Personalized Fashion Design

Cong Yu[†], Yang Hu[§*], Yan Chen[§], Bing Zeng[§]
School of Information and Communication Engineering
University of Electronic Science and Technology of China
[†]congyu@std.uestc.edu.cn, [§]{yanghu,eecyan,eezeng}@uestc.edu.cn

Abstract

Fashion recommendation is the task of suggesting a fashion item that fits well with a given item. In this work, we propose to automatically synthesis new items for recommendation. We jointly consider the two key issues for the task, i.e., compatibility and personalization. We propose a personalized fashion design framework with the help of generative adversarial training. A convolutional network is first used to map the query image into a latent vector representation. This latent representation, together with another vector which characterizes user's style preference, are taken as the input to the generator network to generate the target item image. Two discriminator networks are built to guide the generation process. One is the classic real/fake discriminator. The other is a matching network which simultaneously models the compatibility between fashion items and learns users' preference representations. The performance of the proposed method is evaluated on thousands of outfits composited by online users. The experiments show that the items generated by our model are quite realistic. They have better visual quality and higher matching degree than those generated by alternative methods.

Figure 1. Example outfits our model designs. In the upper case, we design bottoms (items in red box) to go with the given tops, and in the lower case, we design tops (items in red box) to go with the given bottoms. All these designs are user specific.

1. Introduction

With the rapid evolution of the fashion industry toward an online business, fashion related computer vision problems have attracted increasing attention nowadays. One task that is of particular interest is fashion recommendation [4,7,9,10,16,21,35–37], which suggests clothing items that fit well with a given item. The key to fashion recommendation is to model the compatibility between fashion items. Various methods such as distance metric learning [7,25,36], Siamese networks [37] and Recurrent Neural Networks [4, 16] have been explored. Despite of their success in predicting the compatibility, there are still problems when applying them in real world scenarios. Note that these methods only measure the compatibility between existing items. When the given inventory is small or limited, it is possible that there is no item good enough to complement the query. On the other hand, when the inventory is large, generating the recommendation may face some efficiency problem since one needs to compute the compatibility for each item, which is computational expensive due to the usage of deep neural networks by most methods.

In this work, instead of suggesting existing items, we synthesize images of new items that are compatible to a given one. This solves the deficit problem for small inventory. For large inventory, when targeting real items is necessary, we can just search items that are similar with the synthesized one. This is more efficient than exhaustive compatibility evaluation since similarity search can be very fast

*The corresponding author is Yang Hu. This work was supported by the National Natural Science Foundation of China (61602090) and the 111 Project (B17008).

with techniques like hashing. The ability to generate new items for outfit composition also facilitates fashion design and manufacture. It helps producers to create and iterate their designs more quickly.

Besides general compatibility, we also consider the personal issue when synthesizing the complement item. Personalization is an important trend in fashion industry. Given the same query item, different persons may prefer different items to complement it. While personalized recommendation has been prevalent in areas like movie and music recommendation, most recommendations for fashion are still not user specific. Among the few works that explored personalization, Hu [10] showed that their personalized model is more capable of picking out outfits that suit users' taste than unpersonalized methods. Kang [13] presented a model to generate new item images of some category for a user. Although their recommendation was personalized, since no query item was provided in their setting, they did not consider the compatibility between items. Our method synthesizes new fashion items that go together with a given item according to the style preference of a user.

In this paper, we build our personalized fashion designer using the generative adversarial training framework. Generative Adversarial Networks (GANs) [3] have achieved great success in synthesizing realistic images for different applications. Here we apply it for personalized fashion design. We first use an encoder network to map the query image into a latent vector representation. This latent representation, together with another vector which characterizes user's style preference, are taken as the input to the generator network, which generates the target item image. Two discriminator networks are build to guide the generation process. One is the classic discriminator which learns to classify real and fake images. The other is a matching network which models the compatibility between fashion items. This network also learns users' preference representations, which contribute to both compatibility measure and item generation.

We evaluate the performance of our method on thousands of outfits created by online users. We show that modeling users' style preferences and using the compatibility discriminator are important for producing good designs. The images generated by our model are realistic and full of details. They have better visual quality and higher matching degree than those generated by the baseline methods.

2. Related Work

Many studies have been conducted on fashion related vision problems. Exemplary research includes clothing parsing [17, 40], clothing recognition [19, 38], fashion retrieval [18, 42], fashion trend prediction [1, 6], etc.

Compatibility and recommendation Prior works explore various ways to model the compatibility between fashion items [4, 9, 32, 36, 37]. Veit et al. [37] used a Siamese CNN architecture to learn compatibility between co-purchased items. Han et al. [4] trained a bidirectional LSTM model to sequentially predict the next item conditioned on previous ones. Vasileva et al. [36] proposed to learn type-aware embedding for compatibility prediction. All these prior works focused on modeling the compatibility between existing fashion items. Our method, on the other hand, synthesizes garment photos that complement the given items. We use compatibility to guide the generating process. The work most similar to us is [32], which introduced a projected compatibility distance to measure compatibility and a metric-regularized conditional GAN to visualize the learned compatible prototypes. However, they did not consider the personalization issue as our model.

