Fashion Compatibility Modeling through a Multi-modal Try-on-guided Scheme

Xue Dong, Jianlong Wu, Xuemeng Song, Hongjun Dai, Liqiang Nie
Shandong University, Shandong, China
{dongxue.sdu,sxmustc,nieliqiang}@gmail.com,{jlwu1992,dahogn}@sdu.edu.cn

ABSTRACT

Recent years have witnessed a growing trend of fashion compatibility modeling, which scores the matching degree of the given outfit and then provides people with some dressing advice. Existing methods have primarily solved this problem by analyzing the discrete interaction among multiple complementary items. However, the fashion items would present certain occlusion and deformation when they are worn on the body. Therefore, the discrete item interaction cannot capture the fashion compatibility in a combined manner due to the neglect of a crucial factor: the overall try-on appearance. In light of this, we propose a multi-modal try-on-guided compatibility modeling scheme to jointly characterize the discrete interaction and try-on appearance of the outfit. In particular, we first propose a multi-modal try-on template generator to automatically generate a try-on template from the visual and textual information of the outfit, depicting the overall look of its composing fashion items. Then, we introduce a new compatibility modeling scheme which integrates the outfit try-on appearance into the traditional discrete item interaction modeling. To fulfill the proposal, we construct a large-scale real-world dataset from SSENSE, named FOTOS, consisting of 11,000 well-matched outfits and their corresponding realistic try-on images. Extensive experiments have demonstrated its superiority to state-of-the-arts.

CCS CONCEPTS

• **Information systems** → **Retrieval tasks and goals**; *World Wide Web.*

KEYWORDS

Fashion Analysis; Compatibility Modeling; Try-on-guided Scheme

ACM Reference Format:
Xue Dong, Jianlong Wu, Xuemeng Song, Hongjun Dai, Liqiang Nie. 2020. Fashion Compatibility Modeling through a Multi-modal Try-on-guided Scheme. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3397271.3401047

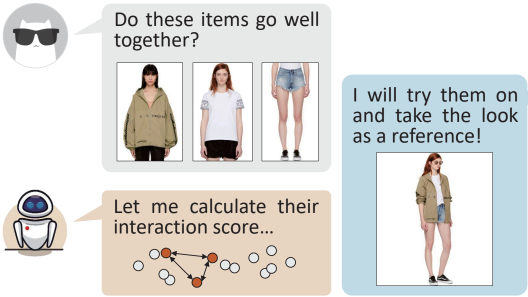
Do these items go well together?
I will try them on and take the look as a reference!
Let me calculate their interaction score...

Figure 1: Previous methods analyze the fashion compatibility by directly modeling the interaction among fashion items, while people usually prefer to try the outfit on to evaluate its practical compatibility.

1 INTRODUCTION

With the recent flourishing of the e-commerce fashion industry, increasing research attention has been paid to studying the automatic fashion compatibility modeling among multiple complementary fashion items (e.g., the khaki jacket, white icon t-shirt and blue denim hot pants), which can benefit many downstream applications in the fashion domain, such as the outfit recommendation [1–3], personal wardrobe creation [4, 5], and fashion-oriented dialogue systems [6, 7]. Existing approaches for the fashion compatibility modeling mainly focus on analyzing the discrete interaction among multiple complementary items [1–3, 8–10]. That is to measure the compatibility of an outfit with certain metric over the latent embeddings of its composing fashion items learned by the matched and ill-matched outfits [10].

Although great success has been achieved by these efforts, they overlook a crucial factor in the fashion compatibility modeling: the overall try-on appearance. As a matter of fact, people usually try the outfit on to evaluate its practical compatibility, where fashion items would present certain occlusion and deformation when they are worn on the body [11, 12], shown in Figure 1. Specifically, in the outfit listed at the figure, all features of these fashion items would be considered in the traditional discrete item interaction modeling. Nevertheless, due to the deformation and occlusion of the loose windbreaker, the center area of the t-shirt contributes mostly to the fashion compatibility while the pattern on the sleeves can be ignored. In light of this, the discrete item interaction may be inadequate to thoroughly model the fashion compatibility.

Therefore, considering the practical concern, we aim to tackle the problem of the compatibility modeling via jointly characterizing the discrete item interaction and try-on appearance. Towards this end, propelled by the recent success of generative networks in enhancing the visual understanding in various tasks [13], we

SIGIR '20, July 25–30, 2020, Virtual Event, China
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-8016-4/20/07...$15.00
https://doi.org/10.1145/3397271.3401047

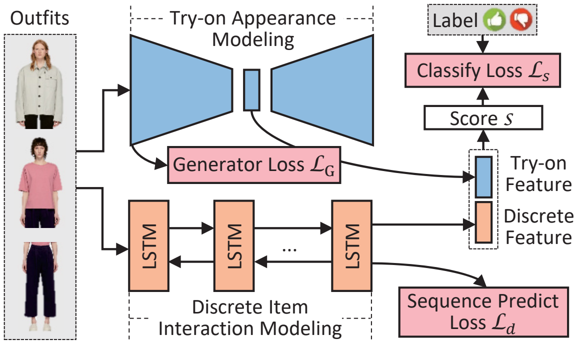
Outfits
Try-on Appearance Modeling
Label
Classify Loss $\mathcal{L}_s$
Score s
Generator Loss $\mathcal{L}_G$
Try-on Feature
Discrete Feature
LSTM
LSTM
...
LSTM
Discrete Item Interaction Modeling
Sequence Predict Loss $\mathcal{L}_d$

Figure 2: Illustration of the proposed TryOn-CM framework, which could analyze the fashion compatibility from both the discrete item interaction and try-on appearance.

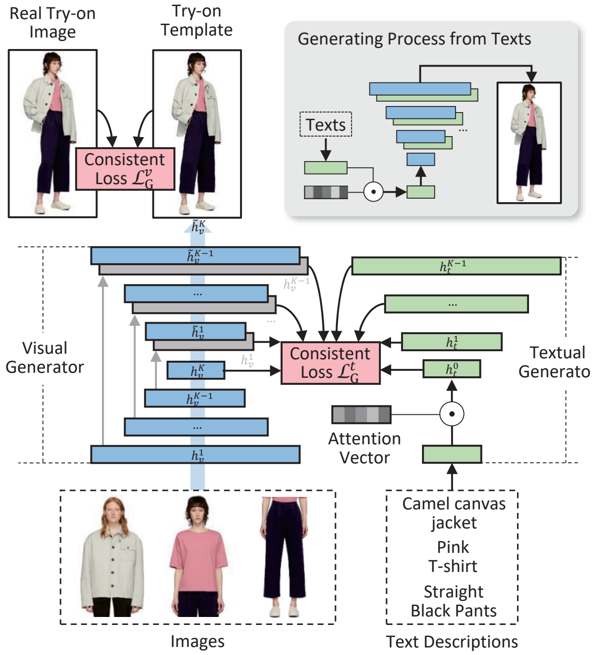

Figure 3: Structure of the multi-modal try-on template generator, comprising a visual and textual generator. The visual generating process is illustrated with the blue arrow while the textual one is shown in the gray box.

propose to enhance the compatibility modeling performance with the generative try-on appearance modeling, where a *try-on template*, depicting the overall look of several complementary fashion items, can be generated to facilitate the fashion compatibility modeling. Nevertheless, this is non-trivial due to the following challenges. 1) How to generate the realistic try-on template with appropriate occlusion and deformation among fashion items poses the major challenge. 2) Both the visual and textual information of fashion items convey important signals to generate the realistic try-on template. Therefore, how to fully explore the multi-modal data of fashion items to synthesize the comprehensive try-on template is the second challenge. 3) How to seamlessly integrate the try-on appearance modeling and discrete item interaction modeling in a unified end-to-end manner constitutes another tough challenge.

