



Personalized Fashion Compatibility Modeling via Metapath-guided Heterogeneous Graph Learning

Weili Guan
Monash University
Melbourne, Australia
weili.guan@monash.edu

Fangkai Jiao
Shandong University
Qingdao, China
jiaofangkai@hotmail.com

Xuemeng Song*
Shandong University
Qingdao, China
sxmusc@gmail.com

Haokun Wen
Shandong University
Qingdao, China
whenhaokun@gmail.com

Chung-Hsing Yeh
Monash University
Melbourne, Australia
chunghsing.yeh@monash.edu

Xiaojun Chang*
University of Technology Sydney
Sydney, Australia
cxj273@gmail.com

ABSTRACT

Fashion Compatibility Modeling (FCM) is a new yet challenging task, which aims to automatically access the matching degree among a set of complementary items. Most of existing methods evaluate the fashion compatibility from the common perspective, but overlook the user's personal preference. Inspired by this, a few pioneers study the Personalized Fashion Compatibility Modeling (PFCM). Despite their significance, these PFCM methods mainly concentrate on the user and item entities, as well as their interactions, but ignore the attribute entities, which contain rich semantics. To address this problem, we propose to fully explore the related entities and their relations involved in PFCM to boost the PFCM performance. This is, however, non-trivial due to the heterogeneous contents of different entities, embeddings for new users, and various high-order relations. Towards these ends, we present a novel metapath-guided personalized fashion compatibility modeling, dubbed as MG-PFCM. 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). Thereafter, we design a multi-modal content-oriented user embedding module to learn user representations by inheriting the contents of their interacted items. Meanwhile, we define the user-oriented and item-oriented metapaths, and perform the metapath-guided heterogeneous graph learning to enhance the user and item embeddings. In addition, we introduce the contrastive regularization to improve the model performance. We conduct extensive experiments on the real-world benchmark dataset, which verifies the superiority of our proposed scheme over several cutting-edge baselines. As a byproduct, we have released our source codes to benefit other researchers.

*This is the corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGIR '22, July 11–15, 2022, Madrid, Spain

© 2022 Association for Computing Machinery.

ACM ISBN 978-1-4503-8732-3/22/07...\$15.00

<https://doi.org/10.1145/3477495.3532038>



Figure 1: Examples of users' outfit compositions.

CCS CONCEPTS

• Information systems → Retrieval tasks and goals.

KEYWORDS

Personalized Compatibility Modeling, Heterogeneous Graph Neural Networks, Metapath-guided Graph Learning

ACM Reference Format:

Weili Guan, Fangkai Jiao, Xuemeng Song, Haokun Wen, Chung-Hsing Yeh, and Xiaojun Chang*. 2022. Personalized Fashion Compatibility Modeling via Metapath-guided Heterogeneous Graph Learning. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*, July 11–15, 2022, Madrid, Spain. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3477495.3532038>

1 INTRODUCTION

Given a set of fashion items, Fashion Compatibility Modeling [21], FCM for short, is to estimate their matching degree towards a proper outfit. Due to its significance and value in E-commerce like fashion item recommendation [17], FCM has gained increasing attention from both academic and industrial communities. Although great success has been achieved by existing efforts [7, 13, 32], they mainly focus on the general fashion compatibility modeling, that is, exploring the compatibility among fashion items from the common perspective while overlooking users' personal preferences. In practice, this is not applicable to the real-world fashion product recommendation scenarios.

Nevertheless, aesthetics can be rather subjective. 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 education backgrounds. For example, as shown in Figure 1, given the same pink shirt, user A prefers to match it with the homochromatic skirt and high-heeled shoes; whereas user B likes to coordinate it with the casual jeans and white sneakers. In light of this, Personalized Fashion Compatibility Modeling, dubbed as PFCM, taking users' preferences into account when measuring the compatibility among fashion items, merits our special attention. In fact, a few pioneer researchers have noticed this phenomenon, and dedicated their efforts to PFCM [6, 20, 29]. These efforts mainly study the user and item entities, as well as their relations. They, however, overlook another important entity type in PFCM, 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 bring in attributes associated with fashion items, and work towards 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 PFCM performance. Without loss of generality, we particularly 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, non-trivial due to the following challenges. **C1:** PFCM involves three kinds of entities with heterogeneous contents, namely, users, items, and attributes. In particular, 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 concrete 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 arrive during the testing phase, even for the case that 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 with the same top may share some common attributes. Another example is that users with similar tastes tend to like the items with similar attributes. In light of this, 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 PFCM 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-PFCM, as shown in Figure 2. This scheme consists of three key components: heterogeneous graph node embedding, metapath-guided heterogeneous graph learning, and personalized fashion compatibility modeling. The first component works on embedding each type of entities of our heterogeneous graph. To represent users, we devise a multi-modal content-oriented user embedding module, which derives the user embedding based on the multi-modal contents of his/her interacted items, a straightforward cue indicating the user's preference. As to the second component, we firstly 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 [33] is used to adaptively fuse the semantic-enhanced user/item embeddings under different metapaths for each user/item. In the last component, apart from the typical cross-entropy loss, we also introduce the contrastive regularization to enhance the embedding learning.

