International ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2662–2670. ACM, 2019.

[14] Haokun Wen, Xuemeng Song, Xin Yang, Yibing Zhan, and Liqiang Nie. Comprehensive linguistic-visual composition network for image retrieval. In *Proceedings of the international ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pages 1369–1378. ACM, 2021.

[15] Xue Dong, Xuemeng Song, Fuli Feng, Peiguang Jing, Xin-Shun Xu, and Liqiang Nie. Personalized capsule wardrobe creation with garment and user modeling. In *Proceedings of the ACM International Conference on Multimedia*, pages 302–310. ACM, 2019.

[16] Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–52. ACM, 2015.

[17] Yujie Lin, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Jun Ma, and Maarten de Rijke. Explainable outfit recommendation with joint outfit matching and comment generation. *IEEE Transactions on Knowledge and Data Engineering*, 32(8): 1502–1516, 2020.

[18] Zeyu Cui, Zekun Li, Shu Wu, Xiaoyu Zhang, and Liang Wang. Dressing as a whole: Outfit compatibility learning based on node-wise graph neural networks. In *Porceedings of the World Wide Web Conference*, pages 307–317. ACM, 2019.

[19] Guillem Cucurull, Perouz Taslakian, and David Vázquez. Context-aware visual compatibility prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12617–12626. IEEE, 2019.

[20] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: bayesian personalized ranking from implicit feedback. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, pages 452–461. AUAI Press, 2009.

[21] Mariya I. Vasileva, Bryan A. Plummer, Krishna Dusad, Shreya Rajpal, Ranjitha Kumar, and David A. Forsyth. Learning type-aware embeddings for fashion compatibility. In *European Conference on Computer Vision*, pages 405–421. Springer, 2018.

[22] Xun Yang, Yunshan Ma, Lizi Liao, Meng Wang, and Tat-Seng Chua. Transnfcm: Translation-based neural fashion compatibility modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 403–410. AAAI Press, 2019.

[23] Jinhuan Liu, Xuemeng Song, Zhaochun Ren, Liqiang Nie, Zhaopeng Tu, and Jun Ma. Auxiliary template-enhanced generative compatibility modeling. In *Proceedings of Joint Conference on Artificial Intelligence*, pages 3508–3514. AAAI Press, 2020.

[24] Dikshant Sagar, Jatin Garg, Prarthana Kansal, Sejal Bhalla, Rajiv Ratn Shah, and Yi Yu. PAI-BPR: personalized outfit recommendation scheme with attribute-wise interpretability. In *IEEE International Conference on Multimedia Big Data*, pages 221–230. IEEE, 2020.

[25] Guang-Lu Sun, Zhi-Qi Cheng, Xiao Wu, and Qiang Peng. Personalized clothing recommendation combining user social circle and fashion style consistency. *Multimedia tools and application*, 77(14):17731–17754, 2018.

[26] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. KGAT: knowledge graph attention network for recommendation. In *Proceedings of the*

25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 950–958. ACM, 2019.

[27] Xuewen Yang, Dongliang Xie, Xin Wang, Jiangbo Yuan, Wanying Ding, and Pengyun Yan. Learning tuple compatibility for conditional outfit recommendation. In *Proceedings of the International ACM Conference on Multimedia*, pages 2636–2644. ACM, 2020.

[28] Haoyu Zhang, Meng Liu, Zan Gao, Xiaoqiang Lei, Yinglong Wang, and Liqiang Nie. Multimodal dialog system: Relational graph-based context-aware question understanding. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 695–703. ACM, 2021.

[29] Ruining He and Julian J. McAuley. VBPR: visual bayesian personalized ranking from implicit feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 144–150. AAAI Press, 2016.

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

[31] Wenhui Yu, Huidi Zhang, Xiangnan He, Xu Chen, Li Xiong, and Zheng Qin. Aesthetic-based clothing recommendation. In *Proceedings of the ACM International Conference on World Wide Web Conference*, pages 649–658. ACM, 2018.

[32] Min Hou, Le Wu, Enhong Chen, Zhi Li, Vincent W. Zheng, and Qi Liu. Explainable fashion recommendation: A semantic attribute region guided approach. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 4681–4688. ijcai.org, 2019.

[33] Wen Chen, Pipei Huang, Jiaming Xu, Xin Guo, Cheng Guo, Fei Sun, Chao Li, Andreas Pfadler, Huan Zhao, and Binqiang Zhao. POG: personalized outfit generation for fashion recommendation at alibaba ifashion. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery*, pages 2662–2670. ACM, 2019.

[34] Yusan Lin, Maryam Moosaei, and Hao Yang. Outfitnet: Fashion outfit recommendation with attention-based multiple instance learning. In *The Web Conference*, pages 77–87. ACM, 2020.

