Improving Outfit Recommendation with Co-supervision of Fashion Generation

Lin, Y.; Ren, P.; Chen, Z.; Ren, Z.; Ma, J.; de Rijke, M.

DOI
10.1145/3308558.3313614

Publication date
2019

Document Version
Final published version

Published in
The Web Conference 2019

Improving Outfit Recommendation with Co-supervision of Fashion Generation

Yujie Lin*
Shandong University
Jinan, China
yu.jie.lin@outlook.com

Pengjie Ren*
University of Amsterdam
Amsterdam, The Netherlands
p.ren@uva.nl

Zhumin Chen
Shandong University
Jinan, China
chenzhumin@sdu.edu.cn

Zhaochun Ren
Shandong University
Jinan, China
zhaochun.ren@sdu.edu.cn

Jun Ma
Shandong University
Jinan, China
majun@sdu.edu.cn

Maarten de Rijke
University of Amsterdam
Amsterdam, The Netherlands
derijke@uva.nl

ABSTRACT

The task of fashion recommendation includes two main challenges: *visual understanding* and *visual matching*. Visual understanding aims to extract effective visual features. Visual matching aims to model a human notion of compatibility to compute a match between fashion items. Most previous studies rely on recommendation loss alone to guide visual understanding and matching. Although the features captured by these methods describe basic characteristics (e.g., color, texture, shape) of the input items, they are not directly related to the visual signals of the output items (to be recommended). This is problematic because the aesthetic characteristics (e.g., style, design), based on which we can directly infer the output items, are lacking. Features are learned under the recommendation loss alone, where the supervision signal is simply whether the given two items are matched or not.

To address this problem, we propose a neural co-supervision learning framework, called the FAshion Recommendation Machine (FARM). FARM improves visual understanding by incorporating the supervision of generation loss, which we hypothesize to be able to better encode aesthetic information. FARM enhances visual matching by introducing a novel layer-to-layer matching mechanism to fuse aesthetic information more effectively, and meanwhile avoiding paying too much attention to the generation quality and ignoring the recommendation performance.

Extensive experiments on two publicly available datasets show that FARM outperforms state-of-the-art models on outfit recommendation, in terms of AUC and MRR. Detailed analyses of generated and recommended items demonstrate that FARM can encode better features and generate high quality images as references to improve recommendation performance.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Outfit matching; Fashion generation; Fashion recommendation

ACM Reference Format:
Yujie Lin, Pengjie Ren*, Zhumin Chen, Zhaochun Ren, Jun Ma, and Maarten de Rijke. 2019. Improving Outfit Recommendation with Co-supervision of Fashion Generation. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3308558.3313614

1 INTRODUCTION

Fashion recommendation has attracted increasing attention [14, 18, 20] for its potentially wide applications in fashion-oriented online communities such as, e.g., Polyvore[1] and Chictopia.[2] By recommending fashionable items that people may be interested in, fashion recommendation can promote the development of online retail by stimulating people's interests and participation in online shopping. In this paper, we target *outfit recommendation*, that is, given a top (i.e., upper garment), we need to recommend a list of bottoms (e.g., trousers or skirts) from a large collection that best match the top, and vice versa. Specifically, we allow users to provide some descriptions as conditions that the recommended items should accord with as much as possible.

Unlike conventional recommendation tasks, outfit recommendation faces two main challenges: *visual understanding* and *visual matching*. Visual understanding aims to extract effective features by building a deep understanding of fashion item images. Visual matching requires modeling a human notion of the compatibility between fashion items [41], which involves matching features such as color and shape etc. Early studies into outfit recommendation rely on feature engineering for visual understanding and traditional machine learning for visual matching [16]. For example, Iwata et al. [15] define three types of feature, i.e., color, texture and local descriptors such as Scale Invariant Feature Transform (SIFT) (for visual understanding), and propose a recommendation model based on Graphical Models (GM) (for visual matching). Liu et al. [29] define five types of feature including Histograms of Oriented Gradient (HOG) [9], Local Binary Pattern (LBP) [1], color moment, color histogram and skin descriptor [5] (for visual understanding), and propose a latent Support Vector Machine (SVM) based recommendation model (for visual matching).

*Co-first author.

[1] http://www.polyvore.com/
[2] http://www.chictopia.com/

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313614

Recently, neural networks have been applied to address the challenges of fashion recommendation: Song et al. [41] use a pre-trained Convolutional Neural Network (CNN) (on ImageNet) to extract visual features (for visual understanding). Then, they employ a separate Bayesian Personalized Ranking (BPR) [35] method to exploit pairwise preferences between tops and bottoms (for visual matching). Lin et al. [28] propose to train feature extraction (for visual understanding) and preference prediction (for visual matching) in a single back-propagation scheme. They introduce a mutual attention mechanism into CNN to improve feature extraction. The visual features captured by these methods only describe basic characteristics (e.g., color, texture, shape) of the input items, which lack aesthetic characteristics (e.g., style, design) to describe the output items (to be recommended). Visual understanding and matching are conducted based on recommendation loss alone, where the supervision signal is just whether two given items are matched or not and no supervision is available to directly connect the visual signals of the fashion items. Recently, some studies have realized the importance of modeling aesthetic information. For example, Ma et al. [30] build a universal taxonomy to quantitatively describe aesthetic characteristics of clothing. Yu et al. [46] propose to encode aesthetic information by pre-training models on aesthetic assessment datasets. However, none of them is for outfit recommendation and none improves visual understanding and matching like we do.

In this paper, we address the challenges of outfit recommendation from a novel perspective by proposing a neural co-supervision learning framework, called *FAshion Recommendation Machine* (FARM). FARM enhances visual understanding and visual matching with the *joint supervision* of generation and recommendation learning. Let us explain. By incorporating the generation process as a supervision signal, FARM is able to encode more aesthetic characteristics, based on which we can directly generate the output items. FARM enhances visual matching by incorporating a novel layer-to-layer matching mechanism to evaluate the matching score of generated and candidate items at different neural layers; in this manner FARM fuses the generation features from different visual levels to improve the recommendation performance. This layer-to-layer matching mechanism also ensures that FARM avoids paying too much attention to the generation quality and ignoring the recommendation performance. To the best of our knowledge, FARM is the first end-to-end learning framework that improves outfit recommendation with joint modeling of fashion generation.

Extensive experimental results conducted on two publicly available datasets show that FARM outperforms state-of-the-art models on outfit recommendation, in terms of AUC and MRR. To further demonstrate the advantages of FARM, we conduct several analyses and case studies.

To sum up, our contributions can be summarized as follows:

- We propose a neural co-supervision learning framework, FARM, for outfit recommendation that simultaneously yields recommendation and generation.
- We propose a layer-to-layer matching mechanism that acts as a bridge between generation and recommendation, and improves recommendation by leveraging generation features.
- Our proposed approach is shown to be effective in experiments on two large-scale datasets.

2 RELATED WORK

We survey related work on fashion recommendation by focusing on the two main challenges in the area: visual understanding and visual matching.

2.1 Visual understanding

One branch of studies aims at extracting better features to improve the visual understanding of fashion items.

For instance, Iwata et al. [15] propose a recommender system for clothing coordinates using full-body photographs from fashion magazines. They extract visual features, such as color, texture and local descriptors such as SIFT, and use a probabilistic topic model for visual understanding of coordinates from these features. Liu et al. [29] target occasion-oriented clothing recommendation. Given a user-input event, e.g., wedding, shopping or dating, their model recommends the most suitable clothing from the user's own clothing photo album. They adopt clothing attributes (e.g., clothing category, color, pattern) for better visual understanding. Jagadeesh et al. [16] describe a visual recommendation system for street fashion images. They mainly focus on color modeling in terms of visual understanding.

The studies listed above achieve visual understanding mostly based on feature engineering and conventional machine learning techniques. With the availability of large scale fashion recommendation datasets and the rapid development of deep learning models, several recent publications turn to neural networks for fashion recommendation. CNNs are certainly widely employed [26, 31]. Ma et al. [30] build a taxonomy based on a theory of aesthetics to describe aesthetic features of fashion items quantitatively and universally. Then they capture the internal correlation in clothing collocations by a novel fashion-oriented multi-modal deep learning based model. Song et al. [41] use a pre-trained CNN on ImageNet to extract visual features. Then, to improve visual understanding with contextual information (such as titles and categories), they propose to use multi-modal auto-encoders to exploit the latent compatibility of visual and contextual features. Han et al. [11] enrich visual understanding by incorporating sequential information by using a Bidirectional Long Short-Term Memory Network (Bi-LSTM) to predict the next item conditioned on previous ones. They further inject attribute and category information as a kind of regularization to learn a visual-semantic space by regressing visual features to their semantic representations. Kang et al. [20] use a CNN-F [7] to learn image representations and train a personalized fashion recommendation system jointly. Besides, they devise a personalized fashion design system based on the learned CNN-F and user representations. Yu et al. [46] propose to introduce aesthetic information into fashion recommendation. To achieve this, they extract aesthetic features using a pre-trained brain-inspired deep structure on the aesthetic assessment task. Lin et al. [28] enhance visual understanding by jointly modeling fashion recommendation and user comment generation, where the visual features learned with a CNN are enriched because they are related to the generation of user comments.

