

Learning to Synthesize Compatible Fashion Items Using Semantic Alignment and Collocation Classification: An Outfit Generation Framework

Dongliang Zhou, Haijun Zhang, Kai Yang, Linlin Liu, Han Yan, Xiaofei Xu, Zhao Zhang, and Shuicheng Yan

Abstract—The field of fashion compatibility learning has attracted great attention from both the academic and industrial communities in recent years. Many studies have been carried out for fashion compatibility prediction, collocated outfit recommendation, artificial intelligence (AI)-enabled compatible fashion design, and related topics. In particular, AI-enabled compatible fashion design can be used to synthesize compatible fashion items or outfits in order to improve the design experience for designers or the efficacy of recommendations for customers. However, previous generative models for collocated fashion synthesis have generally focused on the image-to-image translation between fashion items of upper and lower clothing. In this paper, we propose a novel outfit generation framework, i.e., *OutfitGAN*, with the aim of synthesizing a set of complementary items to compose an entire outfit, given one extant fashion item and reference masks of target synthesized items. *OutfitGAN* includes a semantic alignment module, which is responsible for characterizing the mapping correspondence between the existing fashion items and the synthesized ones, to improve the quality of the synthesized images, and a collocation classification module, which is used to improve the compatibility of a synthesized outfit. In order to evaluate the performance of our proposed models, we built a large-scale dataset consisting of 20,000 fashion outfits. Extensive experimental results on this dataset show that our *OutfitGAN* can synthesize photo-realistic outfits and outperform state-of-the-art methods in terms of similarity, authenticity and compatibility measurements.

Index Terms—Fashion compatibility learning, fashion synthesis, generative adversarial network, image-to-image translation, outfit generation.

I. INTRODUCTION

IN recent years, the fashion sector has undergone a proliferation in economic terms. According to a business report¹ from Statista.com, the economy is expected to maintain an approximate annual growth rate of 7.2% in the future. A Mck-

This work was supported in part by the National Natural Science Foundation of China under Grant no. 61972112 and no. 61832004, the Guangdong Basic and Applied Basic Research Foundation under Grant no. 2021B1515020088, the Shenzhen Science and Technology Program under Grant no. JCYJ20210324131203009, and the HITSZ-J&A Joint Laboratory of Digital Design and Intelligent Fabrication under Grant no. HITSZ-J&A-2021A01.

D. Zhou, H. Zhang, K. Yang, L. Liu, H. Yan and X. Xu are with the Department of Computer Science, Harbin Institute of Technology, Shenzhen, 518055 China; Z. Zhang is with the Department of Computer Science, Hefei University of Technology, Hefei, 230009 China; S. Yan is with Sea AI Lab (SAIL), 119077 Singapore. Corresponding author: Haijun Zhang, e-mail: hjzhang@hit.edu.cn.

¹<https://www.statista.com/outlook/244/100/fashion/worldwide>

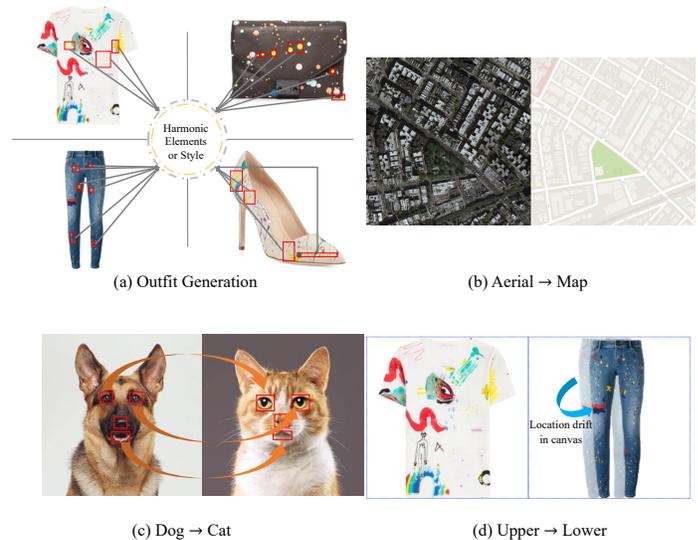


Fig. 1: General image-to-image translation and fashion outfit generation.

insey.com business report² recommends that fashion sellers should pay more attention to cutting-edge techniques, as these offer lucrative opportunities. The key to improving revenue for sellers lies in fashion designers creating more attractive fashion items or outfits for customers. In a traditional design process, however, designers rely on their own creative senses, which may involve subjectivity and uncertainty. With the advent of artificial intelligence (AI) and the era of big data, AI-enabled fashion design has become possible. Fashion designers can create preliminary designs more effectively by relying on machine learning based on numerous extant collocated outfits shared by social media users. The rules of compatibility hidden in these collocated outfits can be learned by a machine learning model to produce new fashion items. In particular, generative adversarial networks (GANs) [1] can assist fashion designers in synthesizing visually plausible images of new fashion items based on extant fashion items. This can be approached as a direct image-to-image translation task in which GANs are fed with pair-wise image data containing extant fashion items and corresponding compatible items during training. The use of GAN-based models has been widely explored in the field of

²<https://www.mckinsey.com/industries/retail/our-insights/the-state-of-fashion-2020-navigating-uncertainty>

image synthesis. For general image-to-image translation, Isola *et al.* [2] proposed a Pix2Pix framework for the synthesis of new images based on extant ones, by adopting a GAN loss and an L1 loss to improve the photo-realism of images with high authenticity. In a later study, Wang *et al.* [3] improved the Pix2Pix framework in a coarse-to-fine manner. In addition to ground-truth images, several unsupervised image-to-image translation methods such as CycleGAN [4], MUNIT [5], DRIT++ [6] and StarGAN-v2 [7] were explored for image-to-image translation using images from two arbitrary domains, without the explicit use of one-to-one correspondent mapping. In the field of fashion synthesis, Liu *et al.* [8] first proposed an Attribute-GAN model, which addressed the task of translation between upper clothing and lower clothing items by considering the latent compatibility between them. They added clothing attributes to guide the process of image generation. In a later study, they [9] extended their collocation discriminator and attribute discriminator framework to form a multi-discriminator framework. Yu *et al.* [10] used the user’s personal preferences to improve the generation quality of clothing images for personalized recommendation. All of the aforementioned works in fashion synthesis focused on image-to-image translation between upper and lower clothing, and rarely considered the generation of a whole outfit.

By taking advantage of the power of GANs, the objective of this research is to explore the issue of how to synthesize an entire compatible outfit, based on extant fashion items with certain alternative reference information, e.g., outline mask information of target fashion items. To accomplish this, we consider a specific scenario: given a particular fashion item, a user may expect to have other collocated fashion items with desirable reference masks in his or her mind, containing outline information on compatible items. Our model aims to generate compatible fashion items to compose an outfit, conditioned on a particular fashion item and the reference masks of other items in the same outfit. The aim of outfit generation is thus to translate harmonic elements or styles from extant fashion garments to synthesized compatible items. As shown in Fig. 1(a), each fashion item in an outfit shares harmonious elements or styles with the other items, to maintain the compatibility. For example, there may be many patches of yellow or elements of the same color co-occurring in an outfit. Meanwhile, in the task of outfit generation, each type of fashion item in an outfit has its own unique outline and style. More specifically, compared with general image-to-image translation methods, our research addresses this problem from the following three perspectives. (i) As shown in Fig. 1(b), traditional supervised image-to-image translation methods such as Pix2Pix [2] or Pix2PixHD [3] need pixel-to-pixel correspondence between an input image and its output image, such as in aerial-to-map translation. For these translation tasks, researchers usually adopt convolutional neural network (CNN)-based generators, which can learn only local patch features of an input image rather than the global features [11]. However, outfit generation has no apparent pixel-to-pixel alignment between extant fashion items and synthesized ones. (ii) Unsupervised image-to-image translation methods such as CycleGAN [4] or MUNIT [5] learn a mapping function

between two domains. For example, as shown in Fig. 1(c), objects need a latent semantic alignment rather than a precise pixel-to-pixel correspondence, in the same way as eye or mouth mapping in dog-to-cat translation. In contrast, in the process of outfit generation, when an extant upper clothing item has a red flower on the top, the target shoes may have a corresponding element or style at the bottom rather than the top of the shoes. This suggests that the mapping relationship between extant fashion items and target items is non-local in space, and the model therefore needs to learn the global feature mapping relationship during training. (iii) In outfit generation, when the target images undergo minor changes in location or size, users may not observe these slight spatial changes. For example, for a given upper clothing image, the target lower clothing image may drift slightly on the canvas, as shown in Fig. 1(d). Although this minor change cannot be immediately observed, it is very important in terms of image-to-image translation, since general image-to-image translation methods with paired images usually supervise the generation process by adopting losses such as L1, L2 [2], and perceptual losses [12], which require a precise spatial alignment between the synthesized and target images.

To address the above issues, we propose a collocated fashion outfit generation framework called OutfitGAN, which adopts a semantic alignment module (SAM) and a collocation classification module (CCM) to guide the process of outfit generation. The SAM fuses a given fashion item with the reference masks of other compatible items, to align the extracted features from the extant fashion item based on the reference masks. Our usage of reference masks is largely inspired by [13] and [14], in which researchers used the key points of the human body to guide the fashion synthesis. Specifically, our OutfitGAN introduces an SAM to improve the quality of synthesized images and provide an explanation of the outfit generation process explicitly. The development of the SAM was motivated by the fact that items mostly convey corresponding areas or style from the same outfit, according to our observations from the collected outfits shown in Fig. 1(a), where we see that many compatible outfits contain elements or styles corresponding to other items in the same outfit. To ensure the compatibility of a synthesized outfit, we also develop a CCM based on bidirectional long short-term memory (Bi-LSTM) [15] to model the compatibility between items. In order to examine the performance of our proposed OutfitGAN, we constructed a large-scale dataset containing 20,000 outfits, each of which was composed of four types of item: upper clothing, bags, lower clothing, and shoes. The results of an extensive set of experiments demonstrate the effectiveness of our proposed framework with respect to various evaluation metrics, in comparison with several state-of-the-art methods. The main contributions of this research can be summarized as follows:

- To the best of our knowledge, this is the first work to synthesize fashion items based on extant ones in order to create a compatible outfit. The overall framework of our OutfitGAN includes an outfit generator, an outfit discriminator, and a CCM. The results of our experiments indicate

that our proposed framework is capable of synthesizing photo-realistic outfit images that are compatible with a given fashion item.

- We propose an SAM to improve the quality of the synthesized outfit and characterize the correspondence between a given fashion item and the synthesized items. This module uses two branches based on CNNs to extract the features of a given fashion item and the reference mask of a target item, respectively; a correspondence layer to calculate the spatial relationship matrix with respect to the features of the given fashion item and the reference mask; and an alignment layer to semantically align the features of the given item with the reference mask.
- We propose a CCM that uses Bi-LSTM to improve the compatibility of the synthesized outfit. This module is primarily applied to guide the compatibility of the synthesized fashion items. It uses a pre-trained CNN to extract the features of the fashion items and two directional LSTMs for compatibility guidance from different directions, from the perspective of human vision.

The remainder of this article is organized as follows. Section II briefly reviews works related to fashion outfit synthesis. Section III describes the overall framework of our OutfitGAN framework and the associated details of the implementation. In Section IV, we conduct extensive experiments to validate the performance of our model. Section V concludes the paper and suggests directions for future work.