To address the aforementioned challenges, we present a multi-modal **Try-On**-guided **C**ompatibility **M**odeling scheme, named TryOn-CM for simplicity, shown in Figure 2. It consists of two key components: the *try-on appearance modeling* (blue part) and *discrete item interaction modeling* (orange part), based on which we can analyze the fashion compatibility from both the discrete and combined manner. 1) As for the former component, to capture the try-on appearance of an outfit, we develop a Multi-modal Try-on Template Generator (MTTG), shown in Figure 3, to synthesize the try-on template, where both visual and textual modalities of fashion items are explored. In particular, the visual generator works on synthesizing the try-on template based on images of composing fashion items with the auto-encoder framework, while the textual generator operates as the encoder of the visual generator taking into account the latent consistency between the textual description and visual image of the same fashion item. And 2) pertaining to the later component, we adopt a bi-directional LSTM to uncover the latent interaction among the list of complementary fashion items. Ultimately, towards the comprehensive compatibility modeling, we feed the compatibility features derived from these two components into the multi-layer perception, and hence obtain the fashion compatibility score, based on which we can recommend compatible outfits for people. Besides, to evaluate the proposed TryOn-CM, we have constructed a large-scale and real-world dataset from an online fashion-oriented community website SSENSE[1], consisting of 11K **F**ashion **O**utfits with their **T**ry-**O**n image**S**, named FOTOS for simplicity.

Our main contributions can be summarized in threefold:

- We present a new compatibility modeling scheme that integrates the outfit try-on appearance into the discrete item interaction modeling, which overcomes the limitation that existing methods mainly neglect the occlusion and deformation factors of fashion items when they are tried on. To the best of our knowledge, we are the first to consider the realistic try-on appearance during the fashion compatibility modeling.
- A multi-modal try-on template generator is designed to produce the try-on template based on the multi-modal data of fashion items, where the latent consistency between the textual description and visual image of the same fashion item is well incorporated.
- We construct a new fashion dataset from SSENSE, named FOTOS, consisting of 11,000 well-matched outfits composed by 20,318 fashion items. Extensive experiments conducted on the dataset demonstrate the superiority of our proposed scheme over the state-of-the-art methods. We have released the data and codes to facilitate other researchers[2].

The remainder of this paper is structured as follows. Section 2 briefly reviews the related work. And then, the proposed TryOn-CM scheme and newly constructed FOTOS dataset are introduced in Section 3 and Section 4, respectively. Finally, we elaborate the experimental results and analyses in Section 5, followed by our concluding remarks and future work in Section 6.

[1] https://www.ssense.com/en-cn.

[2] https://dxresearch.wixsite.com/tryon-cm

2 RELATED WORK

Generally, this work is related to the following two categories: the image synthesis and fashion compatibility modeling.

Image Synthesis. At the initial stage, as the idea of the Auto-Encoder (AE) [14] appeared, many variants, such as the denoising AE [15] and variational AE [16], could cope with the image synthesis task. However, most of them utilize the mean-square loss as the loss function, resulting in fuzzier synthetic images. Later, Goodfellow et al. [17] proposed the Generative Adversarial Nets (GANs) by introducing an adversarial loss, which makes remarkable progress in the image synthesis task. After that, some studies have been dedicated to improve the original GANs. For example, Radford et al. [18] proposed the DCGAN, which provides a specific network topology for the training process. Besides, to incorporate desired properties in generated samples, researchers also utilized different signals, including the text [19] and attributes [20], as priors to condition the image synthesis process. Besides, there are a few studies investigating the problem of image-to-image translation via conditional GANs [21], which transforms the given input image to another one with a different representation. Recently, many researchers have noticed the exciting prospect of GANs in the fashion domain, specifically in the virtual try-on [11], which focuses on generating the new images of the person wearing a new item. Later, great efforts have been made to enhance the virtual try-on by supporting the arbitrary poses [12, 22]. However, most of GANs are very deep neural networks and suffers from the information loss and degradation problem. Therefore, Ronneberger et al. [23] proposed a u-net structure, which utilizes skip connections to merge more features during the image synthesis. In this paper, we found the physiology of the skip connection possesses the great superiority and efficiency, based on which we further proposed a multi-modal try-on template generator to generate the try-on template.

Fashion Compatibility Modeling. Due to the proliferation of various online fashion communities and their importance in fashion analyses, the outfit compatibility modeling has attracted many researchers' attention [1, 2, 8]. For example, Song et al. [1] collected the outfit dataset from Polyvore and introduced a content-based neural framework for the compatibility modeling between the top and bottom. Meanwhile, Li et al. [24] and Chen et al. [9] studied the outfit compatibility that involves multiple (more than two) fashion items. Besides, some auxiliary information, such as the item category [3], aesthetic characteristics [25] and domain knowledge [8, 26], has been explored to promote the performance. Recently, to enhance the practicality, there has been a growing trend to make the compatibility more interpretable, where the attention mechanism [27, 28] and interpretable feature learning [10, 29–31] have been explored. Noticing that existing methods mainly focus on the supervised learning and may present the unreliability of the negative example sampling, several efforts have been made to analyze the compatibility in an unsupervised way. For example, Han et al. [2] and Chaidaroon et al. [32] used a bi-directional LSTM and GRU to uncover the sequential relationship of the outfit, respectively, and Hsiao et al. [4] proposed a style topic model to analyze the relationship among fashion attributes. Despite their effectiveness, the above methods mostly learn the outfit compatibility based on the discrete item interaction. However, in reality, fashion items present the occlusion and deformation when they are tried on, making it hard to accurately model the compatibility simply with the discrete interaction among fashion items. Distinguished from these studies, we propose to analyze the fashion compatibility from a combined manner where the try-on appearance is incorporated into the discrete item interaction.

3 METHODOLOGY

In this section, we detail the proposed TryOn-CM shown in Figure 3. In particular, we first formally define the research problem in Subsection 3.1. Then in Subsection 3.2, we mainly model the try-on appearance by proposing a multi-modal try-on template generator, comprising a visual generator and a textual generator. Following this, we introduce the discrete item interaction modeling in Subsection 3.3. And finally, we present the multi-modal try-on-guided compatibility modeling in Subsection 3.4.

3.1 Problem Formulation

Let $O = [o_1, o_2, ..., o_N]$ denote an outfit, where o_i is the i-th fashion item in the outfit arranged in a predefined order according to its categories, i.e., from the outside to inside and then from the top to bottom. Each fashion item o_i is associated with its product image and text description, which are represented by the pixel array $\boldsymbol{v}_i$ and bag-of-word vector $\boldsymbol{t}_i$, respectively. In this work, we aim to devise a comprehensive fashion compatibility modeling scheme $\mathcal{M}$, which automatically assesses the overall compatibility of the given outfit based on the multi-modal data of its fashion items as follows:

$$s = \mathcal{M}(\{\boldsymbol{v}_i, \boldsymbol{t}_i\}_{i=1}^{N} | \Theta), \tag{1}$$

where s denotes the compatibility score of the outfit and Θ is a set of to-be-learned model parameters.

