# Dress Code: High-Resolution Multi-Category Virtual Try-On

Davide Morelli[1], Matteo Fincato[1], Marcella Cornia[1], Federico Landi[1,*], Fabio Cesari[2], and Rita Cucchiara[1]

[1] University of Modena and Reggio Emilia, Italy
{name.surname}@unimore.it
[2] YOOX NET-A-PORTER, Italy
{name.surname}@ynap.com

**Abstract.** Image-based virtual try-on strives to transfer the appearance of a clothing item onto the image of a target person. Prior work focuses mainly on upper-body clothes (*e.g.* t-shirts, shirts, and tops) and neglects full-body or lower-body items. This shortcoming arises from a main factor: current publicly available datasets for image-based virtual try-on do not account for this variety, thus limiting progress in the field. To address this deficiency, we introduce Dress Code, which contains images of multi-category clothes. Dress Code is more than 3× larger than publicly available datasets for image-based virtual try-on and features high-resolution paired images (1024 × 768) with front-view, full-body reference models. To generate HD try-on images with high visual quality and rich in details, we propose to learn fine-grained discriminating features. Specifically, we leverage a semantic-aware discriminator that makes predictions at pixel-level instead of image- or patch-level. Extensive experimental evaluation demonstrates that the proposed approach surpasses the baselines and state-of-the-art competitors in terms of visual quality and quantitative results. The Dress Code dataset is publicly available at https://github.com/aimagelab/dress-code.

**Keywords:** Dress Code Dataset, Virtual Try-On, Image Synthesis.

## 1 Introduction

Clothes, fashion, and style play a fundamental role in our daily life and allow people to communicate and express themselves freely and directly. With the advent of e-commerce, the variety and availability of online garments have become increasingly overwhelming for the customer. Consequently, user-oriented applications such as virtual try-on, in both 2D [4,12,13,42] and 3D [29,36,46,48] settings, are increasingly important for online shopping, helping fashion companies to tailor the e-commerce experience and maximize customer satisfaction. Image-based virtual try-on aims at synthesizing an image of a reference person wearing a given try-on garment. In this task, while virtually changing clothing,

*Now at Huawei Technologies, Amsterdam Research Center, the Netherlands.


**Fig. 1.** Differently from existing publicly available datasets for virtual try-on, Dress Code features different garments, also belonging to lower-body and full-body categories, and high-resolution images.

the person's intrinsic information such as body shape and pose should not be modified. Also, the try-on garment is expected to properly fit the person's body while maintaining its original texture. All these elements make virtual try-on a very active and challenging research topic.

Due to the strategic role that virtual try-on plays in e-commerce, many rich and potentially valuable datasets are proprietary and not publicly available to the research community [23,24,30,43]. Public datasets, instead, either do not contain paired images of models and garments or feature a very limited number of images [13]. Moreover, the overall image resolution is low (mostly $256 \times 192$). Unfortunately, these drawbacks slow down progress in the field. In this paper, we present *Dress Code*: a new dataset of high-resolution images ($1024 \times 768$) containing more than 50k image pairs of try-on garments and corresponding catalog images where each item is worn by a model. This makes Dress Code more than $3\times$ larger than VITON [13], the most common benchmark for virtual try-on. Differently from existing publicly available datasets, which contain only upper-body clothes, Dress Code features upper-body, lower-body, and full-body clothes, as well as full-body images of human models (Fig. 1, *left*).

Current off-the-shelf architectures for virtual try-on are not optimized to work with clothes belonging to different macro-categories (*i.e.* upper-body, lower-body, and full-body clothes). In fact, this would require learning the correspondences between a particular garment class and the portion of the body involved in the try-on phase. For instance, trousers should match the legs pose, while a dress should match the pose of the entire body, from shoulders to hips and eventually knees. In this paper, we design an image-based virtual try-on architecture that can anchor the given garment to the right portion of the body. As a consequence, it is possible to perform a "complete" try-on over a given person by selecting different garments (Fig 1, *right*). In order to produce high-quality re-

sults rich in visual details, we introduce a parser-based discriminator [26,31,38]. This component can increase the realism and visual quality of the results by learning an internal representation of the semantics of generated images, which is usually neglected by standard discriminator architectures [17,41]. This component works at pixel-level and predicts not only real/generated labels but also the semantic classes for each image pixel. We validate the effectiveness of the proposed approach by testing its performance on both our newly collected dataset and on the most widely used dataset for the task (*i.e.* VITON [13]).