II. RELATED WORK

This research falls into the field of fashion learning, which has a large existing body of literature. In this section, we review related works on image-to-image translation, fashion compatibility learning, and fashion synthesis. We also highlight the features of this research in comparison to those of prior works.

Image-to-Image Translation. This is an important task in computer vision. A model takes an image as input and learns a conditional distribution of the corresponding image with a mapping function. There are many applications for this task, such as image colorization [2], image style transfer [16], super-resolution [12], and virtual try-on [14], [17]. Numerous previous studies have suggested that GANs [1] are capable of producing realistic synthesized images via image-to-image translation. Existing GAN-based translation methods can be roughly divided into two categories: supervised and unsupervised approaches. Using a supervised method, Isola *et al.* [2] proposed a Pix2Pix translation framework to alleviate blurring in this task. Later, Wang *et al.* [3] introduced an improved Pix2Pix model with the aim of achieving more stable and realistic image generation in a coarse-to-fine manner. Using an unsupervised method, Zhu *et al.* [4] proposed a cycle consistency loss to handle a lack of paired images. Subsequently, Huang *et al.* [5] addressed the latent space of image samples using a composition of style and content code, and used two separate encoders to disentangle these components. Lee *et al.* [6] also disentangled the latent space into a shared content

space and an attribute space for each domain. In a later study, Choi *et al.* [7] extended the concepts of style code and content code, employing a multi-layer perceptron (MLP) to synthesize a diverse range of style codes and injecting them into a decoder to synthesize various images.

Fashion Compatibility Learning. With the increasing popularity of online stores, fashion recommendation is now playing an essential role in online retail. Fashion compatibility learning is an important aspect of fashion recommendation, and researchers have adopted metric learning to predict compatibility. Each fashion item in the same outfit is firstly embedded into a shared space, and the compatibility between items is then evaluated based on the distance between them. A shorter distance or a higher similarity indicates better compatibility, and vice versa. To measure the compatibility between items, McAuley *et al.* [18] proposed a method for comparing the distance between the features extracted by a pre-trained CNN. Veit *et al.* [19] then used a SiameseNet to extract visual features to compare the distance between items. These methods regarded the different types of fashion items as the same, and handled them in an embedding space. In order to keep different categories of fashion items with different mappings into embeddings, Vasileva *et al.* [20] tackled this problem by learning the similarity and compatibility simultaneously, in different spaces, for each pair of item categories. Another inspired idea was to regard the fashion items in the outfit as a sequence from the perspective of human vision. Han *et al.* [21] adopted Bi-LSTM to learn the compatibility of an outfit in the form of a sequence. The other mainstream idea that has emerged is the use of graph-based networks to address the issue of compatibility, and these methods have attracted the attention of several researchers. In particular, Cui *et al.* [22] and Li *et al.* [23] employed graph convolutional networks to model the compatibility problem. In this task, fashion compatibility is a crucially important perspective for generating an outfit. In our OutfitGAN, we use Bi-LSTM in our implementation of collocation classification in order to guide the compatibility of the generated items.

Fashion Synthesis. Due to the ever-increasing demand for fashion applications, fashion synthesis has started to become an important aspect of the field of computer vision [24]. Fashion synthesis includes virtual try-on, pose transformation and the synthesis of compatible fashion items. In the field of virtual try-on, Han *et al.* [14] employed a thin plate spline (TPS) and a GAN to synthesize new images, given images of the user's body and the target clothing. Subsequently, a new model called characteristic-preserving image-based virtual try-on network (CP-VTON) [25] was proposed, which included a geometric matching module that could improve the spatial deformation in comparison to TPS. Zhu *et al.* [13] proposed FashionGAN to synthesize clothes on a wearer while maintaining consistency with a text description. In addition to virtual try-on, pose transformation is also an important task in fashion synthesis. A model takes a reference image as input and a target pose based on the key points of the human body, and aims to synthesize a pose-guided image of the person while retaining the personal information of the reference image. A network called PG² [26] was the first to use a two-stage model to address the

problem. Later, Siarohin *et al.* [27] transformed the high-level features for each part of human body using a technique called deformable skipping. Recently, researchers have turned their attention to the generation of fashion items. In particular, Liu *et al.* [8] proposed a network for image-to-image translation between upper and lower clothing using an attribute-based GAN. They extended their model to a more general GAN framework with multiple discriminators by considering rich text descriptions of upper and lower clothing images [9]. Yu *et al.* [10] then exploited a matrix of the user’s personal preferences to improve the quality of image generation. Unlike the works in [8], [9] and [10], we concentrate in this paper on generating an outfit that consists of several compatible fashion items.

Features of Our Model: Several studies have focused on outfit generation using image-to-image translation [2], [3], [4], [5], [6], [7] and compatibility learning [18], [19], [20], [21], [22], [23] for fashion synthesis [8], [9], [10]. Initially, supervised [2], [3] or unsupervised image-to-image translation methods [4], [5], [6], [7] were used with CNN-based generators to carry out image translation from input images to output images, with or without supervised paired images. However, a CNN-based generator is only able to learn local neighborhood relationships, and is unable to learn the long-range dependences between the input and output images [11]. Our outfit generation scheme aims to translate harmonic elements and styles while maintaining their compatibility. In particular, our approach characterizes the long-range dependences between the extant fashion items and the synthesized ones. Unlike the general methods described above, our proposed model is capable of accomplishing cross-domain image translation, in which the images may have no pixel-wise alignment but do have a corresponding spatial alignment mapping for the long-range dependences between the input and output images. In particular, our proposed model uses an SAM which aligns the features of the extant fashion items to those of the target items. In contrast, existing fashion compatibility learning methods [18], [19], [20], [21], [22], [23] are used to predict outfit compatibility and give outfit recommendations for an extant fashion database with discriminative models, and rarely consider the synthesis of new compatible outfits based on extant fashion items. Our proposed model synthesizes compatible fashion items based on extant ones, using a generative model. Finally, although several fashion synthesis methods [8], [9], [10] have been used to synthesize complementary fashion items based on extant ones, these methods only carry out image translation between upper and lower clothing, and cannot synthesize an entire outfit. Our proposed model uses multiple generators to synthesize suitable fashion items for the generation of entire outfits. In addition, a CCM is proposed to supervise the compatibility of the synthesized outfit during the generation process.

III. OUTFITGAN FOR THE GENERATION OF MULTIPLE FASHION ITEMS

In this section, we first formulate our research problem and give the descriptions and definitions needed for outfit

generation. We then present the entire OutfitGAN framework. Finally, the implementation details of our proposed models are discussed.

A. Problem Formulation

In general, previous fashion compatibility learning methods [18], [19], [20], [21], [22], [23] have focused on discriminating the collocation given a set of fashion items. In contrast, generative models allow us to synthesize a entirely new outfit as well as maintaining the collocation for compatibility learning. In this work, we focus on synthesizing a set of compatible items based on a given fashion item in order to compose a complete outfit. Formally, let $\mathcal{O} = [\mathcal{O}_1, \dots, \mathcal{O}_i, \dots, \mathcal{O}_N]$ denote an outfit, where \mathcal{O}_i is the i -th fashion item in the outfit arranged in a fixed order based on its categories, i.e., from top to bottom according to perspective of human vision, e.g., [upper clothing, bag, lower clothing, shoes]. N represents the number of fashion items in an outfit. In addition, each fashion item $\mathcal{O}_i \in \mathcal{O}$ is associated with a mask that indicates the outline of \mathcal{O}_i . For each \mathcal{O}_i , let $Mask_i$ be the corresponding mask of \mathcal{O}_i . Our task is to synthesize a complementary outfit set $\tilde{\mathcal{O}}$ for a user based on a given fashion item \mathcal{O}_k and reference masks $[Mask_1, \dots, Mask_{k-1}, Mask_{k+1}, \dots, Mask_N]$, which represent the user’s rough idea of the outlines of the newly synthesized outfit items. Here, the reference masks may be given by the user, selected by the user from a candidate dataset containing various outlines of fashion items, or produced automatically by a pre-trained generative model.

B. OutfitGAN

Based on the problem formulation presented above, we design a new generative framework called OutfitGAN to accomplish the task of outfit generation. The detailed structure of OutfitGAN is illustrated in Fig. 2. In particular, Fig. 2(a) shows three key modules: an outfit generator G , an outfit discriminator D , and a collocation classification module CCM. For clarity, the outfit generator and collocation classifier that make up the key components of our OutfitGAN are described firstly. The training losses of our model are shown in Fig. 2(b), and are elaborated later in this section.

1) *Outfit Generator:* To synthesize a set of complementary items to make up an outfit, our framework needs to learn a mapping function from extant fashion items to new synthesized ones, by considering the compatibility between fashion items. To accomplish this, we train an outfit generator G to translate a given fashion item into multiple collocated ones. In particular, to synthesize a whole outfit that includes N fashion items, we introduce an outfit generator G , which includes $(N - 1)$ item generators to synthesize a set of fashion items, conditioned on a given item. The i -th item generator G_i includes three components, as shown in Fig. 2(a): an encoder Enc_i , a decoder Dec_i , and a semantic alignment module SAM_i . Here, we employ Enc_i and Dec_i in a similar way to a general image-to-image translation generator [3]. The detailed structures of Enc_i and Dec_i can be found in [3]. The SAM_i was developed to capture the correspondence between the input and output images. In a compatible outfit, harmonizing