As a major novelty, apart from the traditional discrete item interaction modeling, we also take into account the try-on appearance, which involves a try-on template generation for the given outfit. To optimize our scheme, we build the training set of M outfits, i.e., $\Omega = \{(O_i, y_i) | i = 1, ..., M\}$, where O_i is the i-th outfit with a set of complementary fashion items, and y_i stands for the ground truth label, which equals to 1 if O_i is a positive (compatible) outfit, and 0 otherwise. Besides, each positive outfit O_i corresponds to a try-on image P_i, in which all the composing fashion items are put on a fashion model and make a realistic outfit try-on appearance. For convenience, the sets of positive and negative outfits are defined as $\Omega_+ = \{(O_i, y_i, P_i) | y_i = 1\}$ and $\Omega_- = \{(O_i, y_i) | y_i = 0\}$, respectively.

3.2 Try-on Appearance Modeling

It is worth noting that, as a pioneering study on generative fashion compatibility modeling, we focus on the general objective factors, like the occlusion and deformation of fashion items in the try-on appearance, but leave out the subjective factors, like the personal identity and body shape. One naive approach to generating the try-on template for an outfit to capture its try-on appearance is directly conditioned on the visual images of all its fashion items. However, the textural descriptions of fashion items also convey important cues of fashion items, like the category and material, which can guide the item layout and hence promote the try-on template generation. Therefore, we take into account both visual and textual modalities of fashion items in the try-on template generation. Accordingly, we devise the multi-modal try-on template generator with a visual generator and a textual generator.

3.2.1 Visual Generator. We cast the task of the visual generator as an image-to-image translation problem, where the input are images of composing fashion items in the outfit and the output is the try-on template image. Due to the remarkable performance of the auto-encoder structure in this research line [23, 33, 34], we adopt it in our visual generator.

In particular, the visual generator consists of an encoder E_v for compressing the multiple discrete fashion item images into a dense vector, and a decoder D_v for transforming the dense vector into the synthesized try-on template image. The encoder E_v and decoder D_v consist of K convolutional layers and K deconvolutional layers, respectively, where each layer (except the first and last layers) is followed by a ReLU activation function and a batch-normalization layer [35]. Notably, to minimize the information loss during the decoding process, inspired by [23], we combine the feature maps of the decoder and the corresponding ones of the encoder to decode the try-on template. For simplicity, we temporally omit the subscript i of $\boldsymbol{O}_i$, and thus the encoder and decoder of the visual generator can be formulated as follows:

$$\begin{aligned}
\boldsymbol{h}_v^K \;=\; & E_v(\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_N) : \\
& \begin{cases} \boldsymbol{h}_v^1 = \mathrm{conv}(\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_N), \\ \boldsymbol{h}_v^i = \mathrm{bn}(\mathrm{conv}(\mathrm{ReLU}(\boldsymbol{h}_v^{i-1})))|_{i=2}^{K-1}, \\ \boldsymbol{h}_v^K = \mathrm{conv}(\mathrm{ReLU}(\boldsymbol{h}_v^{K-1})), \end{cases} \\
\widetilde{\boldsymbol{h}}_v^K \;=\; & D_v(\boldsymbol{h}_v^K, \{\boldsymbol{h}_v^i\}_{i=1}^{K-1}) : \\
& \begin{cases} \widetilde{\boldsymbol{h}}_v^0 = \mathrm{ReLU}(\mathrm{bn}(\boldsymbol{h}_v^K)), \\ \widetilde{\boldsymbol{h}}_v^i = \mathrm{ReLU}([\mathrm{bn}(\mathrm{dconv}(\widetilde{\boldsymbol{h}}_v^{i-1})), \boldsymbol{h}_v^{K-i+1}])|_{i=1}^{K-1}, \\ \widetilde{\boldsymbol{h}}_v^K = \tanh(\mathrm{dconv}(\boldsymbol{h}_v^{K-1})), \end{cases}
\end{aligned} \tag{2}$$

where $\mathrm{ReLU}(\cdot)$, $\tanh(\cdot)$, $\mathrm{bn}(\cdot)$, $\mathrm{conv}(\cdot)$ and $\mathrm{dconv}(\cdot)$ refer to the ReLU activation function, tanh activation function, batch-normalization layer, convolutional layer and deconvolutional layer, respectively. The encoder E_v takes the stacked images $\{\boldsymbol{v}_i\}_{i=1}^N$ of fashion items as the input and passes the feature maps of each layer $\{\boldsymbol{h}_v^i\}_{i=1}^{K-1}$ to the decoder D_v to generate the final desired try-on template image. We define the output of the decoder is the image-based generated try-on template $P_v = \widetilde{\boldsymbol{h}}_v^K$. To regularize the generated try-on template to imitate the ground truth try-on image P, we adopt the L_1 norm rather than L_2 as L_1 encourages less blurring [21] for the visual try-on template generator as follows:

$$\mathcal{L}_G^v = ||P - P_v||_1. \tag{3}$$

Algorithm 1 Multi-modal Try-on Template Generation

```
Input: The set of positive outfits $\Omega_+$.
Output: The generated try-on template $P_v$ and $P_t$.
Normalize the pixel array of item image $\boldsymbol{v}_i$ into $[0, 1]$.
Initialize the parameters $\boldsymbol{\Theta}_G$ of the MTTG.
repeat
    Randomly draw batch of outfits from $\Omega_+$.
    Update the parameters of the MTTG:
        $\boldsymbol{\Theta}_G^v \leftarrow \boldsymbol{\Theta}_G^v - \eta \frac{\partial \mathcal{L}_G^v}{\partial \boldsymbol{\Theta}_G^v}$,
        $\boldsymbol{\Theta}_G^t \leftarrow \boldsymbol{\Theta}_G^t - \eta \frac{\partial \mathcal{L}_G^t}{\partial \boldsymbol{\Theta}_G^t}$.
until Converge
```

3.2.2 Textual Generator. Intuitively, the textural generator needs to fulfill the task of text-to-image translation, by generating the try-on template image based on the given textual descriptions of composing fashion items. Similar to many existing efforts [19, 36] in this research line, we adopt the deconvolutional network architecture, which consists of an embedding layer and several deconvolutional layers. One naive approach to train the textural generator is to employ the ground truth try-on image to supervise the training of the deconvolutional network, like the visual generator. However, due to the wide domain gap between textual and visual modalities [37], the generated template image can suffer from poor quality [38, 39]. Towards this end, we devise a new scheme for the textual generator by taking into account the latent consistency between the textual description and visual image of the same fashion item. In this manner, the goal of the textual generator is shifted from synthesizing the try-on template image to operating as the encoder of the visual generator and leaving the template generation to the decoder of the visual generator.

Instead of directly feeding the stack of text descriptions $\{\boldsymbol{t}_i\}_{i=1}^N$ of the composing fashion items of the outfit, we introduce the attention mechanism due to the fact that different words can contribute differently to synthesizing the try-on template. For exmaple, "fitted" is more important than "ruffle", as the former one indicates the overall shape of the item while the later one just gives some feature details. Thus, we weight the raw textual information through a to-be-learned attention vector and, similar to [19], we transform it into a dense embedding $\boldsymbol{h}_t^0$ with a fully-connected layer as follows:

$$\boldsymbol{h}_t^0 = \boldsymbol{W}_t^1(\boldsymbol{\alpha} \odot [\boldsymbol{t}_1, \boldsymbol{t}_2, ..., \boldsymbol{t}_N]) + \boldsymbol{b}_t^1, \tag{4}$$

where $\boldsymbol{\alpha}$ is the to-be-learned attention vector, whose different dimension indicates the weight of different word in the different fashion item. $\odot$ is the element-wise multiplication between two vectors. $\boldsymbol{W}_t^1$ and $\boldsymbol{b}_t^1$ are the parameters of the embedding layer. Finally, we deploy $K-1$ deconvolutional layers to decode the latent embeddings of the textual information, which can be formulated as follows:

$$\boldsymbol{h}_t^i = \mathrm{dconv}(\mathrm{ReLU}(\boldsymbol{h}_t^{i-1}))\;|_{i=1}^{K-1}, \tag{5}$$

where $\boldsymbol{h}_t^i$ indicates the intermediate feature maps of the i-th deconvolutional layer.