Our main contributions can be highlighted in threefold:

- We define a heterogeneous graph to creatively unify three types of entities and relations in the PFCM context. To the best of our knowledge, we are the first on organizing the multi-modal content and attribute information of fashion items via a graph towards PFCM.
- We present a metapath-guided personalized compatibility modeling scheme to perform the heterogeneous graph learning. It adopts the pre-defined metapaths to explore the high-order relations among various entities, and hence strengthen the user and item embeddings.
- We derive users' embeddings via fusing their interacted items and introduce a contrastive regularization to improve the embedding learning. As a byproduct, we conduct extensive experiments on the benchmark dataset, which verifies the superiority of our model to several cutting-edge baselines¹.

2 RELATED WORK

This work is related to personalized fashion compatibility modeling and heterogeneous graph learning.

2.1 Personalized Fashion Compatibility Modeling

Existing work on the fashion compatibility modeling [2, 12, 27] can be mainly grouped into three classes: pair-wise, list-wise, and graph-wise methods. To be more specific, 1) pair-wise methods focus on the compatibility modeling for a pair of items, like a top and a bottom. For example, Song et al. [28] proposed a multi-modal pair-wise compatibility modeling scheme with a dual autoencoder network, which aims to answer the question "which bottom matches the given top". 2) List-wise methods treat the outfit, composing of more than two items, as a sequence, and model the outfit compatibility with the sequential neural networks. For example, Han

¹<https://site2750.wixsite.com/pfcm>.

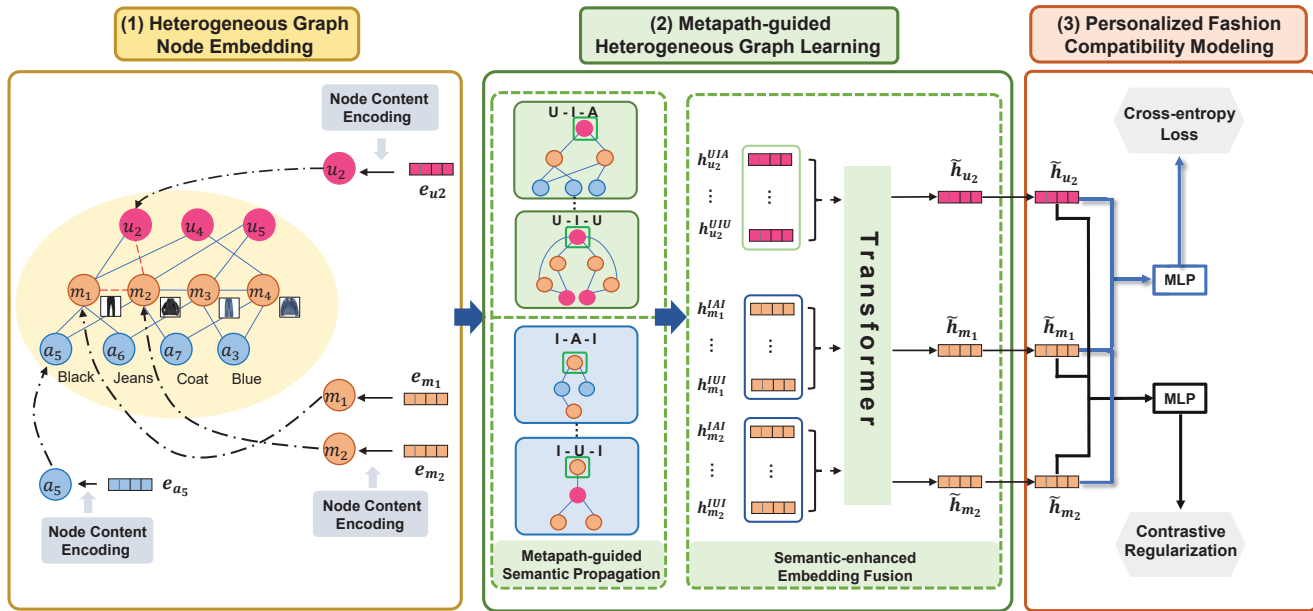


Figure 2: Illustration of the proposed MG-PFCM scheme. It consists of three key components: (1) heterogeneous graph node embedding, (2) metapath-guided heterogeneous graph learning, and (3) personalized fashion compatibility modeling.

et al. [14] proposed to sequentially model the compatibility relationship among the fashion items in a given outfit with a Bi-LSTM. And 3) graph-wise methods deem the outfit as a set of items, and resort to the advanced graph neural networks to explore the outfit compatibility. For example, Cucurull et al. [3] utilized a graph neural network to learn the items' embeddings conditioned on their context, and then estimated outfit compatibility. Despite the significant progress made by these 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 the personalized fashion compatibility modeling. For example, a personalized compatibility modeling scheme for personalized clothing matching, named GP-BPR, is presented in [29], 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 towards the comprehensive modeling. Moving a step forward, Sagar et al. [24] introduced an attribute-wise interpretable personal preference modeling scheme, to strengthen the model interpretability, whereby the images and textual descriptions of items are explored. Besides, Li et al. [19] 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 almost overlook the item attributes when estimating the compatibility. Attributes basically express the key semantics of items and reflect the specific preferences of users. As a complementary effort, in this work, we incorporate the attribute entities and their semantic contents to comprehensively study the PFCM problem.