Even though there is a growing number of studies on better visual understanding for fashion recommendation, none of them takes fashion generation into account like we do in this paper.

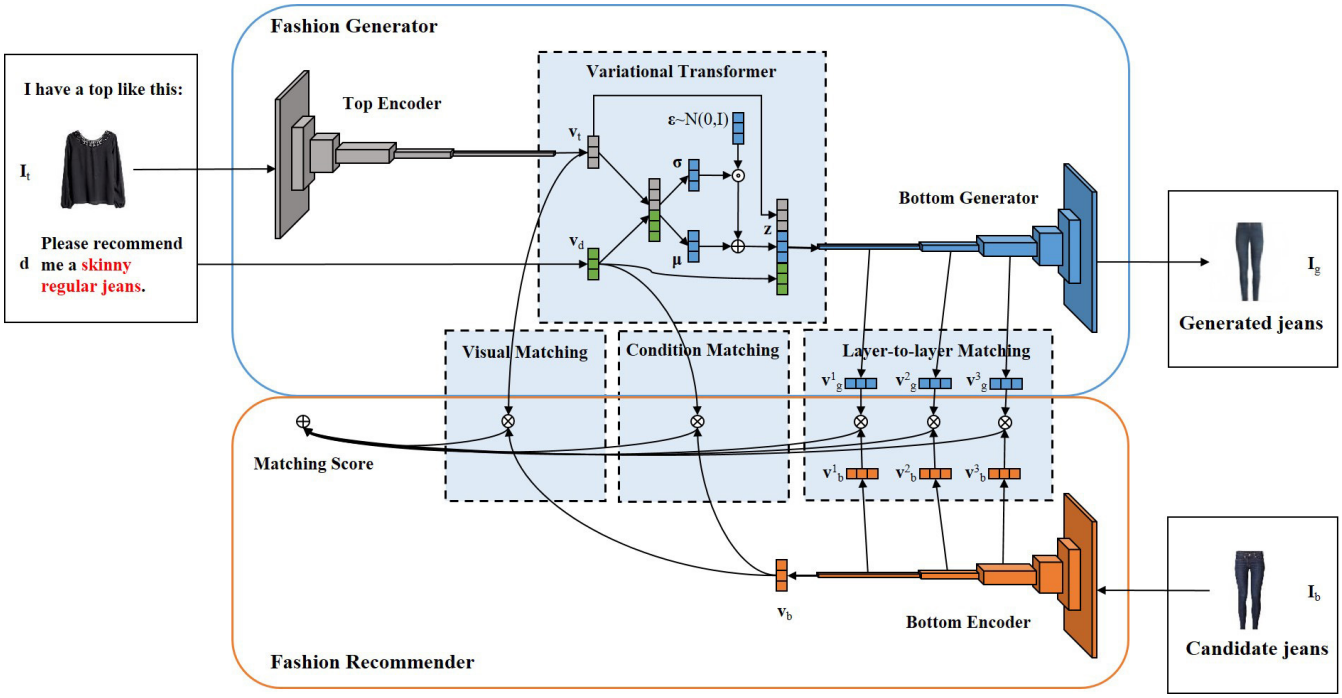


Figure 1: Overview of FARM. The fashion generator (top) uses a variational transformer to learn a special Gaussian distribution for a given top image I_t and a given bottom description d. It then generates a bottom image I_g to match I_t and d. The fashion recommender (bottom) evaluates the matching score between the recommended bottom image I_b and (I_t, d) pair from three angles, i.e., visual matching, description matching, and layer-to-layer matching.

2.2 Visual matching

Early studies into visual matching are based on conventional machine learning methods. Iwata et al. [15] use a topic model to learn the relation between photographs and recommend a bottom that has the closest topic proportions to those of the given top. Liu et al. [29] employ an SVM for recommendation, which has a term describing the relationship between visual features and attributes of tops and bottoms. Simo-Serra et al. [38] predict the popularity of an outfit to implicitly learn its compatibility by a Conditional Random Field (CRF) model. McAuley et al. [31] measure the compatibility between clothes by learning a distance metric with pre-trained CNN features. Hu et al. [14] propose a functional pairwise interaction tensor factorization method to model the interactions between fashion items of different categories. Hsiao and Grauman [13] develop a submodular objective function to capture the key ingredients of visual compatibility in outfits. They propose a topic model namely Correlated Topic Models (CTM) to generate compatible outfits learned from unlabeled images of people wearing outfits.

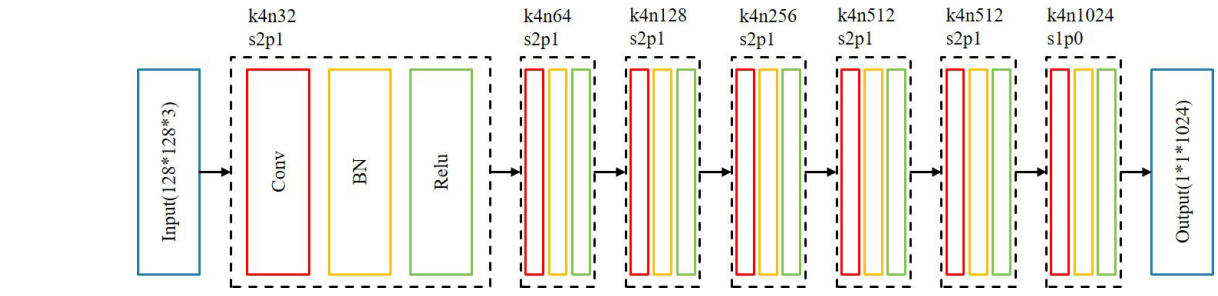
Recently, deep learning methods have been used widely in the fashion recommendation community. Veit et al. [43] train an end-to-end Siamese CNN network to learn a feature transformation from images to a latent compatibility space. Oramas and Tuytelaars [33] mine mid-level elements from CNNs to model the compatibility of clothes. Li et al. [26] use a Recurrent Neural Network (RNN) to predict whether an outfit is popular, which also implicitly learns the compatibility relation between fashion items. Han et al. [11] further train a Bi-LSTM to sequentially predict the next item conditioned on the previous ones for learning their compatibility relationship. Song et al. [41] employ a dual auto-encoder network to learn the latent compatibility space where they use the BPR model to jointly model the relation between visual and contextual modalities and implicit preferences among fashion items. Song et al. [40] consider the knowledge about clothing matching and follow a teacher-student scheme to encode the fashion domain knowledge in a traditional neural network. And they introduce an attentive scheme to the knowledge distillation procedure to flexibly assign rule confidence. Nakamura and Goto [32] present an architecture containing three subnetworks, i.e., VSE (Visual-Semantic Embedding), Bi-LSTM and SE (Style Embedding) modules, to model the matching relation between different items to generate outfits. Lin et al. [28] propose a mutual attention mechanism into CNNs to model the compatibility between different parts of images of fashion items.

Although there are many studies on improving visual matching, none of them considers connecting it with fashion generation.

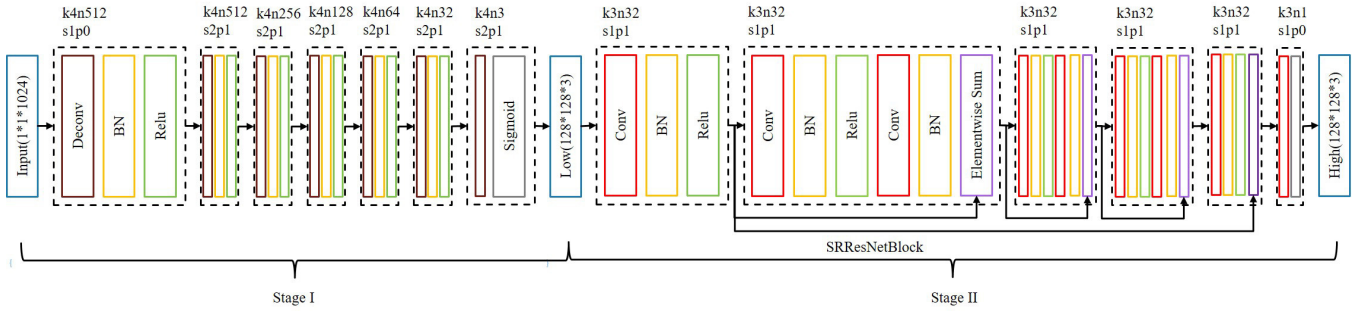
3 NEURAL FASHION RECOMMENDATION

3.1 Overview

Given a top t from a pool $\mathcal{T} = \{t_1, t_2, \ldots, t_{N_t}\}$ and a user's description d for the target bottom, the *bottom recommendation task* is to recommend a list of bottoms from a candidate pool $\mathcal{B}$ =



(a) Encoder



(b) Generator

Figure 2: Details of the encoder and the generator in FARM, where k represents kernel size, n represents the number of channels, s represents strides and p represents padding.