The contributions of this paper are summarized as follows: (1) We introduce Dress Code, a novel dataset for the virtual try-on task. To the best of our knowledge, it is the first publicly available dataset featuring lower-body and full-body clothes. As a plus, all images have high resolution ($1024 \times 768$). (2) To address the challenges of generating high-quality images, we leverage a Pixel-level Semantic-Aware Discriminator (PSAD) that enhances the realism of try-on images. (3) With the aim of presenting a comprehensive benchmark on our newly collected dataset, we train and test up to nine state-of-the-art virtual try-on approaches and three different baselines. (4) Extensive experiments demonstrate that the proposed approach outperforms the competitors and other state-of-the-art architectures both quantitatively and qualitatively, also considering different image resolutions and a multi-garment setting.

## 2 Related Work

The first popular image-based virtual try-on model [13] builds upon a coarse-to-fine network. First, it predicts a coarse image of the reference person wearing the try-on garment, then it refines the texture and shape of the previously obtained result. Wang *et al.* [40] overcame the lack of shape-context precision (*i.e.* bad alignment between clothes and body shape) and proposed a geometric transformation module to learn the parameters of a thin-plate spline transformation to warp the input garment. Following this work, many different solutions were proposed to enhance the geometric transformation of the try-on garment. For instance, Liu *et al.* [27] integrated a multi-scale patch adversarial loss to increase the realism in the warping phase. Minar *et al.* [28] and Yang *et al.* [42] proposed different regularization techniques to stabilize the warping process during training. Instead, other works [8,24] focused on the design of additional projections of the input garment to preserve details and textures of input clothing items.

Another line of work focuses on the improvement of the generation phase of final try-on images [4,7,9,14,18,19]. Among them, Issenuth *et al.* [18] introduced a teacher-student approach: the teacher learns to generate the try-on results using image pairs (sampled from a paired dataset) and then teaches the student how to deal with unpaired data. This paradigm was further improved in [10] with a student-tutor-teacher architecture where the network is trained in a parser-free way, exploiting both the tutor guidance and the teacher supervision. On a different line, Ge *et al.* [9] presented a self-supervised trainable network to reframe the virtual try-on task as clothes warping, skin synthesis, and image composition using a cycle-consistent framework.

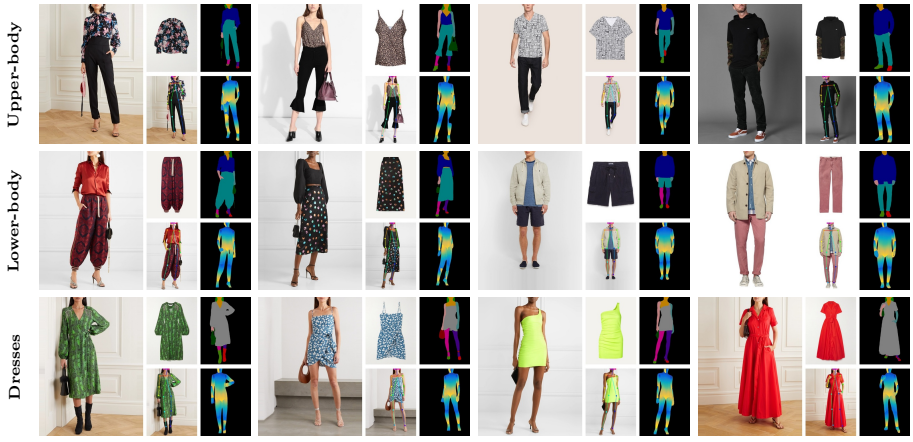


**Fig. 2.** Sample image pairs from the Dress Code dataset with pose keypoints, dense poses, and segmentation masks of human bodies.