2.2 Heterogeneous Graph Embedding

Due to the ubiquity of heterogeneous graph in the real-world setting, containing multiple types of nodes and relations among these nodes [11, 31], increasing research efforts have been dedicated to the 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 [1, 23] and personalized recommendation [9, 40].

To accomplish this task, previous methods mostly rely on the metapath [30], *i.e.*, a sequence of node and edge types, delivering certain semantic information of the graph. For example, Dong et al. [8] developed the metapath-based random walks to construct the heterogeneous neighborhood of a node and then utilized a skip-gram model [22] 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. [25] designed a novel strategy to generate the meaningful node sequences and utilized fusion functions to learn node representation. In addition, Zhang et al. [37] introduced a heterogeneous graph neural network model, named HetGNN, to jointly explore the heterogeneous structures and contents of each node. To get the superior node representation, several researchers [10, 35, 39] have utilized the attention mechanism to softly select the most useful metapath. For example, Wang et al. [35] proposed a heterogeneous graph attention network, which incorporates both node- and semantic-level attention to learn the importance of nodes and metapaths towards the node embedding. Subsequently, Zhang et al. [39] proposed an attentive heterogeneous graph neural network towards the 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. [36] regarded each metapath as a specific view, and borrowed the idea of multi-view learning to comprehensively encode the node representations of different views into a latent representation. To tackle the practical issue of missing attributes, Jin et al. [17] proposed a general framework for heterogeneous graph neural network via attribute completion, comprising two key components: pre-learning of 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 PFCM into a unified heterogeneous graph. It is worth emphasizing that we design a few task-specific metapaths and creatively incorporate the transformer to fuse the semantic-enhanced user/item embeddings.

3 METHODOLOGY

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

3.1 Problem Formulation

In this work, we focus on fulfilling the task of PFCM. 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”. Suppose that we have a set of N_u users $\mathcal{U} = \{u_1, u_2, \dots, u_{N_u}\}$, and a set of N_m items $\mathcal{M} = \{m_1, m_2, \dots, m_{N_m}\}$. For an arbitrary item $m_i, i = 1, 2, \dots, N_m$, 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 $\mathcal{A} = \bigcup_{i=1}^{N_m} \mathcal{A}_i = \{a_1, a_2, \dots, a_{N_a}\}$ represents the entire attribute set in the form of semantic phrases, like *red color*, *wool material*, and *V-Neck design*. Thereinto, the symbol N_a denotes the total number of attributes in our dataset. To simplify the formulation, in this work, we only take the tops and bottoms into consideration. 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 $\mathcal{X}^u = \{(m_{t_1}^u, m_{b_1}^u), (m_{t_2}^u, m_{b_2}^u), \dots, (m_{t_{M_u}}^u, m_{b_{M_u}}^u)\}$, where $m_{t_s}^u \in \mathcal{M}^t$, $m_{b_s}^u \in \mathcal{M}^b$, 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 towards learning the following compatibility estimation function,

$$p_{ij}^k = \mathcal{F}(m_k \in \mathcal{M}^{b(t)} | u_i, m_j \in \mathcal{M}^{t(b)}), \quad (1)$$

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

3.2 MG-PFCM

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

3.2.1 Heterogeneous Graph Node Embedding. This component aims to derive the initial node-level representations in the heterogeneous graph. As the heterogeneous graph has three types of entities, and the node contents differ remarkably. We hence learn their embeddings separately as shown in Figure 3.

Item Entity Embedding. Each item entity is composed of an image and a textual description. The multi-modal cues of each item mutually complement each other. As to an 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 [15], to extract its visual feature. Meanwhile, we adopt the pre-trained BERT to obtain its textual feature², due to its prominent performance in textual representation learning [5, 26]. 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 representation of the textual description. Finally, we concatenate the visual and textual features of each item to derive its final embedding, and use a learnable fully-connected layer to project the item embedding into a lower dimensional space. Mathematically, we have

$$\begin{cases} \mathbf{e}_{v_i} = \text{ResNet}(v_i), \\ \mathbf{e}_{t_i} = \text{BERT}(t_i)_{[\text{CLS}]}, \\ \mathbf{e}_{m_i} = f_t([\mathbf{e}_{v_i}, \mathbf{e}_{t_i}]), \end{cases} \quad (2)$$

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 pre-trained 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,

$$\mathbf{e}_{a_l} = f_a(\text{BERT}(a_l)_{[\text{CLS}]}) \quad (3)$$

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 towards the 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

²Before feeding a text into the BERT, the text is first tokenized into standard vocabularies.

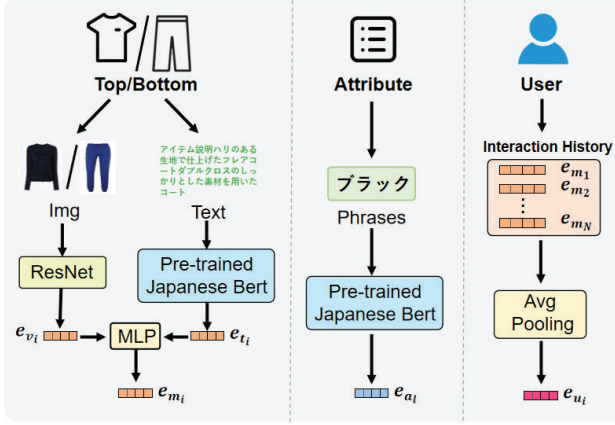


Figure 3: Heterogeneous graph node embedding.