$\{b_1, b_2, \ldots, b_{N_b}\}$. Similarly, the *top recommendation task* is to recommend a ranked list of tops for a given bottom and top description pair. Here, we use bottom recommendation as the setup to introduce our framework FARM.

As shown in Figure 1, FARM consists of two parts, i.e., a fashion generator (for visual understanding) and a fashion recommender (for visual matching), where the fashion generator is actually an auxiliary module for recommendation. For the fashion generator, we use a CNN as the top encoder to extract the visual features from a given top image $\mathbf{I}_t$. We learn the semantic representation for the bag-of-words vector $\mathbf{d}$ of a given bottom description. Then we use a variational transformer to learn the mapping from the bottom distribution to a specific Gaussian distribution that is based on the visual features of $\mathbf{I}_t$ and the semantic representation of $\mathbf{d}$. Finally, we sample a random vector from the Gaussian distribution and input it to a DeConvolutional Neural Network (DCNN) [48] (as bottom generator) to generate a bottom image $\mathbf{I}_g$ that matches $\mathbf{I}_t$ and $\mathbf{d}$, which explicitly forces the top encoder to encode more aesthetic matching information into the visual features. For the fashion recommender, we also employ a CNN as the bottom encoder to extract the visual features from a candidate bottom image $\mathbf{I}_b$. Then we evaluate the matching score between $\mathbf{I}_b$ and $(\mathbf{I}_t, \mathbf{d})$ pair from three angles, namely the visual matching between $\mathbf{I}_b$ and $\mathbf{I}_t$, the description matching between $\mathbf{I}_b$ and $\mathbf{d}$, and the layer-to-layer matching between $\mathbf{I}_b$ and $\mathbf{I}_g$ which leverages the generation information to improve the recommendation. FARM jointly trains the fashion generator and fashion recommender. Next we will detail each of these two main parts.

3.2 Fashion generator

Given an image $\mathbf{I}_t$ of a top t and the bag-of-words vector $\mathbf{d}$ of a bottom description d, the fashion generator needs to generate a bottom image $\mathbf{I}_g$ that not only matches $\mathbf{I}_t$, but also meets $\mathbf{d}$ as much as possible. We enforce the extracted visual features from $\mathbf{I}_t$ to contain the information about its matching bottom by using the generator as a supervision signal. The generated image can be seen as a reference for recommendation.

Specifically, for a generated bottom image $\mathbf{I}_g$ that matches $\mathbf{I}_t$ and $\mathbf{d}$, the aim of the fashion generator is to maximize Eq. 1:

$$p(\mathbf{I}_g|\mathbf{I}_t, \mathbf{d}) = \int_{\mathbf{z}} p(\mathbf{I}_g|\mathbf{z}, \mathbf{I}_t, \mathbf{d}) p(\mathbf{z}|\mathbf{I}_t, \mathbf{d}) \mathrm{d}\mathbf{z}, \tag{1}$$

where $p(\mathbf{z}|\mathbf{I}_t, \mathbf{d})$ is the top encoder, $p(\mathbf{I}_g|\mathbf{z}, \mathbf{I}_t, \mathbf{d})$ is the bottom generator, and $\mathbf{z}$ is the latent variable. Because the integral of the marginal likelihood shown in Eq. 1 is intractable, inspired by variational inference [4], we first find the Evidence Lower BOund (ELBO) of $p(\mathbf{I}_g|\mathbf{I}_t, \mathbf{d})$, as shown in Eq. 2:

$$\begin{aligned} \mathrm{ELBO} = \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z}|\mathbf{I}_t, \mathbf{d})}[\log p(\mathbf{I}_g|\mathbf{z}, \mathbf{I}_t, \mathbf{d})] \\ -\mathrm{KL}[q(\mathbf{z}|\mathbf{I}_t, \mathbf{d}) \| p(\mathbf{z}|\mathbf{I}_t, \mathbf{d})], \end{aligned} \tag{2}$$

where $q(\mathbf{z}|\mathbf{I}_t, \mathbf{d})$ is the approximation of the intractable true posterior $p(\mathbf{z}|\mathbf{I}_g, \mathbf{I}_t, \mathbf{d})$. The following inequality holds for the ELBO:

$$\log p(\mathbf{I}_g|\mathbf{I}_t, \mathbf{d}) \geqslant \mathrm{ELBO}. \tag{3}$$

Hence, we can maximize the ELBO so as to maximize $\log p(\mathbf{I}_g|\mathbf{I}_t, \mathbf{d})$. The ELBO contains three components: $q(\mathbf{z}|\mathbf{I}_t, \mathbf{d})$, $p(\mathbf{z}|\mathbf{I}_t, \mathbf{d})$ and $p(\mathbf{I}_g|\mathbf{z}, \mathbf{I}_t, \mathbf{d})$. Below we explain each component in detail.

3.2.1 $q(\mathbf{z}|\boldsymbol{I}_t, \boldsymbol{d})$ *and* $p(\mathbf{z}|\boldsymbol{I}_t, \boldsymbol{d})$. We propose a variational transformer (as shown in Figure 1) to model these two components, which transforms $\mathbf{I}_t, \mathbf{d}$ into a latent variable $\mathbf{z}$. As with previous work [23, 37], we assume that $q(\mathbf{z}|\mathbf{I}_t, \mathbf{d})$ and $p(\mathbf{z}|\mathbf{I}_t, \mathbf{d})$ are Gaussian distributions, i.e.,

$$q(\mathbf{z}|\mathbf{I}_t, \mathbf{d}) \sim \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2), \quad p(\mathbf{z}|\mathbf{I}_t, \mathbf{d}) \sim \mathcal{N}(0, 1), \tag{4}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ denote the variational mean and standard deviation respectively, which are calculated with our top encoder and variational transformer as follows.

Specifically, for a top image $\mathbf{I}_t$ of size 128×128 with 3 channels, we first use a CNN, i.e., *the top encoder* (as shown in Figure 2(a)) to extract visual features $\mathbf{F}_t$:

$$\mathbf{F}_t = \mathbf{CNN}(\mathbf{I}_t), \tag{5}$$

where $\mathbf{F}_t \in \mathbb{R}^{W \times H \times D}$, W and H are the width and height of the output feature maps, respectively, and D is the number of output feature maps. And we flatten $\mathbf{F}_t$ into a vector $\mathbf{f}_t \in \mathbb{R}^N$, where $N = W \times H \times D$, and project $\mathbf{f}_t$ to the visual representation $\mathbf{v}_t$:

$$\mathbf{v}_t = \text{sigmoid}(\mathbf{W}_{vt}\mathbf{f}_t + \mathbf{b}_{vt}), \tag{6}$$

where $\mathbf{W}_{vt} \in \mathbb{R}^{e \times N}$, $\mathbf{v}_t$ and $\mathbf{b}_{vt} \in \mathbb{R}^e$, and e is the size of the representation.

Besides the top image, FARM also allows users to give a natural language description $\mathbf{d}$, which describes the ideal bottom they want. In order to take into account the description $\mathbf{d}$, we follow Eq. 7 to get the semantic representation $\mathbf{v}_d$:

$$\mathbf{v}_d = \text{sigmoid}(\mathbf{W}_d\mathbf{d}), \tag{7}$$

where $\mathbf{v}_d \in \mathbb{R}^e$, $\mathbf{d} \in \mathbb{R}^{D_d}$, D_d is the vocabulary size, and $\mathbf{W}_d \in \mathbb{R}^{e \times D_d}$ is the visual semantic word embedding matrix [32], which transforms words from the textual space to the visual space. Specially, when d is an empty description, $\mathbf{v}_d$ is a zero vector.