Fashion synthesis Due to the high demand for real-life applications, fashion synthesis has gained increasing popularity recently. Given an image of a person and a piece of texture description, Zhu et al. [43] proposed a two-stage GAN approach for generating new clothing on the person. Han et al. [5] presented an image-based virtual try-on network to transfer a clothing item onto a person. Yoo [41] generated product photos of clothings from images of dressed persons. In [29], a SwapNet that interchanged garments between images of two people was presented. There were also many interests in synthesizing images of people in arbitrary poses while keeping their clothing unchanged [22, 28, 33]. In most of these synthesis, the fashion items were kept the same. Only their appearances or modalities were changed after the translation. In our case, the output is different from the input even in categories. They are linked through the underlying coordinating relationships.

Conditional GANs GANs have been vigorously studied in the conditional setting, which learn a conditional generative model of data. Previous works have conditioned GANs on discrete labels [26], text [30], as well as images [2, 15, 20, 24]. Isola et al. [11] proposed a generic solution for different image-to-image translation problems. Promising results were obtained for a variety of mappings like labels to street scene, edges to photo, etc. Kim et al. [14] designed a model to learn bidirectional mapping between two unpaired image domains. Note that the semantic structure of the images were mostly kept the same during the translations in these works. Our work differs in that the semantic layouts of the images generated are different from those of their corresponding inputs.

3. Our Approach

The task of personalized fashion design is to produce a fashion item for a specific user given an input query item. There are two general requirements for the design: (1) realness requirement, which means that the designed item should look realistic; (2) compatibility requirement, i.e. the

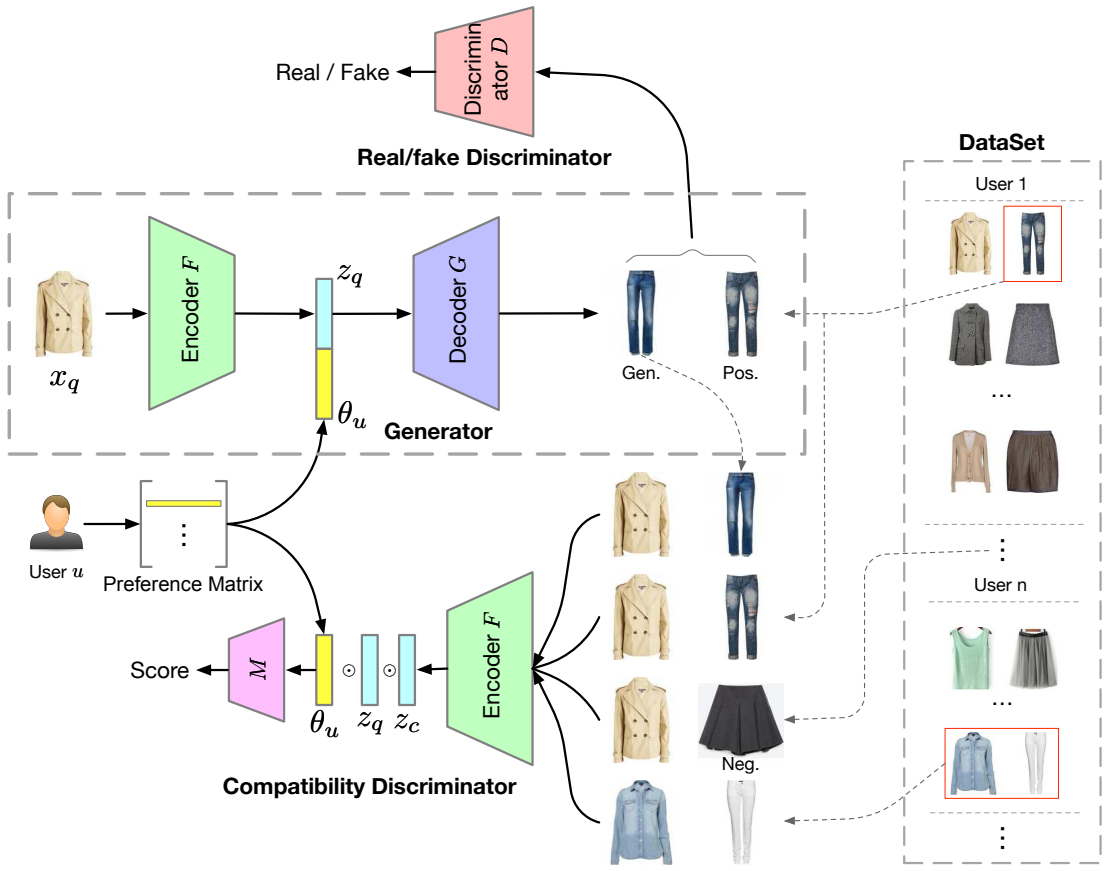

Figure 2. Network architecture for personalized fashion design. It contains one generator and two discriminators. The generator uses an encoder-decoder architecture. One of the discriminators is for real/fake supervision. And the other one is for compatibility prediction.

designed item should be compatible with the query item.