A third direction of research estimates the person semantic layout to improve the visual quality of generated images [12,20,42,45]. In this context, Jandial *et al.* [20] proposed to generate a conditional segmentation mask to handle occlusions and complex body poses effectively. Very recently, Chen *et al.* [3] introduced a new scenario where the try-on results are synthesized in sequential poses with spatio-temporal smoothness. Using a recurrent approach, Cui *et al.* [5] designed a person generation framework for pose transfer, virtual try-on, and other fashion-related tasks. While almost all these methods generate low-resolution results, a limited subset of works focuses on the generation of higher-resolution images instead. Unfortunately, these works employ non-public datasets to train and test the proposed architectures [24,43].

## 3 Dress Code Dataset

Publicly available datasets for virtual try-on are often limited by one or more factors such as lack of variety, small size, low-resolution images, privacy concerns, or from the fact of being proprietary. We identify four main desiderata that the ideal dataset for virtual try-on should possess: (1) it should be publicly available for research purposes; (2) it should have corresponding images of clothes and reference human models wearing them (*i.e.* the dataset should consist of paired images); (3) it should contain high-resolution images and (4) clothes belonging to different macro-categories (tops and t-shirts belong to the upper-body category, while skirts and trousers are examples of lower-body clothes and dresses are full-body garments). In addition to this, a dataset for virtual try-on with a large number of images is more preferable than other datasets with the same overall characteristics but smaller size. By looking at Table 1, we can see that Dress Code complies with all of the above desiderata, while featuring more than three times the number of images of VITON [13]. To the best of our knowledge,

**Table 1.** Comparison between Dress Code and the most widely used datasets for virtual try-on and other related tasks.

| Dataset | Public | Multi-Category | # Images | # Garments | Resolution |
|---|---|---|---|---|---|
| O-VITON [30] | ✗ | ✓ | 52,000 | - | $512 \times 256$ |
| TryOnGAN [23] | ✗ | ✓ | 105,000 | - | $512 \times 512$ |
| Revery AI [24] | ✗ | ✓ | 642,000 | 321,000 | $512 \times 512$ |
| Zalando [43] | ✗ | ✓ | 1,520,000 | 1,140,000 | $1024 \times 768$ |
| VITON-HD [4] | ✓ | ✗ | 27,358 | 13,679 | $1024 \times 768$ |
| FashionOn [16] | ✓ | ✗ | 32,685 | 10,895 | $288 \times 192$ |
| DeepFashion [27] | ✓ | ✗ | 33,849 | 11,283 | $288 \times 192$ |
| MVP [6] | ✓ | ✗ | 49,211 | 13,524 | $256 \times 192$ |
| FashionTryOn [47] | ✓ | ✗ | 86,142 | 28,714 | $256 \times 192$ |
| LookBook [44] | ✓ | ✓ | 84,748 | 9,732 | $256 \times 192$ |
| VITON [13] | ✓ | ✗ | 32,506 | 16,253 | $256 \times 192$ |
| **Dress Code** | ✓ | ✓ | 107,584 | 53,792 | $1024 \times 768$ |

this is the first publicly available virtual try-on dataset comprising multiple macro-categories and high-resolution image pairs. Additionally, it is the biggest available dataset for this task at present, as it includes more than 100k images evenly split between garments and human reference models.

**Image collection and annotation.** All images are collected from different fashion catalogs of YOOX NET-A-PORTER containing both casual clothes and luxury garments. To create a coarse version of the dataset, we select images of different garment categories for a total of about 250k fashion items, each containing 2-5 images of different views of the same product. Since our goal is to create a dataset for virtual try-on and not all fashion items were released with the image pair required to perform the task, we select only those products where the front-view image of the garment and the corresponding full figure of the model are available. We exploit an automatic selection procedure: we only store the clothing items for which at least one image with the entire body of the model is present, using a human pose estimator to verify the presence of the neck and feet joints. In this way, all products without valid image pairs are automatically discarded. After this automatic stage, we manually validate all images and remove the remaining invalid image pairs, including those pairs for which the garment of interest is mostly occluded by other overlapping clothes. Finally, we group the annotated products into three categories: upper-body clothes (composed of tops, t-shirts, shirts, sweatshirts, and sweaters), lower-body clothes (composed of skirts, trousers, shorts, and leggings), and dresses. Overall, the dataset is composed of 53,795 image pairs: 15,366 pairs for upper-body clothes, 8,951 pairs for lower-body clothes, and 29,478 pairs for dresses.