In order to mimic the work of the encoder in the visual generator, the deconvolutional network in the textual generator should output the counterpart of feature maps derived from the encoder in the visual generator. Accordingly, we define the following objective function to optimize the textual generator:

$$\mathcal{L}_G^t = ||\boldsymbol{h}_v^K - (\boldsymbol{W}_t^2 \boldsymbol{h}_t^0 + \boldsymbol{b}_t^2)||_2 + \sum_{i=i}^{K-1} ||\boldsymbol{h}_v^i - \boldsymbol{h}_t^{K-i}||_2, \tag{6}$$

where $\boldsymbol{W}_t^2$ and $\boldsymbol{b}_t^2$ are transformation parameters to project the latent textual embedding $\boldsymbol{h}_t^0$ into the visual code $\boldsymbol{h}_v^K$. We choose the Euclidean norm $||\cdot||_2$ to make the two vectors close as the most studies [1, 5, 31] do.

Thereafter, once the deconvolutional network has been trained by minimizing the loss function in Eqn. (6), the textual generator can output the try-on template image P_t with the help of the visual decoder D_v as follows:

$$P_t = D_v((\boldsymbol{W}_t^2 \boldsymbol{h}_t^0 + \boldsymbol{b}_t^2), \{\boldsymbol{h}_t^i\}_{i=1}^{K-1}). \tag{7}$$

Ultimately, we reach the final loss function for the multi-modal try-on template generator:

$$\mathcal{L}_G = \mathcal{L}_G^v + \mathcal{L}_G^t. \tag{8}$$

Note that the textual generator works on mimicking the encoder of the visual generator, thereafter, their parameters are optimized independently. The detailed training process of the proposed multi-modal try-on template generator is summarized in Algorithm 1.

3.3 Discrete Item Interaction Modeling

Similar to existing efforts [1, 8], we model the outfit compatibility by uncovering the latent interaction among discrete composing items. Due to the fact that outfits can have different number of fashion items, it is intractable to take the pair-wise scheme that works on evaluating the compatibility between two items, e.g., the top and bottom. We thus resort to the list-wise manner, where each outfit can be treated as a sequence of fashion items with unfixed length. Due to the remarkable success of the bi-directional LSTM [2] in the comprehensive sequence dependency modeling, we involve it for uncovering the latent interaction reside in the well-matched outfits. In particular, we first extract the latent visual feature $\hat{\boldsymbol{v}}_i$ from the image of an item through a pre-trained CNN network and then feed it into the bi-directional LSTM. Here, we take the forward LSTM as an example , while the backward LSTM can be defined similarly. The forward LSTM recurrently takes a visual feature $\hat{\boldsymbol{v}}_i$ as the input and outputs a hidden state $\boldsymbol{h}_d^i$ from $i = 1$ to $i = N$ as follows:

$$\boldsymbol{h}_d^i = \text{LSTM}(\hat{\boldsymbol{v}}_i), i = 1, 2, ..., N. \tag{9}$$

Similarly, the backward LSTM takes the visual feature in a reverse order and maps it to backward output $\widetilde{\boldsymbol{h}}_d^i$. In our context of discrete item interaction modeling, following [2], we maximize the probability of the next item in the outfit given the previous ones in the dual directions to uncover the latent interaction reside in outfits. Accordingly, we have the following loss function for the discrete item interaction modeling:

$$\begin{aligned} \mathcal{L}_d = &- \frac{1}{N} \sum_{i=1}^{N} \log\left(\frac{\exp(\boldsymbol{h}_d^i \hat{\boldsymbol{v}}_{i+1})}{\sum_{\hat{\boldsymbol{v}} \in \hat{\mathcal{V}}} \exp(\boldsymbol{h}_d^i \hat{\boldsymbol{v}})}\right) \\ &- \frac{1}{N} \sum_{i=N-1}^{0} \log\left(\frac{\exp(\widetilde{\boldsymbol{h}}_d^{i+1} \hat{\boldsymbol{v}}_i)}{\sum_{\hat{\boldsymbol{v}} \in \hat{\mathcal{V}}} \exp(\widetilde{\boldsymbol{h}}_d^{i+1} \hat{\boldsymbol{v}})}\right), \end{aligned} \tag{10}$$

where these two terms of loss denote the probability of the prediction in the forward and backward LSTM, respectively. $\hat{\mathcal{V}}$ contains all images of the current batch.

3.4 TryOn-CM

As aforementioned, fashion items have occlusion and deformation when they are put on a person to make a real outfit, which is hard to be modeled by simply modeling items in the discrete manner. Towards this end, we incorporate the try-on appearance to enhance the compatibility modeling of complementary fashion items in a combined manner.

Algorithm 2 Multi-modal Try-on-guided Compatibility Modeling

```
Input: The training set $\Omega$.
Output: The parameters $\Theta$ of the TryOn-CM.
(a) Normalize the pixel array of item image $v_i$ into $[0, 1]$,
(b) Derive $\hat{v}_i$ from a pre-trained CNN model.
Initialize the parameters $\Theta$ of the TryOn-CM.
repeat
    Randomly draw a batch of outfits from $\Omega$.
    (a) Calculate $\mathcal{L}_G$ and $\mathcal{L}_d$ with all positive outfits of the batch,
    (b) Calculate $\mathcal{L}_s$ with the batch of outfits.
    Update the parameters of the TryOn-CM:
        $\Theta \leftarrow \Theta - \eta \frac{\partial \mathcal{L}}{\partial \Theta}$.
until Converge
```

Inspired by the work [40] that utilizes different dense feature to represent different view of the object, in this work, we similarly adopt different features to represent the fashion compatibility of different manners, i.e., the discrete and combined manners. Intuitively, towards the discrete compatibility modeling, we adopt the output of the last time step of the forward LSTM, i.e., $\boldsymbol{h}_d^N$, which encodes the latent dependency of the whole sequence of discrete fashion items in the outfit, as an indicator of the outfit compatibility from the discrete interaction perspective. To incorporate the try-on appearance in the compatibility modeling, one naive approach is to employ an extra CNN-based network to extract the features of the generated try-on template images, which can be fused with $\boldsymbol{h}_d^N$. However, in this manner, the extra network would lead to the performance degradation [41] by deepening the depth of the network. Therefore, we resort to the intermediate outputs of the MTTG, i.e., $\boldsymbol{h}_v^K$ and $\boldsymbol{h}_t^0$, as the references of the try-on appearance. Accordingly, we comprehensively measure the compatibility score for the outfit with a fully-connected layer as follows:

$$s = \sigma(\boldsymbol{W}_s([\boldsymbol{h}_d^N, \boldsymbol{h}_v^K, \boldsymbol{h}_t^0]) + \boldsymbol{b}_s), \tag{11}$$

where $\boldsymbol{W}_s$ and $\boldsymbol{b}_s$ are the layer parameters. σ is the sigmoid function for scaling the outfit compatibility score to [0, 1].

Similar to [24], we cast the compatibility modeling as a binary classification problem and hence utilize the cross entropy loss function to learn the parameters Θ of the proposed scheme. Formally, we have:

$$\mathcal{L}_s = -y \log(s) - (1 - y) \log(1 - s), \tag{12}$$

where y is the ground truth label of the outfit.