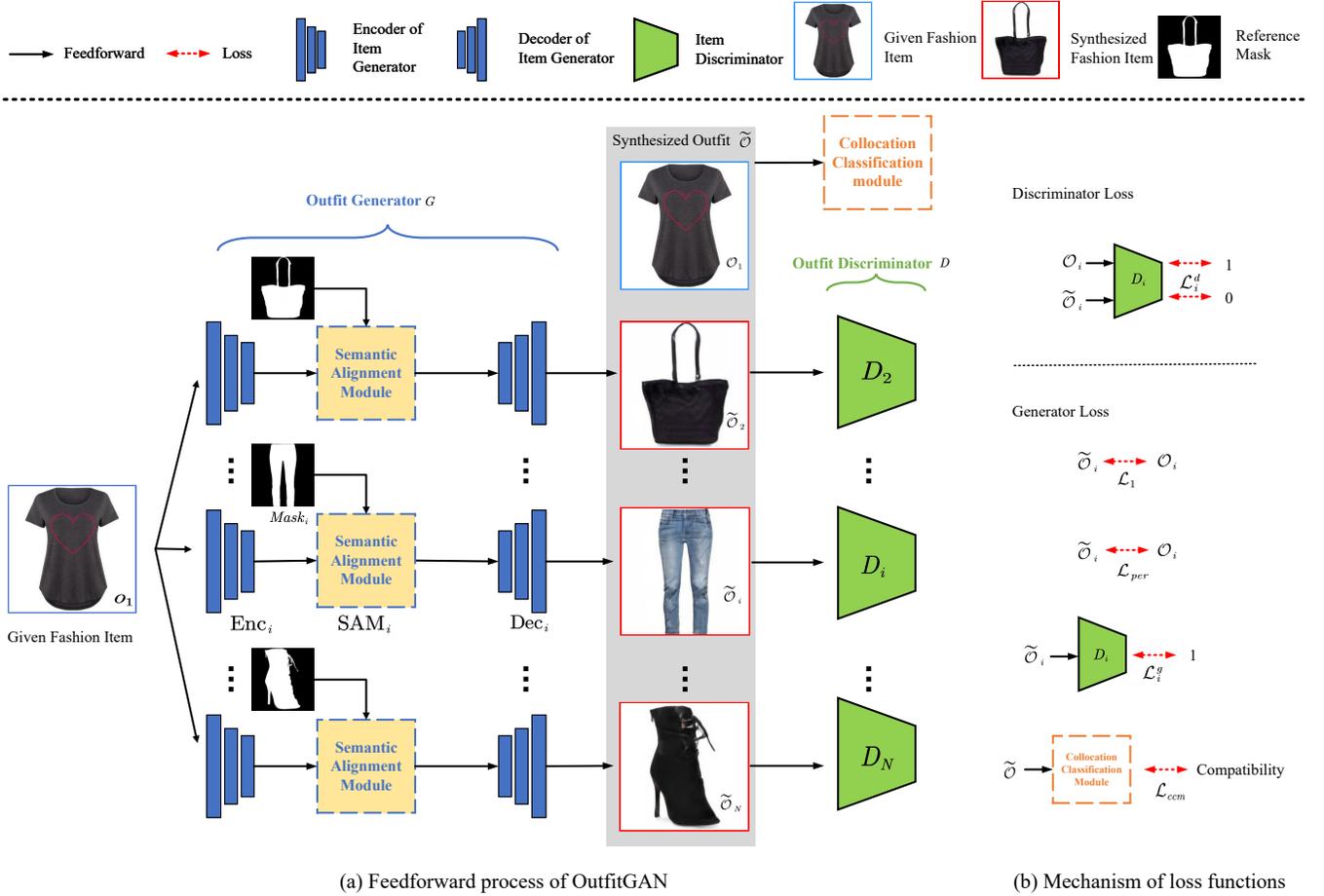


Fig. 2: The structure of OutfitGAN framework, which sets $\mathcal{O}_1(k=1)$ as the given fashion item, including: (a) the feedforward process of our OutfitGAN during training and (b) training losses of our model.

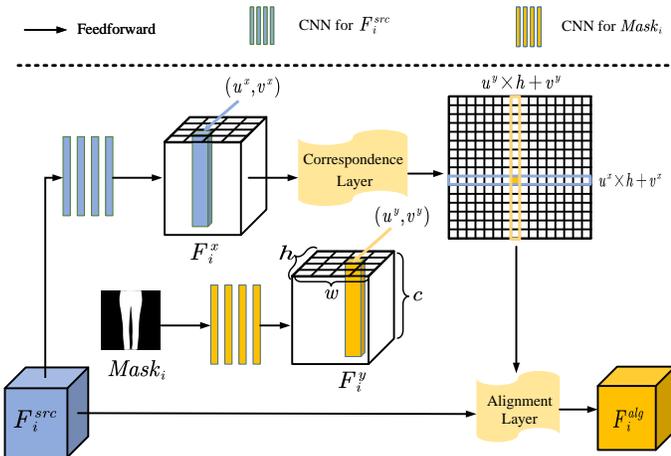


Fig. 3: Feedforward process of semantic alignment module.

elements or styles are often shared by each fashion item. To characterize these shared elements or styles, the SAM_i is used to capture the collocation correspondences among the fashion items in a certain outfit. In order to fully capture the spatial mapping relationships between the given fashion items and synthesized ones, we use SAM_i to learn these

relationships during the training of OutfitGAN. As shown in Fig. 3, SAM_i has four components: two branches consisting of CNNs, a correspondence layer and an alignment layer. These two CNN branches are used for feature extraction, while the correspondence layer with a differentiable module [28] is used to calculate the degree of spatial correlation for each pair of locations for the features extracted by the two CNNs, and the alignment layer aligns the features from Enc_i based on the degree of spatial correlation. We first use two separate CNNs to extract the features of a given fashion item and the reference mask of a target item, which are denoted by F_i^x and F_i^y , respectively. Here, F_i^x and F_i^y all lie in the space $\mathbb{R}^{h \times w \times c}$, where h and w are the height and width of F_i^x and F_i^y , respectively, and c is the number of channels. The correspondence layer is then applied to calculate the correspondence matrix $M_i^{corr} \in \mathbb{R}^{(h \times w) \times (h \times w)}$ for these two types of features. In particular, the operation used by the correspondence layer to obtain the correspondence matrix M_i^{corr} can be expressed as follows:

$$M_i^{corr}(u, v) = \frac{F_i^x(u)^T F_i^y(v)}{\|F_i^x(u)\| \cdot \|F_i^y(v)\|}, \quad (1)$$

where u and v are the row and column indexes for F_i^x and F_i^y , respectively. Each position in M_i^{corr} represents the degree

of correlation between two positions F_i^x and F_i^y . As shown in Fig. 3, the correlation degree of the c -dimensional feature in (u^x, v^x) of F_i^x and that in (u^y, v^y) of F_i^y is represented as a scalar value in $(u^x \times h + v^x, u^y \times h + v^y)$ of M_i^{corr} . A higher value indicates a higher degree of correlation.

After obtaining the correspondence matrix for the input and output images, the feature of a given fashion item from Enc_i , represented as F_i^{src} , needs to be aligned according to M_i^{corr} in order to synthesize a complementary fashion item. To achieve this, the alignment layer is formulated as follows:

$$F_i^{alg}(u) = \sum_v F_i^{src}(u) \cdot softmax(M_i^{corr}(u, v)), \quad (2)$$

where F_i^{alg} is the aligned feature from F_i^{src} based on M_i^{corr} . The vector at each position in F_i^{alg} is the result of a weighted summation of the vectors in the source feature F_i^{src} . We then feed F_i^{alg} into the decoder Dec_i to synthesize a fashion item \mathcal{O}_i .

2) *Collocation Classification Module*: In this subsection, we describe the CCM, which is used to model the compatibility prediction in order to supervise the compatibility during the outfit generation process.

More specifically, to ensure that the synthesized outfits fall into the collocation domain, we pre-train a CCM (see Section III-C3), which is leveraged to identify whether or not a synthesized outfit is compatible. During the training of OutfitGAN, we fix the parameters of the pre-trained CCM to supervise the compatibility of synthesized items. If the synthesized items are compatible, the CCM applies a smaller penalty to the outfit generator G , and if not, the penalty is larger. In particular, the CCM is designed as a sequence model that regards the outfit as a sequence from the perspective of human vision [21]. To synthesize compatible outfits, we employ a pre-trained sequence model to maintain the compatibility for outfit generation. This includes a pre-trained CNN and two directional LSTMs [15], in order to supervise the compatibility from two directions. Formally, given a fashion outfit $\mathcal{O} = [\mathcal{O}_1, \dots, \mathcal{O}_i, \dots, \mathcal{O}_N]$, we regard it as a sequence, where \mathcal{O}_i is the i -th fashion item in \mathcal{O} . As shown in Fig. 4, we first extract the latent feature f_i for \mathcal{O}_i using a pre-trained CNN, and this is then fed into a Bi-LSTM module. For example, the forward LSTM recurrently takes the feature f_{i-1} and the last hidden state \vec{h}_{i-1} as input and outputs a hidden state \vec{h}_i from $i = 2$ to N , as follows:

$$\vec{h}_i = LSTM(f_1, \dots, f_{i-1}). \quad (3)$$

Similarly, the backward LSTM takes the features in the reverse order and outputs the hidden state \overleftarrow{h}_i from $i = N - 1$ to 1. We then attempt to maximize the probability of the next item in the outfit given the previous sequence. More formally, we minimize the following compatibility objective function using a cross-entropy loss [29] in the form:

$$\begin{aligned} \mathcal{L}_{ccm} = & -\frac{1}{N-1} \sum_{i=2}^N \log\left(\frac{\exp(\vec{h}_i f_i)}{\sum_{f \in \mathcal{F}} \exp(\vec{h}_i f)}\right) \\ & -\frac{1}{N-1} \sum_{i=N-1}^1 \log\left(\frac{\exp(\overleftarrow{h}_i f_i)}{\sum_{f \in \mathcal{F}} \exp(\overleftarrow{h}_i f)}\right), \end{aligned} \quad (4)$$

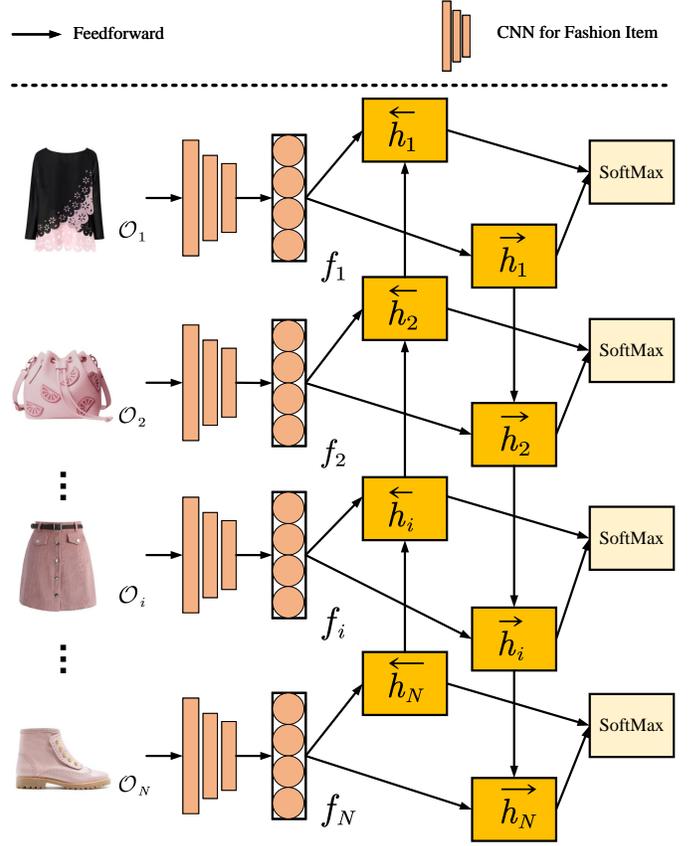


Fig. 4: Illustration of collocation classification module.

where these two loss terms represent the probabilities of the predictions from the forward and backward LSTM, respectively, and \mathcal{F} denotes all the features f of the current batch. In the pre-training phase, all of the parameters involved in the CCM are learnable. We fix all the learned parameters of the pre-trained module during the training of OutfitGAN.