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. Concretely, we reach the user embedding below,

$$e_{u_i} = \frac{1}{|\mathcal{N}^{u_i}|} \sum_{m_i \in \mathcal{N}^{u_i}} e_{m_i}, \quad (4)$$

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

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 [30], a metapath is defined as a path in the form of $X_1 \xrightarrow{R_1} X_2 \xrightarrow{R_2} \dots \xrightarrow{R_n} X_{n+1}$, which describes a composite relation between entities. In our work, as illustrated in Figure 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^3 implies that a user historically prefers an item and that item poses 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 by the same user. Pertaining to the PFCM context, we only adopt metapaths that start from user entities and item entities. Formally, let $\mathcal{P}_{user} = \{r_1, \dots, r_Y\}$ and $\mathcal{P}_{item} =$

³Due to the limited space, we omit the relation types between entities.

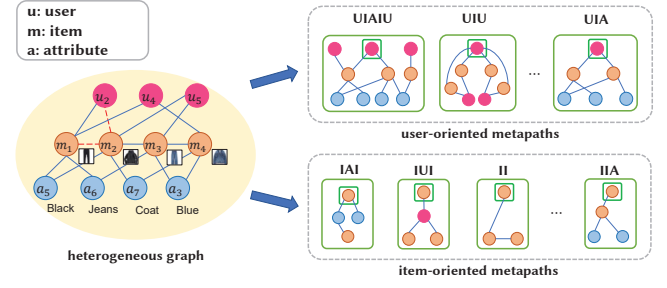


Figure 4: Illustration of user- and item-oriented metapaths.

$\{s_1, \dots, s_Z\}$ denote the set of pre-defined user-oriented and item-oriented metapaths, respectively. Y and Z stand for the total number of user-oriented and item-oriented metapaths, respectively.

Metapath-guided Semantic Propagation. Based on the pre-defined user- and item-oriented metapaths, we are capable of deriving 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.

Suppose that the metapath UIA is applicable to the user entity u_i . We then build a subgraph $\mathcal{G}_{u_i}^{UIA}$ 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 $\mathcal{N}_{u_i}^{UIA(1)}$ consisting of all the items the user once interacted. In the same way, we denote the two-hop neighbors of the user entity u_i as $\mathcal{N}_{u_i}^{UIA(2)}$, comprising all the attributes associated with items in $\mathcal{N}_{u_i}^{UIA(1)}$. 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:

$$\begin{cases} e_{m_i}^{UIA} = \mathcal{H}(e_{a_i} | a_i \in \mathcal{N}_{u_i}^{UIA(2)}), m_i \in \mathcal{N}_{u_i}^{UIA(1)}, \\ h_{u_i}^{UIA} = \mathcal{H}(e_{m_i}^{UIA} | m_i \in \mathcal{N}_{u_i}^{UIA(1)}), \end{cases} \quad (5)$$

where \mathcal{H} is the aggregation function, and $e_{m_i}^{UIA}$ denotes the semantic-enhanced embedding of the item entity m_i , which is a one-hop neighbor of the user entity u_i . $h_{u_i}^{UIA}$ represents the semantic-enhanced embedding of the user entity u_i . It is worth highlighting that, during each hop aggregation, different neighbors may play different roles in characterizing the center entity. Concretely, some attributes may be more important in conveying the item’s properties, while some items may contribute more in reflecting the user’s preference. In light of this, we adopt the graph attention mechanism of GAT [34] 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 in a similar way. Specifically, the aggregation operation \mathcal{H} over the

one-hop neighbors of the user entity u_i can be written as follows,

$$\begin{cases} \mathbf{h}_{u_i}^{\text{UIA}} = \mathbf{e}_{u_i} + \sigma \left(\sum_{m_j \in \mathcal{N}_{u_i}^{\text{UIA}}} \alpha_{ij} \mathbf{e}_{m_j} \right), \\ \alpha_{ij} = \frac{\exp(\sigma(\mathbf{W}^{\text{UIA}}[e_{u_i}, e_{m_j}]))}{\sum_{m_j \in \mathcal{N}_{u_i}^{\text{UIA}}} \exp(\sigma(\mathbf{W}^{\text{UIA}}[e_{u_i}, e_{m_j}]))}, \end{cases} \quad (6)$$

where $\sigma(\cdot)$ denotes the activation function, $[\cdot, \cdot]$ refers to the concatenation operation, and $\mathbf{W}^{\text{UIA}} \in \mathbb{R}^{2D \times 1}$ 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 will not be able to derive the subgraph according to the meta-path UIU. Accordingly, we use $\mathcal{P}^{u_i} = \{r_{i_1}, \dots, r_{i_{Y_i}}\}$ 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 $r_{i_n} \in \mathcal{P}_{user}$, $n = 1, \dots, Y_i$. 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 $\mathcal{P}^{m_i} = \{s_{i_1}, \dots, s_{i_{Z_i}}\}$ 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 $s_{i_z} \in \mathcal{P}_{item}$, $z = 1, \dots, Z_i$. 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 to fuse the multiple embeddings of each user or item. In particular, we leverage the transformer [33] without the positional coding to perform the multi-semantic embedding fusion, mainly 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 fulfil both user and item entities' embedding fusion as follows,

$$\begin{cases} \tilde{\mathbf{h}}_{u_i} = \text{Transformer}(\mathbf{h}_{u_i}^p | p \in \mathcal{P}^{u_i}) \\ \tilde{\mathbf{h}}_{m_i} = \text{Transformer}(\mathbf{h}_{m_i}^p | p \in \mathcal{P}^{m_i}) \end{cases} \quad (7)$$

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.