Then *the variational transformer* uses the visual representation $\mathbf{v}_t$ and the semantic representation $\mathbf{v}_d$ to calculate the mean $\boldsymbol{\mu}$ and standard deviation $\boldsymbol{\sigma}$ for $q(\mathbf{z}|\mathbf{I}_t, \mathbf{d})$:

$$\begin{aligned} \boldsymbol{\mu} &= \mathbf{W}_{\mu t}\mathbf{v}_t + \mathbf{W}_{\mu d}\mathbf{v}_d + \mathbf{b}_\mu \\ \log \boldsymbol{\sigma}^2 &= \mathbf{W}_{\sigma t}\mathbf{v}_t + \mathbf{W}_{\sigma d}\mathbf{v}_d + \mathbf{b}_\sigma, \end{aligned} \tag{8}$$

where $\mathbf{W}_{\mu t}$, $\mathbf{W}_{\mu d}$, $\mathbf{W}_{\sigma t}$ and $\mathbf{W}_{\sigma d} \in \mathbb{R}^{k \times e}$, $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$, $\mathbf{b}_\mu$ and $\mathbf{b}_\sigma \in \mathbb{R}^k$, and k is the size of latent variable $\mathbf{z}$. The latent variable $\mathbf{z}$ can be calculated by the reparameterization trick [23, 37]:

$$\boldsymbol{\epsilon} \sim \mathcal{N}(0, 1), \quad \mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \otimes \boldsymbol{\epsilon}, \tag{9}$$

where $\boldsymbol{\epsilon}$ and $\mathbf{z} \in \mathbb{R}^k$, and $\boldsymbol{\epsilon}$ is the auxiliary noise variable. By the reparameterization trick, we make sure $\mathbf{z}$ is a random vector sampled from $\mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2)$.

3.2.2 $p(\boldsymbol{I}_g|\mathbf{z}, \boldsymbol{I}_t, \boldsymbol{d})$. We use the bottom generator (as shown in Figure 2(b)) to generate $\mathbf{I}_g$ from the variable $\mathbf{z}$. We also assume $p(\mathbf{I}_g|\mathbf{z}, \mathbf{I}_t, \mathbf{d})$ is a Gaussian distribution [23, 37], i.e.,

$$p(\mathbf{I}_g|\mathbf{z}, \mathbf{I}_t, \mathbf{d}) \sim \mathcal{N}(g(\mathbf{z}, \mathbf{I}_t, \mathbf{d}), \boldsymbol{\sigma}^2), \tag{10}$$

where g is the bottom generator.

Specifically, we first follow Eq. 11 to obtain the basic visual feature vector $\mathbf{f}_g$:

$$\mathbf{f}_g = \text{relu}(\mathbf{W}_{gz}\mathbf{z} + \mathbf{W}_{gt}\mathbf{v}_t + \mathbf{W}_{gd}\mathbf{v}_d + \mathbf{b}_g), \tag{11}$$

where $\mathbf{f}_g$ and $\mathbf{b}_g \in \mathbb{R}^N$, $\mathbf{W}_{gz}\mathbb{R}^{N \times k}$, $\mathbf{W}_{gt}$ and $\mathbf{W}_{gd} \in \mathbb{R}^{N \times e}$. Then we reshape $\mathbf{f}_g$ into a 3-D tensor $\mathbf{F}_g \in \mathbb{R}^{W \times H \times D}$, which is the reverse operation to what we do for $\mathbf{F}_t$. Finally, we use a DCNN, i.e., *the bottom generator* to generate the bottom image $\mathbf{I}_g$:

$$\mathbf{I}_g = \mathbf{DCNN}(\mathbf{F}_g), \tag{12}$$

where $\mathbf{I}_g \in \mathbb{R}^{128 \times 128 \times 3}$. To avoid generating blurry images [3], we divide the process of image generation into two stages [6, 49]. The first stage is an ordinary deconvolutional neural network that generates low-resolution images. The second stage is similar to the super-resolution residual network (SRResNet) [24], which accepts the images from the first stage and refines them to generate high quality ones. The DCNN is meant to capture high-level aesthetic features of the bottoms to be recommended [47, 48]. Besides, in order to generate the bottom, the generation process also forces the top encoder to capture more aesthetic information.

During training, we first sample a z from $q(\mathbf{z}|\mathbf{I}_t, \mathbf{d})$. Then we generate $\mathbf{I}_g$ with $g(\mathbf{z}, \mathbf{I}_t, \mathbf{d})$. During testing, in order to avoid the randomness introduced by $\boldsymbol{\epsilon}$, we directly generate $\mathbf{I}_g$ by $g(\mathbf{z} = \boldsymbol{\mu}, \mathbf{I}_t, \mathbf{d})$.

3.3 Fashion recommender

Given the image $\mathbf{I}_b$ of a bottom b, the fashion recommender needs to evaluate the matching score between $\mathbf{I}_b$ and the pair $(\mathbf{I}_t, \mathbf{d})$. Specifically, we first use the *bottom encoder* (as shown in Figure 2(a)), which has the same structure as the top encoder (parameters not shared), to extract visual features $\mathbf{F}_b \in \mathbb{R}^{W \times H \times D}$ from $\mathbf{I}_b$. Then we flatten $\mathbf{F}_b$ into a vector $\mathbf{f}_b \in \mathbb{R}^N$ and project $\mathbf{f}_b$ to the visual representation $\mathbf{v}_b$. Next, we calculate the matching score between $\mathbf{I}_b$ and the pair $(\mathbf{I}_t, \mathbf{d})$ in three ways.

3.3.1 Visual matching. We propose visual matching to evaluate the compatibility between $\mathbf{I}_b$ and $\mathbf{I}_t$ based on their visual features. Specifically, we calculate the visual matching score s_v between $\mathbf{I}_b$ and $\mathbf{I}_t$ by Eq. 13:

$$s_v = \mathbf{v}_b^T\mathbf{v}_t. \tag{13}$$

3.3.2 Description matching. For evaluating the matching degree between $\mathbf{I}_b$ and $\mathbf{d}$, we propose to match descriptions. The description matching score s_d between $\mathbf{I}_b$ and $\mathbf{d}$ is calculated by Eq. 14:

$$s_d = \mathbf{v}_b^T\mathbf{v}_d. \tag{14}$$

Note that if d does not contain any word, s_d equals 0.

3.3.3 Layer-to-layer matching. As we will demonstrate in our experiments in Section 6.2, a simple combination of generation and recommendation is not able to improve the recommendation performance. The reason is that there is no direct connection between generation and recommendation, which results in two issues. First, the aesthetic information from the generation process cannot be used effectively. Second, the generation process might introduce features that are only helpful for generation while unhelpful for recommendation. To overcome these issues, we propose a layer-to-layer matching mechanism. Specifically, we denote the visual features of the l-th CNN layer in the bottom encoder as $\mathbf{F}_b^l \in \mathbb{R}^{W^l \times H^l \times D^l}$. And we denote the visual features of the corresponding DCNN layer, which has the same size as $\mathbf{F}_b^l$, in the bottom generator as

$\mathbf{F}_g^l \in \mathbb{R}^{W^l \times H^l \times D^l}$. Then, we reshape $\mathbf{F}_b^l = [\mathbf{f}_{b,1}^l, \ldots, \mathbf{f}_{b,S}^l]$ by flattening the width and height of the original $\mathbf{F}_b^l$, where $S = W^l \times H^l$ and $\mathbf{f}_{b,i}^l \in \mathbb{R}^{D^l}$. And we can consider $\mathbf{f}_{b,i}^l$ as the visual features of the i-th location of $\mathbf{I}_b$. We perform global-average-pooling in $\mathbf{F}_b^l$ to get the global visual features $\mathbf{f}_b^l \in \mathbb{R}^{D^l}$:

$$\mathbf{f}_b^l = \frac{1}{S} \sum_{i=1}^{S} \mathbf{f}_{b,i}^l. \tag{15}$$

We project $\mathbf{f}_b^l$ to the visual representation $\mathbf{v}_b^l \in \mathbb{R}^e$:

$$\mathbf{v}_b^l = \text{sigmoid}(\mathbf{W}_{vb}^l \mathbf{f}_b^l + \mathbf{b}_{vb}^l), \tag{16}$$

where $\mathbf{W}_{vb}^l \in \mathbb{R}^{e \times D^l}$ and $\mathbf{b}_{vb}^l \in \mathbb{R}^e$. The same operations apply to $\mathbf{F}_g^l$ to get $\mathbf{v}_g^l$. Then we calculate the dot product between $\mathbf{v}_b^l$ and $\mathbf{v}_g^l$, which represents the matching degree s_g^l between $\mathbf{I}_b$ and $\mathbf{I}_g$ in the l-th visual level:

$$s_g^l = {\mathbf{v}_b^l}^T \mathbf{v}_g^l. \tag{17}$$

For different visual levels, we sum all s_g^l to get the matching score s_g between $\mathbf{I}_b$ and $\mathbf{I}_g$:

$$s_g = \sum_{l \in L} s_g^l, \tag{18}$$

where L is the selected CNN layer set for layer-to-layer matching.