3) *Training Losses*: In addition to the components of OutfitGAN mentioned above, the training losses are of the utmost importance in terms of supervising the training process. As shown in Fig. 2(b), the loss function for OutfitGAN includes two types of losses: the outfit discriminator loss and the outfit generator loss. We first discuss the adversarial training loss for the outfit discriminator. As shown in Fig. 2(a), our outfit discriminator uses $(N - 1)$ independent item discriminators to guide the outfit generation. In a similar way to MUNIT [5], for each item discriminator D_i we adopt a multi-scale discriminator architecture [3] and an LSGAN objective [30] to guide the training of our generator. We take the discriminator for the i -th item \mathcal{O}_i as an example. We first downsample the real and synthesized images by factors of two and four. The item discriminator $D_i = \{D_{i,1}, D_{i,2}, D_{i,3}\}$ is then applied to distinguish between the real and synthesized images at three different scales. Formally, the objective function of our adversarial loss for the training of each item discriminator is expressed as follows:

$$\mathcal{L}_i^d = \sum_{s=1}^3 \mathcal{L}_{i,s}^d(D_i), \quad (5)$$

where $D_i = \{D_{i,1}, D_{i,2}, D_{i,3}\}$ is the discriminator of the real fashion item \mathcal{O}_i and the synthesized fashion item $\tilde{\mathcal{O}}_i$, and $\mathcal{L}_{i,s}^d$ is the LSGAN objective for training D_i at a down-sampling scale s . In particular, for each scale s , the $\mathcal{L}_{i,s}^d$ is formulated as follows:

$$\begin{aligned} \mathcal{L}_{i,s}^d(D_{i,s}) &= \mathbb{E}_{\mathcal{O}_i \sim p_{data}(\mathcal{O}_i)}(D_{i,s}(\nabla(\mathcal{O}_i, s-1) - 1))^2 + \\ &\quad \mathbb{E}_{\mathcal{O}_k \sim p_{data}(\mathcal{O}_k)}(D_{i,s}(\nabla(G_i(\mathcal{O}_k, Mask_i), s-1)))^2, \end{aligned} \quad (6)$$

where p_{data} is the distribution of real data, $\nabla(x, s)$ represents down-sampling an image x by a factor of 2^s , and \mathcal{O}_k is the given fashion item.

In addition to the outfit discriminator loss, as shown in Fig. 2(b), the loss used in the outfit generator has four parts: the adversarial loss for the generator (\mathcal{L}^g) [1], the L1 loss (\mathcal{L}_1) [2], the perceptual loss (\mathcal{L}_{per}) [12], and the CCM loss (\mathcal{L}_{ccm}). More specifically, the objective function for our outfit generator loss is defined as follows:

$$\mathcal{L}_{total} = \frac{1}{N-1} \sum_{\substack{i=1 \\ i \neq k}}^N \mathcal{L}_i^g + \lambda_1 \mathcal{L}_1 + \lambda_2 \mathcal{L}_{per} + \mathcal{L}_{ccm}, \quad (7)$$

where λ_1 and λ_2 are two coefficients used to balance each loss. In the following, we introduce these losses used in the outfit generator. The compatibility loss was introduced in Eq. (4) and is not discussed here.

Adversarial loss: The objective function of the adversarial loss used in the training of G includes $(N-1)$ losses for each G_i . Each adversarial loss for G_i can be expressed as follows:

$$\begin{aligned} \mathcal{L}_i^g(G_i) &= \\ &\quad \sum_{s=1}^3 \mathbb{E}_{\mathcal{O}_k \sim p_{data}(\mathcal{O}_k)} [D_{i,s}(\nabla(G_i(\mathcal{O}_k, Mask_i), s-1)) - 1]^2, \end{aligned} \quad (8)$$

where \mathcal{L}_i^g is the LSGAN objective function for G_i and p_{data} is the distribution of real data.

L1 loss: To minimize the difference between the target outfits and the synthesized ones, we use a reconstruction loss (L1) to capture the overall structure of the images from the target domain. Specifically, we keep the discriminator unchanged and add the L1 loss to calculate the absolute distance between the synthesized images and the target ones [2]. This is defined as follows:

$$\mathcal{L}_1 = \frac{1}{N-1} \sum_{\substack{i=1 \\ i \neq k}}^N \|\tilde{\mathcal{O}}_i - \mathcal{O}_i\|_1, \quad (9)$$

where $\tilde{\mathcal{O}}_i \in \mathbb{R}^{256 \times 256 \times 3}$ denotes a synthesized image of the i -th fashion item and $\mathcal{O}_i \in \mathbb{R}^{256 \times 256 \times 3}$ is a target image for the same category.

Perceptual loss: Unlike the L1 loss, the perceptual loss [12] is introduced to ensure that the synthesized images are close to the target ones in high-level feature space. It also measures the perceptual difference between the images in terms of their content and style. Here, we adopt the perceptual loss to ensure that our OutfitGAN produce images that are similar to the ground truths. We compute the perceptual loss in the *relu1_2*,

relu2_2, *relu3_3* and *relu4_3* layers of the VGG-16 network ϕ which was pre-trained on ImageNet [31]. We then apply an auxiliary benchmark from DeepFashion [32] consisting of 50 categories of fashion items with 289,229 images, each of which is annotated with 1,000 descriptive attributes. This benchmark was used to classify the attributes, in order to fine-tune our network ϕ . Specifically, the perceptual loss adopted here is defined as:

$$\mathcal{L}_{per} = \frac{1}{N-1} \sum_{\substack{i=1 \\ i \neq k}}^N \sum_l \|\phi_l(\tilde{\mathcal{O}}_i) - \phi_l(\mathcal{O}_i)\|_1, \quad (10)$$

where $l \in \{\text{relu1_2}, \text{relu2_2}, \text{relu3_3}, \text{relu4_3}\}$ is the aforementioned layer of VGG-16, and ϕ_l represents the function of layer l .

C. Implementation Details

In this subsection, we introduce two reference mask generation strategies used to synthesize the alternative masks. Pix2Pix mask generation is used to synthesize reference masks via a pre-trained generative model, and random mask generation is used to return the sampled reference masks from the training set. In this following, we discuss the details of the detailed network architecture of OutfitGAN. Finally, we illustrate the overall adversarial algorithm used to train OutfitGAN.

1) *Strategies for Reference Mask Generation:* The reference mask is an essential component of OutfitGAN, and provides important guidance information in terms of supervising the generation of compatible fashion items. In addition to the reference masks given by users, we may in practice need to synthesize reference masks based on models, such that they can then be fed into OutfitGAN. To overcome this issue, we design two strategies for the synthesis of a diverse range of masks: Pix2Pix and random mask generation. These two methods mimic the phase in which fashion designers or common users generate reference masks, and extend the basic functions of OutfitGAN.

Mask generation using Pix2Pix: In order to take advantage of reference masks to improve the effectiveness of OutfitGAN, we propose a method of synthesizing reference masks using a pre-trained generative model to extend the function of OutfitGAN. In fact, reference mask generation can be regarded as another image-to-image translation task, in which the input is an RGB image of a given fashion item and the output consists of the corresponding masks of compatible fashion items. There are many methods that are capable of synthesizing reference masks for a given fashion item [2][3]. In this research, Pix2Pix [2], as a representative framework among these methods, was chosen for this task. We constructed a large-scale dataset called OutfitSet (more details are given in Section IV-A), which consisted of 20,000 outfits with their associated masks, and used this to train the mask generator. Using $(N-1)$ fashion items for each complete outfit (i.e., excluding the given fashion item), we pre-trained $(N-1)$ independent Pix2Pix mask generators on the training set of OutfitSet to synthesize the reference masks for the target

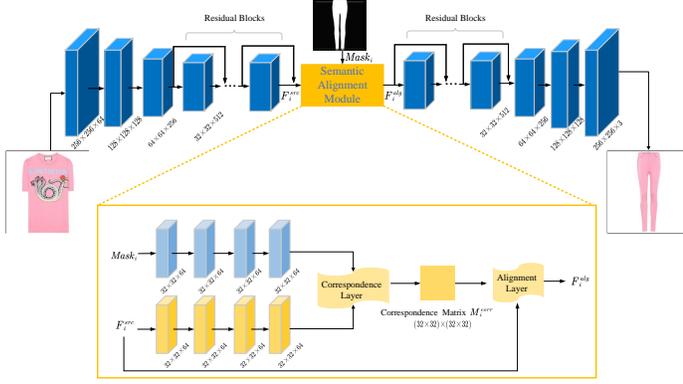


Fig. 5: Network architecture of the i -th item generator G_i .

fashion items. For example, given an RGB upper clothing image, we needed to synthesize the corresponding masks for the bag, lower clothing and shoes. Each mask generator included an encoder, several residual blocks and a decoder. Each discriminator used the PatchGAN architecture [2], and the LSGAN [30] and L1 losses were employed to guide the process of reference mask generation. We used the default setting given in [2] to pre-train the Pix2Pix mask generator. The detailed structure of the network can be found in [2]. In the testing phase of OutfitGAN, the reference masks for compatible fashion items can be automatically synthesized by the pre-trained Pix2Pix mask generator without the need for assistance from users.

Random mask generation: In a real-world clothing collocation application, users can use different reference masks to guide the generation of outfits. In order to meet these personalized requirements, we can use a random mask generator that randomly selects a reference mask for a compatible fashion item from a source dataset. In our implementation, the training set for our constructed OutfitSet was used as a source dataset for reference masks. Given a fashion item, reference masks corresponding to target compatible fashion items can be randomly selected from the source dataset based on the categories of target collocation items. This strategy reflects the different tastes of users in terms of the selection of reference masks, and increases the diversity of the generated compatible fashion items to some extent. In our experiments, we also performed a detailed empirical study of the effects of different reference mask generation strategies on the results.

2) *Network Architecture:* In this subsection, we describe the detailed network architecture of OutfitGAN. For the i -th item generator G_i , as shown as an example in Fig. 5, we employ the architecture of encoder Enc_i and the decoder Dec_i from [5], in which their effectiveness in image-to-image translation is proven [4]. The encoder Enc_i includes four convolutional blocks (conv-blocks) and three residual blocks (res-blocks), whereas the decoder contains three res-blocks, three upsampling and conv-block modules, and one conv-block followed by a Tanh function. We apply a ReLU activation function to all the conv-blocks. As illustrated in Fig. 5, the SAM applies four conv-blocks to each branch of the feature extractor, followed by a correspondence layer and an

Algorithm 1: Adversarial training algorithm for OutfitGAN

Input: Extant fashion item \mathcal{O}_k , reference masks $[Mask_1, \dots, Mask_{k-1}, Mask_{k+1}, \dots, Mask_N]$, and target fashion items $[\mathcal{O}_1, \dots, \mathcal{O}_{k-1}, \mathcal{O}_{k+1}, \dots, \mathcal{O}_N]$