[35] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826. IEEE, 2016.

[36] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[37] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.

[38] Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard H. Hovy, and Eric P. Xing. Harnessing deep neural networks with logic rules. In *Proceedings of the the Annual Meeting of the Association for Computational Linguistics*. The Association for Computer Linguistics, 2016.

[39] Peng Zhang, Li Su, Liang Li, BingKun Bao, Pamela C. Cosman, Guorong Li, and Qingming Huang. Training efficient saliency prediction models with knowledge distillation. In *Proceedings of the ACM International Conference on Multimedia*, pages 512–520. ACM, 2019.

[40] Xianjing Han, Xuemeng Song, Yiyang Yao, Xin-Shun Xu, and Liqiang Nie. Neural compatibility modeling with probabilistic knowledge distillation. *IEEE Transactions on Image Processing*, 29:871–882, 2020.

[41] Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. Deep mutual learning. In *IEEE Conference on Computer Vision and Pattern Recognition,*, pages 4320–4328. IEEE, 2018.

[42] Hao Luo, Wei Jiang, Xuan Zhang, Xing Fan, Jingjing Qian, and Chi Zhang. Alignedreid++: Dynamically matching local information for person re-identification. *Pattern Recognition.*, 94:53–61, 2019.

[43] Yunpeng Zhai, Qixiang Ye, Shijian Lu, Mengxi Jia, Rongrong Ji, and Yonghong Tian. Multiple expert brainstorming for domain adaptive person re-identification. In *Proceedings of the European Conference on Computer Vision*, pages 594–611. Springer, 2020.

[44] Yixiao Ge, Dapeng Chen, and Hongsheng Li. Mutual mean-teaching: Pseudo label refinery for unsupervised domain adaptation on person re-identification. In *International Conference on Learning Representations*, pages 1–12. OpenReview.net, 2020.

[45] Qianming Xue, Wei Zhang, and Hongyuan Zha. Improving domain-adapted sentiment classification by deep adversarial mutual learning. In *The AAAI Conference on Artificial Intelligence,*, pages 9362–9369. AAAI Press, 2020.

[46] Wonpyo Park, Wonjae Kim, Kihyun You, and Minsu Cho. Diversified mutual learning for deep metric learning. In *Proceedings of the European Conference on Computer Vision*, pages 709–725. Springer, 2020.

[47] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9. IEEE, 2015.

[48] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778. IEEE Computer Society, 2016.

[49] Meng Liu, Liqiang Nie, Meng Wang, and Baoquan Chen. Towards micro-video understanding by joint sequential-sparse modeling. In *Proceedings of the ACM International Conference on Multimedia*, pages 970–978. ACM, 2017.

[50] Yuxiao Chen, Jianbo Yuan, Quanzeng You, and Jiebo Luo. Twitter sentiment analysis via bi-sense emoji embedding and attention-based LSTM. In *Proceedings of the ACM International Conference on Multimedia*, pages 117–125. ACM, 2018.

[51] Zhiwei Jin, Juan Cao, Han Guo, Yongdong Zhang, and Jiebo Luo. Multimodal fusion with recurrent neural networks for rumor detection on microblogs. In *Proceedings of the ACM on Multimedia Conference*, pages 795–816. ACM, 2017.

[52] Reuben Tan, Mariya I. Vasileva, Kate Saenko, and Bryan A. Plummer. Learning similarity conditions without explicit supervision. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10372–10381. IEEE, 2019.

[53] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.

[54] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*, pages 1–15. OpenReview.net, 2015.

[55] Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, and Wenwu Zhu. Disentangled graph convolutional networks. In *Proceedings of International Conference on Machine Learning,*, volume 97, pages 4212–4221. PMLR, 2019.

[56] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. Learning disentangled representations for recommendation. In *Advances in Neural Information Processing Systems*, pages 5712–5723, 2019.

[57] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. Explicit factor models for explainable recommendation based on phrase-level

sentiment analysis. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 83–92. ACM, 2014.

[58] Liqian Ma, Qianru Sun, Stamatios Georgoulis, Luc Van Gool, Bernt Schiele, and Mario Fritz. Disentangled person image generation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 99–108. IEEE, 2018.

[59] Hao Chen, Yongjian Deng, Youfu Li, Tzu-Yi Hung, and Guosheng Lin. RGBD salient object detection via disentangled cross-modal fusion. *IEEE Transactions on Image Processing*, 29:8407–8416, 2020.

[60] Fu-En Yang, Jing-Cheng Chang, Chung-Chi Tsai, and Yu-Chiang Frank Wang. A multi-domain and multi-modal representation disentangler for cross-domain image manipulation and classification. *IEEE Transactions on Image Processing*, 29:2795–2807, 2020.

