2017 IEEE International Conference on Data Mining

Visually-Aware Fashion Recommendation and Design with Generative Image Models

Wang-Cheng Kang
UC San Diego
wckang@eng.ucsd.edu

Chen Fang
Adobe Research
cfang@adobe.com

Zhaowen Wang
Adobe Research
zhawang@adobe.com

Julian McAuley
UC San Diego
jmcauley@eng.ucsd.edu

***Abstract*—Building effective recommender systems for domains like fashion is challenging due to the high level of subjectivity and the semantic complexity of the features involved (i.e., fashion styles). Recent work has shown that approaches to 'visual' recommendation (e.g. clothing, art, etc.) can be made more accurate by incorporating visual signals directly into the recommendation objective, using 'off-the-shelf' feature representations derived from deep networks. Here, we seek to extend this contribution by showing that recommendation performance can be significantly improved by learning 'fashion aware' image representations directly, i.e., by training the image representation (from the pixel level) and the recommender system jointly; this contribution is related to recent work using Siamese CNNs, though we are able to show improvements over state-of-the-art recommendation techniques such as BPR and variants that make use of pre-trained visual features. Furthermore, we show that our model can be used *generatively*, i.e., given a user and a product category, we can generate new images (i.e., clothing items) that are most consistent with their personal taste. This represents a first step towards building systems that go beyond recommending existing items from a product corpus, but which can be used to suggest styles and aid the design of new products.**

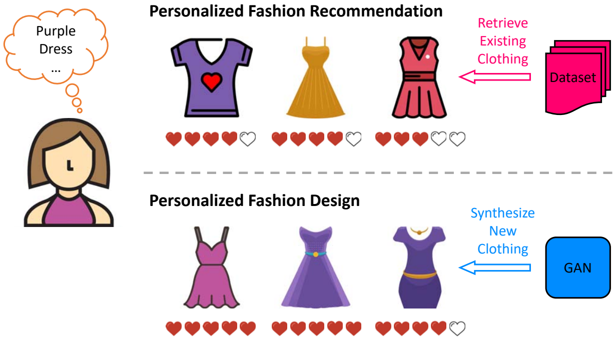


Fig. 1. Our model can be used for both personalized recommendation and design. Personalized recommendation is achieved by using a 'visually aware' recommender based on Siamese CNNs; generation is achieved by using a Generative Adversarial Net to synthesize new clothing items in the user's personal style. (Icons made by Madebyoliver, Roundicons and Freepik from www.flaticon.com)

I. Introduction

The goal of a recommender system is to provide personalized suggestions to users, based on large volumes of historical feedback, by uncovering hidden dimensions that describe the preferences of users and the properties of the items they consume. Traditionally, this means training predictive algorithms that can identify (or rank) items that are likely to be clicked on, purchased (or co-purchased), or given a high rating. In domains like fashion, this can be particularly challenging for a number of reasons: the vocabulary of items is long-tailed and new items are continually being introduced (cold-start); users' preferences and product styles change over time; and more critically, the semantics that determine what is 'fashionable' are incredibly complex.

Recently, there has been an effort to address these challenges by building recommender systems that are 'visually aware,' i.e., which incorporate visual signals directly into the recommendation objective. In the simplest case, visual features can be taken 'off the shelf,' and incorporated into a content-aware recommender system [1]. Although the features captured by such methods (e.g. CNN features extracted from Caffe [2]) describe high-level characteristics (e.g. color, texture, shape), and not necessarily the subtle characteristics of 'fashion,' these methods are nevertheless effective, especially in 'cold-start' settings where high-level image characteristics can be informative. Such work has been extended to incorporate temporal and social signals, and to model visual factors in other settings, like artistic recommendation [3], [4].

A parallel line of work has sought to develop new image representations using convolutional neural networks. This includes image classification, as well as models specifically designed for fashion, e.g. to identify compatible items by training on large corpora of co-purchase dyads [5].

In this paper: we seek to combine these two lines of work, by training image representations specifically for the purpose of fashion recommendation. In other words, we seek to use image content (at the pixel level) to build recommender systems. This follows the recent trend of incorporating representation learning techniques into recommender systems [6], and methodologically is most similar to the work of [7] where comparative judgments between images are modeled using a certain type of Siamese network. We adapt the popular formulation from Bayesian Personalized Ranking (BPR) [8] to include image content via a Siamese net and show significant improvements over BPR itself, as well as extensions of BPR that make use of pre-trained representations [1].

2374-8486/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDM.2017.30

A. Toward Recommender Systems for Design

Beyond demonstrating improved recommendation and retrieval performance, our second contribution is to show that our model can be combined with Generative Adversarial Networks (GANs) [9] to *generate* images that are consistent with those in the training corpus, and can be conditioned on a particular user to maximize their personal objective value. In other words, given a particular user and category, we can generate a new (image of an) item that is most consistent with a user's preference model. This idea consists of using a GAN to take us from the discrete space of fashion images to a *continuous* space, allowing us to explore the potential range of fashion items, suggest modifications to existing items, or generate items that are tailored to a specific user.

The items generated look plausible yet substantially different from those in the training corpus; this represents a first step toward using recommender systems not just to suggest existing items, but also to aid in the design of new ones.

Methodologically, our system builds upon BPR [8] and Siamese networks [10] to build a visually-aware personalized recommender system that for each user optimizes pairwise preferences between purchased versus non-purchased items, based on latent properties of the items and their product images. This allows us to generate personalized preference models and generate rankings of likely purchases. Next, we combine the learned system with a Generative Adversarial Net, so that given a user and product category, we can synthesize new items (or item images) that are maximally consistent with each user's preference model. Similar to the method of activation maximization [11], our synthesis component iteratively finds a low dimensional latent code, which generates an item image that maximizes a personal objective value.

Quantitatively, we perform experiments on a popular fashion recommendation corpus from *Amazon* [12], which includes hundreds of thousands of users, items, and reviews. We show state-of-the-art results in terms of recommendation performance (in terms of the AUC) compared to BPR and variants that make use of product images via pre-trained features [1], [8]. We also show that our GAN architecture is able to synthesize images with substantially higher objective values for each user than existing items in the corpus.

Qualitatively, the clothing images generated by our system are plausible and diverse, meaning that the system can feasibly be used in an exploratory way to understand the potential space of fashion items, and to identify items that match individual users' preferences. This is in line with recent work that aims to make recommender systems more interpretable and explainable, and to generate 'richer' recommendations—rather than simply recommending products from an existing corpus, we can help users and designers explore new styles and design new items.

The overall idea of our system is shown in Figure 1. We envision a system capable of performing both recommendation (retrieval) and synthesis: Our system architecture is based on the Siamese CNN framework, where the final layer of the CNN is treated as an item representation that can be used in a preference prediction framework; for image synthesis we use the framework of Generative Adversarial Nets, where a generator and discriminator are simultaneously trained so as to generate images that follow the same distribution as (and hence can't be distinguished from) those in the training corpus. We demonstrate this framework in a variety of design scenarios.