Finally, the total matching score s between $\mathbf{I}_b$ and the pair $(\mathbf{I}_t, \mathbf{d})$ is defined as follows:

$$s = s_v + s_d + s_g. \tag{19}$$

3.4 Co-supervision learning framework

For FARM, we train the fashion generator and the fashion recommender jointly with a co-supervision learning framework.

Specifically, for the generation part, we regard the image $\mathbf{I}_p$ of a positive bottom p, which not only matches the given top $\mathbf{I}_t$ but also meets the given description $\mathbf{d}$, as the generation target. And we denote the generated bottom image in the first stage as $\mathbf{I}_g^1$, and denote the generated bottom image in the second stage as $\mathbf{I}_g^2$. Then, the first loss is to maximize the first term in ELBO, which is Eq. 20:

$$\mathcal{L}_{gen}(t, d, p) = \frac{1}{2} \|\mathbf{I}_g^1 - \mathbf{I}_p\|_2^2 + \|\mathbf{I}_g^2 - \mathbf{I}_p\|. \tag{20}$$

The second loss is to minimize the second term in ELBO, which is Eq. 21:

$$\mathcal{L}_{kl}(t, d, p) = \frac{1}{2} \sum_{i=1}^{k} (1 + \log \boldsymbol{\sigma}_i^2 - \boldsymbol{\mu}_i^2 - \boldsymbol{\sigma}_i^2), \tag{21}$$

where $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i$ are the i-th elements in $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ respectively.

For the recommendation part, we employ BPR [35] as the loss:

$$\mathcal{L}_{bpr}(t, d, p, n) = -\log(\text{sigmoid}(s_p - s_n)), \tag{22}$$

where s_p and s_n are the matching scores of a positive bottom $\mathbf{I}_p$ and a negative bottom $\mathbf{I}_n$, respectively (calculated with Eq. 19). $\mathbf{I}_n$ (image of bottom n) is randomly sampled.

The total loss function can be defined as follows:

$$\mathcal{L} = \sum_{(t,d,p,n) \in \mathcal{D}} \mathcal{L}_{gen}(t, d, p) + \mathcal{L}_{kl}(t, d, p) + \mathcal{L}_{bpr}(t, d, p, n), \tag{23}$$

where $\mathcal{D} = \{(t, d, p, n) | t \in \mathcal{T}, d \in \mathcal{D}_b, p \in \mathcal{B}_{t,d}, n \in \mathcal{B} \setminus \mathcal{B}_{t,d}\}$, $\mathcal{D}_b$ is the bottom description set, $\mathcal{B}_{t,d}$ is the positive bottom set for the pair $(\mathbf{I}_t, \mathbf{d})$ and $\mathcal{B} \setminus \mathcal{B}_{t,d}$ is the negative bottom set for the pair $(\mathbf{I}_t, \mathbf{d})$. The whole framework can be efficiently trained using back-propagation in an end-to-end paradigm.

For top recommendation, we follow the same way to build and train the model, but exchange the roles of tops and bottoms.

4 EXPERIMENTAL SETUP

We set up a series of experiments to evaluate the recommendation performance of FARM. Details of our experimental settings are listed below. All code and data used to run the experiments in this paper are available at https://bitbucket.org/Jay_Ren/www2019_fashionrecommendation_yujie/src/master/farm/.

4.1 Datasets

Existing fashion datasets include *WoW* [29], *Exact Street2Shop* [21], *Fashion-136K* [16], *FashionVC* [41] and *ExpFashion* [28]. *WoW*, *Exact Street2Shop*, and *Fashion-136K* have been collected from street photos[3] on the web and involve (visual) parsing of clothing, which still remains a great challenge in the computer vision domain [41, 44, 45] and which is beyond the scope of this paper. *FashionVC* and *ExpFashion* have been collected from the fashion-oriented online community Polyvore[4] and contain both images and texts. The images are of good quality and the texts include descriptions like names and categories. For our experiments, we choose *FashionVC* and *ExpFashion*. The statistics of the two datasets are given in Table 1. We preprocess *FashionVC* or *ExpFashion* with the following steps, taking bottom recommendation as an example. For each tuple (*top*, *top description*, *bottom*, *bottom description*), we regard (*top*, *bottom description*) as input and the *bottom* as the ground truth output. We follow existing studies [41] and randomly select bottoms to generate 100 candidates along with the ground truth bottoms in the validation and test set. Similar processing steps are used for top recommendation.

Table 1: Dataset statistics.

Dataset	Tops	Bottoms	Outfits
FashionVC [41]	14,871	13,663	20,726
ExpFashion [28]	168,682	117,668	853,991

4.2 Implementation details

The parameters W, H, D and N of the encoder and the generator are set to 1, 1, 1024 and 1024, respectively. The size e of the visual semantic word embedding, the semantic representation and the visual representation is set to 100. And the latent variable size k is set to 100 too. The 7th, the 6th and the 5th layers of the encoder CNN are adopted to compute the layer-to-layer matching with the input, the 1st and the 2nd layers of the generator DCNN. To build descriptions, we first filter out words whose frequency is less than 100. Then, we manually go through the rest to only keep words that can describe tops or bottoms. Finally, the remaining vocabulary

[3] http://www.tamaraberg.com/street2shop/
[4] http://www.polyvore.com/

size D_d is 547. During training, we initialize model parameters randomly with the Xavier method [10]. We choose Adam [22] as our optimization algorithm. For the hyper-parameters of the Adam optimizer, we set the learning rate $\alpha = 0.001$, two momentum parameters $\beta 1 = 0.9$ and $\beta 2 = 0.999$, and $\epsilon = 10^{-8}$. We apply dropout [42] to the output of our encoder and set the rate to 0.5. We also apply gradient clipping [34] with range $[-5, 5]$ during training. We use a mini-batch size 64 by grid search to both speed up the training and converge quickly. We test the model performance on the validation set for every epoch. Our framework is implemented with MXNet [8]. All experiments are conducted on a single Titan X GPU.

4.3 Methods used for comparison

We choose the following methods for comparison.

- *LR*: Logistic Regression (LR) is a standard machine learning method [17]. We use it to predict whether a candidate bottom matches a given (*top*, *bottom description*) pair or not. Specifically, we employ a pre-trained CNN to extract visual features from images. Then we follow Eq. 24 to calculate the matching probability p:

$$p = \text{sigmoid}(\mathbf{w}_t^T \mathbf{v}_t + \mathbf{w}_b^T \mathbf{v}_b + \mathbf{w}_d^T \mathbf{d}), \tag{24}$$

  where $\mathbf{v}_t$ and $\mathbf{v}_b \in \mathbb{R}^{D_v}$ are the visual features of the top and the bottom respectively, $\mathbf{w}_t$ and $\mathbf{w}_b \in \mathbb{R}^{D_v}$, and $\mathbf{w}_d \in \mathbb{R}^{D_d}$. D_v is set to 4096 in our experiments.
- *IBR$_d$*: IBR [31] learns a visual style space in which related objects are close and unrelated objects are far. In order to consider the given descriptions at the same time, we modify IBR by projecting descriptions to the visual style space. As a result, we can evaluate the matching degree between objects and descriptions by their distance in the space. Specifically, the distance function between the candidate bottom b and the given (*top*, *bottom description*) pair (t, d) is as follows:

$$m_{tdb} = \|\mathbf{W}_v \mathbf{v}_t - \mathbf{W}_v \mathbf{v}_b\|_2^2 + \|\mathbf{W}_d \mathbf{v}_d - \mathbf{W}_v \mathbf{d}\|_2^2, \tag{25}$$

  where $\mathbf{W}_v \in \mathbb{R}^{K \times D_v}$, $\mathbf{W}_d \in \mathbb{R}^{K \times D_d}$, $\mathbf{v}_t$ and $\mathbf{v}_b \in \mathbb{R}^{D_v}$ are the visual features extracted by a pre-trained CNN, and K is the dimension of the visual style space. D_v is 4096, and K is 100 in our experiments. We refer to the modified version as IBR$_d$.
- *BPR-DAE$_d$*: BPR-DAE [41] can jointly model the implicit matching preference between items in visual and textual modalities and the coherence relation between different modalities of items. In our task, we do not have other text information except descriptions, so we first remove the part of BPR-DAE that is related to text information. Then, for evaluating the matching score between the given description and the candidate item, we project the description representation and the item representation to the same latent space:

$$\mathbf{v}'_d = \text{sigmoid}(\mathbf{W}_d \mathbf{d}), \quad \mathbf{v}'_i = \text{sigmoid}(\mathbf{W}_v \mathbf{v}_i), \tag{26}$$

  where $\mathbf{W}_d \in \mathbb{R}^{K \times D_d}$, $\mathbf{W}_v \in \mathbb{R}^{K \times D_v}$, and $\mathbf{v}_i \in \mathbb{R}^{D_v}$ is the latent representation of item i learned by BPR-DAE. Finally, we follow Eq. 27 to evaluate the compatibility between a candidate bottom b and a given (*top*, *bottom description*) pair (t, d):

$$m_{tdb} = \mathbf{v}_t^T \mathbf{v}_b + \mathbf{v}'^T_d \mathbf{v}'_b. \tag{27}$$