3.3 Personalized Fashion Compatibility Modeling

To accomplish the task of PFCM, we first build the training set $\Omega = \{(u_i, m_j, m_{k+}, m_{k-}) | m_j \in \mathcal{M}^{(b)}, m_{k+}, m_{k-} \in \mathcal{M}^{(t)}, y_{ij}^{k+} = 1, y_{ij}^{k-} = 0\}$, 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 the 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. (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,

$$\hat{p}_{ij}^{k+(-)} = \text{MLP}_0([\tilde{\mathbf{h}}_{u_i}, \tilde{\mathbf{h}}_{m_j}, \tilde{\mathbf{h}}_{m_{k+(-)}}]), \quad (8)$$

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

$$\mathcal{L}^{(i,j,k+,k-)} = -\log\left(\frac{\exp(\hat{p}_{ij}^{k+})}{\exp(\hat{p}_{ij}^{k+}) + \exp(\hat{p}_{ij}^{k-})}\right). \quad (9)$$

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 the contrastive regularization to regulate the similarity between latent representations of different triplet pairs. Suppose 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 one. We utilize two MLPs to obtain the latent representations for these three triplets as follows,

$$\begin{cases} \tilde{\mathbf{h}}_{p_1^+} = \text{MLP}_1([\tilde{\mathbf{h}}_{u_{i_1}^+}, \tilde{\mathbf{h}}_{m_{j_1}^+}, \tilde{\mathbf{h}}_{m_{k_1}^+}]), \\ \tilde{\mathbf{h}}_{p_2^+} = \text{MLP}_2([\tilde{\mathbf{h}}_{u_{i_2}^+}, \tilde{\mathbf{h}}_{m_{j_2}^+}, \tilde{\mathbf{h}}_{m_{k_2}^+}]), \\ \tilde{\mathbf{h}}_{n^-} = \text{MLP}_2([\tilde{\mathbf{h}}_{u_{i^-}}, \tilde{\mathbf{h}}_{m_{j^-}}, \tilde{\mathbf{h}}_{m_{k^-}}]), \end{cases} \quad (10)$$

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,

$$\mathcal{L}_{cons}^{(p_1^+, p_2^+, n^-)} = -\log \frac{\exp(\text{sim}(\tilde{\mathbf{h}}_{p_1^+}, \tilde{\mathbf{h}}_{p_2^+}))}{\exp(\text{sim}(\tilde{\mathbf{h}}_{p_1^+}, \tilde{\mathbf{h}}_{p_2^+})) + \exp(\text{sim}(\tilde{\mathbf{h}}_{p_1^+}, \tilde{\mathbf{h}}_{n^-}))}, \quad (11)$$

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

$$\mathcal{L} = \sum_{(u_i, m_j, m_{k+}, m_{k-})} \mathcal{L}^{(i,j,k+,k-)} + \lambda \sum_{(p_1^+, p_2^+, n^-)} \mathcal{L}_{cons}^{(p_1^+, p_2^+, n^-)}, \quad (12)$$

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

4 EXPERIMENT

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

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

4.1 Experimental Settings

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

Table 1: 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

4.1.1 Dataset. To justify our model, similar to existing methods [24, 29], we also resorted to the public benchmark dataset IQON3000 [29], 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 [24, 29] 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 is arbitrarily switchable for a given user. That is to say, 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 own dataset from IQON3000. In particular, we only remained the outfits that contain a top and a bottom, and users who have interacted with no less than 10 and no more than 200 outfits to keep the dataset relatively balanced. Finally, there are 82,079 user-top-bottom triplets involved 1,769 users. The detailed statistics are summarized in Table 1. Meanwhile, the attributes and their corresponding value examples of the derived dataset are shown in Table 2.

Notably, all these retained triplets are positive ones, 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 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) one. Following that, we replaced the target (positive) item with a randomly sampled one sharing the same category with the target one, to derive a negative triplet. It is worth mentioning that to ensure the fairness, considering the baseline methods do not need the specific graph construction set, we train them with both the graph construction set and training set, where the negative triplets in the graph construction set are derived in the same manner.

4.1.2 Evaluation Tasks and Metrics. Towards comprehensive evaluation, similar to previous studies [4, 12, 14, 24, 29], we justified our proposed MG-PFCM scheme with two tasks: the compatibility estimation task and the complementary item retrieval task. The

Table 2: Attributes and their possible value examples.

Attribute	Possible Value Examples	Total Number
Color	Grey, 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

former 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) [38] as the evaluation metric. The latter task is to retrieve the target complementary and compatible item from a set of item candidates for a given user and a given item (either a top or a bottom). Specifically, for each positive triplet, we randomly selected one item as the target item, and additionally introduced T negative items to constitute the whole set of item candidates. These item candidates will be ranked according to their compatibility scores to the given user and item, calculated by Eqn.(8). In the complementary item retrieval task, we utilized the Mean Reciprocal Ranking (MRR) [16] as the evaluation metric.