Output: OutfitGAN generator G

- 1 Pre-train collocation classification module CCM on our training set and select the best model through validation set, update the parameters θ_{CCM} of CCM with
- 2 $\theta_{CCM} \leftarrow \theta_{CCM} - \eta_{CCM} \nabla_{\theta_{CCM}} (\mathcal{L}_{CCM})$; // See Eq. (4)
- 3 Fine-tune the VGG-16 for attributes classification on DeepFashion; // Prepare for perceptual loss
- 4 Initialize the parameters θ_G, θ_D of G, D , respectively; fix all parameters of CCM and VGG-16;
- 5 **for** $iter \leftarrow 1$ **to** N_{iter} **do**
- 6 sample a batch of $\mathcal{O} = [\mathcal{O}_1, \dots, \mathcal{O}_N]$ and reference masks $\{Mask_1, \dots, Mask_N\}$ from training set;
- 7 **for** $i \in \{1, \dots, k-1, k+1, \dots, N\}$ **do**
- 8 $\mathcal{L}_i^d \leftarrow \sum_{s=1}^3 \mathcal{L}_{i,s}^d(D_i)$; // See Eq. (5)
- 9 update $\theta_{D_i} \in \theta_D$ with
- 10 $\theta_{D_i} \leftarrow \theta_{D_i} - \eta \nabla_{\theta_{D_i}} (\mathcal{L}_i^d)$;
- 11 **end**
- 12 **for** $i \in \{1, \dots, k-1, k+1, \dots, N\}$ **do**
- 13 $\mathcal{L}_i^g \leftarrow \sum_{s=1}^3 \mathcal{L}_{i,s}^g(D_i)$; // See Eq. (8)
- 14 **end**
- 15 $\mathcal{L}_1 \leftarrow \frac{1}{N-1} \sum_{i=1}^N \|\tilde{\mathcal{O}}_i - \mathcal{O}_i\|_1$; // See Eq. (9)
- 16 $\mathcal{L}_{per} = \frac{1}{N-1} \sum_{i=1}^N \sum_l \|\phi_l(\tilde{\mathcal{O}}_i) - \phi_l(\mathcal{O}_i)\|_1$; // See Eq. (10)
- 17 $\mathcal{L}_{CCM} \leftarrow -\frac{1}{N-1} \sum_{i=2}^N \log\left(\frac{\exp(\vec{h}_i f_i)}{\sum_{f \in \mathcal{F}} \exp(h_i f)}\right) - \frac{1}{N-1} \sum_{i=N-1}^1 \log\left(\frac{\exp(h_i f_i)}{\sum_{f \in \mathcal{F}} \exp(h_i f)}\right)$; // See Eq. (4)
- 18 update θ_G with
- 19 $\theta_G \leftarrow \theta_G - \eta \nabla_{\theta_G} \left(\frac{1}{N-1} \sum_{i=1}^N \mathcal{L}_i^g + \lambda_1 \mathcal{L}_1 + \lambda_2 \mathcal{L}_{per} + \mathcal{L}_{CCM}\right)$;
- 20 // See Eq. (7)
- 20 **end**

alignment layer. Our discriminator is designed using a multi-scale architecture, in the same way as in [5]. In the CCM, we extract image features with a pre-trained ResNet-50 [33] provided by PyTorch [34] and a fully-connected network for 512-dimensional embeddings. A Bi-LSTM (which includes forward and backward LSTM) is then used to model the collocation relationship. In the same way as in [35], the number of layers for each LSTM is set to one, and the number of hidden features is set to 512.

3) *Adversarial Training Process:* In this subsection, we present the design of an adversarial training scheme which is used to optimize the generator G of OutfitGAN. For clarity, the entire training process of OutfitGAN is summarized in Algorithm 1. We first pre-train the collocation classification module on our training set by minimizing the loss in Eq. (4) with a learning rate η_{CCM} . We then select the best CCM model with our validation set (see Section IV-A) by calculating the smallest \mathcal{L}_{CCM} according to Eq. (4) (shown in lines 1-2). Following this, we fine-tune the VGG-16, which was pre-trained on ImageNet, by applying the attribute classification method used in DeepFashion (shown in line 3). We initialize the parameters of G and D , and fix all the parameters of the

CCM and VGG-16 during the training of OutfitGAN (shown in line 4). The subsequent training process is carried out by applying a gradient descent step to D and G in alternate steps, and using the gradient descent method to update the parameters θ_D and θ_G of D and G , respectively (shown in lines 5-20). Specifically, given a batch of outfit images and reference masks, we train each $D_i \in D$ by reducing the loss in Eq. (5) (shown in line 10). We then fix θ_D and calculate the adversarial loss (\mathcal{L}_i^g) for each $G_i \in G$ (shown in line 13) and L1 loss (\mathcal{L}_1) (shown in line 15), the perceptual loss (\mathcal{L}_{per}) (shown in line 16) and the CCM loss (\mathcal{L}_{ccm}) (shown in line 17) for G . Finally, we optimize θ_G by reducing the loss in Eq. (7) (shown in line 19). We train D and G over N_{iter} iterations with a learning rate of η .

IV. EXPERIMENTS

In this section, we first describe the construction of our dataset in detail. Parameter settings of models and evaluation metrics are then described sequentially. The performance of our proposed OutfitGAN is compared against several competitive image-to-image translation baselines, and we perform an ablation study to verify the effectiveness of the main modules in OutfitGAN. Furthermore, we conduct a parametric study on our model and an extra study on different sequences of fashion items used in the collocation classification module. Finally, the limitation of our framework is discussed.

A. Dataset

When carrying out fashion outfit generation, accurate fashion datasets are of the utmost importance in terms of providing the ground truths for model training and evaluation. Although many public fashion outfit compatibility datasets are available for fashion modeling, such as UT Zappos50K [36], the Maryland Polyvore dataset [21], FashionVC [37], and IQON3000 [38], all of these lack explicit common category annotations for fashion items and clear compositions for outfits. To overcome these issues in current datasets and to verify the effectiveness of our proposed outfit generation models, we collected fashion outfits from a fashion matching website, Polyvore.com, which contained numerous compatible outfits constructed by fashion experts. These outfits were put together based on the preferences of fashion experts, with the aim of clearly and attractively presenting specific fashion styles. The original dataset consisted of over 344,560 outfits, which were composed of 2,131,607 fashion items. We selected four types of fashion items (upper clothing, bag, lower clothing and shoes) that are common components of outfits worn in daily life. We used the upper clothing as the given fashion item in order to exploit the richer information on styles that can be obtained from the upper clothing compared with the other fashion items in the same outfit. This means that for each outfit set [upper clothing, bag, lower clothing, shoes] (i.e., $N = 4$), the extant given fashion item represents upper clothing. We therefore kept only those outfits that included all four of these categories. Since images of shoes have diverse orientations due to the different shooting angles used, we filtered out images that only contained one shoe, and flipped all

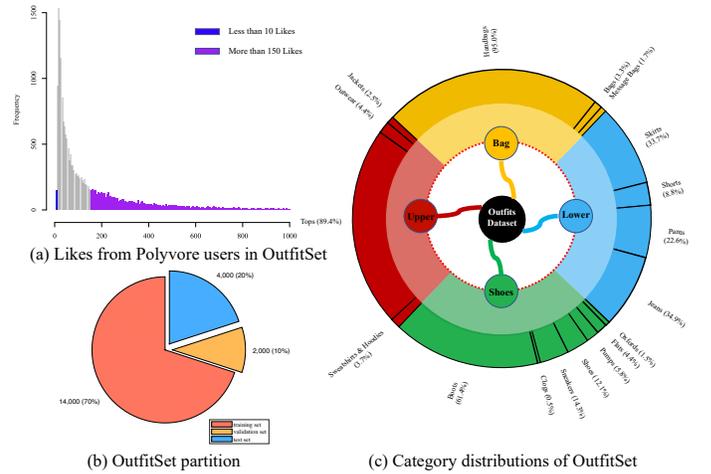


Fig. 6: Statistics of OutfitSet.

left-oriented shoes horizontally to form right-oriented shoes. After this process was complete, 32,043 outfits remained in the dataset. We selected the top 20,000 outfits based on the number of likes given by Polyvore users. As shown in Fig. 6(a), most of the outfits had more than 10 likes and less than 150. We then partitioned these outfits randomly into three folds, to form a training set, a validation set and a test set, and these constituted our OutfitSet dataset. The training set contained 14,000 outfits (70%), the validation set 2,000 outfits (10%) and the test set 4,000 outfits (20%), as shown in Fig. 6(b). In particular, the training set was used to train the CCM, OutfitGAN, and the Pix2Pix mask generation strategy; the validation set was used to select the best pre-trained CCM; and the test set was used to evaluate OutfitGAN at the testing stage. It is worth noting that the OutfitSet dataset also contained many sub-classes for each class, as illustrated in Fig. 6(c), and had relatively rich category-based annotations in comparison to existing datasets. For automatic reference mask generation (see Section III-C1), we employed a saliency detector [39] to detect the masks of fashion items for use as reference masks to guide the mask generation. Each fashion item in our dataset had a corresponding mask in the form $[m]^{256 \times 256}$, $m \in \{0, 1\}$, where a value of one denotes the segmentation of fashion items and zero denotes the background of an image.

B. Experimental Setup and Parameter Settings

In the experiments, all images were resized to 256×256 , and we used random cropping for data augmentation during training. In the training phase, the batch size was set to four, and the number of training iterations for the model was set to 200,000. All experiments were performed on a single NVIDIA GeForce RTX 3090, and the implementation was carried out in PyTorch [34]. We set the coefficients to balance the losses as follows: $\lambda_1 = 100$ and $\lambda_1 = 10$ for OutfitGAN with real reference masks, and $\lambda_1 = 10$ and $\lambda_1 = 10$ for OutfitGAN with reference masks produced by mask generation strategies. The CCM was trained with an SGD [40] optimizer with a learning rate η_{cmp} of 0.2 and a momentum of 0.9. OutfitGAN was trained with an Adam [41] optimizer with $\beta_1 = 0$ and

$\beta_2 = 0.99$, and the learning rate η for G and D was set to 10^{-4} .

C. Evaluation Metrics

To evaluate the performance of our proposed model, we used a variety of evaluation metrics from three perspectives, as follows:

- 1) A similarity measurement was used to measure the similarity between the synthesized images and the target ones. We adopted two metrics: a structural similarity (SSIM) [42] and a learned perceptual image patch similarity (LPIPS) [43]. SSIM [42] is a traditional, widely used image quality index for image comparison. Given two local patches extracted from input images, i.e., a real image patch x and a synthesized image patch y , SSIM measures the luminance, contrast and similarity of x and y , where a higher score indicates a higher similarity. LPIPS [43] is another common metric used to evaluate the image similarity between two images, particularly for a synthesized image and a target one, with a pre-trained deep model. We used the default pre-trained AlexNet [44] provided by the authors [43] to calculate the LPIPS metric. Here, a higher score indicates a lower similarity, and vice versa.
- 2) An authenticity measurement was applied to reflect the quality of the synthesized images in terms of their authenticity. Previous studies [7] have suggested that the Fréchet inception distance (FID) can be used to estimate the authenticity of synthesized images in feature space. More specifically, the FID measures the similarity between two domains of images, and is particularly suitable for real images and images synthesized by GANs. To calculate the FID between two image domains \mathcal{Y} and \mathcal{Y}' , we first embed both images into the same feature space F given by an Inception model [45]. The FID can be defined as follows:

$$\text{FID}(\mathcal{Y}, \mathcal{Y}') = \|\mu_{\mathcal{Y}} - \mu_{\mathcal{Y}'}\|_2^2 + \text{Tr}(\Sigma_{\mathcal{Y}} + \Sigma_{\mathcal{Y}'} - 2(\Sigma_{\mathcal{Y}}\Sigma_{\mathcal{Y}'})^{\frac{1}{2}}), \quad (11)$$

where $\mu_{\mathcal{Y}}$ and $\mu_{\mathcal{Y}'}$ are the average values of the feature space F for \mathcal{Y} and \mathcal{Y}' , respectively; $\Sigma_{\mathcal{Y}}$ and $\Sigma_{\mathcal{Y}'}$ are their variances, respectively; and $\text{Tr}(\cdot)$ is the trace of the matrix. A lower FID score indicates a higher visual authenticity for the synthesized images, and vice versa.