Optimization. Ultimately, based on our constructed training set $\Omega = \Omega_+ \cup \Omega_-$, our final objective function of the proposed TryOn-CM scheme can be defined as follows:

$$\min_{\Theta} \mathcal{L} = \sum_{\Omega_+} (\mathcal{L}_G + \mathcal{L}_d) + \sum_{\Omega} \mathcal{L}_s. \tag{13}$$

Notably, $\mathcal{L}_G$ is optimized with only the set of positive outfits Ω_+ rather than the whole training set because the try-on image of the negative outfit is not available. Meanwhile, as mentioned in Subsection 3.3, the bi-directional LSTM is designed to uncovering the latent interaction reside in the well-matched outfits, we also only

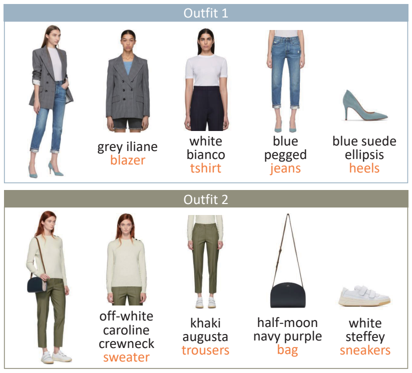

Figure 4: Outfit examples in FOTOS dataset.

utilize Ω_+ to optimize $\mathcal{L}_d$. To give a clear illustration of the training process of the proposed multi-modal try-on-guided compatibility modeling, we summarize the training process in Algorithm 2.

4 FOTOS DATASET

Although several fashion datasets have been made publicly available towards the fashion compatibility modeling, such as FashionVC [1] and Polyvore dataset [2], all of them can only support the discrete item interaction modeling but not the try-on appearance modeling as they lack the try-on ground truth image. To bridge this gap and facilitate the multi-modal try-on-guided compatibility modeling research, we constructed a dataset, named FOTOS, based upon the online fashion-oriented community website SSENSE. In particular, we first collected a seed set of popular fashion items on SSENSE. Then by tracking the "STYLED WITH" section of each seed item's profile, we can obtain the outfit composition, i.e., the set of complementary fashion items, while the ground truth try-on image for this outfit can be found in any item's display page. Some outfit examples are listed in Figure 4.

In order to guarantee the quality of our dataset, we screened out the duplicated outfits as well as those comprising a only single piece of item. The final dataset consists of $11,000$ outfits with $20,318$ fashion items. For each item, we crawled its product image, title and description. As fine-grained categories provided by the website are non-standard, such as "Fur & Shearling" and "V-Necks", we resorted to the last word of the item title that reveals the fine-grained category of the item, e.g., the t-shirt, blouse and hoodie, to derive the item category. For preprocessing, we merged the same meaning words, e.g., jean and jeans, and standardized the word spelling, e.g., skort to skirt. And finally, we derived 141 fine-grained categories. Considering that the clothing items contribute greater to the outfit compatibility than other items, such as shoes and sunglasses, we further divided the fashion items into the clothing items and others. The statistics of our dataset is listed in Table 1, where the total and average indicate the total number of items in the dataset and the average number of items in an outfit, respectively.

5 EXPERIMENTS

To evaluate our proposed method, we conducted extensive experiments on FOTOS by answering the following research questions:

Table 1: Statistics of the FOTOS dataset.

	Category	Item Number	
		Total	Average
Clothing	blazer, parka, jacket, coat, t-shirt, blouse, shirt, polo, sweatshirt, trousers, skirt, gown, bodysuit, ... (55)	13,348	2.57
Other	oxfords, sneakers, loafers, sunglasses, gloves, satchel, briefs, bracelet, ... (86)	6,970	1.45
Total	..., (141)	20,318	4.05

- **RQ1:** Does our TryOn-CM outperform the state-of-the-arts?
- **RQ2:** How about the try-on template generation ability of the proposed MTTG?
- **RQ3:** Does and how does the try-on template help the outfit compatibility modeling?

5.1 Experimental Settings

In this subsection, we first detailed the feature extraction of the visual and textual information in Subsection 5.1.1. And then, the process of building the training set is introduced in Subsection 5.1.2 followed by the scheme structure description in Subsection 5.1.3.

5.1.1 Feature Extraction. In this work, we utilized the advanced deep convolutional neural networks, which have been proven to be the state-of-the-art methods for the image representation learning [42, 43]. In particular, we chose the pre-trained ConvNet [44], which consists of 16 convolutional layers followed by 3 fully connected layers. We used the output of the second fully connected layer, a 4096D vector, as the visual feature $\hat{\boldsymbol{v}}_i$ of an item.

For the text description $\boldsymbol{t}_i$ of the fashion items, we employed the bag-of-words method [45] for its simplicity and robust performance. Analogous to [46], we first constructed an attribute vocabulary based on the words of item titles. We filtered out the low-frequency words of the text since they tend to be the noise. As for the FOTOS dataset, we empirically found that setting the filtered threshold to 40 (i.e., removing attributes whose frequency is less than 40) can deliver an appropriate vocabulary with size of 256, retaining few noisy words but relatively adequate attributes to describe the fashion item. Accordingly, each fashion item can be represented as a 256D boolean vector, where each element indicates whether the corresponding word is in the item textual description.

5.1.2 Training Set Processing. In a sense, our FOTOS dataset only comprises positive outfits. As to build the training set Ω, we need to compose the set of negative outfits Ω_- artificially. In particular, instead of composing negative outfits from scratch, we randomly replaced one item of the positive outfit with another item of the same clothing category. Notably, considering that most outfits comprise less than 5 fashion items, we set the maximum number of items in an outfit as 4. In case that the outfit comprises less than 4 items, we will pad zeros at the end. In addition, we unified the size of fashion item images to 256×256.

5.1.3 Scheme Structures. The layer number K of the visual generator of MTTG is set to 8. The kernel size and stride of the filter in

Table 2: Performance comparison among baselines. † and ‡ denote the statistical significance for p value < 0.05 and p value < 0.01, respectively, compared to the best baseline.

	AUC	MRR	HR@1	HR@10	HR@100	HR@200
RAND	0.502	0.014	0.002	0.020	0.204	0.403
POP	0.496	0.016	0.006	0.024	0.221	0.447
NCR	0.646	0.034	0.012	0.064	0.376	0.616
BPR-DAE	0.742	0.087	0.046	0.165	0.552	0.741
PAICM	0.692	0.057	0.024	0.110	0.468	0.662
LSTM-VSE	0.794	0.118	**0.065**	0.226	0.642	0.809
TryOn-CM	**0.832**‡	**0.134**†	0.061	**0.290**‡	**0.721**‡	**0.852**‡

each layer are set to 5 and 2, respectively. As for the encoder E_v, the number of filters for each convolutional layer is set to 64, 128, 256, 512, 512, 512, 512 and 512, respectively, leading the output $\boldsymbol{h}_v^K$ with size of 1×1×512. Regarding the decoder, the number of filters for D_v is 512, 512, 512, 512, 256, 128, 64 and 3, respectively. As to the textual generator, the textual information of the outfit is first transformed to its 512D embedding $\boldsymbol{h}_d^0$ with a fully-connected layer according to Eqn.(4). The rest structure of the textual generator is the same with the first 7 deconvolutional layers of D_v.

For the discrete item interaction modeling, we first mapped the visual feature $\hat{\boldsymbol{v}}_i$ into a 512D vector with a fully connected layer and then fed it into the bi-directional LSTM network. The number of hidden units of the LSTM is set to 512.