As shown in Figure 2, we use an encoder-decoder architecture to synthesize the complementary item. The encoder F progressively downsamples the query image until it is compressed into a low dimensional latent space representation $\boldsymbol{z}_q$. The vector $\boldsymbol{z}_q$ captures the semantic attributes, e.g., category, color, style, of the query item, and serves as the basis for designing the target item. The encoder F used in this paper is similar to the VGG16 network [34] except for the last three fully connected layers, which have 1024, 512 and 64 channels respectively.

To achieve personalized design, one method is to take user's identification information as a discrete label to the generator, just as many work do for conditional GANs. The discrete labels, however, are not capable enough to describe users' style preferences. We therefore use a vector $\boldsymbol{\theta}_u$, which is learned from historical data, to represent user u. The latent vector $\boldsymbol{z}_q$ and the user vector $\boldsymbol{\theta}_u$ are concatenated and then fed into the decoder G to generate the image of the complementary item. The architecture of the decoder G is illustrated in Figure 3. It is composed of several deconvolution layers, similar to the generator in [13].

To make sure that the designed item output by the decoder G meets the realness and compatibility requirements, we train the decoder G using the generative adversarial training framework with two discriminators, each of which serve one requirement. We discuss them in detail in the following two subsections.

3.1. Real/fake Discriminator

The real/fake discriminator is utilized to train the decoder G such that the designed item images look realistic. To overcome the problems of traditional GANs, [12] proposed to use a "relativistic discriminator" which estimates the probability that the given real data is more realistic than fake data. Following this work and combining the idea of LSGANs [23] , we use the following loss for the real/fake discriminator:

$$\begin{aligned}\mathcal{L}_D^{RaLSGAN} =& \frac{1}{2}\mathbb{E}_{\boldsymbol{x}_r\sim\mathbb{P}}\left[(D(\boldsymbol{x}_r)-\mathbb{E}_{\boldsymbol{x}_f\sim\mathbb{Q}}D(\boldsymbol{x}_f)-1)^2\right] \\ &+\frac{1}{2}\mathbb{E}_{\boldsymbol{x}_f\sim\mathbb{Q}}\left[(D(\boldsymbol{x}_f)-\mathbb{E}_{\boldsymbol{x}_r\sim\mathbb{P}}D(\boldsymbol{x}_r)+1)^2\right],\end{aligned} \tag{1}$$

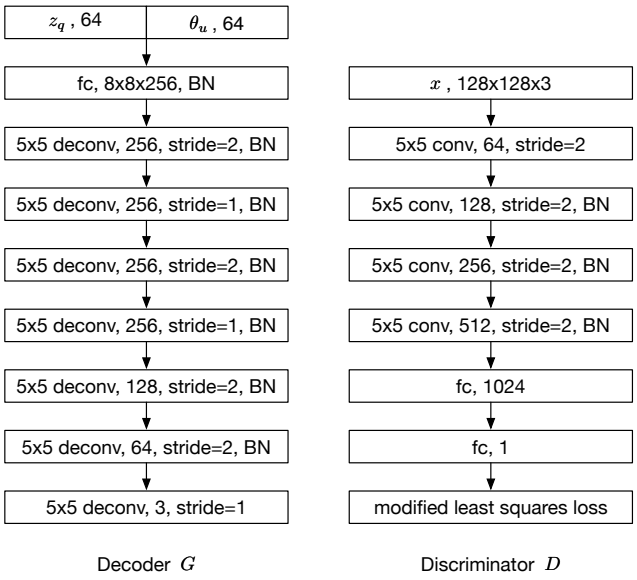

Figure 3. Network structures of the decoder G and the real/fake discriminator D. We use LeakyReLU activation with slope 0.2 in both networks for all hidden layers, and use Tanh in the decoder G for the output layer.