II. Related Work

We extend work on visually-aware recommendation with recent advances on representation learning and generative models of images. We highlight a few of the main ideas from each area below.

Recommender Systems: At their core, recommender systems seek to model 'compatibility' between users and items, based on historical observations of clicks, purchases, or interactions. *Matrix Factorization* (MF) methods relate users and items by uncovering latent dimensions such that users have similar representations to items they rate highly, and are the basis of many state-of-the-art approaches. We are concerned with personalized ranking from *implicit* feedback (e.g. distinguishing purchases from non-purchases, rather than estimating ratings), which can be challenging due to the ambiguity of interpreting 'non-observed' (e.g. non-purchased) data. Recently, *point-wise* and *pairwise* methods have successfully adapted MF to address such challenges.

Point-wise methods assume non-observed feedback to be inherently negative, and cast the task in terms of regression, either by associating 'confidence levels' to feedback [13], or by sampling non-observed feedback as negative instances [14].

Pairwise methods are based on a weaker but possibly more realistic assumption that positive feedback must only be 'more preferable' than non-observed feedback. Such methods directly optimize the ranking (in terms of the AUC) of the feedback and are to our knowledge state-of-the-art for implicit feedback datasets. In particular, Bayesian Personalized Ranking (BPR), has experimentally been shown to outperform a variety of competitive baselines [8]. Furthermore, BPR-MF (i.e., BPR with MF as the underlying predictor) has been successfully extended to incorporate (pre-trained) visual signals [1], and is thus the framework on which we build.

Besides images, others have developed content- and context-aware models that make use of a variety of information sources, including temporal dynamics [15], and geographic information [16], [17]. Like our work these fall under the umbrella of 'content aware' recommender systems, though are orthogonal due to the application domain and choice of signals considered.

Deep Learning-Based Recommender Systems: A variety of approaches have sought to incorporate 'deep' learning into recommender systems [6], including systems that make use of content-based signals, such as for recommendation of *YouTube* videos [18]. Our work is also a form of deep recommender system, though is orthogonal to existing approaches in terms of data modality, except for a few notable exceptions described below.

Visually-aware Recommender Systems: Recent works have introduced visually-aware recommender systems where users' rating dimensions are modeled in terms of visual signals in the system (product images). Such visual dimensions were demonstrated to be successful at link prediction tasks, such as recommending alternative (e.g. two similar t-shirts) and complementary (e.g. a t-shirt and a matching pair of pants) items [12], and POI recommendation [19].

The most closely related work is a recent method that extended BPR-MF to incorporate visual dimensions to facilitate item recommendation tasks [1]. We build upon this framework, showing that substantially improved performance can be obtained by using an 'end-to-end' learning approach (rather than pre-trained features).

Fashion and Clothing Style: Beyond the methods mentioned above, modeling fashion or style characteristics has emerged as a popular computer vision task in settings other than recommendation [20]–[22], e.g. with a goal to categorize or extract features from images, without necessarily building any model of a 'user.' This includes categorizing images as belonging to a certain style [23], as well as assessing items (or individuals [24]) for compatibility [12], [25].

Siamese Networks and Comparative Image Models: Convolutional Neural Networks (CNNs) have experienced a resurgence due to their success as general-purpose image models, especially for classification [26]. In particular, *Siamese nets* [10] have become popular for metric learning and retrieval, as they allow CNNs to be trained 'comparatively;' here two 'copies' of a CNN are joined by a function that compares their outputs. This type of architecture has been applied to discriminative tasks (like face verification) [27], as well as comparative tasks, such as modeling preference judgments between images [7]. Such models have also been used to learn notions of 'style' [28], including for fashion, by modeling the notion of item compatibility; however such systems are focused on comparison and retrieval, rather than personalized recommendation.

The closest works to ours in this line are those of Veit *et al.* [29] and Lei *et al.* [7]. The former considers item-to-item recommendation tasks on the same *Amazon* dataset used here, though is concerned with learning a *global* notion of compatibility and has no personalization component. The latter uses a similar triplet-embedding framework to ours, albeit for a different objective and a different domain. However neither work considers using the model *generatively* for image synthesis and design as we do here.

Image Generation and Generative Adversarial Networks: Generative Adversarial Networks (GANs) are an unsupervised learning framework in which two components 'compete' to generate realistic looking outputs (in particular, images) [9]. One component (a generator) is trained to generate images, while another (a discriminator) is trained to distinguish real versus generated images. Thus the generated images are trained to look 'realistic' in the sense that they are indistinguishable from those in the dataset.

TABLE I
NOTATION.

Notation	Explanation
$\mathcal{U}, \mathcal{I}$	user and item set
$\mathcal{I}_u^+$	positive item set for user u
$x_{u,i} \in \mathbb{R}$	predicted score user u gives to item i
$K \in \mathbb{N}$	latent factor dimensionality
$\boldsymbol{\theta}_u \in \mathbb{R}^K$	latent factor for user u
Φ	convolutional network for feature extraction
D	discriminator in GAN
G	generator in GAN
$\mathbf{z} \in [-1, 1]^{100}$	latent input code for the generator
Δ, ∇	image upscaling/downscaling operator
$\mathbf{X}_i \in \mathbb{R}^{224\times224\times3}$	product image for item i
X_c	product image set for category c

Such systems can also be conditioned on additional inputs, in order to sample outputs with certain characteristics [30]. We follow a similar approach based on activation maximization [11] to generate images that best match a selected user's personal style. GANs have been used for broad applications, though to our knowledge we are the first to apply this architecture to fashion image generation.

A. Key Differences

In summary, our method is distinct from most recommendation approaches through the use of visual signals, and specifically extends work on visually-aware recommendation by using richer image representations; this leads to state-of-the-art performance in terms of personalized ranking. Most novel is the ability to use our system *generatively*, suggesting a form of recommender system that can help with the exploration and design of new items.

III. METHODOLOGY

Our system has two important parts: a component for visual recommendation (essentially a combination of a Siamese CNN with Bayesian Personalized Ranking), and a component for image generation (based on Generative Adversarial Networks). Our notation is defined in Table I. We describe each component separately.

A. Visually-Aware Recommender Systems

We start by developing an end-to-end visually-aware deep Bayesian personalized ranking method (called *DVBPR*) to simultaneously extract task-guided visual features and to learn user latent factors. We first define the visually-aware recommendation problem with implicit feedback and introduce our preference predictor function. Afterward we outline our procedure for model training.

Problem Definition: We consider recommendation problems with implicit feedback (e.g. purchase/click histories, as opposed to ratings). In the implicit feedback setting, there is no negative feedback; instead our goal is to rank items such that 'observed' items are ranked higher than non-observed ones [8]. This approach was shown to be successful in previous work on visual recommendation [1], to which we compare our results.