Pertaining to the multi-modal try-on-guided compatibility modeling, we mapped all the compatibility indicators, i.e., $\boldsymbol{h}_d^N$, $\boldsymbol{h}_v^K$ and $\boldsymbol{h}_t^0$, to 128D vectors with respective fully connected layers in order to enhance the ability of the model in dealing with the complex fashion compatibility.

5.2 On Model Comparison (RQ1)

We evaluated the performance on top-n recommendation tasks [47]. For each testing outfit, we randomly picked one composing clothing item as the ground truth item and randomly sampled additional 499 fashion items in the same category with the ground truth item as the candidate items. All candidate items are ranked based on their compatibility scores derived in Eqn. (11), and we adopted the Mean Reciprocal Ranking (MRR) and Hit Rate (HR) at 1, 10, 100, and 200 to assess the complementary item retrieval performance. Meanwhile, we adopted the Area Under Curve (AUC) as another metric to verify the positive/negative outfit classification capability of the model. To prove the effectiveness of the proposed TryOn-CM, we chose the following baselines:

- **RAND.** We randomly ranked the candidate items for the query fashion items.
- **POP.** We ranked the candidate items directly based on its popularity, which is defined as the the number of occurrences of the item in the dataset.
- **BPR-DAE.** We selected the content-based neural scheme [1], which models the coherent relation between different modalities of fashion items via a dual auto-encoder network.
- **LSTM-VSE.** According to [2], we used a bidirectional LSTM to uncover the latent discrete item interaction and the visual-semantic space to inject attribute and category information as a regularization for training the LSTM. We chose the loss of the bi-directional LSTM as an indicator of the outfit score.
- **PAICM.** The method [31] involves matrix factorization to learn some compatible and incompatible prototypes, which can be used to guide the outfit compatibility modeling.
- **NCR.** This work [32] makes full use of the textual information of fashion items and models the outfit compatibility from the semantic and lexical aspects.

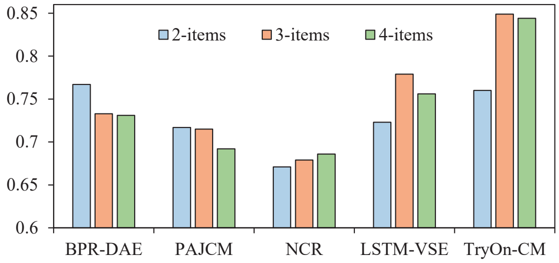

Figure 5: Results of the number test, which is used to evaluate the ability of different methods to handle the outfits with different numbers of fashion items.

It is worth noting that BPR-DAE and PAICM are designed for analyzing the fashion compatibility between item pairs (e.g., the top-bottom pair). To fit them with the context of outfit compatibility modeling that involves more than two items, we divided items in FOTOS into two groups: tops and bottoms, according to their fine-grained categories. And then, we used the average compatibility score of top-bottom pairs as the outfit compatibility. For example, suppose there is a jacket, shirt and skirt in an outfit. Then the fashion compatibility is measured as the average compatibility of the jacket-skirt pair and the shirt-skirt pair.

Table 2 shows the performance comparison among different approaches. Overall, TryOn-CM achieves the best performance with respect to almost all evaluation metrics, demonstrating the superiority of the proposed method over these baselines. Compared with the naive methods (i.e., POP and RAND), NCR promotes the performance by fully exploring the textual information of items. However, the neglect of the visual appearance, which is also the valuable cue for the compatibility modeling, makes it inferior to the methods that utilize the multi-modal data (i.e., BPR-DAE, Bi-LSTM). Besides, the pair-wise methods (i.e., BPR-DAE, PAICM) present worse performance than the list-wise methods (e.g., Bi-LSTM, TryOn-CM). One of the possible reasons is that the pair-wise methods are designed to cope with top-bottom pairs, therefore, they are not suitable for handling outfits with multiple items. However, the list-wise methods are designed for a list of items, which performs better in outfits with multiple items.

To gain a thorough understanding of the aforementioned pair-wise methods and list-wise methods fit for the outfits with different item numbers, we further involved an extra evaluation, named "number test". In particular, we divided our dataset into 3 parts: the outfits with 1) two items, 2) three items and 3) four items. Without loss of generality, we listed the performance of different methods with respect to AUC in Figure 5. From the figure we can see that BPR-DAE performs better for outfits with two items as compared to those with multiple (i.e., 3 and 4) ones, indicating its superiority in

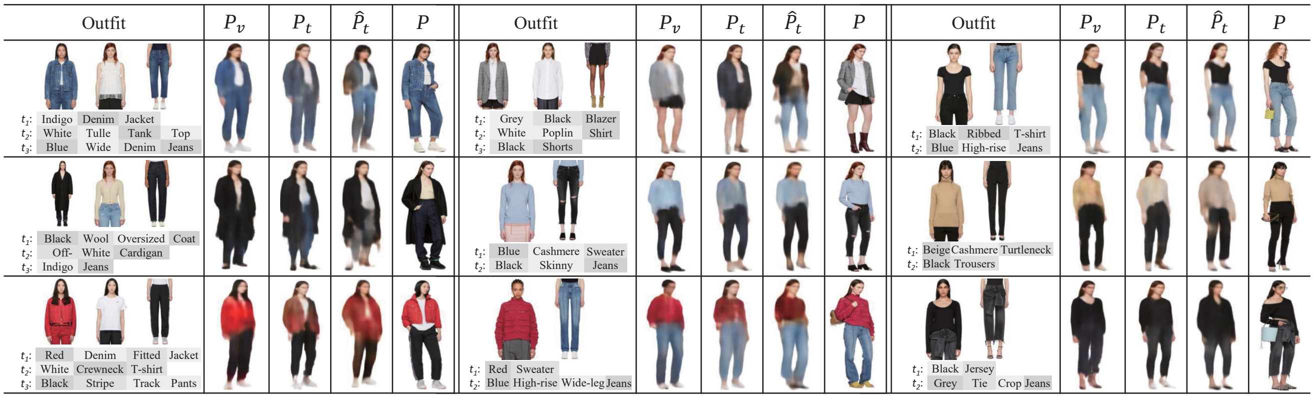
Outfit
P_v
P_t
$\hat{P}_t$
P
t_1: Indigo Denim Jacket
t_2: White Tulle Tank Top
t_3: Blue Wide Denim Jeans
t_1: Grey Black Blazer
t_2: White Poplin Shirt
t_3: Black Shorts
t_1: Black Ribbed T-shirt
t_2: Blue High-rise Jeans
t_1: Black Wool Oversized Coat
t_2: Off- White Cardigan
t_3: Indigo Jeans
t_1: Blue Cashmere Sweater
t_2: Black Skinny Jeans
t_1: Beige Cashmere Turtleneck
t_2: Black Trousers
t_1: Red Denim Fitted Jacket
t_2: White Crewneck T-shirt
t_3: Black Stripe Track Pants
t_1: Red Sweater
t_2: Blue High-rise Wide-leg Jeans
t_1: Black Jersey
t_2: Grey Tie Crop Jeans

Figure 6: Examples of the multi-modal try-on template generation. P_v and P_t are the generated templates from the visual and textual modality, respectively. $\hat{P}_t$ is the template generated by the naive text-to-image method.

the pair-wise compatibility modeling. In addition, PAICM narrows this gap probably because that it measures the outfit compatibility by the auxiliary compatible and incompatible prototypes in stead of the merely item interaction. The list-wise methods consistently do better on outfits with multiple items while worse on outfits with two items. Therefore, we concluded that the pair-wise methods work better on the outfits with two items while the list-wise methods perform well with multiple items. Besides, we found that BPR-DAE and TryOn-CM achieve comparable performance while dealing with the outfits with two items (i.e., BPR-DAE: 0.767 and TryOn-CM: 0.760). This may be because that outfits with two items has less occlusion or deformation when tried on. Therefore, the benefit of incorporating the try-on appearance modeling in the compatibility modeling for outfits with only two items is rather limited.