The discriminator distinguishes real and fake data by keeping a margin between them. The generator is trained to eliminate this gap by minimizing the following loss:

$$\begin{aligned}\mathcal{L}_G^{RaLSGAN} =& \frac{1}{2}\mathbb{E}_{\boldsymbol{x}_f\sim\mathbb{Q}}\left[(D(\boldsymbol{x}_f)-\mathbb{E}_{\boldsymbol{x}_r\sim\mathbb{P}}D(\boldsymbol{x}_r))^2\right] \\ &+\frac{1}{2}\mathbb{E}_{\boldsymbol{x}_r\sim\mathbb{P}}\left[(D(\boldsymbol{x}_r)-\mathbb{E}_{\boldsymbol{x}_f\sim\mathbb{Q}}D(\boldsymbol{x}_f))^2\right].\end{aligned} \tag{2}$$

3.2. Compatibility Discriminator

The compatibility discriminator is utilized to model users' style preferences and guide the training of the generator so that the item designed is compatible to the query. This discriminator consists of two parts. The first is a Siamese network [37] that takes a pair of fashion images, i.e., the query image $\boldsymbol{x}_q$ and the complementary item image $\boldsymbol{x}_c$, as input. Each image is passed through an encoder F that transfers it into a latent representation:

$$\boldsymbol{z}_i = F(\boldsymbol{x}_i), i \in \{q, c\}. \tag{3}$$

Note that this encoder shares parameters with the encoder in the generator. In the second part, we link the two latent representations to get a score that reflects how well the two items are compatible. We first take element-wise product of $\boldsymbol{z}_q$ and $\boldsymbol{z}_c$ to get a latent space representation of the outfit:

$$\boldsymbol{z}_o = \boldsymbol{z}_q \odot \boldsymbol{z}_c. \tag{4}$$

To take personalization into consideration, for each user u, we use a vector $\boldsymbol{\theta}_u$ to characterize his/her fashion preference. $\boldsymbol{\theta}_u$ is part of the network parameters and is learned during training. $\boldsymbol{\theta}_u$ is combined with $\boldsymbol{z}_o$ also through element-wise product, which works better than vector concatenation in our initial experiments. The result is fed into a metric network M, which consists of fully-connected layers, to get the final compatibility score, i.e.,

$$s_{u,o} = M(\boldsymbol{\theta}_u \odot \boldsymbol{z}_o). \tag{5}$$

To train the compatibility discriminator, we split our dataset into positive set $\mathcal{O}^+$ and negative set $\mathcal{O}^-$

$$\mathcal{O}^+ = \{o^+ | o^+ \rightarrow (\boldsymbol{x}_q^+, \boldsymbol{x}_c^+ | u)\}, \tag{6}$$

$$\mathcal{O}^- = \{o^- | o^- \rightarrow (\boldsymbol{x}_q^+, \boldsymbol{x}_c^- | u) \; or \; (\boldsymbol{x}_q^-, \boldsymbol{x}_c^- | u)\}, \tag{7}$$

where o^+ is an outfit that user u crafts online. We take it as positive outfit for u; o^- is a negative outfit, which is formed by a query item $\boldsymbol{x}_q^+$ and a random item $\boldsymbol{x}_c^-$ from the complementary category, or it is an outfit $\{\boldsymbol{x}_q^-, \boldsymbol{x}_c^-\}$ created by users other than u. The negative outfit $\{\boldsymbol{x}_q^+, \boldsymbol{x}_c^- | u\}$ reflects the general 'incompatibility' between the query $\boldsymbol{x}_q^+$ and the random item $\boldsymbol{x}_c^-$, while the negative outfit $\{\boldsymbol{x}_q^-, \boldsymbol{x}_c^- | u\}$ depicts the mismatch between the outfit $\{\boldsymbol{x}_q^-, \boldsymbol{x}_c^-\}$ and the user u.

The designed item images can also compose a new dataset $\mathcal{O}^*$ with the input query images:

$$\mathcal{O}^* = \{o^* | o^* \rightarrow (\boldsymbol{x}_q^+, \boldsymbol{x}_c^* | u)\}, \tag{8}$$

where o^* is the output outfit of our system, which contains the query $\boldsymbol{x}_q^+$ and the designed item $\boldsymbol{x}_c^*$.

The compatibility discriminator should be able to distinguish positive outfits from negative outfits, i.e., assign higher compatibility scores to positive outfits:

$$s_{u,o^+} > s_{u,o^-}. \tag{9}$$

To achieve this, the encoder F and the metric network M should seek to reduce the loss

$$\mathcal{L}_{FM} = -\mathbb{E}_{\substack{o^+\sim\mathcal{O}^+, \\ o^-\sim\mathcal{O}^-}}[\ln\sigma(s_{u,o^+} - s_{u,o^-})] + \lambda_{\theta_{FM}}\|\theta_{FM}\|^2, \tag{10}$$

where $\sigma(\cdot)$ is the sigmoid function, θ_{FM} includes the parameters in the encoder F and the metric network M, and $\lambda_{\theta_{FM}}$ is a regularization parameter. $\boldsymbol{\theta}_u$ is also learned during this process.