We use $\mathcal{U}$ and $\mathcal{I}$ to denote the set of users and items (respectively) in our platform. For our implicit feedback dataset, $\mathcal{I}_u^+$ denotes a set that includes all items about which user u has expressed positive feedback (e.g. purchased, clicked). Each item $i \in \mathcal{I}$ is associated with an image, denoted $\mathbf{X}_i$. Our goal is to generate for each user u a personalized ranking over items with which the user u has not yet interacted.

Preference Predictor: Matrix Factorization (MF) methods have shown state-of-the-art performance both for rating prediction and modeling implicit feedback [8], [31]. A basic predictor (that we use as a building block) to predict the preference of a user u about an item i is

$$x_{u,i} = \alpha + \beta_u + \beta_i + \boldsymbol{\gamma}_u^T \boldsymbol{\gamma}_i, \tag{1}$$

where α is an offset, β_u and β_i are user/item biases, and $\boldsymbol{\gamma}_u$ and $\boldsymbol{\gamma}_i$ are latent factors that describe user u and item i respectively. These latent factors can be thought of as the 'preferences' of the user and the 'properties' of an item, such that a user is likely to interact with an item if their preferences are compatible with its properties.

Previous models for visually-aware recommendation made use of *pre-trained* visual features, like [1], adding additional item factors by embedding the visual features into a low-dimensional space:

$$x_{u,i} = \alpha + \beta_u + \beta_i + \overbrace{\boldsymbol{\gamma}_u^T \boldsymbol{\gamma}_i}^{\textit{latent}\text{ user-item preference}} + \underbrace{\boldsymbol{\theta}_u^T (\mathbf{E}\mathbf{f}_i)}_{\textit{visual}\text{ user-item preference}}. \tag{2}$$

Above $\boldsymbol{\theta}_u$ is a (latent) visual preference vector for user u, $\mathbf{f}_i \in \mathbb{R}^{4096}$ is a pre-extracted CNN feature representation of item i, and $\mathbf{E} \in \mathbb{R}^{K \times 4096}$ is an embedding matrix that projects $\mathbf{f}_i$ into a K-dimensional latent space, whose dimensions correspond to facets of 'fashion style' that explain variance in users' opinions.

The main issue we seek to address with the method above is its reliance on *pre-trained* features that were optimized for a different task (namely, classification on ImageNet). Thus we wish to evaluate the extent to which task-specific features can be used to improve performance within an end-to-end framework. This builds upon recent work on comparative image models [7], [29], though differs in terms of the formulation and training objective used.

To achieve this, we replace the pre-trained visual features and embedding matrix with a CNN network $\Phi(\cdot)$ to extract visual features directly from the images themselves:

$$x_{u,i} = \alpha + \beta_u + \boldsymbol{\theta}_u^T \Phi(\mathbf{X}_i). \tag{3}$$

Unlike standard MF (eq. 1) and VBPR (eq. 2), we exclude item bias terms β_i and non-visual latent factors γ_i. Empirically, we found that doing so improved performance, and that the remaining terms in the model are sufficient to capture these factors implicitly.

Feature Extraction: Our Convolutional Neural Network $\Phi(\cdot)$ is based on the *CNN-F* architecture from [32]. This is a CNN architecture designed specifically for efficient training. Using more powerful architectures (e.g. ResNet [33]), may achieve better performance; however, we found that CNN-F is sufficient to show the effectiveness of our method, while allowing for training on standard desktop hardware. Specifically, CNN-F consists of 8 learnable layers, 5 of which are convolutional, while the last 3 are fully-connected. The input image size is 224×224.

Details of our architecture are shown below (see [32], Table 1 for details; st=stride; pad=spatial padding):

conv1	conv2	conv3	conv4	conv5	full6	full7	full8
64x11x11	256x5x5	256x3x3	256x3x3	256x3x3	4096	4096	K
st. 4, pad 0	st. 1, pad 2	st. 1, pad 1	st. 1, pad 1	st. 1, pad 1	drop-	drop-	-
x2 pool	x2 pool	-	-	x2 pool	out	out	-

In our experiments, the probability of dropout is set to 0.5, and the weight decay term is set to 10^{-3}.

Note one difference between the above architecture and that of CNN-F: our last layer has K dimensions, whereas that of CNN-F has 1000. Here, instead of seeking a final layer that can be adapted to general-purpose prediction tasks, we hope to learn a representation whose dimensions explain the variance in users' fashion preferences.

B. Learning the Model

Bayesian Personalized Ranking (BPR) is a state-of-the-art ranking optimization framework for implicit feedback. In BPR, the main idea is to optimize rankings by considering triplets $(u, i, j) \in \mathcal{D}$, where

$$\mathcal{D} = \{(u, i, j) | u \in \mathcal{U} \wedge i \in \mathcal{I}_u^+ \wedge j \in \mathcal{I} \setminus \mathcal{I}_u^+\}.$$

Here $i \in \mathcal{I}_u^+$ is an item about which the user u has expressed interest, whereas $j \in \mathcal{I} \setminus \mathcal{I}_u^+$ is one about which they have not. Thus intuitively, for a user u, the predictor should assign a larger preference score to item i than item j. Hence BPR defines

$$x_{u,i,j} = x_{u,i} - x_{u,j} \tag{4}$$

as the difference between preference scores, and seeks to optimize an objective function given by

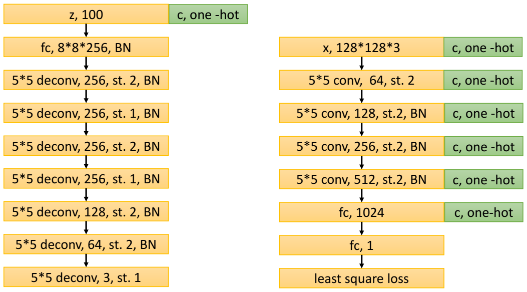
$$\max \sum_{(u,i,j) \in \mathcal{D}} \ln \sigma(x_{u,i,j}) - \lambda_\Theta \|\Theta\|^2, \tag{5}$$

where $\sigma(\cdot)$ is the sigmoid function, Θ includes all model parameters, and λ_Θ is a regularization hyperparameter.

By considering a large number of samples of non-observed items $j \in \mathcal{I} \setminus \mathcal{I}_u^+$, this method can be shown to approximately optimize the AUC in terms of ranking observed feedback for each user [8].

Note that in (eq. 4), the global bias term α and user bias term β_u can be discarded, as they cancel between $x_{u,i}$ and $x_{u,j}$. Hence the final form of our preference predictor can be simplified as

$$x_{u,i} = \boldsymbol{\theta}_u^T \Phi(\mathbf{X}_i). \tag{6}$$



(a) Generator G(z,c)

(b) Discriminator D(x,c)

Fig. 2. Model architecture of our GAN models.