5.3 On Qualitative Analysis (RQ2)

To intuitively show the try-on template generation ability of MTTG, we visualized some examples in Figure 6. The visual and textual information of the outfit are listed in the column "Outfit". The try-on template generated by the visual and textual modality are displayed in the columns "P_v" and "P_t", respectively, while the real try-on image is shown in the column "P". In addition, we marked the contribution of different words by shading the color, where a darker color indicates a greater contribution.

From the figure we can see, both P_v and P_t are representative for the real try-on image. In particular, P_v is closer to the real try-on appearance probably because the image of the fashion item brings more information than its textual description. For example, as for the outfit in the lower right of the figure, the text "Black Jersey" of the fashion item cannot cover the wide-collar attribute of the item delivered by its image. Consequently, P_v successfully synthesizes this feature while P_t fails. Pertaining to the outfit in the mid-top, the textual generator mainly focuses on the "Black" of the outer and generates the wrong color for P_t. Besides, during the template generation, we found that the color and category play more important roles than other attributes, for example, as for the outfit in the mid bottom, the "Red", "Blue" and "Jeans" take the bigger weights than "High-rise" and "Wide-leg". The reason may be that they represent the main features of fashion items, while other words describe the trivial details.

Besides, to validate the effectiveness of our proposed textual generator in bridging the domain gap between textual and visual modalities, we compared the try-on templates generated by our MTTG and the naive text-to-image method, which directly employs the ground truth image to supervise the textual try-on template generator. As we can see from Figure 6, the templates generated by the naive method (see the column $\hat{P}_t$) suffer from either the poor image quality or the low consistence with the real try-on appearance. This proves that our proposed textual generator successfully narrows the domain gap between textual and visual modalities and generates images with the higher quality.

5.4 On Ablation Study (RQ3)

To evaluate the importance of the try-on appearance in the compatibility modeling, we further compared TryOn-CM with its derivation: the discrete compatibility modeling (DCM), which can be effortlessly derived by removing $\mathcal{L}_G$ from the loss function in Eqn. (13). Moreover, to obtain a thorough understanding, we conducted the comparative experiments with different modality configurations: VCM and TCM, which adopt the try-on template generated by the only visual and textual information, respectively. In particular, VCM and TCM can be obtained by changing the outfit representation to $[\boldsymbol{h}_N, \boldsymbol{h}_v^K]$ and $[\boldsymbol{h}_N, \boldsymbol{h}_t^0]$ in Eqn. (11), respectively. Furthermore, we conducted an extra experiment on the comparison of the proposed textual generator and the normal text-to-image method, named orig-TCM that optimizes the textual generator via the ground truth image. We extracted the $\hat{\boldsymbol{h}}_t^0$, which corresponds to $\boldsymbol{h}_t^0$, as the compatibility indicator to form the outfit representation $[\boldsymbol{h}_N, \hat{\boldsymbol{h}}_t^0]$. Note that all the derivations are retrained from scratch.

Table 3 shows the results of the ablation study, where "V" and "T" indicate the visual and textual modality, respectively. From the table, we can see that our model consistently surpasses all methods across all metrics, which verifies the effectiveness of our proposed method. In particular, both VCM and TCM exceed DCM in varying degrees, suggesting that it is necessary to consider the try-on appearance during the outfit compatibility modeling. More

Table 3: Results of the ablation study. Each method is ticked with its involved components, where "DCM", "V" and "T" correspond to the discrete compatibility modeling, the try-on template from the visual and textual information, respectively.

	DCM	V	T	AUC	MRR	HR@1	HR@10	HR@100	HR@200
DCM	√			0.816	0.118	0.059	0.237	0.687	0.843
orig-TCM	√		√	0.812	0.110	0.055	0.220	0.678	0.835
TCM	√		√	0.821	0.120	0.060	0.240	0.697	0.845
VCM	√	√		0.824	0.123	**0.061**	0.244	0.706	0.846
TryOn-CM	√	√	√	**0.832**	**0.134**	**0.061**	**0.290**	**0.721**	**0.852**

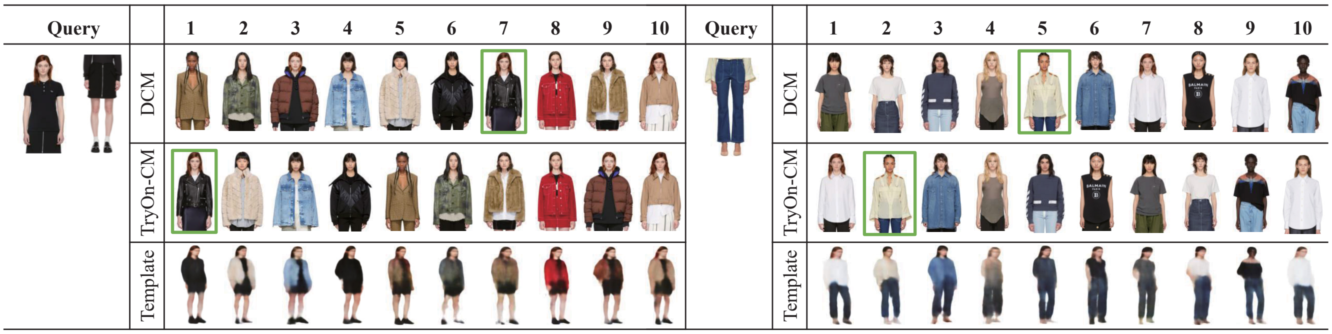

Figure 7: Ranking results of the discrete compatibility modeling (DCM) and our multi-modal try-on-guided compatibility modeling. The positive complementary item of the query is circled in the green box and we visualize the try-on template of the outfit generated by the try-on template generator in the last line.

specifically, VCM slightly outperforms TCM which implies that the visual information plays a more important role than the textual information in the try-on template generation. Finally, we found the orig-TCM performs the worst, even worse than the pure discrete interaction modeling method DCM. One reason may be that this method cannot fully bridge the wide gap between the visual and textual modality and hence fail to generate the proper template (see the column $\hat{P}_t$) directly with the text generator supervised by the ground truth image. Moreover, the misleading try-on templates generated by orig-TCM degrade the performance of the discrete item interaction modeling.

To have a deep understanding of how the try-on template help the compatibility modeling, we compare the ranking results between DCM and the proposed TryOn-CM in Figure 7. Without loss of generality, we only select 10 candidates to rank for the query, including one positive item and 9 negative items. Towards the more intuitive illustration, we also visualized the synthesized try-on template of each candidate item paired up with the query items at the last row of the Figure 7. As we can see, benefited from the synthesized try-on templates, our proposed TryOn-CM ranked the positive complementary item (circled by the green box) higher than the discrete item interaction modeling.

6 CONCLUSION AND FUTURE WORK

Fashion items often present occlusion and deformation when they are tried on, which complicates the challenge of the fashion compatibility modeling merely with the discrete item interaction modeling. In this work, we take into account the try-on appearance of the outfit into the discrete item interaction modeling to comprehensively analyze the compatibility from both discrete and combined manners. In particular, we propose a multi-model try-on-guided compatibility modeling scheme that first generates the try-on template of an outfit and then combines it with the discrete item interaction to model the outfit compatibility. To evaluate the proposed method, we construct a new dataset from the fashion website SSENSE, consisting of 11,000 well-matched outfits and their corresponding try-on images. Extensive experiments have been conducted over our newly collected dataset, and verified the necessity of considering the try-on appearance during the outfit compatibility modeling. In addition, we find that the visual information of fashion items brings more details than the textual information during the try-on template generation.