To make sure that the designed item fits well with the query item and the outfit they make satisfy the user's preference, we let o^* get similar compatibility score as the positive outfits o^+. This is achieved by optimizing the parameters of the decoder G so that the following loss is minimized

$$\begin{aligned}\mathcal{L}_G^{FM} =& \frac{1}{2}\mathbb{E}_{o^+\sim\mathcal{O}^+}[(s_{u,o^+} - \mathbb{E}_{o^*\sim\mathcal{O}^*}(s_{u,o^*}))^2] \\ &+\frac{1}{2}\mathbb{E}_{o^*\sim\mathcal{O}^*}[(s_{u,o^*} - \mathbb{E}_{o^+\sim\mathcal{O}^+}(s_{u,o^+}))^2].\end{aligned} \tag{11}$$

3.3. Adversarial Training

The overall objective of our approach is to minimize the following loss function

$$\mathcal{L} = \mathcal{L}_{FM} + \lambda_1 \mathcal{L}_G^{FM} + \lambda_2 \mathcal{L}_D^{RaLSGAN} + \lambda_3 \mathcal{L}_G^{RaLSGAN}, \tag{12}$$

where $\mathcal{L}_{FM}$ is related to the encoder F, the metric network M and the user preference vectors $\boldsymbol{\theta}_u$. $\mathcal{L}_D^{RaLSGAN}$ is only related to the real/fake discriminator D. Both $\mathcal{L}_G^{FM}$ and $\mathcal{L}_G^{RaLSGAN}$ are related to the decoder G. $\lambda_1, \lambda_2, \lambda_3$ are the model tradeoff parameters. All these losses are complementary to each other, and ultimately enable our algorithm to obtain pleasant results.

Given a batch of real images from the training set, we first train the compatibility discriminator by reducing the loss of Eq.(10). The real/fake discriminator is then trained to reduce the loss of Eq.(1). After that, we keep the discriminator parameters fixed, and optimize the parameters of the decoder G by reducing the loss in Eq.(2) and Eq.(11). The whole training procedure is summarized in Algorithm 1.

Algorithm 1 Adversarial training algorithm for personalized fashion design

Set: The number of iterations for the D network $n_D = 2$, the batch size $m = 64$, $\lambda_{\theta_{FM}} = 10^{-6}$.
Initialize: Initialize the network parameters, i.e. θ_{FM} for F and M, θ_G for G, θ_D for D, and the user preference vectors $\boldsymbol{\theta}_u$.

while θ_G has not converged **do**
  Sample a batch of $o^+ = \{\boldsymbol{x_q^+}, \boldsymbol{x_c^+}|u\}$ from the positive set
  Sample a batch of $o^* = \{\boldsymbol{x_q^+}, \boldsymbol{x_c^*}|u\}$ from the designed set
  Sample a batch of $o^- = \{\boldsymbol{x_q^+}, \boldsymbol{x_c^-}|u\}$ or $o^- = \{\boldsymbol{x_q^-}, \boldsymbol{x_c^-}|u\}$ from the negative set
  Update θ_{FM} with
    $\theta_{FM} \leftarrow \theta_{FM} - \eta \nabla_{\theta_{FM}} \mathcal{L}_{FM}$
  for $t = 1, \ldots, n_D$ **do**
    Update θ_D with
      $\theta_D \leftarrow \theta_D - \lambda_2 \eta \nabla_{\theta_D} \mathcal{L}_D^{RaLSGAN}$
  end for
  Update θ_G with
    $\theta_G \leftarrow \theta_G - \eta \nabla_{\theta_G} (\lambda_1 \mathcal{L}_G^{FM} + \lambda_3 \mathcal{L}_G^{RaLSGAN})$
end while

4. Experiments

In this section, we conduct experiments to evaluate the proposed method. We compare it with several state-of-the-art methods quantitatively and also through their visual quality performance. We implement our method using TensorFlow and all experiments are run on a commodity workstation with a single GTX-1080 graphics card.

Our dataset is crawled from the community-powered fashion website Polyvore. In total, we collected 208,814 outfits crafted by 797 online users. For each user, 221 and 41 outfits are used for training and testing respectively. Each outfit consists of two items, i.e., a top and a bottom. We test on two tasks: given a top, designing a bottom item to go with it; and given a bottom, designing a top item for it. Some statistics of our dataset are given in Table 1.

	Users	Top	Bottom	Outfits
Training	797	102,217	76,245	176,137
Testing	797	26,899	23,642	32,677

Table 1. Statistics of our dataset.

We use Adam optimizer and the learning rate is 0.0002. The model tradeoff parameters are set to $\lambda_1 = \lambda_2 = \lambda_3 = 1$. We set the batch size to 64 and train the model for 25 epochs, 2750 iterations per step.

4.1. Baselines

To validate the effectiveness of our method, we compare it with the following methods. The first two are general frameworks for image-to-image translation problems and the last two are designed specific for fashion problems.

Pix2pix [11] A U-Net architecture is used for the generator and a single discriminator is learned to classify real and fake tuples ({query, complementary item}).