Since all parts of the objective are differentiable, we perform optimization by stochastic gradient ascent using the *Adam* optimizer [34], a state-of-the-art stochastic optimization method with adaptive estimation of moments.

During each iteration of stochastic gradient ascent, we sample a user u, a positive item $i \in \mathcal{I}_u^+$, and a negative item $j \in \mathcal{I} \setminus \mathcal{I}_u^+$. Thus the convolutional network Φ has *two* images to consider, $\mathbf{X}_i$ and $\mathbf{X}_j$. Following the Siamese setup [10], both CNNs $\Phi(\mathbf{X}_i)$ and $\Phi(\mathbf{X}_j)$ share the same weights.

C. From Recommendation to Generation

DVBPR evaluates a user preference score for a given image. But a richer form of recommendation might consist of guiding users and designers by helping them to explore the space of *potential* fashion images and styles. This requires a method that can efficiently sample arbitrary realistic-looking images from a prior distribution. Generative Adversarial Networks (GANs) [9] offer an effective way to capture such a distribution and have demonstrated promising results in a variety of applications [35]. In particular, conditional GANs [30] are a useful variant, which can generate images according to semantic inputs like class labels [36], textual descriptions [37], etc. In our case, we train a GAN conditioned on product's top-level category.

A GAN consists of a generator G and a discriminator D, which are usually implemented as multi-layer convolutional or deconvolutional neural networks. The generator $G(\mathbf{z}, c)$ takes a random noise vector $\mathbf{z} \sim U(-\mathbf{1}, \mathbf{1})$ and a category c as inputs and synthesizes an image. The discriminator takes an image $\mathbf{x}$ sampled either from training data X_c or synthesized examples $G(\mathbf{z}, c)$, and predicts the likelihood of the image being 'real' (i.e., belonging to the training set X_c). We train our GAN by using a least squares loss as in an LSGAN [38], which is a newly developed GAN with high image quality. The objective functions of our discriminator and generator are defined as:

$$\begin{aligned} \min_D V(D) &= \mathbb{E}_{\mathbf{x}, c \sim p_{\text{data}}(\mathbf{x}, c)} L_{real}(\mathbf{x}, c) \\ &\quad + \mathbb{E}_{c \sim p(c), \mathbf{z} \sim p(\mathbf{z})} L_{fake}(G(\mathbf{z}, c), c), \qquad (7) \\ \min_G V(G) &= \mathbb{E}_{c \sim p(c), \mathbf{z} \sim p(\mathbf{z})} L_{real}(G(\mathbf{z}, c), c), \end{aligned}$$

where $L_{real}(\mathbf{x}, c) = [D(\mathbf{x}, c) - 1]^2$ and $L_{fake}(\mathbf{x}, c) = [D(\mathbf{x}, c)]^2$. This means the discriminator D tries to predict '1' for real images and '0' for fake images, while the generator G wants to generate 'realistic' images to fool D. These two opposing objectives are optimized alternately until the quality of generated images is acceptable (around 25 epochs in our case). In Figure 2, we show the details of our model architectures which are deeper conditional networks modified from [35], [38].

Compared to some GAN results that generate 'dream like' hallucinations of natural images, our generated images look realistic, presumably because we operate in a relatively circumscribed domain (clothing images viewed in one of a few canonical poses). This means that the samples generated by our network are distinct from, yet similar in quality to, existing images in the training corpus.

D. Personalized Design

Although recommender systems are concerned with modeling users' preferences, this is typically limited to identifying items to which users would assign a high rating (or high purchase probability). Although latent variable models (and in particular matrix factorization) are able to uncover surprisingly complex semantics, they are limited in their interpretability. By using our model to generate images, we can begin to reason about what the model 'knows,' and in particular explore the space of potentially desirable items that do not yet exist.

Such technology could support applications such as (1) Sampling new items, including items tailored to a specific user's preferences; (2) Tailoring existing items, i.e., making small modifications such that an item better matches the preferences of a user. These applications are useful for interpreting users' preferences learned from our model, and also for providing inspiration to designers. We describe each of these scenarios in detail in our experiments.

Our model associates preference scores between users and items (images), suggesting that by generating images that maximize this preference value we might be able to produce items that best match a user's personal style. This process, called *preference maximization*, is similar to activation maximization methods [11], which try to interpret deep neural networks by finding inputs that maximize a particular neuron (e.g. a classification neuron in the final layer).

Preference Maximization: Formally, given a user u and a category c, we can straightforwardly retrieve existing items in the dataset to maximize a user's preference score:

$$\delta(u, c) = \operatorname*{argmax}_{e \in X_c} \; x_{u,e} = \operatorname*{argmax}_{e \in X_c} \; \boldsymbol{\theta}_u^T \Phi(e), \qquad (8)$$

where X_c is the set of item images belonging to category c.

While the above selects a 'real' image from the training corpus, our GAN provides us with an approximated distribution of the original data. Thus we can synthesize an image by optimizing

$$\begin{aligned}\widehat{\delta}(u,c) &= \operatorname*{argmax}_{e\in G(\cdot,c)}\ \widehat{x}_{u,e} = \operatorname*{argmax}_{e\in G(\cdot,c)}\ x_{u,e} - \eta L_{real}(e,c)\\ &= G\left[\operatorname*{argmax}_{\mathbf{z}\in[-1,1]^{100}}\ \boldsymbol{\theta}_u^T\Phi\big[\Delta G_c(\mathbf{z})\big] - \eta\big[D_c(G_c(\mathbf{z}))-1\big]^2, c\right],\end{aligned} \tag{9}$$

where $D_c(\mathbf{x}) = D(\mathbf{x},c)$, $G_c(\mathbf{z}) = G(\mathbf{z},c)$, $\mathbf{z}$ is a latent input code for the generator G and Δ is an image upscaling operator which resizes the RGB image from 128×128 to 224×224. We used nearest-neighbor scaling, though any differentiable image scaling algorithm would work. We also add a term $\eta L_{real}(e,c)$ to control image quality via the learned discriminator. The hyper-parameter η controls the trade-off between preference score and image quality, which we will analyze in our experiments.

To optimize (eq. 9) given the constraint $\mathbf{z} \in [-1,1]^{100}$, we introduce an auxiliary variable $\mathbf{z}' \in \mathbb{R}^{100}$, and let $\mathbf{z} = \tanh(\mathbf{z}')$. Thus our optimization problem becomes

$$\max_{\mathbf{z}'\in\mathbb{R}^{100}}\ \boldsymbol{\theta}_u^T\Phi\big[\Delta G_c(\tanh(\mathbf{z}'))\big] - \eta\big[D_c(G_c(\tanh(\mathbf{z}')))-1\big]^2, \tag{10}$$

which we solve via gradient ascent.