[61] Linmei Hu, Siyong Xu, Chen Li, Cheng Yang, Chuan Shi, Nan Duan, Xing Xie, and Ming Zhou. Graph neural news recommendation with unsupervised preference disentanglement. In *Proceedings of the Association for Computational Linguistics*, pages 4255–4264. ACL, 2020.

[62] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. Disentangled graph collaborative filtering. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1001–1010. ACM, 2020.

[63] Na Zheng, Xuemeng Song, Qingying Niu, Xue Dong, Yibing Zhan, and Liqiang Nie. Collocation and try-on network: Whether an outfit is compatible. In *Proceedings of the International ACM Conference on Multimedia*, pages 309–317. ACM, 2021.

[64] Weili Guan, Haokun Wen, Xuemeng Song, Chung-Hsing Yeh, Xiaojun Chang, and Liqiang Nie. Multimodal compatibility modeling via exploring the consistent and complementary correlations. In *Proceedings of the International ACM Conference on Multimedia*, pages 2299–2307. ACM, 2021.

[65] M. Gori, G. Monfardini, and F. Scarselli. A new model for learning in graph domains. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, pages 729–734. IEEE, 2005.

[66] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard S. Zemel. Gated graph sequence neural networks. In *International Conference on Learning Representations*. OpenReview.net, 2016.

[67] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, pages 1–15. OpenReview.net, 2017.

[68] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034. Curran Associates Inc., 2017.

[69] Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat-Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *IEEE International Conference on Computer Vision*, pages 2272–2281. IEEE, 2019.

[70] Lei Zhong, Juan Cao, Qiang Sheng, Junbo Guo, and Ziang Wang. Integrating semantic and structural information with graph convolutional network for controversy detection. pages 515–526. ACL, 2020.

[71] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *Proceedings of SIGIR Conference on Research and Development in Information Retrieval*, pages 165–174. ACM, 2019.

[72] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6857–6866. IEEE Computer Society, 2018.

[73] Tiancheng Sun, Yulong Wang, Jian Yang, and Xiaolin Hu. Convolution neural networks with two pathways for image style recognition. *IEEE Transactions on Image Processing*, 26(9):4102–4113, 2017.

[74] Yupeng Hu, Meng Liu, Xiaobin Su, Zan Gao, and Liqiang Nie. Video moment localization via deep cross-modal hashing. *IEEE Transactions on Image Processing*, 30:4667–4677, 2021.

[75] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations*. OpenReview.net, 2016.

[76] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *Proceedings of the International Conference on Learning Representations*, pages 45–61. OpenReview.net, 2018.

[77] Tianyu Su, Xuemeng Song, Na Zheng, Weili Guan, Yan Li, and Liqiang Nie. Complementary factorization towards outfit compatibility modeling. In *Proceedings of the ACM International Conference on Multimedia*, pages 4073–4081. ACM, 2021.

[78] Xuemeng Song, Shi-Ting Fang, Xiaolin Chen, Yinwei Wei, Zhongzhou Zhao, and Liqiang Nie. Modality-oriented graph learning toward outfit compatibility modeling. In *Proceedings of the IEEE Transactions on Multimedia*, pages 1–1. IEEE, 2021.

[79] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9992–10002. IEEE, 2021.

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

[81] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. LINE: large-scale information network embedding. In *Proceedings of the World Wide Web Conference*, pages 1067–1077. ACM, 2015.

[82] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the International ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 855–864. ACM, 2016.

[83] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. In *Proceedings of the International Conference on Learning Representations*, pages 1–17. OpenReview.net, 2020.

[84] Sami Abu-El-Haija, Amol Kapoor, Bryan Perozzi, and Joonseok Lee. N-GCN: multi-scale graph convolution for semi-supervised node classification. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 841–851. AUAI Press, 2019.

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

[86] Huan Zhao, Quanming Yao, Jianda Li, Yangqiu Song, and Dik Lun Lee. Meta-graph based recommendation fusion over heterogeneous information networks. In *Proceedings of the International ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 635–644. ACM, 2017.

[87] Yizhou Sun and Jiawei Han. Mining heterogeneous information networks: principles and methodologies. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 3(2):1–159, 2012.

[88] Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the International ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 135–144. ACM, 2017.

[89] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations*, pages 1–12. OpenReview.net, 2013.

[90] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and Philip S. Yu. Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 31(2):357–370, 2019.

[91] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. Heterogeneous graph neural network. In *Proceedings of the International ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 793–803. ACM, 2019.

[92] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. MAGNN: metapath aggregated graph neural network for heterogeneous graph embedding. In *Proceedings of the World Wide Web Conference*, pages 2331–2341. ACM, 2020.