Existing datasets for virtual try-on show the face and physiognomy of the human models. While this feature is not essential for virtual try-on, it also causes potential privacy issues. To preserve the models' identity, we partially anonymize all images by cutting them at the level of the nose. In this way, information about the physiognomy of the human models is not available. To further enrich our dataset, we compute the joint coordinates, the dense pose, and

the segmentation mask for the human parsing of each model. In particular, we use OpenPose [2] to extract 18 keypoints for each human body, DensePose [11] to compute the dense pose of each reference model, and SCHP [25] to generate a segmentation mask representing the human parsing of model body parts and clothing items. Sample human model and garment pairs from our dataset with the corresponding additional information are shown in Figure 2.

**Comparison with other datasets.** Table 1 reports the main characteristics of the Dress Code dataset in comparison with existing datasets for virtual try-on and fashion-related tasks. Although some proprietary and non-publicly available datasets have also been used [23,24,43], almost all virtual try-on literature [10,40,42] employs the VITON dataset [13] to train the proposed models and perform experiments. We believe that the use of Dress Code could greatly increase the performance and applicability of virtual try-on solutions. In fact, when comparing Dress Code with the VITON dataset, it can be seen that our dataset jointly features a larger number of image pairs (*i.e.* 53,792 vs 16,253 of the VITON dataset), a wider variety of clothing items (*i.e.* VITON only contains t-shirts and upper-body clothes), a greater variance in model images (*i.e.* Dress Code images can contain challenging backgrounds, accessories like bags, scarfs, and belts, and both male and female models), and a greater image resolution (*i.e.* $1024 \times 768$ vs $256 \times 192$ of VITON images).

## 4 Virtual Try-On with Pixel-level Semantics

Architectures for virtual try-on address the task of generating a new image of the reference person wearing the input try-on garment. Given the generative nature of this task, virtual try-on methods are usually trained using adversarial losses that typically work at image- or patch-level and do not consider the semantics of generated images. Differently from previous works, we introduce a Pixel-level Semantic Aware Discriminator (PSAD) that can build an internal representation of each semantic class and increase the realism of generated images. In this section, we first describe the baseline generative architecture and then detail PSAD which improves the visual quality and overall performance.

### 4.1 Baseline Architecture

To tackle the virtual try-on task, we begin by building a baseline generative architecture that performs three main operations: (1) garment warping, (2) human parsing estimation, and finally (3) try-on. First, the warping module employs geometric transformations to create a warped version of the input try-on garment. Then, the human parsing estimation module predicts a semantic map for the reference person. Last, the try-on module generates the image of the reference person wearing the selected garment. Our baseline model is shown in Fig. 3 and detailed in the following.

**Network Inputs and Notation.** Here, we define the different inputs for our network and related notation. We denote with $c$ an image depicting a clothing

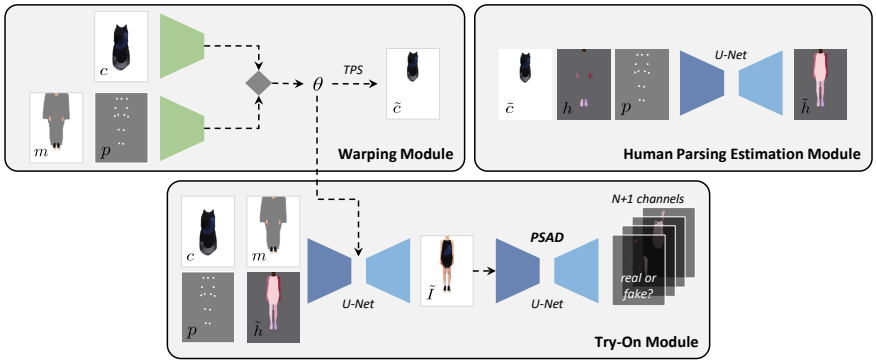


**Fig. 3.** Overview of the proposed architecture.

item alone. This image contains information about the shape, texture, and color of the try-on garment. Details about the reference human model come in different forms: $p$, $m$, and $h$ are three images containing respectively the pose of that person ($p$), the background and appearance of the portions of the body and outfit that are not involved in the try-on phase such as hands, feet, and part of the face ($m$), and the semantic labels of each of these regions ($h$). Our architecture can employ two different representations for the body pose: keypoints or dense pose [11]. In this section, as well as in Fig. 3, we consider the case of pose keypoints. However, it is possible to switch and use dense pose representation by accounting for the different number of channels. Finally, we denote with $I$ the image depicting the person described by $(p, m, h)$ wearing the garment $c$.