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, the ResNet18 is also fine-tuned with the whole model. Regarding the textual feature extraction of items, we resorted to the implementation of BERT⁴ for Japanese text considering our dataset is in Japanese, and embed each item’s textual description into a 768-D vector. The dimension of final item embedding $D = 512$. Similarly, using this BERT implementation, each semantic attribute is 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 adopted the following set of user-oriented metapaths $\mathcal{P}_{user} = \{UIAIU, UIU, UIA\}$, and item-oriented metapaths $\mathcal{P}_{item} = \{IAL, IUI, II, IIA\}$. 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 [18]). The learning rate is warmed up to the peak value, which is set to $1e-4$, in the 6% steps and then linearly decayed to 0. The hyperparameter λ is set to 1, and the batch size is set to 24. The number of negative outfits in the complementary item retrieval task, *i.e.*, T is set to 4. All the experiments are implemented by PyTorch over a server equipped with 4 A100-PCIE-40GB GPUs.

4.2 On Model Comparison (RQ1)

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

⁴<https://huggingface.co/cl-tohoku/bert-base-japanese-char/tree/main>.

Table 3: Performance comparison between our proposed MG-PFCM 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	PAI-BPR	HFGN	GP-BPR	MG-PFCM
AUC	0.6096	0.6783	<u>0.7146</u>	0.7730
MRR	0.5456	0.6173	<u>0.6346</u>	0.6427

- **GP-BPR** [29] is a comprehensive personal preference modeling scheme, where the multi-modal data (e.g., the image and text description) of fashion items are jointly explored.
- **PAI-BPR** [24] is an attribute-wise interpretable compatibility modeling scheme, which solves the problem of interpretability in clothing matching by locating the discordant and harmonious attributes between fashion items.
- **HFGN** [19] refers to a hierarchical fashion graph network, which simultaneously models the relationships among users, items, and outfits.

Table 3 shows the performance comparison among different methods in terms of AUC and MRR. From this table, we have the following observations. 1) our proposed MG-PFCM scheme consistently outperforms all baseline methods over different metrics. In particular, MG-PFCM 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 PFCM into a unified heterogeneous graph, and utilizes the metapath-guided heterogeneous graph learning towards personalized fashion compatibility modeling. 2) Our method surpasses the heterogeneous graph based method HFGN remarkably over both metrics, implying the necessity of considering the items' attributes. And 3) GP-BPR outperforms HFGN, which may be due to that HFGN only utilizes the visual information of fashion items, while GP-BPR jointly considers the images and textual descriptions of items.

4.3 On Ablation Study (RQ2)

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 item for PFCM, 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 PFCM, 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-PFCM is 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 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.(12).

Table 4: Ablation study of our proposed MG-PFCM 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-PFCM	0.7730	0.6427

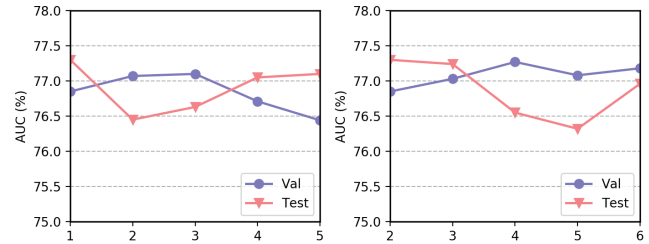


Figure 5: Sensitivity analysis of our model performance in terms of AUC (%) with respect to (a) the number of transformer layers, and (b) the number of GAT layers.

- **w mean pooling**: To evaluate the function of transformer component in the semantic embedding fusion, we replaced the transformer component with the mean pooling function.

Table 4 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-PFCM. Specifically, we have the following detailed observations. 1) Both w/o text and w/o image perform inferior to MG-PFCM, which suggests that it is essential to consider both modalities of fashion items to boost the item representation learning. 2) w/o attribute presents the worst performance, reflecting the benefit of incorporating the attribute entities as well as their semantic contents into personalized fashion compatibility modeling. 3) w/o (II,UIA) performs worse than our MG-PFCM, which suggests that different metapaths do deliver different semantics, and it is advisable to consider the sub meta-paths. 4) The performance of w/o contrastive drops, as compared to MG-PFCM, indicating that the contrastive regularization is indeed helpful to strengthen the fashion entity representation learning. And 5) w mean pooling also performs worse than our MG-PFCM, reflecting the effectiveness of the transformer in fusing the unfixed number of semantic-enhanced embeddings of users/items.

4.4 On Sensitivity Analysis (RQ3)

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 thus changed the number of GAT layers from 2 to 6 with the step of 1. Figure 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,













	User History Preference	Given Item	Positive Item	Negative Item
User 1		 Brown Skirt	 Gray Blouse Floral pattern Ruffle	 Black Blouse Suede
		MG-PFCM	0.8164 ✓	0.1836
		w/o attribute	0.5332 ✓	0.4668
User 2		 Beige Cardigan Tops Wool	 White Long pants Stripe	 Black Skirt Floral pattern
		MG-PFCM	0.7388 ✓	0.2612
		w/o attribute	0.5742 ✓	0.4258
User 3		 Beige Blouse Tops	 Black Long skirt	 Green Long pants
		MG-PFCM	0.6714 ✓	0.3283
		w/o attribute	0.4639	0.5361 ✗

Figure 6: Illustration of several PFCM results obtained by our MG-PFCM and w/o attribute derivative.