To sample initial points within the space we draw $\mathbf{z} \sim U(-\mathbf{1},\mathbf{1})$, and let $\mathbf{z}' = \tanh^{-1}(\mathbf{z}) = \frac{1}{2}[\ln(1+\mathbf{z}) - \ln(1-\mathbf{z})]$ where $\tanh^{-1}(\cdot)$ and $\ln(\cdot)$ are applied elementwise. Note that such an objective can potentially have many local optima. To get a high quality solution, we repeat the optimization process from m random initial points ($m = 64$ in our experiments; larger m could yield better solutions but is also more time-consuming), choosing whichever has the highest objective value after optimization.

The method $\widehat{\delta}(u,c)$ can also easily adapt to generate multiple images. For example, after optimization, we rank the m images $\{e_1, e_2, ..., e_m\}$ according to their objective value (i.e., $\widehat{x}_{u,e}$), and return the top-k images. However, we found results from this method have poor diversity. Hence we perform sampling when returning multiple images, with the probability of choosing e_t as

$$p(e_t) = \frac{\exp(\widehat{x}_{u,e_t})}{\sum_{d=1}^{d=m}\exp(\widehat{x}_{u,e_d})}. \tag{11}$$

Adjusting Existing Items: Rather than sampling completely new items, it may be desirable to make minor alterations to those in the existing corpus, to suggest how items might be 'tailored' to a user's personal style. Using generated images from GANs allows us to manipulate images through $\mathbf{z}$, which requires us to first find an approximated image (equivalent to finding a latent code $\mathbf{z}$) for a query image in the GAN's latent space. This can be achieved by the following optimization problem, in terms of an ℓ_1 reconstruction error:

$$\min_{\mathbf{z}'\in\mathbb{R}^{100}}\ \left\|G_c\left(\tanh(\mathbf{z}')\right) - \nabla\mathbf{X}_{\text{query}}\right\|_1. \tag{12}$$

We consider each of the two cases above in our qualitative evaluation in the following section.

IV. Experiments

We perform both quantitative experiments, to evaluate ranking performance and generated images, as well as qualitative experiments to demonstrate the capacity of our system to perform image synthesis and design. (Data and code are available on the first author's webpage.)

A. Experimental Setting

All experiments of our method were conducted on a commodity workstation with a 4-core Intel CPU and a single GTX-1080 graphics card. Note that although our dataset contains hundreds of thousands of images, and around a million user-item interactions, it is still possible to train on commodity hardware requiring around one day of training.

Dataset: Our first group of datasets were introduced in [12] and consist of reviews of clothing items crawled from *Amazon.com*. We first extract a subset called *Amazon Fashion*, which contains six representative fashion categories (men/women's tops, bottoms and shoes). We also consider two comprehensive datasets, containing all subcategories (gloves, scarves, sunglasses, etc.), named *Amazon Women* and *Amazon Men*. We treat users' reviews as implicit feedback.

Another dataset is crawled from *Tradesy.com* [1], which is a c2c online platform for buying and selling used fashion items. The dataset includes several kinds of feedback, like clicks, purchases, sales, etc. We treat these as implicit feedback as in [1]. For all four datasets, each item has exactly one associated image and a pre-extracted CNN feature representation.

For data preprocessing, we discard inactive users u for whom $|\mathcal{I}_u^+| < 5$, as in VBPR [1]. For each user, we randomly withhold one action for validation $\mathcal{V}_u$, and another for testing $\mathcal{T}_u$. All remaining items are used for training $\mathcal{P}_u$, and we always report performance for the model that achieved the best performance on our validation set. Statistics of our datasets are given in Table II. Note that we focus our qualitative evaluation on *Amazon Fashion*, while other datasets are used to compare recommendation performance.

TABLE II
Dataset statistics (after preprocessing)

Dataset	#Users	#Items	#Interactions	#Categories
Amazon Fashion	64583	234892	513367	6
Amazon Women	97678	347591	827678	53
Amazon Men	34244	110636	254870	50
Tradesy.com	33864	326393	655409	N/A

Evaluation Metrics: We calculate the AUC to measure recommendation performance of our method and that of baselines. The AUC measures the quality of a ranking based on pairwise comparisons (and is the measure that BPR-like methods are trained to optimize). Formally, we have

$$AUC = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\frac{1}{|\mathcal{D}_u|}\sum_{(i,j)\in\mathcal{D}_u}\xi(x_{u,i} > x_{u,j}),$$

where $\mathcal{D}_u = \{(i,j)|(u,i) \in \mathcal{T}_u \wedge (u,j) \notin (\mathcal{P}_u \cup \mathcal{V}_u \cup \mathcal{T}_u)\}$ and $\xi(\cdot)$ is an indicator function. In other words, we are counting the fraction of times that the 'observed' items i are preferred over 'non-observed' items j.

When generating images, we use three metrics to evaluate preference scores, image quality, and diversity. For a given user u, a large preference score $x_{u,i}$ suggests that the user u would be interested in item i. From the perspective of our model, this ought to be true even for generated images. Hence, the compared methods (i.e., our retrieval-based method $\delta(u,c)$ and synthesis-based method $\widehat{\delta}(u,c)$) should find an item with its objective value as large as possible. Here we randomly sample a user u and set c to the category of the item in the test set $\mathcal{T}_u$, and compute the mean objective value (i.e., eq. 6) of compared methods. For image quality, we report the inception score [39], which is a commonly used heuristic approach to measure image quality, based on a standard pre-trained inception network. Higher scores typically mean better quality. For diversity, similar to [36], we calculate the visual similarity of pairs of returned images for each query (i.e., a user and a category), and then take the average of multiple queries. The visual similarity is measured by structural similarity (SSIM) [40], which is more consistent with human visual perception than the mean squared error and other traditional measures. SSIM ranges from 0 to 1, where higher values indicate more similar images. We report Opposite Mean SSIM, which is one minus mean SSIM, to represent diversity. Thus a higher value means better diversity.

Both metrics for measuring image quality and diversity have limitations and are subjective. Thus we also show qualitative results of synthetic images under different image generation scenarios, which we describe further in Section IV-D.

Ultimately while we are unable to directly evaluate the 'designed products' against real users, our evaluation shows that (1) Our end-to-end learning approach is state-of-the-art in terms of recommendation performance (AUC); (2) Our optimization process is able to identify plausible images that are distinct from those in the training corpus, and have higher preference scores than those of existing products.

B. Baselines

When evaluating methods in terms of their AUC, we compare our method against the following baselines:

Random (RAND) Images are ranked in a random order. By definition this method has AUC $= 0.5$.

PopRank Images are ranked in order of their popularity within the system.

WARP Matrix factorization for optimizing top-k performance with weighted approximated ranking pairwise (WARP) loss [41].