[93] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S. Yu. Heterogeneous graph attention network. In *Proceedings of the World Wide Web Conference*, pages 2022–2032. ACM, 2019.

[94] Jintao Zhang and Quan Xu. Attention-aware heterogeneous graph neural network. *Big Data Mining and Analytics*, 4(4):233–241, 2021.

[95] Yuying Xing, Zhao Li, Pengrui Hui, Jiaming Huang, Xia Chen, Long Zhang, and Guoxian Yu. Link inference via heterogeneous multi-view graph neural networks. In *Proceedings of the International Conference on Database Systems for Advanced Applications*, pages 698–706. Springer, 2020.

[96] Di Jin, Cuiying Huo, Chundong Liang, and Liang Yang. Heterogeneous graph neural network via attribute completion. In *Proceedings of the World Wide Web Conference*, pages 391–400. ACM, 2021.

[97] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In

Proceedings of the Association for Computational Linguistics, pages 4171–4186. ACL, 2019.

[98] Hanwang Zhang, Zheng-Jun Zha, Yang Yang, Shuicheng Yan, Yue Gao, and Tat-Seng Chua. Attribute-augmented semantic hierarchy: towards bridging semantic gap and intention gap in image retrieval. In *Proceedings of the ACM International Conference on Multimedia*, pages 33–42. ACM, 2013.

[99] Lu Jiang, Shoou-I Yu, Deyu Meng, Yi Yang, Teruko Mitamura, and Alexander G. Hauptmann. Fast and accurate content-based semantic search in 100m internet videos. In *Proceedings of the International ACM Conference on Multimedia*, pages 49–58. ACM, 2015.

[100] Zhi Lu, Yang Hu, Yunchao Jiang, Yan Chen, and Bing Zeng. Learning binary code for personalized fashion recommendation. In *Conference on Computer Vision and Pattern Recognition,*, pages 10562–10570. IEEE, 2019.

[101] Fartash Faghri, David J. Fleet, Jamie Ryan Kiros, and Sanja Fidler. VSE++: improving visual-semantic embeddings with hard negatives. In *British Machine Vision Conference*, pages 12–12. BMVA Press, 2018.

[102] Jorge Sánchez and Florent Perronnin. High-dimensional signature compression for large-scale image classification. In *the Conference on Computer Vision and Pattern Recognition,*, pages 1665–1672. IEEE, 2011.

[103] Andrea Vedaldi and Andrew Zisserman. Sparse kernel approximations for efficient classification and detection. In *Conference on Computer Vision and Pattern Recognition,*, pages 2320–2327, 2012.

[104] Haomiao Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. Deep supervised hashing for fast image retrieval. In *Conference on Computer Vision and Pattern Recognition*, pages 2064–2072. IEEE, 2016.

[105] Han Zhu, Mingsheng Long, Jianmin Wang, and Yue Cao. Deep hashing network for efficient similarity retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2415–2421. AAAI Press, 2016.

[106] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Philip S. Yu. Hashnet: Deep learning to hash by continuation. In *International Conference on Computer Vision*, pages 5609–5618. IEEE, 2017.

[107] Ke Zhou and Hongyuan Zha. Learning binary codes for collaborative filtering. In *The ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 498–506. ACM, 2012.

[108] Hui Cui, Lei Zhu, Jingjing Li, Yang Yang, and Liqiang Nie. Scalable deep hashing for large-scale social image retrieval. *IEEE Trans. Image Process.*, 29:1271–1284, 2020.

[109] Changchang Sun, Xuemeng Song, Fuli Feng, Wayne Xin Zhao, Hao Zhang, and Liqiang Nie. Supervised hierarchical cross-modal hashing. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 725–734. ACM, 2019.

[110] Jingdong Wang, Ting Zhang, Jingkuan Song, Nicu Sebe, and Heng Tao Shen. A survey on learning to hash. *Trans. Pattern Anal. Mach. Intell.*, 40(4):769–790, 2018.

[111] Zhi Lu, Yang Hu, Yan Chen, and Bing Zeng. Personalized outfit recommendation with learnable anchors. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 12722–12731. IEEE, 2021.

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

[113] Zhengzhong Zhou, Xiu Di, Wei Zhou, and Liqing Zhang. Fashion sensitive clothing recommendation using hierarchical collocation model. In *Proceedings of the International ACM Conference on Multimedia*, pages 1119–1127. ACM, 2018.

[114] Paulo E. Rauber, Alexandre X. Falcão, and Alexandru C. Telea. Visualizing time-dependent data using dynamic t-sne. In Enrico Bertini, Niklas Elmqvist, and Thomas Wischgoll, editors, *Eurographics Conference on Visualization*, pages 73–77. Eurographics Association, 2016.