**Warping Module.** The warping module transforms the input try-on garment $c$ into a warped image of the same item that matches the body pose and shape expressed respectively by $p$ and $m$. As warping function we use a thin-plate spline (TPS) geometric transformation [33], which is commonly used in virtual try-on models [8,40,42]. Inside this module, we aim to learn the correspondence between the inputs $(c, p, m)$ and the set $\theta$ of parameters to be used in the TPS transformation. Specifically, we follow the warping module proposed in [40] and compute a correlation map between the encoded representations of the try-on garment $c$ and the pose and cloth-agnostic person representation ($p$ and $m$), obtained using two separate convolutional networks. Then, we predict the spatial transformation parameters $\theta$ corresponding to the $(x, y)$-coordinate offsets of TPS anchor points. These parameters are used in the TPS function to generate the warped version $\tilde{c}$ of the input try-on garment:

$$\tilde{c} = \mathrm{TPS}_\theta(c). \tag{1}$$

To train this network, we minimize the $L_1$ distance between the warped result $\tilde{c}$ and the cropped version of the garment $\hat{c}$ obtained from the ground-truth image $I$. In addition, to reduce visible distortions in the warped result, we employ the second-order difference constraint introduced in [42]. Overall, the loss function

used to train this module is defined as follows:

$$\mathcal{L}_{warp} = \|\tilde{c} - \hat{c}\|_1 + \lambda_{const}\mathcal{L}_{const}, \tag{2}$$

where $\mathcal{L}_{const}$ is the second-order difference constraint and $\lambda_{const}$ is used to weigh the constraint loss function [42].

**Human Parsing Estimation Module.** This module, based on the U-Net architecture [34], takes as input a concatenation of the warped try-on clothing item $\tilde{c}$ (Eq. 1), the pose image $p$, and the masked semantic image $h$, and predicts the complete semantic map $\tilde{h}$ containing the human parsing for the reference person:

$$\tilde{h} = \text{U-Net}_{\mu}(c, h, p), \tag{3}$$

where $\mu$ denotes the set of learnable weights in the network. Every pixel of $\tilde{h}$ contains a probability distribution over 18 semantic classes, which include *left/right arm*, *left/right leg*, *background*, *dress*, *shirt*, *skirt*, *neck*, and so on. We optimize the set of weights $\mu$ of this module using a pixel-wise cross-entropy loss between the generated semantic map $\tilde{h}$ and the ground-truth $\hat{h}$.

**Try-On Module.** This module produces the image $\tilde{I}$ depicting the reference person described by the triple $(p, m, \tilde{h})$ wearing the input try-on clothing item $c$. To this end, we employ a U-Net model [34] which takes as input $c$, $p$, $m$, and the one-hot semantic image obtained by taking the pixel-wise argmax of $\tilde{h}$. During training, instead, we employ the ground-truth human parsing $\hat{h}$. This artifice helps to stabilize training and brings better results.

At this stage, we take advantage of the previously learned geometric transformation $\text{TPS}_\theta$ to facilitate the generation of $\tilde{I}$. Specifically, we employ a modified version of the U-Net model featuring a two-branch encoder that generates two different representations for the try-on garment $c$ and the reference person, and a decoder that combines these two representations to generate the final image $\tilde{I}$. The input of the first branch is the original try-on garment $c$, while the input of the second branch is a concatenation of the reference model and corresponding additional information. In the first branch, we apply the previously learned transformation $\text{TPS}_\theta$. Thus, the skip connections, which are typical of the U-Net design, no longer perform an identity mapping, but compute:

$$E_i(c) = \text{TPS}_\theta(E_i(c)), \tag{4}$$

where $E_i(c)$ are the features extracted from the $i^{th}$ layer of the U-Net encoder.