BPR-MF Introduced by [8], is a state-of-the-art method for personalized ranking on implicit feedback datasets. It uses standard MF (i.e., eq. 1) as the underlying predictor.

Besides standard ranking methods that only use implicit feedback, we also include stronger methods that are able to exploit item visual features:

VisRank A simple content-based method based on visual similarity (i.e., distance of pre-trained CNN features) among items. Images are ranked according to their average distance to items bought by the user.

Factoization Machines (FM) Proposed by [42], provides a generic factorization approach that can be used to estimate the score of an item given by a user. We use the same BPR loss to optimize personalized ranking. Also as in [43], we only consider interactions between (user, item) and (user, item's CNN feature).

VBPR A state-of-the-art method for visually-aware personalized ranking from implicit feedback [1]. VBPR is a form of hybrid content-aware recommendation that makes use of pre-trained CNN features of product images.

DVBPR The method proposed in this paper.

These baselines are intended to show (a) the importance of learning personalized notions of compatibility for fashion recommendation (MF methods vs. PopRank); (b) the improvement to be gained by incorporating visual features directly into the recommendation pipeline (FM/VBPR/DVBPR vs. MF methods); and (c) the improvement to be gained by learning image representations expressly for the task of fashion recommendation, rather than relying on features from a pre-trained model (DVBPR vs. FM/VBPR). Note that there are alternative methods that make use of other side information (e.g. social or temporal) for recommendation [3], [4], though these are orthogonal to our approach. We also considered related methods based on the Siamese setup [7], [29], though these were not directly comparable as they optimize different objectives or consider different problem settings (item-to-item recommendation, user features, etc.).

For WARP and FM, we use LightFM's [43] implementation; for other baselines we use our own implementation. DVBPR is implemented in *TensorFlow*.

For WARP, BPR-MF, FM and VBPR, we tune hyper-parameters via grid search on a validation set, with a regularizer selected from $\{0, 0.001, 0.01, 0.1, 1, 10, 30\}$ and the number of latent factors selected from $\{10, 30, 50\}$. We found that DVBPR is not overly sensitive to hyper-parameter tuning, hence we use a single setting for all datasets. We used a mini-batch size of 128 for DVBPR and 64 for GAN training. We set the number of latent factors $K = 50$ and regularization hyper-parameter $\lambda_{\theta_u} = 1$. Regularization of item factors is achieved via weight decay and dropout of the CNN-F network.

C. Quantitative Evaluation

Recommendation Performance: We report recommendation performance in terms of the AUC in Table III. We consider two settings, *All Items* and *Cold Items*. For the latter, we seek to estimate relative preference scores among items that have few observations at training time (fewer than 5).

On average, our method outperforms the second-best method by 5.13% across all datasets, and 2.73% in cold-start scenarios. Our method outperforms the strongest content-*un*aware method (BPR-MF) substantially; in fact, even our simple 'nearest-neighbor' style baseline (VisRank) outperforms content-unaware methods on this data. The poor performance of MF based methods is largely due to the extreme

TABLE III

RECOMMENDATION PERFORMANCE IN TERMS OF THE AUC (HIGHER IS BETTER). THE BEST PERFORMING METHOD IN EACH ROW IS BOLDFACED.

Dataset	Setting	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	Improvement	
		RAND	PopRank	WARP	BPR-MF	VisRank	FM	VBPR	DVBPR	h vs. d	h vs. best
Amazon Fashion	All Items	0.5	0.5849	0.6065	0.6278	0.6839	0.7093	0.7479	**0.7964**	26.9%	6.5%
	Cold Items	0.5	0.3905	0.5030	0.5514	0.6807	0.7088	0.7319	**0.7718**	40.0%	5.5%
Amazon Women	All Items	0.5	0.6192	0.5998	0.6543	0.6512	0.6678	0.7081	**0.7574**	15.8%	7.0%
	Cold Items	0.5	0.3822	0.5017	0.5196	0.6387	0.6682	0.6885	**0.7137**	37.4%	3.7%
Amazon Men	All Items	0.5	0.6060	0.6081	0.6450	0.6589	0.6654	0.7089	**0.7410**	14.9%	4.5%
	Cold Items	0.5	0.3903	0.5005	0.5132	0.6545	0.6705	0.6863	**0.6923**	34.9%	0.9%
Tradesy.com	All Items	0.5	0.5268	0.6176	0.5860	0.6457	0.7662	0.7500	**0.7857**	34.1%	2.5%
	Cold Items	0.5	0.3946	0.5333	0.5418	0.6084	0.7730	0.7525	**0.7793**	43.8%	0.8%

sparsity of the data (2-3 observations per item on average) and is consistent with previous reported results on the same datasets [1]. This confirms the importance of content-aware recommendation methods on extremely long-tailed applications like fashion recommendation.

Evaluation of Generated Images: Next, our goal is to quantitatively compare real versus generated images. As we stated in Section IV-A, we use the mean objective value (i.e., eq. 6), inception score [39] and Opposite SSIM [36], [40] to evaluate how well images match user preferences, image quality, and diversity, respectively.

Table IV shows evaluation results of images from the four sources. The first two methods are random methods, which draw real images (from X_c), or generated images (from $G(\cdot, c)$), by feeding uniform random vectors $\mathbf{z}$. Both methods exhibit poor performance in term of preference score, since they don't consider the given user in each query. However, the generated images can achieve comparable performance regarding quality and diversity, which verifies the effectiveness of GAN training in that the GAN can generate realistic and diverse (rather than noisy or identical) images.

The last two methods $\delta(u, c)$ and $\widehat{\delta}(u, c)$ are personalized methods, hence they both achieve higher preference scores compared to random methods. In particular, even in a comprehensive dataset like *Amazon's* clothing catalog, a synthesis-based method ($\widehat{\delta}(u, c)$) can gain 6.8% improvement over a retrieval-based method ($\delta(u, c)$) in terms of preference score. This shows the generative model can uncover potentially desirable items that don't exist in the dataset. Furthermore, $\widehat{\delta}(u, c)$ achieves similar quality and slightly lower diversity compared to $\delta(u, c)$.

In the method $\widehat{\delta}(u, c)$, there is an important hyper-parameter η to control the trade-off between preference score and quality. In Figure 3, we plot the performance of the three metrics under different η. It is clear that as η increases, the preference score drops and the quality becomes better, which is consistent with our intention. However η doesn't have an obvious impact on diversity. According to these results, we choose $\eta = 1$ as the 'sweet spot' in all experiments, as it results in better preference scores and similar quality compared to $\delta(u, c)$.

TABLE IV

COMPARISON OF TOP-3 RETURNED IMAGES FOR A GIVEN USER AND CATEGORY (1000 TRIALS). FOR ALL THREE METRICS, A LARGER VALUE IS BETTER. THE VALUE AFTER THE ± IS THE STANDARD DEVIATION.