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.

4.5 On Case Study (RQ4)

To gain more intuitive insights into our model, we also conducted the case study of our method and the w/o attribute derivative. Figure 6 shows three testing samples, where the users' historical preferred top-bottom pairs and items' attributes are also listed to facilitate the experimental result analysis⁵. As can be seen, for the case of the first user with the given brown skirt, although both our MG-PFCM and its derivative w/o attribute give the correct prediction, our MG-PFCM assigns a much higher score to the positive item than the negative one. By contrast, w/o attribute gives the former a slightly higher score than the latter one. Namely, our model has the high confidence than its derivative. This may be due to that incorporating the attribute entities in PFCM enables our model MG-PFCM 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-PFCM rather than the w/o attribute derivative. Meanwhile, as the negative item is a black skirt, which cannot go well with the beige cardigan, our MG-PFCM 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 one. As for the last case, as we can see that the third user prefers "long skirts" for blouses. The positive item is a black long skirt, looking like the long pants, while the negative

⁵Due to the limited space, we did not provide the text description of the items.

one is the long pants looking like a long skirt. Then with the help of their category attributes, our MG-PFCM correctly selects the compatible item for the given top, while the w/o attribute method gives the wrong judgment. Overall, based upon these case studies, we can confirm the effectiveness of our method, and the benefit of incorporating the attribute information in the context of PFCM.

5 CONCLUSION AND FUTURE WORK

In this work, 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 metapath-guided personalized compatibility modeling (MG-PFCM) 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 contrastive regularization as well as using a non-position transformer to fulfil 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.

The limitation of our work is that currently we are only able to judge the compatibility degree of a bottom (top) to the given top (bottom) for a specific user. In fact, each outfit usually involves not only the top and bottom, but also other items like shoes and accessories. Accordingly, in future, we will extend our work to explore PFCM for outfits with unfixed number of composing items.

6 ACKNOWLEDGEMENTS

This work was supported in part by Australian Research Council (ARC) Discovery Early Career Researcher Award (DECRA) under DE190100626.