  We set $D_v = 512$, and $K = 100$ in experiments. We refer to the modified version as BPR-DAE$_d$.
- *DVBPR$_d$*: DVBPR [20] learns the image representations and trains the recommender system jointly to recommend fashion items for users. We adopt DVBPR to our task and refer to it as DVBPR$_d$. Specifically, we first follow DVBPR to use a CNN-F to learn image representations of tops and bottoms. Then we calculate the matching score between a bottom and the given (*top*, *bottom description*) pair by Eq. 28:

$$m_{tdb} = \mathbf{v}_t^T \mathbf{v}_b + \mathbf{v}_d^T \mathbf{v}_b, \tag{28}$$

  where $\mathbf{v}_t$ and $\mathbf{v}_b \in \mathbb{R}^K$ are the image representations of the top and bottom respectively, $\mathbf{v}_d \in \mathbb{R}^K$ is the description representation learned in the same way as FARM, and K is set to 100 in experiments.

4.4 Evaluation metrics

We employ *Mean Reciprocal Rank* (MRR) and *Area Under the ROC Curve* (AUC) to evaluate the recommendation performance, which are widely used in recommender systems [25, 36, 50].

In the case of bottom recommendations, for example, MRR and AUC are calculated as follows:

$$\text{MRR} = \frac{1}{|Q^{td}|} \sum_{i=1}^{|Q^{td}|} \frac{1}{rank_i}, \tag{29}$$

where Q^{td} is the (*top*, *bottom description*) collection as queries, and $rank_i$ refers to the rank position of the first positive bottom for the i-th (*top*, *bottom description*) pair. Furthermore,

$$\text{AUC} = \frac{1}{|Q^{td}|} \sum_{(t,d) \in Q^{td}} \frac{1}{|E(t,d)|} \sum_{(p,n) \in E(t,d)} \delta(s_p > s_n), \tag{30}$$

where $E(t, d)$ is the set of all positive and negative candidate bottoms for the given top t and the given bottom description d, s_p is the matching score of a positive bottom p, s_n is the matching score of a negative bottom n, and $\delta(\alpha)$ is an indicator function that equals 1 if α is true and 0 otherwise.

5 RESULTS

The recommendation results on the FashionVC and ExpFashion datasets of FARM and the methods used for comparison are shown in Table 2. We can see that FARM consistently outperforms all baselines in terms of AUC and MRR on both datasets. We have five main observations from Table 2.

(1) FARM significantly outperforms all baselines and achieves the best results on all metrics. There are three main reasons. First, FARM contains a fashion generator as an auxiliary module for recommendation. With its co-supervision learning framework, FARM can encode more aesthetic characteristics and use this extra information to improve recommendation performance; see Section 6.1 for further analysis. Second, we propose a layer-to-layer matching scheme to make sure that FARM can effectively use the aesthetic features in the fashion generator to improve recommendation results; see Section 6.2 for a further analysis. Third, LR, IBR$_d$ and BPR-DAE$_d$ employ pre-trained CNNs (all

Table 2: Recommendation results on the FashionVC and ExpFashion datasets (%).

Method	FashionVC			
	Top		Bottom	
	AUC	MRR	AUC	MRR
LR	48.7	4.5	46.4	4.4
IBR_d	52.8	6.1	62.9	10.3
BPR-DAE_d	62.9	8.6	70.2	10.9
$DVBPR_d$	64.6	9.1	76.9	13.0
FARM	**71.2***	**12.6***	**77.8**	**15.3***
Method	ExpFashion			
	Top		Bottom	
	AUC	MRR	AUC	MRR
LR	50.5	5.4	48.4	4.4
IBR_d	56.1	7.1	68.9	12.0
BPR-DAE_d	73.0	12.3	79.9	14.7
$DVBPR_d$	82.4	18.5	83.7	15.4
FARM	**85.2***	**25.1***	**88.4***	**24.3***

The superscript * indicates that FARM significantly outperforms $DVBPR_d$, using a paired t-test with $p < 0.05$.

AlexNet [19] trained on ImageNet[5]) to extract visual features from images, but they do not fine-tune the CNNs during experiments. However, in FARM, we jointly train the top encoder, the bottom encoder and the top/bottom generator, which can extract better visual features.

(2) $DVBPR_d$ performs better than other baseline methods. The reason is that $DVBPR_d$ employs a CNN-F to jointly learn image representations during recommendation. Hence, it can extract more effective visual features to improve recommendation performance.

(3) Although BPR-DAE_d, IBR_d and LR all use visual features extracted by a pre-trained CNN as input, BPR-DAE_d performs much better than the other two. This is because BPR-DAE_d learns a more sophisticated latent space using an auto-encoder neural network to represent the fashion items. However, IBR_d only applies a linear transformation to inputs, which restricts the expressive ability of the visual style space. And LR directly uses the visual features and the bag-of-words vectors as inputs, making it hard to learn an effective matching relation.

(4) The performance of all methods on the ExpFashion dataset is better than on the FashionVC dataset. The most important reason is that the average length of the descriptions in the ExpFashion dataset is 5.6 words, however, it is only 3.7 words in the FashionVC dataset. That means that the descriptions in the ExpFashion dataset contain more details that can provide more information for recommendation and generation, which boosts the recommendation performance.

(5) The bottom recommendation performance is better than the top recommendation performance for most methods. The number of tops is larger than the number of bottoms and the styles of tops are also richer than those of bottoms on both datasets. That makes bottom recommendation and bottom generation easier.

[5]http://www.image-net.org/

Table 3: Analysis of co-supervision learning. Recommendation results on the FashionVC and ExpFashion datasets (%).

Method	FashionVC			
	Top		Bottom	
	AUC	MRR	AUC	MRR
FARM-G	54.8	8.4	60.9	9.8
FARM-R	68.0	9.8	77.2	12.8
FARM	**71.2***	**12.6***	**77.8**	**15.3***
Method	ExpFashion			
	Top		Bottom	
	AUC	MRR	AUC	MRR
FARM-G	64.4	14.2	72.4	21.3
FARM-R	82.3	18.9	84.2	15.2
FARM	**85.2***	**25.1***	**88.4***	**24.3***

The superscript * indicates that FARM significantly outperforms FARM-R, using a paired t-test with $p < 0.05$.

In summary, FARM significantly outperforms state-of-the-art methods on both datasets. The improvements mainly come from the co-supervision of generation and the layer-to-layer mechanism, which we will demonstrate in the next section.

6 ANALYSIS

We provide two types of analyses (concerning co-supervision learning and layer-to-layer matching) and two cases studies (recommendation and generation).

6.1 Co-supervision learning

To demonstrate the superiority of incorporating the extra supervision of the generator, we compare FARM with FARM-G and FARM-R, where FARM-G is FARM without the recommendation part and FARM-R is FARM without the generation part. The results are shown in Table 3. To be able to apply FARM-G to the recommendation task, we first use FARM-G to generate a bottom image for a given (*top*, *bottom description*) pair. Then, similar to [2, 27], we use a pre-trained AlexNet to get the representations of the generated bottom and the candidate bottoms. Finally, we compute the similarity between the generated bottom and a candidate bottom based on their representations.

From Table 3, we can see that FARM achieves significant improvements over FARM-R. On the FashionVC dataset, for top recommendation, AUC increases by 3.2%, MRR increases by 2.8%, and for bottom recommendation, AUC increases by 0.6%, MRR increases by 2.5%. On the ExpFashion dataset, for top recommendation, AUC increases by 2.9%, MRR increases by 6.2%, and for bottom recommendation, AUC increases by 4.2%, MRR increases by 9.1%. Thus, FARM is able to improve recommendation performance by using the generator as a supervision signal.