DiscoGAN [14] It is a state-of-the-art method for discovering relations between two domains. It consists of two generators for bilateral cross-domain generation and two discriminators with one for each domain respectively.

Pixel-level transfer [41] It uses two discriminators to guide the domain transfer. The source and target images are concatenated along the channels when input to the domain-discriminator.

MrCGAN [32] It first obtains a compatible prototype using a pre-learned projection function, then the prototype is used to generate images of compatible items.

4.2. Ablation Study

Compared with previous methods, the main differences of our model are: (1) We learn users' style preferences from historical data and incorporate them into the image generation process; (2) we design a compatibility discriminator to ensure that the generated item fit well with the query. Therefore, we conduct the following ablation studies to evaluate the effect of these important components of our model.

Discrete user label As many vanilla conditional GANs do,

	Methods	IS ↑	Opposite SSIM ↑	FID ↓
Baselines	Real	4.6372±.05	0.5500±.15	0.0000
	Pix2pix [11]	3.0793±.04	0.4077±.12	43.5215
	DiscoGAN [14]	4.3430±.05	0.5717±.14	51.8099
	Pixel-level transfer [41]	3.8569±.03	0.5735±.16	57.0461
	MrCGAN [32]	3.9078±.06	0.5576±.13	26.3591
Ablation Study	Ours (discrete user label)	3.9137±.04	0.5207±.14	45.4453
	Ours (remove $\boldsymbol{\theta}_u$)	4.2375±.05	0.5447±.14	31.3604
	Ours (remove $\mathcal{L}_G^{FM}$)	4.3157±.06	0.5499±.07	22.3429
	Ours (full)	4.2626±.05	0.5644±.13	18.1023

Table 2. Quantitative evaluation of the generated images by different methods. The values after ± are the standard deviations.

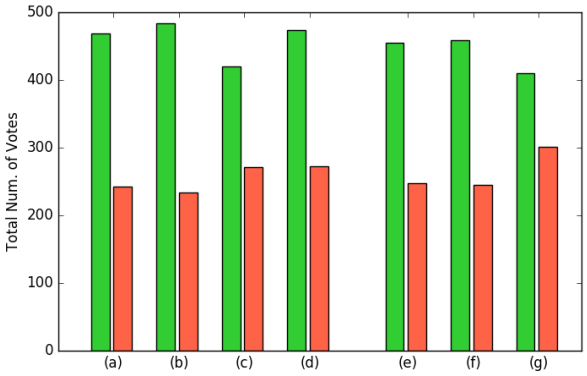

Figure 4. Survey results of the user study. The green bars indicate votes got by our full model, and the orange bars are votes of other methods. Panel (a, b, c, d) correspond to Pix2pix [11] , DiscoGAN [14], Pixel-level transfer [41] and MrCGAN [32] respectively. Panel (e, f, g) are different versions of our model, i.e. discrete user label, remove $\boldsymbol{\theta}_u$ and remove $\mathcal{L}_G^{FM}$.

instead of learning and using $\boldsymbol{\theta}_u$, we use a discrete label to represent the users for the conditional generation.

User unaware design We only model the general compatibility but unaware of the preferences of different users. In this case, we omit $\boldsymbol{\theta}_u$ in the network. The input of the decoder G is only the latent vector $\boldsymbol{z}_q$ and the compatibility score $s_o = M(\boldsymbol{z}_q \odot \boldsymbol{z}_c)$.

Remove $\mathcal{L}_G^{FM}$ The compatibility discriminator guarantees that the items designed by the generator fit well with the queries. To analyze the importance of this component, we remove $\mathcal{L}_G^{FM}$ while training the decoder G. Note that the user preference vector $\boldsymbol{\theta}_u$ is kept and still fed into the decoder as a condition.

4.3. Quantitative Evaluation

The designed item images are evaluated from the perspective of realness, diversity and compatibility. To measure the realness, the Inception Score [31] based on a standard pre-trained inception network is utilized. The higher the score the better the quality. For diversity, similar to [27], we calculate the visual similarity of pairs of generated images, which is measured by the structural similarity (SSIM) [39]. Following [13], we report the Opposite Mean SSIM, which is one minus the mean SSIM, to show diversity. A higher value means better diversity. Furthermore, we also compute the Fréchet Inception Distance (FID) [8] between the sets of generated images and the ground truth images. The smaller the value, the closer the two image distributions are.

For compatibility, we conduct user surveys to see whether our model could produce images that are perceived as compatible to the query item. 20 subjects are involved in the study. Each is assigned 300 randomly selected queries (150 tops and 150 bottoms). We make pairwise comparison between our method and the baselines. For each query image, two complementary item images, one generated by our full model and the other by a randomly selected baseline method, are given. The users are asked to select the item that is more compatible with the query. The total number of votes got by each method are computed. We also provide pairwise comparison between our full model and the models for ablation study.