- 3) A compatibility measurement was used to gauge the degree of matching between the synthesized outfits. In order to perform a fair evaluation in terms of the compatibility of each outfit, we developed a new metric called the fashion compatibility test score (FCTS). For this metric, we used an open-source toolbox MMFashion³ to evaluate the fashion compatibility between the items making up an outfit. The fashion compatibility predictor module of MMFashion was developed on the basis of the work in [20] on fashion compatibility prediction. To enable a fair comparison, this fashion

compatibility predictor ψ was trained on the Maryland Polyvore dataset [21], meaning that its training set was different from our OutfitSet, and the pre-trained model was provided by MMFashion. We calculated the FCTS for all models as follows. Firstly, both positive and negative samples were constructed. We defined positive samples as outfits synthesized by generative models, and the negative samples were randomly composed of synthesized fashion items which are not from the same outfit. We assume the synthesized outfits (positive samples) are more compatible than the randomly composed ones (negative samples). We tested the compatibility score between positive and negative samples based on FCTS, which can be defined as:

$$\text{FCTS} = \frac{\sum_{j=1}^{N_{\text{cmp}}} [\text{Comp}(\text{outfit}_j^p) > \text{Comp}(\text{outfit}_j^n)]}{N_{\text{cmp}}}, \quad (12)$$

where $\text{Comp}(\cdot)$ is the fashion compatibility score computed by the compatibility predictor ψ ; N_{cmp} denotes the number of comparisons between the positive and negative samples; outfit_j^p and outfit_j^n denote a positive and negative outfit sample, respectively. A higher score indicates better compatibility.

D. Performance Comparison

1) *Compared Methods*: To examine the effectiveness of our proposed OutfitGAN, we compared it with six state-of-the-art methods: Pix2Pix [2], Pix2PixHD [3], CycleGAN [3], MUNIT [5], DRIT++ [6], and StarGAN-v2 [7]. These include both supervised and unsupervised models. For completeness, we give a brief introduction to these methods as follows:

Pix2Pix [2] is the first framework developed for supervised image-to-image translation, and uses a U-Net architecture for the generator and a single discriminator to classify real and fake image pairs.

Pix2PixHD [3] is an improved version of Pix2Pix framework based on a coarse-to-fine approach, which uses a coarse-to-fine generator, a multi-scale discriminator and a feature matching loss.

CycleGAN [4] is an unsupervised image-to-image translation method with a cycle reconstruction loss, and was the first framework to address the issue of unpaired image-to-image translation.

MUNIT [5] is based on the idea that an image representation can be decomposed into a style code and a content code. It can learn disentangled representations for image-to-image translation.

DRIT++ [6] is an improved version of DRIT [46], which disentangles the latent spaces into a shared content space and an attribute space for each domain and was developed to synthesize diverse images for image-to-image translation.

StarGAN-v2 [7] is an improved version of StarGAN [47] that employs an MLP to synthesize different styles and then injects them into decoders to synthesize a diverse range of images.

It should be noted that except for StarGAN-v2, these baseline methods can synthesize only one target domain image

³<https://github.com/open-mmlab/mmfashion>

given an image from the source domain. We therefore trained $(N - 1)$ models for each baseline model independently, except for StarGAN-v2. The implementations of these models were all based on original codes released by the authors, and the hyperparameters were tuned to adapt to our OutfitSet.

2) *Comparison of Results*: A quantitative comparison of the results for all of the evaluation metrics is given in Table I. As described in Section III-C1, reference masks that represent the outlines of target fashion items play an important role in guiding our model. These reference masks can be divided into three types based on their source, i.e., whether they were provided by a user, synthesized by a generative model or randomly selected by the system. As shown in Table I, we use OutfitGAN, OutfitGAN-P, and OutfitGAN-R to denote our model with real reference masks, synthesized reference masks from Pix2Pix mask generation and reference masks from random selection, respectively, in the following discussion. It should be noted that since different reference masks can produce different synthesized fashion items, we compare OutfitGAN-P and OutfitGAN-R with the baselines in terms of only the authenticity and compatibility measures, i.e., FID and FCTS. Table I shows that our proposed OutfitGAN consistently outperforms other image-to-image translation methods in terms of all three metrics (similarity, authenticity and compatibility). Fig. 7 shows examples of synthesized fashion items from our models and other baseline models. Since the reference masks of OutfitGAN and OutfitGAN-R are from the same domain, the results of OutfitGAN-R are omitted here. From Fig. 7, it is clear that our model produces superior results, particularly in terms of the textural details and the harmony of the styles with the given fashion items. The results of a quantitative evaluation show that OutfitGAN yields approximate performance improvements of 0.175, 0.098, and 0.185 in the SSIM for the categories of bag, lower clothing and shoes, respectively, in comparison to the second-best method; it also reduces approximate 0.168, 0.085, and 0.179 values of LPIPS for the generation of bag, lower clothing, and shoes, respectively, compared with other methods. This suggests that our synthesized images can maintain the overall image structure and visual similarity better than other methods. Fig. 7 shows that OutfitGAN can synthesize the most similar results in terms of visual appearance. This means that our approach not only surpasses other methods in terms of the quantitative similarity metrics, but also outperforms them in terms of visual observations.

We also compared our models with baseline methods with respect to the authenticity measurement, i.e., the FID. We evaluated the FID for each category of synthesized and target fashion items. For this metric, OutfitGAN reduces approximate 5.512, 25.724, and 6.649 for the generation of bag, lower clothing and shoes, respectively, in comparison with the second-best method. Our models with synthesized reference masks, OutfitGAN-P/OutfitGAN-R, reduce approximate 18.154, 9.825, and 7.088/ (10.886, 3.621, and 4.345) values of FID for the generation of bag, lower clothing and shoes compared with the second-best method. From the synthesized results in Fig. 7, we can see that the images produced by our models have higher authenticity based on human perceptual

observations. In particular, the Pix2Pix method sometimes produces spots on the borderline between the bag and lower clothing, and its synthesized images are not well contoured. The synthesis results from Pix2PixHD, MUNIT and DRIT++ are blurred, and the MUNIT method exhibits mode collapse in the synthesis of bags. CycleGAN always translates an upper clothing image into an outfit of compatible items while maintaining very similar styles, even for textual logos or lines. This can be attributed to the cycle reconstruction loss in CycleGAN. Of the methods compared here, StarGAN-v2 produces the best fashion items for lowers. Our OutfitGAN is able to synthesize the most visually plausible results based on real reference masks. Using synthesized masks, our OutfitGAN-P can also synthesize plausible results. With respect to the compatibility measurement for the synthesized outfits, the results for the FCTS for our OutfitGAN suggest that a generator supervised by a CCM can produce synthetic outfits with a superior degree of matching in comparison to other baselines. CycleGAN also produced promising results for the FCTS, as shown in Table I; however, the outfits synthesized by CycleGAN are based on styles that are extremely similar to those of the input upper clothing images, as can be observed from Fig. 7. The outfits synthesized by CycleGAN therefore did not achieve a high compatibility score from a human perspective, due to the lack of difference in style from the given fashion items.

E. Ablation Study

In this subsection, two sets of experiments are carried out to validate the effectiveness of the SAM and the CCM, which are the main components of OutfitGAN.

Effectiveness of the SAM: In order to investigate the effectiveness of the SAM, we validated it from two perspectives. Firstly, we trained our OutfitGAN without the SAM. In Table II, ‘OutfitGAN w/o SAM’ means that we concatenated a reference mask with only the feature from the i -th encoder Enc_i and fed the concatenated feature into the i -th decoder Dec_i . The results show that the OutfitGAN model with the SAM consistently outperformed the model without the SAM in terms of the SSIM, LPIPS, and FID. This indicates that the SAM in our original framework was able to learn a correspondence relationship between a given fashion item and the targeted collocation items, allowing the visual similarity and authenticity to be significantly improved. To further examine the impacts of the SAM, we elaborate the explanation of the correspondence M_i^{corr} for the i -th synthesized fashion item during the generation process in Fig. 8. As shown in Fig. 8(a), there is a selected mapping relationship between the extant upper clothing and the synthesized lower clothing. For clarity, the corresponding four highest semantic regions for the areas of each white block in the synthesized lower images are annotated in Fig. 8(a). We can see that the red regions of synthesized images are always from the red patches or other salient patches of the extant upper clothing. This suggests that the SAM captures the translation correspondence relationship. In addition to the precise mapping between the given fashion items and the synthesized ones, we also average the correspondence matrix for visualization. It can be seen that the

TABLE I: Results of compared methods (here, for all metrics except LPIPS and FID, higher is better)

Method	SSIM(\uparrow)			LPIPS(\downarrow)			FID(\downarrow)			FCTS(\uparrow)
	bag	lower	shoes	bag	lower	shoes	bag	lower	shoes	
Pix2Pix [2]	0.384	0.562	0.512	0.576	0.415	0.489	58.024	55.399	57.315	57.5%
Pix2PixHD [3]	0.468	0.620	0.542	0.556	0.365	0.448	151.683	191.154	134.126	57.5%
CycleGAN [4]	0.392	0.447	0.504	0.570	0.515	0.468	51.695	56.625	43.497	86.4%
MUNIT [5]	0.444	0.553	0.563	0.553	0.403	0.443	177.605	88.730	85.521	52.0%
DRIT++ [6]	0.392	0.470	0.514	0.585	0.510	0.480	74.474	108.055	80.916	52.2%
StarGAN-v2 [7]	0.355	0.610	0.477	0.590	0.355	0.465	153.603	116.923	116.467	50.3%
OutfitGAN	0.643	0.718	0.748	0.385	0.270	0.264	46.183	29.675	36.848	87.1%
OutfitGAN-P	-	-	-	-	-	-	33.541	35.324	36.409	91.9%
OutfitGAN-R	-	-	-	-	-	-	40.809	41.528	39.152	91.4%

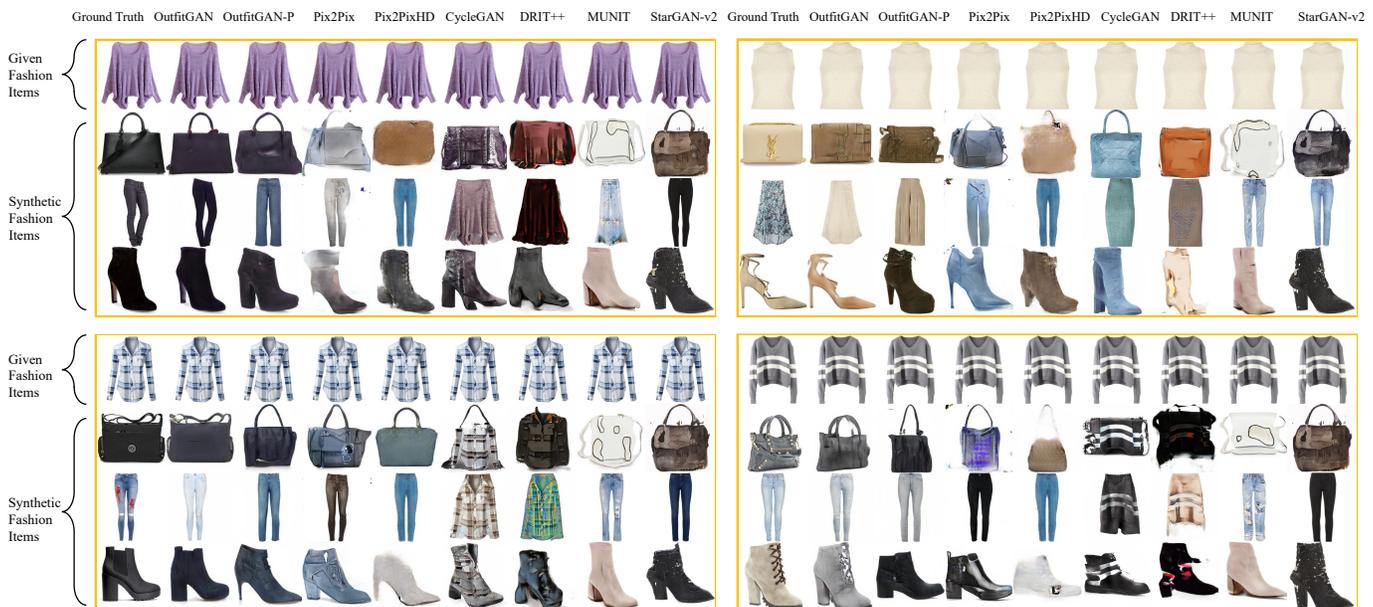


Fig. 7: Synthesized samples of our models and baselines.