Comparing FARM-G with all baselines, we notice that FARM-G achieves better performance, and especially it performs better than IBR_d in most settings. Hence, the images generated by FARM-G and FARM reflect some key factors of the items to be recommended, which is why the generator can help improve recommendation.

Table 4: Analysis of layer-to-layer matching. Recommendation results on the FashionVC and ExpFashion datasets (%).

Method	FashionVC			
	Top		Bottom	
	AUC	MRR	AUC	MRR
FARM-WL	59.8	7.6	67.8	8.2
FARM	**71.2***	**12.6***	**77.8***	**15.3***
Method	**ExpFashion**			
	Top		Bottom	
	AUC	MRR	AUC	MRR
FARM-WL	68.6	9.9	74.3	10.3
FARM	**85.2***	**25.1***	**88.4***	**24.3***

The superscript * indicates that FARM significantly outperforms FARM-WL, using a paired t-test with $p < 0.05$.

Additionally, we find that FARM-R outperforms LR, IBR$_d$ and BPR-DAE$_d$. And it achieves comparable performance with DVBPR$_d$, whose difference against FARM-R is mainly in the CNN part. If FARM employs more powerful CNN architectures such as VGG [39] or ResNet [12], it should perform even better.

6.2 Layer-to-layer matching

To analyze the effect of the layer-to-layer matching scheme, we compare FARM with FARM-WL which only uses the visual matching and the description matching to evaluate the matching degree. We can see from Table 4 that FARM performs significantly better than FARM-WL according to all metrics on both datasets, which confirms that layer-to-layer matching does indeed improve the performance of recommendation.

To help understand the effect of layer-to-layer matching, we list some real and generated images in Figure 3. FARM generates good quality images that are similar to real images. This means that the generated images can tell us what kind of bottoms can match the given (*top*, *bottom description*) pair from the perspective of generation, so layer-to-layer matching can direct the recommender by evaluating the matching degree between the candidate images and the generated images. That is why layer-to-layer matching is able to improve the performance of recommendation.

Additionally, we notice that FARM-WL performs worse than FARM-R, which means that a simple combination of recommendation and generation is not able to improve recommendation performance significantly. This may be because, without layer-to-layer matching, FARM-WL pays too much attention to the generation quality and ignores recommendation performance. We are able to improve this situation with layer-to-layer matching. Layer-to-layer matching builds a connection between the bottom generator and the bottom encoder in different layers. As a result, the bottom encoder pushes the bottom generator to learn useful matching information for improving recommendation performance.

6.3 Recommendation case studies

We list some recommendation produced by FARM in Figure 4. For each input, we list the top-10 recommended items. We highlight

(a) Top generation.

(b) Bottom generation.

Figure 3: Comparison between real and generated images.

the positive items with red boxes. We can see that most recommended items not only match the given items, but also meet the given descriptions. For example, in the second case of the top recommendation, the given top description is "sleeve black blazer outerwear jackets,"so most recommended tops are jackets, and especially almost all recommended tops are black. Also in the first case of the bottom recommendation, the given bottom description is "distressed straight leg jean," so the recommended bottoms are all jeans, most of which are straight leg and some are distressed. By comparing the generated items with the recommended items, such as in the first case of the top recommendation and the second case of the bottom recommendation, we can see that the generated images are able to provide good guidance for the recommendation.

We also notice that not all recommended items meet the given description, mostly because FARM recommends items not only based on the given description, but also based on the given item. For example, in the third case of the bottom recommendation, the sixth recommended bottom is a denim jeans instead of a daydress. The given top is a denim coat, which makes FARM believe that recommending a denim jeans is also reasonable. Besides, not all positive items are ranked in the first position. See, e.g., the third case of the top recommendation., where the top recommended item and the given bottom have the same color green, which looks more compatible. In these failure cases, the quality of the generated images is poor so they are likely less helpful for recommendation.

6.4 Generation case studies

Although this paper focuses on improving recommendation by incorporating generation, we also list some generation cases in Figure 5. Overall, the generated items are able to match the given input.

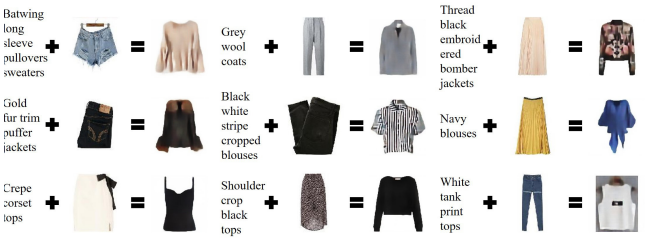
(a) Top recommendation.

(b) Bottom recommendation.

Figure 4: Case studies of recommendation. The items highlighted in the red boxes are the positive ones.

For example, in the sixth case of the top generation, the generated navy blouse with the yellow keen length skirt looks beautiful and elegant. Although there are many kinds of navy blouses like sailor suits, the style of the generated top seems to be more suitable for the given bottom. And in the eighth case of the bottom generation, the given description does not give the specific pattern of the generated bottom.

But the generated bottom has a flame-like pattern, which makes it more compatible with the bright yellow camisole. From these samples we can see that FARM is able to generate fashion items based on the relation between the visual features of different fashion items.



(a) Top generation.

(b) Bottom generation.

Figure 5: Case studies of generation. Each case is in the form: "given description + given item = generated item".

The generated items can accord with the given descriptions no matter what they are about. For example, in the second case of the top generation, the description is "grey wool coats," so the generated top is a grey coat which also looks like wool. And in the fourth case, the description is "gold fur trim puffer jackets", so the generated jacket has fur in its collar and cuff. In the bottom generation, we also observe that FARM is able to distinguish between skinny jeans and bootcut jeans from the first and the second cases. Another example is the sixth case, where the description contains "floral print." FARM generates a black long pencil skirt with flower pattern. In short, FARM is able to build a cross-modal connection between text and images in order to generate fashion items.

Generation is a challenging process, which means that powerful features are needed in order to generate a matching item. We can see from the examples provided that FARM is able to generate aesthetically matching outfits. FARM is able to improve recommendation performance through jointly modeling generation.

7 CONCLUSION

In this paper, we have studied the task of outfit recommendation, which has two main challenges: visual understanding and visual matching. To tackle these challenges, we propose a co-supervision learning framework, namely FARM. For visual understanding, FARM captures aesthetic characteristics with the supervision of generation learning. For visual matching, FARM incorporates a layer-to-layer matching mechanism to evaluate the matching score at different neural layers.

We have conducted experiments to confirm the effectiveness of FARM. It achieves significant improvements over state-of-the-art baselines in terms of AUC and MRR. We also show that the proposed layer-to-layer matching mechanism can make effective use of generation information to improve recommendation performance. We further exhibit some cases to analyze the performance of FARM.

Our results can be used to improve users' experience in fashion-oriented online communities by providing better recommendation and to promote the research into fashion generation by demonstrating a novel application in outfit recommendation.

A limitation of FARM is that its recommendation performance is affected by the quality of the generated images. If the quality of the generated images is not high, the generation part cannot provide effective guidance for the recommendation part.

As to future work, we plan to improve the recommendation and the generation of FARM when the descriptions are lacking. And we want to extend FARM to recommend and generate whole outfits that not only contain tops and bottoms but also include shoes and hats, etc. We will also try more powerful CNN and DCNN architectures for recommendation and generation.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their helpful comments.

This work is supported by the Natural Science Foundation of China (61672324, 61672322), the Natural Science Foundation of Shandong province (2016ZRE27468), the Fundamental Research Funds of Shandong University, Ahold Delhaize, the Association of Universities in the Netherlands, and the Innovation Center for Artificial Intelligence (ICAI). All content represents the opinion of

the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. 2006. Face Description with Local Binary Patterns: Application to Face Recognition. *IEEE Trans on Pattern Analysis and Machine Intelligence (TPAMI)* 28, 12 (2006).

[2] Artem Babenko, Anton Slesarev, Alexander Chigorin, and Victor S. Lempitsky. 2014. Neural Codes for Image Retrieval. In *European Conf. on Computer Vision (ECCV'14)*.

[3] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. 2017. CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training. In *International Conf. on Computer Vision (ICCV'17)*. 2764–2773.

[4] David M. Blei, Alp Kucukelbir, and Jon D. Mcauliffe. 2017. Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association (JASA)* 112, 518 (2017).

[5] Lubomir D. Bourdev, Subhransu Maji, and Jitendra Malik. 2011. Describing people: A Poselet-based Approach to Attribute Classification. In *International Conf. on Computer Vision (ICCV'11)*. 1543–1550.