However, the try-on template generated by MTTG is vague and loses certain details compared with the real try-on image. Therefore, in the future, we plan to devise a more robust generator to synthesize the try-on template and model the outfit compatibility directly based on it. Besides, as the personalized factors, such as the personal identity and body shape, also play the important role in the fashion analysis, we plan to take them into the try-on template generation and synthesize a personalized template.

ACKNOWLEDGEMENTS

This work is supported by the National Key Research and Development Project of New Generation Artificial Intelligence, No.: 2018AAA0102502; the National Natural Science Foundation of China, No.: 61772310, No.: 61702300 and No.: U1936203; the Shandong Provincial Natural Science Foundation, No.: ZR2019JQ23; the Shandong Provincial Key Research and Development Program, No.: 2019JZZY010118; the Innovation Teams in Colleges and Universities in Jinan, No.: 2018GXRC014; the Fundamental Research Funds of Shandong University.

REFERENCES

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

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

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

[4] Wei-Lin Hsiao and Kristen Grauman. Creating capsule wardrobes from fashion images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7161–7170, 2018.

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

[6] Lizi Liao, Yunshan Ma, Xiangnan He, Richang Hong, and Tat-Seng Chua. Knowledge-aware multimodal dialogue systems. In *ACM Conference on Multimedia*, pages 801–809, 2018.

[7] Liqiang Nie, Wenjie Wang, Richang Hong, Meng Wang, and Qi Tian. Multimodal dialog system: Generating responses via adaptive decoders. In *ACM Conference on Multimedia*, pages 1098–1106, 2019.

[8] Xuemeng Song, Fuli Feng, Xianjing Han, Xin Yang, Wei Liu, and Liqiang Nie. Neural compatibility modeling with attentive knowledge distillation. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 5–14, 2018.

[9] Long Chen and Yuhang He. Dress fashionably: Learn fashion collocation with deep mixed-category metric learning. In *AAAI Conference on Artificial Intelligence*, pages 2103–2110, 2018.

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

[11] Xintong Han, Zuxuan Wu, Zhe Wu, Ruichi Yu, and Larry S. Davis. VITON: an image-based virtual try-on network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7543–7552, 2018.

[12] Na Zheng, Xuemeng Song, Zhaozheng Chen, Linmei Hu, Da Cao, and Liqiang Nie. Virtually trying on new clothing with arbitrary poses. In *ACM Conference on Multimedia*, pages 266–274, 2019.

[13] Yujie Lin, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Jun Ma, and Maarten de Rijke. Improving outfit recommendation with co-supervision of fashion generation. In *World Wide Web Conference*, pages 1095–1105, 2019.

[14] Rumelhart D E, Hinton G E, and Williams R J. Learning representations by back-propagating errors. *Nature*, 303(6088):533–536, 1986.

[15] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning*, pages 1096–1103, 2008.

[16] Kingma D P and Welling M. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

[17] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Conference Neural Information Processing Systems*, pages 2672–2680, 2014.

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

[19] Scott E. Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International Conference on Machine Learning*, pages 1060–1069, 2016.

[20] Wei Shen and Rujie Liu. Learning residual images for face attribute manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1225–1233, 2017.

[21] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5967–5976, 2017.

[22] Zhonghua Wu, Guosheng Lin, Qingyi Tao, and Jianfei Cai. M2e-try on net: Fashion from model to everyone. In *ACM Conference on Multimedia*, pages 293–301, 2019.

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention Conference*, pages 234–241, 2015.

[24] Yuncheng Li, Liangliang Cao, Jiang Zhu, and Jiebo Luo. Mining fashion outfit composition using an end-to-end deep learning approach on set data. *IEEE Trans. Multimedia*, 19(8):1946–1955, 2017.

[25] Yujie Lin, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Jun Ma, and Maarten de Rijke. Improving outfit recommendation with co-supervision of fashion generation. In *World Wide Web Conference*, pages 1095–1105, 2019.

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

[27] Wenguan Wang, Yuanlu Xu, Jianbing Shen, and Song-Chun Zhu. Attentive fashion grammar network for fashion landmark detection and clothing category classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4271–4280, 2018.

[28] Jingyuan Liu and Hong Lu. Deep fashion analysis with feature map upsampling and landmark-driven attention. In *European Conference on Computer Vision Workshops*, pages 30–36, 2018.

[29] Zunlei Feng, Zhenyun Yu, Yezhou Yang, Yongcheng Jing, Junxiao Jiang, and Mingli Song. Interpretable partitioned embedding for customized multi-item fashion outfit composition. In *ACM International Conference on Multimedia Retrieval*, pages 143–151, 2018.

[30] Maryam Ziaeefard, Jaime R. Camacaro, and Carolina Bessega. Hierarchical feature map characterization in fashion interpretation. In *Conference on Computer and Robot Vision*, pages 88–94, 2018.

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

[32] Suthee Chaidaroon, Yi Fang, Min Xie, and Alessandro Magnani. Neural compatibility ranking for text-based fashion matching. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1229–1232, 2019.

[33] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *International Conference on Computer Vision*, pages 2242–2251, 2017.

[34] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Conference on Neural Information Processing Systems*, pages 465–476, 2017.

[35] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[36] Seunghoon Hong, Dingdong Yang, Jongwook Choi, and Honglak Lee. Inferring semantic layout for hierarchical text-to-image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7986–7994, 2018.

[37] Richang Hong, Lei Li, Junjie Cai, Dapeng Tao, Meng Wang, and Qi Tian. Coherent semantic-visual indexing for large-scale image retrieval in the cloud. *IEEE Trans. Image Processing*, 26(9):4128–4138, 2017.

[38] Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. DM-GAN: dynamic memory generative adversarial networks for text-to-image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5802–5810, 2019.

[39] Yali Cai, Xiaoru Wang, Zhihong Yu, Fu Li, Peirong Xu, Yueli Li, and Lixian Li. Dualattn-gan: Text to image synthesis with dual attentional generative adversarial network. *IEEE Access*, 7:183706–183716, 2019.

[40] Richang Hong, Zhenzhen Hu, Ruxin Wang, Meng Wang, and Dacheng Tao. Multi-view object retrieval via multi-scale topic models. *IEEE Trans. Image Processing*, 25(12):5814–5827, 2016.

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

[42] Jingyuan Chen, Xuemeng Song, Liqiang Nie, Xiang Wang, Hanwang Zhang, and Tat-Seng Chua. Micro tells macro: Predicting the popularity of micro-videos via a transductive model. In *ACM Conference on Multimedia*, pages 898–907, 2016.

[43] Hanwang Zhang, Xindi Shang, Wenzhuo Yang, Huan Xu, Huan-Bo Luan, and Tat-Seng Chua. Online collaborative learning for open-vocabulary visual classifiers. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2809–2817, 2016.

[44] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[45] Rongrong Ji, Xing Xie, Hongxun Yao, and Wei-Ying Ma. Mining city landmarks from blogs by graph modeling. In *ACM Conference on Multimedia*, pages 105–114, 2009.

[46] Liqiang Nie, Meng Wang, Zheng-Jun Zha, Guangda Li, and Tat-Seng Chua. Multimedia answering: enriching text QA with media information. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 695–704, 2011.

[47] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *ACM Conference on Recommender Systems*, pages 39–46, 2010.