The evaluation results are shown in Table 2 and Figure 4. We can see that the proposed method performs better than most other methods in IS and Opposite SSIM. As for FID, our method outperforms other methods with a large margin. Although DiscoGAN performs a little better in IS and Opposite SSIM. Its FID value is much worse than ours. It is also less preferable to our method in the user study. According to the ablation study, our full model performs much

Figure 5. Qualitative comparison of different methods. Results of the top-to-bottom task are shown on the left and results of the bottom-to-top task are shown on the right. For each task, the first row are the query images. From the 2nd to the 5th rows are the results of pix2pix [11], DiscoGAN [14], pixel-level transfer [41] and MrCGAN [32] respectively. From the 6th to the 8th rows are results of our part models. The last two rows are the items generated by our full model and the ground truth items selected by online users.

better than the alternative methods. Modeling users' preference properly and using it to guide item design help the generator better estimate the data distribution and improve the results a lot. Using discrete labels as the user condition is not capable enough to capture users' style preferences and it performs even worse than the unpersonalized model. By using the compatibility discriminator, the matching degree between the generated items and the queries is improved. Therefore, both the personalized modeling and the compatibility module are important for enhancing the performance.

4.4. Qualitative Evaluation

We also evaluate the visual quality of the designed items. In Figure 5, we show some results of the two design tasks: (a) top to bottom and (b) bottom to top. We can see that compared with the baseline methods and our abbreviated models, the items designed by our full model look more realistic. They are good shaped and full of texture details. They have fewer visual artifacts than the other methods and are also more similar with the corresponding ground truth items. The results demonstrate that our method is capable

Figure 6. Generated images and their nearest neighbors in the real image dataset.

Figure 7. Designed items for different users given the same query.

of generating realistic and compatible items.

In Figure 6, we illustrate the 5 nearest neighbors of some designed items in the real image dataset. From the figures, we find that the designed items are very similar with the real items, i.e., almost undistinguishable from the real ones. For applications such as online shopping, where real items are needed, we can use this design-and-retrieval method to efficiently identify the targets.

Our framework can model users' preferences on fashion styles and make the designs personalized. Therefore for the same query, the designs for different users are different. This is validated in Figure 7. It shows that quite different items are generated for different users. This is very desirable in practice.

4.5. Learning preferences for new users

When new users join after the model has been trained, it would be un-affordable to retrain the whole model. With our framework, we can keep all other parameters of the network fixed and only learn $\boldsymbol{\theta}_u$ for the newcomers. Note that $\boldsymbol{\theta}_u$ is only a 64 dimensional vector, which can be computed efficiently with all other parameters fixed.

We have tested the performance with this setting. We first use outfits from 700 users to train the whole network. For the remaining 97 users, we learn their $\boldsymbol{\theta}_u$ without updating other network parameters. We compare the images generated for these 97 users under this setting with our previous setting where all 797 users were trained together. The quantitative comparison is showed in (Table 3). In general, we find that the qualities of the images generated under these two settings are similar.

	IS ↑	Opposite SSIM ↑	FID ↓
Original setting	4.2899±.18	0.5922±.10	19.4098
New setting	4.1198±.13	0.5672±.11	21.6177

Table 3. Evaluation of the images generated for the 97 new users.

5. Conclusion

In this paper, we have proposed a personalized fashion design framework with the help of generative adversarial training. Our framework can automatically model user's fashion taste and design a fashion item that is compatible to a given query item. It is composed of an encoder that maps the query image into a latent vector representation, a decoder that generates the target item image, and two discriminators which guide the generation process. Experiments on thousands of outfits crafted by online users show that the proposed method outperforms alternative methods in terms of capturing users' personal tastes, modeling the compatibility between items, and the visual quality of the designed items.

References

[1] Ziad Al-Halah, Rainer Stiefelhagen, and Kristen Grauman. Fashion forward: Forecasting visual style in fashion. In *ICCV*, 2017.

[2] Yang Chen, Yu-Kun Lai, and Yong-Jin Liu. CartoonGAN: Generative adversarial networks for photo cartoonization. In *CVPR*, 2018.

[3] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[4] Xintong Han, Zuxuan Wu, Yu-Gang Jiang, and Larry S. Davis. Learning fashion compatibility with bidirectional LSTMs. In *ACM Multimedia*, 2017.

[5] Xintong Han, Zuxuan Wu, Zhe Wu, Ruichi Yu, and Larry S. Davis. VITON: An image-based virtual try-on network. In *CVPR*, 2018.

[6] Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *WWW*, 2016.

[7] Ruining He, Charles Packer, and Julian McAuley. Learning compatibility across categories for heterogeneous item recommendation. In *ICDM*, 2016.

[8] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a Nash equilibrium. In *NIPS*, 2017.