Item Source	Preference Score	Quality	Diversity
Random Methods			
X_c	-1.9547±3.8	6.8685±.36	0.5585±.09
$G(\cdot, c)$	-1.9392±3.7	6.8121±.37	0.5589±.09
Personalized Methods			
$\delta(u, c)$ in (eq. 8)	7.1876±3.2	7.6492±.27	0.5393±.11
$\widehat{\delta}(u, c)$ in (eq. 9)	7.6760±4.1	7.6524±.24	0.5265±.12

D. Qualitative Evaluation

The above experiments suggest a form of recommender system that can aid in the design and exploration of new items. Finally, we assess our system qualitatively, by demonstrating image generation results following each of the settings in Section III-D: image sampling, personalized design, and prototype-based tailoring.

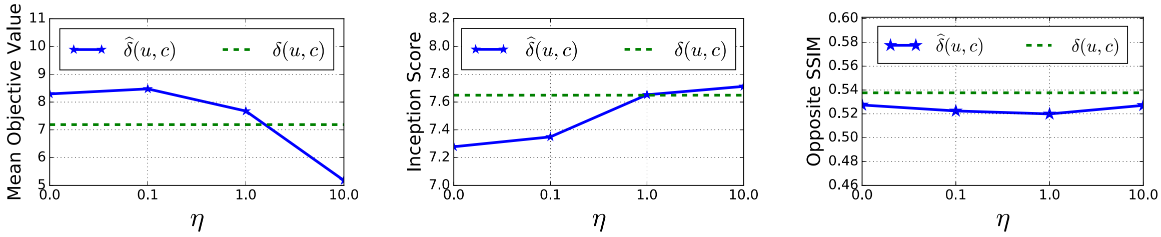
First, Figure 4 compares sampled images from each of six categories (women/men's tops, bottoms and shoes) to their nearest neighbors in the dataset. This demonstrates that the generated images are realistic and plausible, yet are quite different from any images in the original dataset—they have common shape and color profiles, but quite different styles.

Fashion design for a given user: Next, we examine images that have been personalized to a single user (i.e., Top-3 returned images from $\delta(u, c)$ and $\widehat{\delta}(u, c)$).

Results for six users are shown in Figure 5. We select the top 3 images generated by the GAN (in terms of personalized objective value) with the top 3 images in the dataset. Note that while the images generated are again quite different from those in the training corpus, they seem to belong to a similar style, indicating that the users' individual styles have been effectively captured.

Fashion design based on a prototype: The second type of personalized design consists of making small modifications to existing items so that they more closely match users' preferences.

Figure 6 shows the results of this optimization process (described in eqs. 12 and 10) for six users based on a given t-shirt or pair of pants. There are a few relevant observations



(a) Preference Score

(b) Image Quality

(c) Image Diversity

Fig. 3. Effect of hyper-parameter η.

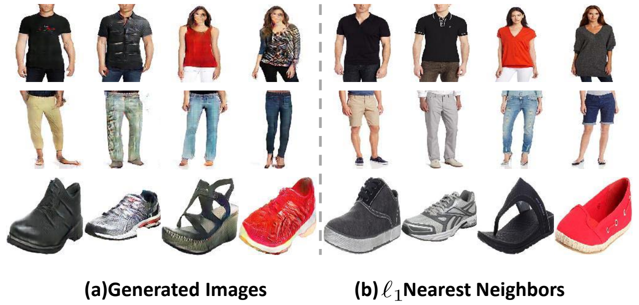


Fig. 4. Generated image samples and their ℓ_1 nearest neighbors in the dataset (ℓ_1 and ℓ_2 nearest neighbors were equivalent in almost all cases). Note that all images are rescaled to be square during preprocessing.

to make: First, the optimization process substantially increases the preference score over that of the original image. Second, the L1-approximated images are similar to the given prototype, and most images are in the vicinity of the original image. Third, the process highlights the continuous nature of the space learned by the GAN. The types of modifications vary for different users, including but not limited to changing color, extending sleeve length (row 3), 'distressing' pants (row 5), shortening pants (row 6) and other minor stylistic changes.

V. Conclusions and Future Work

We have presented a system for fashion recommendation that is capable not only of suggesting existing items to a user, but which is also capable of generating new, plausible fashion images that match user preferences. This suggests a new type of recommendation approach that can be used for both prediction and design.

We extended previous work on visually-aware recommendation using an end-to-end learning approach based on the Siamese-CNN framework. This leads to substantially more accurate performance than methods that make use of pre-trained representations. We then used the framework of Generative Adversarial Networks to learn the distribution of fashion images and generate novel fashion items that maximize users' preferences. This framework can be used in a variety of recommendation scenarios, to explore the space of possible fashion items, to modify existing items, or to generate items tailored to an individual.

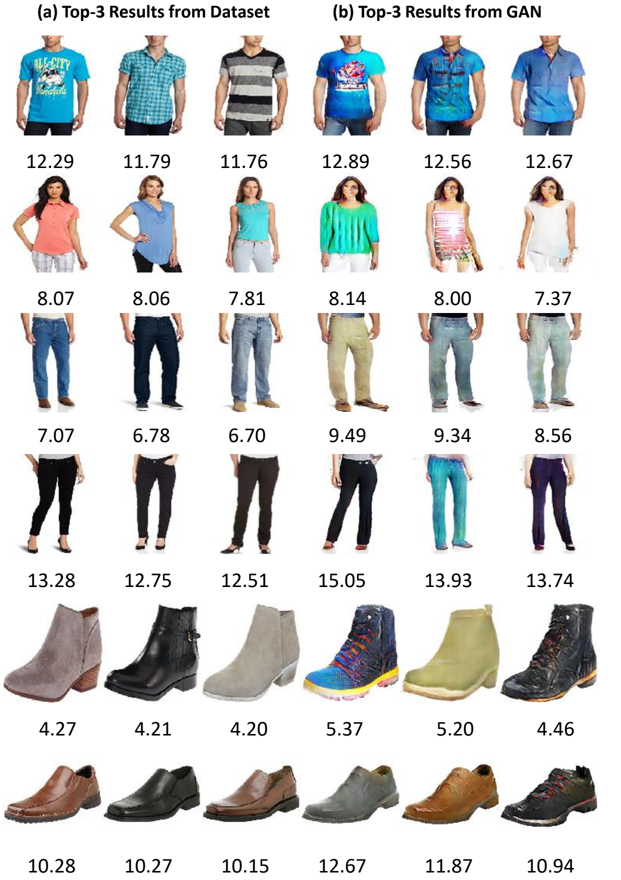


Fig. 5. Top-3 results from the dataset and GAN. Each row is a separate retrieval/optimization process for a given user and product category. Left: real images; right: synthetic images. The values shown are preference scores for each image.