[6] Lei Cai, Hongyang Gao, and Shuiwang Ji. 2017. Multi-Stage Variational Auto-Encoders for Coarse-to-Fine Image Generation. *CoRR* abs/1705.07202 (2017). http://arxiv.org/abs/1705.07202

[7] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Return of the Devil in the Details: Delving Deep into Convolutional Nets. In *British Machine Vision Conf. (BMVC'14)*.

[8] Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. 2015. MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems. In *Annual Conf. on Neural Information Processing Systems (NIPS'15)*.

[9] Navneet Dalal and Bill Triggs. 2005. Histograms of Oriented Gradients for Human Detection. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'05)*. 886–893.

[10] Xavier Glorot and Yoshua Bengio. 2010. Understanding the Difficulty of Training Deep Feedforward Neural Networks. *Journal of Machine Learning Research (JMLR)* 9 (2010), 249–256.

[11] Xintong Han, Zuxuan Wu, Yu-Gang Jiang, and Larry S. Davis. 2017. Learning Fashion Compatibility with Bidirectional LSTMs. In *ACM International Conf. on Multimedia (MM'17)*. 1078–1086.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'16)*.

[13] Wei Lin Hsiao and Kristen Grauman. 2018. Creating Capsule Wardrobes from Fashion Images. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'18)*.

[14] Yang Hu, Xi Yi, and Larry S. Davis. 2015. Collaborative Fashion Recommendation: A Functional Tensor Factorization Approach. In *ACM International Conf. on Multimedia (MM'15)*. 129–138.

[15] Tomoharu Iwata, Shinji Watanabe, and Hiroshi Sawada. 2011. Fashion Coordinates Recommender System Using Photographs from Fashion Magazines. In *International Joint Conf. on Artificial Intelligence (IJCAI'11)*. 2262–2267.

[16] Vignesh Jagadeesh, Robinson Piramuthu, Anurag Bhardwaj, Wei Di, and Neel Sundaresan. 2014. Large Scale Visual Recommendations from Street Fashion Images. In *ACM Knowledge Discovery and Data Mining (KDD'14)*. 1925–1934.

[17] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning*. Springer.

[18] Shatha Jaradat. 2017. Deep Cross-Domain Fashion Recommendation. In *ACM Conf. on Recommender Systems (RecSys'17)*. 407–410.

[19] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional Architecture for Fast Feature Embedding. In *ACM International Conf. on Multimedia (MM'14)*. 675–678.

[20] Wang-Cheng Kang, Chen Fang, Zhaowen Wang, and Julian McAuley. 2017. Visually-Aware Fashion Recommendation and Design with Generative Image Models. In *International Conf. on Data Mining (ICDM'17)*. 207–216.

[21] M. Hadi Kiapour, Xufeng Han, Svetlana Lazebnik, Alexander C. Berg, and Tamara L. Berg. 2015. Where to Buy It: Matching Street Clothing Photos in Online Shops. In *International Conf. on Computer Vision (ICCV'15)*. 3343–3351.

[22] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conf. on Learning Representations (ICLR'15)*. http://arxiv.org/abs/1412.6980

[23] Diederik P. Kingma and Max Welling. 2014. Auto-encoding Variational Bayes. In *International Conf. on Learning Representations (ICLR'14)*.

[24] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, and Zehan Wang. 2017. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'17)*. 105–114.

[25] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural Attentive Session-based Recommendation. In *ACM International Conf. on Information and Knowledge Management (CIKM'17)*. 1419–1428.

[26] Yuncheng Li, Liangliang Cao, Jiang Zhu, and Jiebo Luo. 2017. Mining Fashion Outfit Composition Using an End-to-End Deep Learning Approach on Set Data. *IEEE Transactions on Multimedia (TMM)* 19, 8 (2017), 1946–1955.

[27] Kevin Lin, Huei Fang Yang, Jen Hao Hsiao, and Chu Song Chen. 2015. Deep Learning of Binary Hash Codes for Fast Image Retrieval. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'15)*. 27–35.

[28] Yujie Lin, Pengjie. Ren, Zhumin Chen, Zhaochun Ren, Jun Ma, and Maarten de Rijke. 2018. Explainable Fashion Recommendation with Joint Outfit Matching and Comment Generation. *CoRR* abs/1806.08977 (2018).

[29] Si Liu, Jiashi Feng, Zheng Song, Tianzhu Zhang, Hanqing Lu, Changsheng Xu, and Shuicheng Yan. 2012. Hi, Magic Closet, Tell Me What to Wear!. In *ACM International Conf. on Multimedia (MM'12)*. 619–628.

[30] Yihui Ma, Jia Jia, Suping Zhou, Jingtian Fu, Yejun Liu, and Zijian Tong. 2017. Towards Better Understanding the Clothing Fashion Styles: A Multimodal Deep Learning Approach. In *AAAI Conf. on Artificial Intelligence (AAAI'17)*. 38–44.

[31] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. In *International Conf. on Research on Development in Information Retrieval (SIGIR'15)*. 43–52.

[32] Takuma Nakamura and Ryosuke Goto. 2018. Outfit Generation and Style Extraction via Bidirectional LSTM and Autoencoder. In *ACM Knowledge Discovery and Data Mining (KDD'18)*.

[33] Jose Oramas and Tinne Tuytelaars. 2016. Modeling Visual Compatibility through Hierarchical Mid-level Elements. *CoRR* (2016). http://arxiv.org/abs/1604.00036

[34] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the Difficulty of Training Recurrent Neural Networks. In *International Conf. on Machine Learning (ICML'13)*. III–1310–III–1318.

[35] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *International Conf. on Uncertaintyin Artificial Intelligence (UAI'09)*. 452–461.

[36] Steffen Rendle and Lars Schmidt-Thieme. 2010. Pairwise Interaction Tensor Factorization for Personalized Tag Recommendation. In *ACM International Conf. on Web Search and Data Mining (WSDM'10)*. 81–90.

[37] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *International Conf. on Machine Learning (ICML'14)*. 1278–1286.

[38] Edgar Simo-Serra, Sanja Fidler, Francesc Moreno-Noguer, and Raquel Urtasun. 2015. Neuroaesthetics in fashion: Modeling the perception of fashionability. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'15)*, Vol. 00. 869–877.

[39] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conf. on Learning Representations (ICLR'15)*.

[40] Xuemeng Song, Fuli Feng, Xianjing Han, Xin Yang, Wei Liu, and Liqiang Nie. 2018. Neural Compatibility Modeling with Attentive Knowledge Distillation. In *International Conf. on Research on Development in Information Retrieval (SIGIR'18)*.

[41] Xuemeng Song, Fuli Feng, Jinhuan Liu, Zekun Li, Liqiang Nie, and Jun Ma. 2017. NeuroStylist: Neural Compatibility Modeling for Clothing Matching. In *ACM International Conf. on Multimedia (MM'17)*. 753–761.

[42] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research (JMLR)* 15, 1 (2014), 1929–1958.

[43] Andreas Veit, Balazs Kovacs, Sean Bell, Julian Mcauley, Kavita Bala, and Serge Belongie. 2015. Learning Visual Clothing Style with Heterogeneous Dyadic Co-Occurrences. In *International Conf. on Computer Vision (ICCV'15)*. 4642–4650.

[44] Kota Yamaguchi. 2012. Parsing Clothing in Fashion Photographs. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'12)*. 3570–3577.

[45] Kota Yamaguchi, M. Hadi Kiapour, Luis E. Ortiz, and Tamara L. Berg. 2015. Retrieving Similar Styles to Parse Clothing. *IEEE Trans on Pattern Analysis and Machine Intelligence (TPAMI)* 37, 5 (2015), 1028–1040.

[46] Wenhui Yu, Huidi Zhang, Xiangnan He, Xu Chen, Li Xiong, and Zheng Qin. 2018. Aesthetic-based Clothing Recommendation. In *International World Wide Web Conferences (WWW'18)*. 649–658.

[47] Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and Understanding Convolutional Networks. In *European Conf. on Computer Vision (ECCV'14)*. 818–833.

[48] Matthew D. Zeiler, Graham W. Taylor, and Rob Fergus. 2011. Adaptive Deconvolutional Networks for Mid and High Level Feature Learning. In *International Conf. on Computer Vision (ICCV'11)*. 2018–2025.

[49] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. 2017. StackGAN: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks. In *International Conf. on Computer Vision (ICCV'17)*. 5908–5916.

[50] 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 *ACM International Conf. on Multimedia (MM'13)*. 33–42.