[9] Wei-Lin Hsiao and Kristen Grauman. Creating capsule wardrobes from fashion images. In *CVPR*, 2018.

[10] Yang Hu, Xi Yi, and Larry S. Davis. Collaborative fashion recommendation: a functional tensor factorization approach. In *ACM Multimedia*, 2015.

[11] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

[12] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard GAN. In *arXiv preprint arXiv:1807.00734*, 2018.

[13] Wang-Cheng Kang, Chen Fang, Zhaowen Wang, and Julian McAuley. Visually-aware fashion recommendation and design with generative image models. In *ICDM*, 2017.

[14] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *ICML*, 2017.

[15] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017.

[16] Yuncheng Li, LiangLiang Cao, Jiang Zhu, and Jiebo Luo. Mining fashion outfit composition using an end-to-end deep learning approach on set data. *IEEE Trans. Multimedia*, 2017.

[17] Xiaodan Liang, Chunyan Xu, Xiaohui Shen, Jianchao Yang, Jinhui Tang, Liang Lin, and Shuicheng Yan. Human parsing with contextualized convolutional neural network. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017.

[18] Si Liu, Zheng Song, Guangcan Liu, Changsheng Xu, Hanqing Lu, and Shuicheng Yan. Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In *CVPR*, 2012.

[19] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *CVPR*, 2016.

[20] Pauline Luc, Camille Couprie, Soumith Chintala, and Jakob Verbeek. Semantic segmentation using adversarial networks. In *arXiv preprint arXiv:1611.08408*, 2016.

[21] Jose Oramas M. and Tinne Tuytelaars. Modeling visual compatibility through hierarchical mid-level elements. In *arXiv preprint arXiv:1604.00036*, 2016.

[22] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *NIPS*, 2017.

[23] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, and Zhen Wang. Multi-class generative adversarial networks with the L2 loss function. In *ICCV*, 2017.

[24] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. In *ICLR*, 2016.

[25] Julian McAuley, Christopher Targett, Javen Shi, and Anton van den Hengel. Image-based recommendations on styles and substitutes. In *SIGIR*, 2015.

[26] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. In *arXiv preprint arXiv:1411.1784*, 2014.

[27] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier GANs. In *ICML*, 2017.

[28] Albert Pumarola, Antonio Agudo, Alberto Sanfeliu, and Francesc Moreno-Noguer. Unsupervised person image synthesis in arbitrary poses. In *CVPR*, 2018.

[29] Amit Raj, Patsorn Sangkloy, Huiwen Chang, James Hays, Duygu Ceylan, and Jingwan Lu. SwapNet: Image based garment transfer. In *ECCV*, 2018.

[30] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *ICML*, 2016.

[31] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *NIPS*, 2016.

[32] Yong-Siang Shih, Kai-Yueh Chang, Hsuan-Tien Lin, and Min Sun. Compatibility family learning for item recommendation and generation. In *AAAI*, 2018.

[33] Aliaksandr Siarohin, Enver Sangineto, Stephane Lathuiliere, and Nicu Sebe. Deformable GANs for pose-based human image generation. In *CVPR*, 2018.

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[35] Xuemeng Song, Fuli Feng, Jinhuan Liu, Zekun Li, Liqiang Nie, and Jun Ma. NeuroStylist: Neural compatibility modeling for clothing. In *ACM Multimedia*, 2017.

[36] Mariya I. Vasileva, Bryan A. Plummer, Krishna Dusad, Shreya Rajpal, Ranjitha Kumar, and David Forsyth. Learning type-aware embeddings for fashion compatibility. In *ECCV*, 2018.

[37] Andreas Veit, Balazs Kovacs, Sean Bell, Julian McAuley, Kavita Bala, and Serge Belongie. Learning visual clothing style with heterogeneous dyadic co-occurrences. In *ICCV*, 2015.

[38] Wenguan Wang, Yuanlu Xu, Jianbing Shen, and Song-Chun Zhu. Attentive fashion grammar network for fashion landmark detection and clothing category classification. In *CVPR*, 2018.

[39] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.*, 2004.

[40] Kota Yamaguchi, M. Hadi Kiapour, and Tamara L. Berg. Paper doll parsing: Retrieving similar styles to parse clothing items. In *ICCV*, 2013.

[41] Donggeun Yoo, Namil Kim, Sunggyun Park, Anthony S. Paek, and In So Kweon. Pixel-level domain transfer. In *ECCV*, 2016.

[42] Bo Zhao, Jiashi Feng, Xiao Wu, and Shuicheng Yan. Memory-augmented attribute manipulation networks for interactive fashion search. In *CVPR*, 2017.

[43] Shizhan Zhu, Sanja Fidler, Raquel Urtasun, Dahua Lin, and Chen Change Loy. Be your own Prada: Fashion synthesis with structural coherence. In *ICCV*, 2017.