TABLE II: Comparative results for OutfitGAN in terms of semantic alignment module

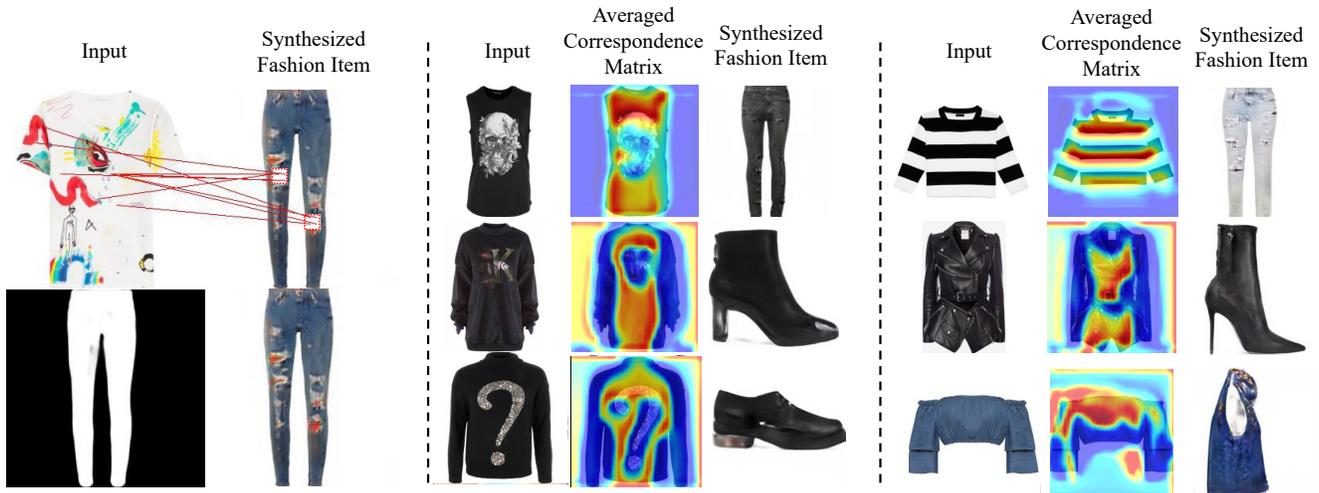
Method	SSIM(\uparrow)			LPIPS(\downarrow)			FID(\downarrow)		
	bag	lower	shoes	bag	lower	shoes	bag	lower	shoes
OutfitGAN w/o SAM	0.621	0.718	0.729	0.399	0.284	0.279	48.584	34.435	50.973
OutfitGAN w/ SAM	0.643	0.718	0.748	0.385	0.270	0.264	46.183	29.675	36.848

TABLE III: Results of OutfitGAN in terms of collocation classification module on FCTS

Method	FCTS(\uparrow)
OutfitGAN w/o CCM	67.5%
OutfitGAN w/ CCM	87.1%

SAM cognitively processes the specific patterns of the given fashion items to some extent, as shown in Figs. 8(b) and (c). We can divide the mapping relationship for the given fashion items into two types: ‘ignorance’ and ‘concentration’. Fig. 8(b) shows that the SAM can overlook some specific patterns for ignorance in given fashion items. Fig. 8(c) shows that the SAM concentrates more on certain patterns for outfit generation, and particularly on black-and-white lines rather than the other patterns. The above analysis of the SAM indicates that it is able to learn the correspondence relationships between the given fashion items and the synthesized ones.

Effectiveness of the CCM: We also explore the impact of the CCM in OutfitGAN. Specifically, we examine its effect on the FCTS in terms of visual compatibility. A comparison of the results is given in Table III, where ‘OutfitGAN w/ CCM’ and ‘OutfitGAN w/o CCM’ denote models with and without collocation classification, respectively. We can see that the model without the CCM gives a significant decrease in the FCTS, from 87.1% to 67.5%, thus demonstrating that the CCM markedly improves the compatibility of synthesized outfits. In addition, Fig. 10 shows that OutfitGAN with the CCM synthesizes more compatible outfits with more harmonious styles than the model without the CCM. The collocation module enhances the frequency of co-occurrence of compatible elements or style for compatible outfits. These quantitative and qualitative results suggest that the CCM can effectively improve the compatibility of the synthesized outfits.



(a) Precise mapping between given fashion items and synthesized ones (b) Average mapping of ‘ignorance’ for given fashion items (c) Average mapping of ‘concentration’ for given fashion items

Fig. 8: Illustration of explanation of our semantic alignment module for outfit generation.

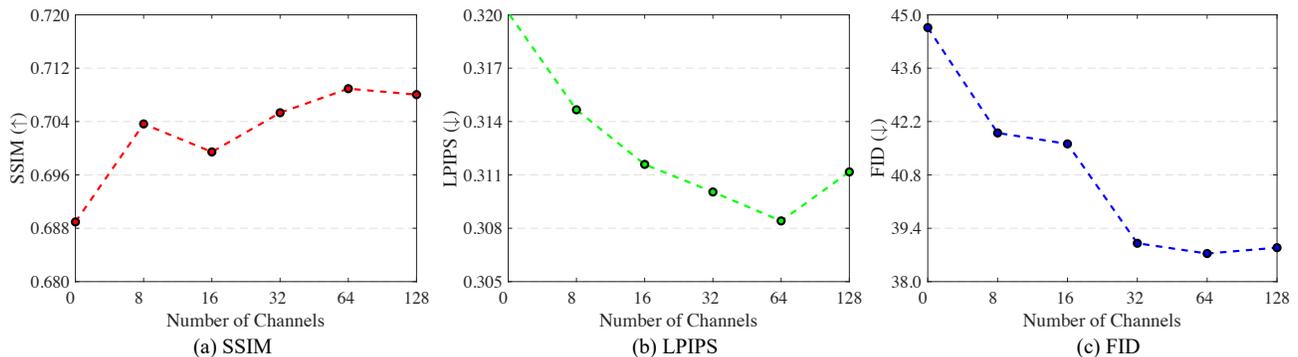


Fig. 9: Similarity of synthesis measurements over all categories with the number of feature channels for semantic alignment module.



Fig. 10: Synthesized samples of OutfitGAN with respect to collocation classification module.

F. Parametric Study

The main hyperparameters used in OutfitGAN are the number of feature channels for the SAM and the coefficients in the training losses.

Number of feature channels for the SAM. We first

investigate the influence of the number of feature channels on the results of outfit generation. We set the number of feature channels c to $[0, 8, 16, 32, 64, 128]$ in OutfitGAN. The results for the SSIM, LPIPS and FID over all categories are illustrated in Fig. 9. In particular, the use of zero channel means that we concatenate only the reference mask with the feature extracted by the i -th encoder Enc_i in OutfitGAN. From Fig. 9, we can see that an increase in the number of feature channels within the range $[0, 64]$ generally increases the performance of OutfitGAN in terms of the SSIM, LPIPS and FID. When the number of feature channels is increased beyond 64, the performance of OutfitGAN may become slightly worse. We ascribe this to the fact that the outfit generation process requires a much larger exploration space when the number of feature channels becomes large. In our experiments, setting the parameter c to 64 was sufficient to deliver satisfactory results.

Coefficients of the training loss. To further investigate the impacts of the coefficients used in weighting the training losses, we present the results from OutfitGAN with different weight parameters λ_1 and λ_2 for \mathcal{L}_1 and \mathcal{L}_{per} , respectively,

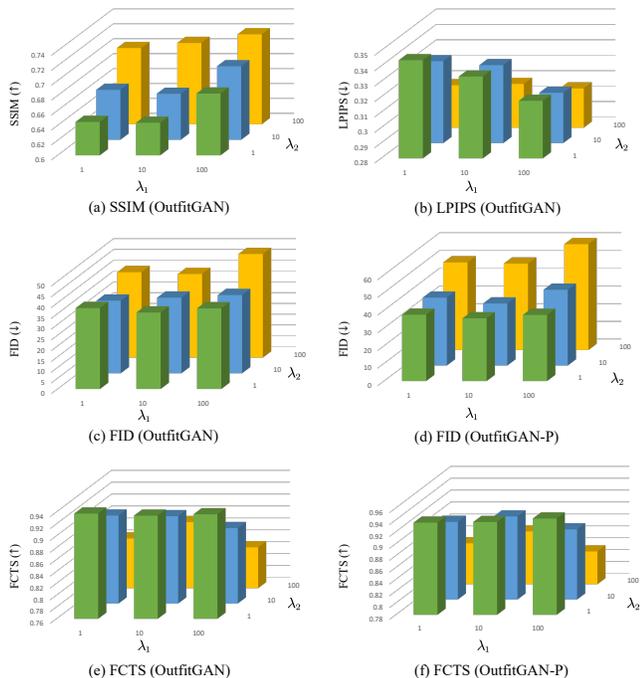


Fig. 11: Results of OutfitGAN and OutfitGAN-P against different settings of weight coefficients of training losses.

which are the reconstruction losses at the pixel-level and feature-level, as shown in Eqs. (9) and (10). The results are illustrated in Figs. 11 (a)-(f), in which the coefficients λ_1 and λ_2 are varied in the range [1, 10, 100]. Figs. 11 (a)-(c) show the results from OutfitGAN with different coefficients in terms of the metrics SSIM, LPIPS and FID; Fig. 11(d) shows the FID results for OutfitGAN with Pix2Pix mask generation (i.e., OutfitGAN-P) for different coefficients; and Figs. 11(e) and (f) show the FCTS results for OutfitGAN and OutfitGAN-P, respectively. From Fig. 11, we observe that the parameters λ_1 and λ_2 have a significant impact on the similarity measurements, and an increase in these values can strongly improve the similarity between the synthesized fashion items and the target ones. However, λ_1 and λ_2 produce the opposite impact on the authenticity and compatibility of the synthesized fashion items. This can be ascribed to the fact that an increase in these parameters weakens the influence of the discriminator and the CCM in OutfitGAN. In our implementation, the selection of these coefficients was made based on a tradeoff between the similarity and authenticity measurements. The results show that settings of $\lambda_1 = 100$ and $\lambda_2 = 10$ for OutfitGAN and $\lambda_1 = 100$ and $\lambda_2 = 10$ for OutfitGAN-P gave the best synthesized images in terms of their similarity and authenticity. For simplicity, the coefficient settings for OutfitGAN-R were the same as those for OutfitGAN-P in our experiments.

G. Study on Different Sequences of Fashion Items

As previously stated, the fashion compatibility task can be addressed with a sequence model which is motivated by the human observation perspective [21]. However, the sequence of the fashion items in an outfit has many possible arrangements.