REFERENCES

- [1] Sami Abu-El-Haija, Amol Kapoor, Bryan Perozzi, and Joonseok Lee. 2019. N-GCN: Multi-scale Graph Convolution for Semi-supervised Node Classification. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 841–851.
- [2] Suthesh Chaidaroon, Yi Fang, Min Xie, and Alessandro Magnani. 2019. Neural Compatibility Ranking for Text-based Fashion Matching. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1229–1232.
- [3] Guillem Cucurull, Perouz Taslakian, and David Vázquez. 2019. Context-Aware Visual Compatibility Prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 12617–12626.
- [4] Zeyu Cui, Zekun Li, Shu Wu, Xiaoyu Zhang, and Liang Wang. 2019. Dressing as a Whole: Outfit Compatibility Learning Based on Node-wise Graph Neural Networks. In *Proceedings of the World Wide Web Conference*. ACM, 307–317.
- [5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*. ACL, 4171–4186.
- [6] Xue Dong, Xuemeng Song, Fuli Feng, Peiguang Jing, Xin-Shun Xu, and Liqiang Nie. 2019. Personalized Capsule Wardrobe Creation with Garment and User Modeling. In *Proceedings of the International ACM Conference on Multimedia*. ACM, 302–310.
- [7] Xue Dong, Jianlong Wu, Xuemeng Song, Hongjun Dai, and Liqiang Nie. 2020. 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*. ACM, 771–780.
- [8] Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable Representation Learning for Heterogeneous Networks. In *Proceedings of the International ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 135–144.
- [9] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Yihong Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph Neural Networks for Social Recommendation. In *Proceedings of the World Wide Web Conference*. ACM, 417–426.
- [10] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. 2020. MAGNN: Metapath Aggregated Graph Neural Network for Heterogeneous Graph Embedding. In *Proceedings of the World Wide Web Conference*. ACM, 2331–2341.
- [11] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. In *Proceedings of the International ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 855–864.
- [12] Weili Guan, Haokun Wen, Xuemeng Song, Chung-Hsing Yeh, Xiaojun Chang, and Liqiang Nie. 2021. Multimodal Compatibility Modeling via Exploring the Consistent and Complementary Correlations. In *Proceedings of the International ACM Conference on Multimedia*. ACM, 2299–2307.
- [13] Xianjing Han, Xuemeng Song, Jianhua Yin, Yinglong Wang, and Liqiang Nie. 2019. Prototype-guided Attribute-wise Interpretable Scheme for Clothing Matching. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 785–794.
- [14] Xintong Han, Zuxuan Wu, Yu-Gang Jiang, and Larry S. Davis. 2017. Learning Fashion Compatibility with Bidirectional LSTMs. In *Proceedings of the International ACM Conference on Multimedia*. ACM, 1078–1086.
- [15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 770–778.
- [16] Lu Jiang, Shou-I Yu, Deyu Meng, Yi Yang, Teruko Mitamura, and Alexander G. Hauptmann. 2015. Fast and Accurate Content-based Semantic Search in 100M Internet Videos. In *Proceedings of the International ACM Conference on Multimedia*. ACM, 49–58.
- [17] Di Jin, Cuiying Huo, Chungong Liang, and Liang Yang. 2021. Heterogeneous Graph Neural Network via Attribute Completion. In *Proceedings of the World Wide Web Conference*. ACM, 391–400.
- [18] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the International Conference on Learning Representations*. OpenReview.net, 1–15.
- [19] Xingchen Li, Xiang Wang, Xiangnan He, Long Chen, Jun Xiao, and Tat-Seng Chua. 2020. Hierarchical Fashion Graph Network for Personalized Outfit Recommendation. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 159–168.
- [20] Zhi Lu, Yang Hu, Yunchao Jiang, Yan Chen, and Bing Zeng. 2019. Learning Binary Code for Personalized Fashion Recommendation. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 10562–10570.
- [21] Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 43–52.
- [22] Tomáš Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the International Conference on Learning Representations*. OpenReview.net, 1–12.
- [23] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. 2020. DropEdge: Towards Deep Graph Convolutional Networks on Node Classification. In *Proceedings of the International Conference on Learning Representations*. OpenReview.net, 1–17.
- [24] Dikshant Sagar, Jatin Garg, Prarthana Kansal, Sejal Bhalla, Rajiv Ratn Shah, and Yi Yu. 2020. PAI-BPR: Personalized Outfit Recommendation Scheme with Attribute-wise Interpretability. In *IEEE International Conference on Multimedia Big Data*. IEEE, 221–230.
- [25] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and Philip S. Yu. 2019. Heterogeneous Information Network Embedding for Recommendation. *IEEE Transactions on Knowledge and Data Engineering* 31, 2 (2019), 357–370.
- [26] Xue Song, Jingjing Chen, Zuxuan Wu, and Yu-Gang Jiang. 2021. Spatial-temporal graphs for cross-modal text2video retrieval. *IEEE Transactions on Multimedia* (2021).
- [27] Xuemeng Song, Fuli Feng, Xianjing Han, Xin Yang, Wei Liu, and Liqiang Nie. 2018. Neural Compatibility Modeling with Attentive Knowledge Distillation. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 5–14.
- [28] Xuemeng Song, Fuli Feng, Jinhuan Liu, Zekun Li, Liqiang Nie, and Jun Ma. 2017. NeuroStylist: Neural Compatibility Modeling for Clothing Matching. In *Proceedings of the International ACM Conference on Multimedia*. ACM, 753–761.
- [29] Xuemeng Song, Xianjing Han, Yunkai Li, Jingyuan Chen, Xin-Shun Xu, and Liqiang Nie. 2019. GP-BPR: Personalized Compatibility Modeling for Clothing Matching. In *Proceedings of the International ACM Conference on Multimedia*. ACM, 320–328.
- [30] Yizhou Sun and Jiawei Han. 2012. Mining heterogeneous information networks: principles and methodologies. *Synthesis Lectures on Data Mining and Knowledge Discovery* 3, 2 (2012), 1–159.
- [31] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale Information Network Embedding. In *Proceedings of the World Wide Web Conference*. ACM, 1067–1077.
- [32] Mariya I. Vasileva, Bryan A. Plummer, Krishna Dusad, Shreya Rajpal, Ranjitha Kumar, and David A. Forsyth. 2018. Learning Type-Aware Embeddings for Fashion Compatibility. In *Proceedings of the European Conference on Computer Vision*. Springer, 405–421.
- [33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Proceedings of the Conference on Neural Information Processing Systems*. MIT Press, 5998–6008.
- [34] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *Proceedings of the International Conference on Learning Representations*. OpenReview.net, 1–12.
- [35] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S. Yu. 2019. Heterogeneous Graph Attention Network. In *Proceedings of the World Wide Web Conference*. ACM, 2022–2032.
- [36] Yuying Xing, Zhao Li, Pengrui Hui, Jiaming Huang, Xia Chen, Long Zhang, and Guoxian Yu. 2020. Link Inference via Heterogeneous Multi-view Graph Neural Networks. In *Proceedings of the International Conference on Database Systems for Advanced Applications*. Springer, 698–706.
- [37] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. 2019. Heterogeneous Graph Neural Network. In *Proceedings of the International ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 793–803.
- [38] Hanwang Zhang, Zheng-Jun Zha, Yang Yang, Shuicheng Yan, Yue Gao, and Tat-Seng Chua. 2013. Attribute-augmented Semantic Hierarchy: Towards Bridging Semantic Gap and Intention Gap in Image Retrieval. In *Proceedings of the International ACM Conference on Multimedia*. ACM, 33–42.
- [39] Jintao Zhang and Quan Xu. 2021. Attention-aware Heterogeneous Graph Neural Network. *Big Data Mining and Analytics* 4, 4 (2021), 233–241.
- [40] Huan Zhao, Quanming Yao, Jianda Li, Yangqiu Song, and Dik Lun Lee. 2017. Meta-Graph Based Recommendation Fusion over Heterogeneous Information Networks. In *Proceedings of the International ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 635–644.