In the future, we believe this opens up a promising line of work in using recommender systems for design. Other than improving the quality of the generated images and providing control of fine-grained styles, the same ideas can be applied to visual data besides fashion images, or even to non-visual forms of content. We believe that such frameworks can lead to richer forms of recommendation, where content recommendation and content generation are more closely linked.

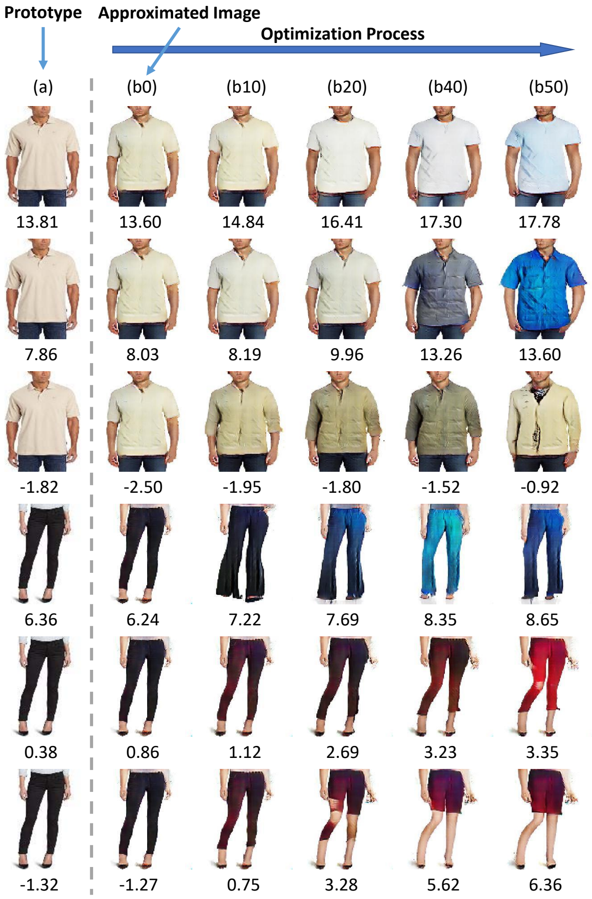


Fig. 6. Optimization for different users given the same initial image. Each row is a separate process for a different user, given the real image in column (a). Columns (b0) through (b50) show the image after successive iterations of GAN optimization. The preference score of each image is shown below the image.

References

[1] R. He and J. McAuley, "VBPR: visual bayesian personalized ranking from implicit feedback," in *AAAI*, 2016.

[2] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *MM*, 2014.

[3] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *WWW*, 2016.

[4] R. He, C. Fang, Z. Wang, and J. McAuley, "Vista: A visually, socially, and temporally-aware model for artistic recommendation," in *RecSys*, 2016.

[5] A. Veit, B. Kovacs, S. Bell, J. McAuley, K. Bala, and S. Belongie, "Learning visual clothing style with heterogeneous dyadic cooccurrences," in *ICCV*, 2015.

[6] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative deep learning for recommender systems," in *SIGKDD*, 2015.

[7] C. Lei, D. Liu, W. Li, Z.-J. Zha, and H. Li, "Comparative deep learning of hybrid representations for image recommendations," in *CVPR*, 2016.

[8] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: bayesian personalized ranking from implicit feedback," in *UAI*, 2009.

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014.

[10] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *CVPR*, 2006.

[11] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune, "Synthesizing the preferred inputs for neurons in neural networks via deep generator networks," in *NIPS*, 2016.

[12] J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, "Image-based recommendations on style and substitutes," in *SIGIR*, 2015.

[13] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *ICDM*, 2008.

[14] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang, "One-class collaborative filtering," in *ICDM*, 2008.

[15] J. Yang, C. Liu, M. Teng, H. Xiong, M. Liao, and V. Zhu, "Exploiting temporal and social factors for B2B marketing campaign recommendations," in *ICDM*, 2015.

[16] D. Lian, Y. Ge, F. Zhang, N. J. Yuan, X. Xie, T. Zhou, and Y. Rui, "Content-aware collaborative filtering for location recommendation based on human mobility data," in *ICDM*, 2015.

[17] Z. Yao, Y. Fu, B. Liu, Y. Liu, and H. Xiong, "POI recommendation: A temporal matching between POI popularity and user regularity," in *ICDM*, 2016.

[18] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *RecSys*, 2016.

[19] S. Wang, Y. Wang, J. Tang, K. Shu, S. Ranganath, and H. Liu, "What your images reveal: Exploiting visual contents for point-of-interest recommendation," in *WWW*, 2017.

[20] Y. Li, L. Cao, J. Zhu, and J. Luo, "Mining fashion outfit composition using an end-to-end deep learning approach on set data," *TMM*, 2017.

[21] X. Han, Z. Wu, Y. Jiang, and L. S. Davis, "Learning fashion compatibility with bidirectional lstms," in *MM*, 2017.

[22] Z. Al-Halah, R. Stiefelhagen, and K. Grauman, "Fashion forward: Forecasting visual style in fashion," in *ICCV*, 2017.

[23] L. Bossard, M. Dantone, C. Leistner, C. Wengert, T. Quack, and L. Van Gool, "Apparel classification with style," in *ACCV*, 2013.

[24] A. C. Murillo, I. S. Kwak, L. Bourdev, D. Kriegman, and S. Belongie, "Urban tribes: Analyzing group photos from a social perspective," in *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2012.

[25] S. Vittayakorn, K. Yamaguchi, A. C. Berg, and T. L. Berg, "Runway to realway: Visual analysis of fashion," in *WACV*, 2015.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[27] J. Hu, J. Lu, and Y.-P. Tan, "Discriminative deep metric learning for face verification in the wild," in *CVPR*, 2014.

[28] S. Bell and K. Bala, "Learning visual similarity for product design with convolutional neural networks," in *TOG*, 2015.

[29] A. Veit, B. Kovacs, S. Bell, J. McAuley, K. Bala, and S. Belongie, "Learning visual clothing style with heterogeneous dyadic cooccurrences," in *ICCV*, 2015.

[30] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[31] Y. Koren and R. Bell, "Advances in collaborative filtering," in *Recommender Systems Handbook*. Springer, 2011.

[32] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *BMVC*, 2014.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[35] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[36] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *ICML*, 2017.

[37] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *ICML*, 2016.

[38] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," *arXiv preprint ArXiv:1611.04076*, 2016.

[39] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *NIPS*, 2016.

[40] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *TIP*, 2004.

[41] J. Weston, S. Bengio, and N. Usunier, "Wsabie: Scaling up to large vocabulary image annotation," in *IJCAI*, 2011.

[42] S. Rendle, "Factorization machines," in *ICDM*, 2010.

[43] M. Kula, "Metadata embeddings for user and item cold-start recommendations," *arXiv preprint arXiv:1507.08439*, 2015.