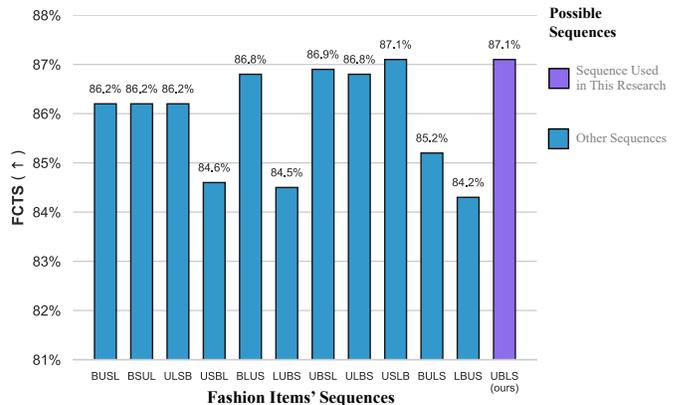


Fig. 12: Fashion compatibility measurements against different settings of possible fashion items' sequences (here, each item in the abscissa represents a possible sequence, where 'UBLS' represents an order of [upper, bag, lower, shoes], and other items have similar definitions).

For an outfit with N fashion items, it has $N!$ possible orders to model the fashion compatibility, where $N!$ denotes factorial N . In this section, we further investigate all possible sequences under our problem settings on the performance of OutfitGAN in terms of the fashion compatibility metric, FCTS. Considering that the used collocation classification module is based on Bi-LSTM, here we only have $\frac{N!}{2}$ possible unique orders in our task. We implemented different variants of OutfitGAN with all possible orders which were trained to validate the effectiveness of our pre-defined order, i.e., [upper, bag, lower, shoes]. Additional eleven versions of OutfitGAN were carried out in total. As shown in Fig. 12, we observe that our pre-defined order used in Section IV-A obtains the best fashion compatibility in comparison to other possible orders, despite that 'USLB' and 'UBLS' have the same FCTS values (see Fig. 12). Moreover, most models with other orders show relatively decent performance on fashion compatibility. This may be ascribed to the fact that there only exist four fashion items in an outfit in our current research and Bi-LSTM may have sufficient ability to build the compatibility relation among fashion items in the same outfit even if we provide an arbitrary order.

H. Limitation

Although the proposed method achieves state-of-the-art performance in outfit generation, OutfitGAN still has certain limitations at the current stage. Firstly, an outfit includes N fashion items, where $N = 4$ in our implementation. During the process of our dataset construction, we crawled outfits which are composed by fashion experts from Polyvore.com. To cover as many fashion items as possible, we define our outfit generation on four commonly used items by women – upper, bag, lower, and shoes. It is possible to build a large-scale dataset with more kinds of fashion items when more relevant fashion compatibility-related resources are available in the future. Secondly, for an outfit with N fashion items, OutfitGAN needs $(N - 1)$ item generators to synthesize the

complementary fashion items based on the given item. The number of item generators increases with the number of fashion items, indicating that the computational complexity of OutfitGAN is $O(N)$. Actually, even if a shared item generator is used for synthesizing all kinds of fashion items, the model needs $(N - 1)$ feedforward times for synthesizing $(N - 1)$ fashion items. Moreover, once the outfit generator is trained, an arbitrary number of item generators can be selected for synthesizing desired fashion items. It is worth noting that every item generator is able to be used separately for synthesizing its targeted fashion item. For synthesizing multiple fashion items with lightweight models in more real-life applications, we leave this for future work.

V. CONCLUSION

This paper has presented an outfit generation framework with the aim of synthesizing photo-realistic fashion items that are compatible with a given item. In particular, in order to exploit the harmonious elements and styles shared in a compatible outfit, OutfitGAN uses a mask-guided strategy for image synthesis which can overcome the issue of spatial misalignment that arises in general image-to-image translation tasks. OutfitGAN consists of an outfit generator, an outfit discriminator and a CCM. An SAM is adopted to capture the mapping relationships between the extant fashion items and the synthesized ones, in order to improve the quality of fashion synthesis. A CCM is developed to improve the compatibility of the synthesized outfits. To evaluate the effectiveness of the proposed model, we constructed a large-scale dataset that consists of 20,000 outfits. Extensive experimental results show that our method can achieve state-of-the-art performance on the task of outfit generation and outperforms other methods. In the future, we plan to concentrate on synthesizing outfits with finer detail, and to use other reference information such as textual descriptions in a multi-modal manner to guide the process of outfit generation.

REFERENCES

- [1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. of NeurIPS*, 2014, pp. 2672–2680.
- [2] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. of CVPR*, 2017, pp. 5967–5976.
- [3] T. Wang, M. Liu, J. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proc. of CVPR*, 2018, pp. 8798–8807.
- [4] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. of ICCV*, 2017, pp. 2242–2251.
- [5] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proc. of ECCV*, 2018.
- [6] H.-Y. Lee, H.-Y. Tseng, Q. Mao, J.-B. Huang, Y.-D. Lu, M. K. Singh, and M.-H. Yang, "DRIT++: Diverse image-to-image translation via disentangled representations," *International Journal of Computer Vision*, pp. 1–16, 2020.
- [7] Y. Choi, Y. Uh, J. Yoo, and J. Ha, "StarGAN v2: Diverse image synthesis for multiple domains," in *Proc. of CVPR*, 2020, pp. 8185–8194.
- [8] L. Liu, H. Zhang, Y. Ji, and Q. J. Wu, "Toward AI fashion design: An Attribute-GAN model for clothing match," *Neurocomputing*, vol. 341, pp. 156–167, 2019.
- [9] L. Liu, H. Zhang, X. Xu, Z. Zhang, and S. Yan, "Collocating clothes with generative adversarial networks cosupervised by categories and attributes: A multidiscriminator framework," *IEEE Trans. Neural Netw. and Learn. Syst.*, vol. 31, no. 9, pp. 3540–3554, 2020.
- [10] C. Yu, Y. Hu, Y. Chen, and B. Zeng, "Personalized fashion design," in *Proc. of ICCV*, 2019, pp. 9045–9054.
- [11] X. Wang, R. B. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. of CVPR*, 2018, pp. 7794–7803.
- [12] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. of CVPR*, 2017, pp. 105–114.
- [13] S. Zhu, S. Fidler, R. Urtasun, D. Lin, and C. C. Loy, "Be your own prada: Fashion synthesis with structural coherence," in *Proc. of ICCV*, 2017, pp. 1689–1697.
- [14] X. Han, Z. Wu, Z. Wu, R. Yu, and L. S. Davis, "VITON: An image-based virtual try-on network," in *Proc. of CVPR*, 2018, pp. 7543–7552.
- [15] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. on Signal Process.*, vol. 45, no. 11, pp. 2673–2681, 1997.
- [16] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proc. of CVPR*, 2016, pp. 2414–2423.
- [17] H. Zhang, X. Wang, L. Liu, D. Zhou, and Z. Zhang, "Warpcloutngout: A stepwise framework for clothes translation from the human body to tiled images," *IEEE MultiMedia*, vol. 27, no. 4, pp. 58–68, 2020.
- [18] J. J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, "Image-based recommendations on styles and substitutes," in *Proc. of SIGIR*, 2015, pp. 43–52.
- [19] A. Veit, B. Kovacs, S. Bell, J. J. McAuley, K. Bala, and S. J. Belongie, "Learning visual clothing style with heterogeneous dyadic co-occurrences," in *Proc. of ICCV*, 2015, pp. 4642–4650.
- [20] M. I. Vasileva, B. A. Plummer, K. Dusad, S. Rajpal, R. Kumar, and D. Forsyth, "Learning type-aware embeddings for fashion compatibility," in *Proc. of ECCV*, 2018, pp. 390–405.
- [21] X. Han, Z. Wu, Y. Jiang, and L. S. Davis, "Learning fashion compatibility with bidirectional lstms," in *Proc. of ACM MM*, 2017, pp. 1078–1086.
- [22] Z. Cui, Z. Li, S. Wu, X. Zhang, and L. Wang, "Dressing as a whole: Outfit compatibility learning based on node-wise graph neural networks," in *Proc. of WWW*, 2019, pp. 307–317.
- [23] X. Li, X. Wang, X. He, L. Chen, J. Xiao, and T. Chua, "Hierarchical fashion graph network for personalized outfit recommendation," in *Proc. of SIGIR*, 2020, pp. 159–168.
- [24] W.-H. Cheng, S. Song, C.-Y. Chen, S. Chusnul Hidayati, and J. Liu, "Fashion meets computer vision: A survey," *arXiv preprint*, p. arXiv:2003.13988, 2020.
- [25] B. Wang, H. Zheng, X. Liang, Y. Chen, L. Lin, and M. Yang, "Toward characteristic-preserving image-based virtual try-on network," in *Proc. of ECCV*, 2018, pp. 589–604.
- [26] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. V. Gool, "Pose guided person image generation," in *Proc. of NeurIPS*, 2017, pp. 406–416.
- [27] A. Siarohin, E. Sangineto, S. Lathuilière, and N. Sebe, "Deformable GANs for pose-based human image generation," in *Proc. of CVPR*, 2018, pp. 3408–3416.
- [28] I. Rocco, R. Arandjelovic, and J. Sivic, "Convolutional neural network architecture for geometric matching," in *Proc. of CVPR*, 2017, pp. 39–48.
- [29] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*, 2012.
- [30] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. of ICCV*, 2017, pp. 2813–2821.
- [31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
- [32] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "DeepFashion: Powering robust clothes recognition and retrieval with rich annotations," in *Proc. of CVPR*, 2016, pp. 1096–1104.
- [33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of CVPR*, 2016, pp. 770–778.
- [34] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.
- [35] X. Wang, B. Wu, and Y. Zhong, "Outfit compatibility prediction and diagnosis with multi-layered comparison network," in *Proc. of ACM MM*, 2019, pp. 329–337.
- [36] A. Yu and K. Grauman, "Fine-grained visual comparisons with local learning," in *Proc. of CVPR*, 2014, pp. 192–199.

- [37] X. Song, F. Feng, J. Liu, Z. Li, L. Nie, and J. Ma, "Neurostylist: Neural compatibility modeling for clothing matching," in *Proc. of ACM MM*, 2017, pp. 753–761.
- [38] X. Song, X. Han, Y. Li, J. Chen, X. Xu, and L. Nie, "GP-BPR: personalized compatibility modeling for clothing matching," in *Proc. of ACM MM*, 2019, pp. 320–328.
- [39] J. Liu, Q. Hou, M. Cheng, J. Feng, and J. Jiang, "A simple pooling-based design for real-time salient object detection," in *Proc. of CVPR*, 2019, pp. 3917–3926.
- [40] M. D. Zeiler, "AdaDelta: An adaptive learning rate method," *arXiv preprint*, 2012.
- [41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of ICLR*, 2015.
- [42] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.
- [43] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. of CVPR*, 2018, pp. 586–595.
- [44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. of NeurIPS*, 2012, pp. 1106–1114.
- [45] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. of CVPR*, 2016, pp. 2818–2826.
- [46] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. K. Singh, and M.-H. Yang, "Diverse image-to-image translation via disentangled representations," in *Proc. of ECCV*, 2018.
- [47] Y. Choi, M. Choi, M. Kim, J. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proc. of CVPR*, 2018, pp. 8789–8797.