During training, we exploit a combination of three different loss functions: an $L_1$ loss between the generated image $\tilde{I}$ and the ground-truth image $I$, a perceptual loss $\mathcal{L}_{vgg}$, also know as VGG loss [21], to compute the difference between the feature maps of $\tilde{I}$ and $I$ extracted with a VGG-19 [39], and the adversarial loss $\mathcal{L}_{adv}$:

$$\mathcal{L}_{try\text{-}on} = \left\|\tilde{I} - I\right\|_1 + \mathcal{L}_{vgg} + \lambda_{adv}\mathcal{L}_{adv}, \tag{5}$$

where $\lambda_{adv}$ is used to weigh the adversarial loss. For a formulation of $\mathcal{L}_{adv}$ using our proposed Pixel-level Parsing-Aware Discriminator (PSAD), we refer the reader to the next subsection (Eq. 6).

### 4.2 Pixel-level Semantic-Aware Discriminator

Virtual try-on models are usually enriched with adversarial training strategies to increase the realism of generated images. However, most of the existing discriminator architectures work at image- or patch-level, thus neglecting the semantics of generated images. To address this issue, we draw inspiration from semantic image synthesis literature [26,31,38] and train our discriminator to predict the semantic class of each pixel using generated and ground-truth images as fake and real examples respectively. In this way, the discriminator can learn an internal representation of each semantic class (*e.g.* tops, skirts, body) and force the generator to improve the quality of synthesized images.

The discriminator is built upon the U-Net model [34], which is used as an encoder-decoder segmentation network. For each pixel of the input image, the discriminator predicts its semantic class and an additional label (real or generated). Overall, we have $N + 1$ classes (*i.e.* $N$ classes corresponding to the ground-truth semantic classes plus one class for fake pixels) and thus we train the discriminator with a $(N + 1)$-class pixel-wise cross-entropy loss. In this way, the discriminator prediction shifts from a patch-level classification, typical of standard patch-based discriminators [17,41], to a per-pixel class-level prediction.

Due to the unbalanced nature of the semantic classes, we weigh the loss class-wise using the inverse pixel frequency of each class. Formally, the loss function used to train this Pixel-level Parsing-Aware Discriminator (PSAD) can be defined as follows:

$$\begin{aligned}\mathcal{L}_{adv} = &-\mathbb{E}_{(I,\hat{h})}\left[\sum_{k=1}^{N} w_k \sum_{i,j}^{H\times W} \hat{h}_{i,j,k} \log D(I)_{i,j,k}\right] \\ &-\mathbb{E}_{(p,m,c,\hat{h})}\left[\sum_{i,j}^{H\times W} \log D(G(p,m,c,\hat{h}))_{i,j,k=N+1}\right],\end{aligned} \tag{6}$$

where $I$ is the real image, $\hat{h}$ is the ground-truth human parsing, $p$ is the model pose, $m$ and $c$ are respectively the person representation and the try-on garment given as input to the generator, and $w_k$ is the class inverse pixel frequency.

## 5 Experiments

### 5.1 Experimental Setup

**Datasets.** First, we perform experiments on our newly proposed dataset, Dress Code, using 48,392 image pairs as training set and the remaining as test set (*i.e.* 5,400 pairs, 1,800 for each category). During evaluation, image pairs of the test set are rearranged to form unpaired pairs of clothes and front-view models. On Dress Code, we use three different image resolutions: $256 \times 192$ (*i.e.* the one typical used by virtual try-on models), $512 \times 384$, and $1024 \times 768$. Following our experiments on Dress Code, we evaluate our model on the standard VITON

**Table 2.** Try-on results on the Dress Code test set. Top-1 results are highlighted in bold, underlined denotes second-best.

| | Upper-body | | | Lower-body | | | Dresses | | | All | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Model** | **SSIM ↑** | **FID ↓** | **KID ↓** | **SSIM ↑** | **FID ↓** | **KID ↓** | **SSIM ↑** | **FID ↓** | **KID ↓** | **SSIM ↑** | **FID ↓** | **KID ↓** | **IS ↑** |
| CP-VTON [40] | 0.812 | 46.99 | 3.236 | 0.782 | 54.66 | 3.656 | 0.816 | 34.95 | 1.759 | 0.803 | 35.16 | 2.245 | 2.817 |
| CP-VTON+ [28] | 0.863 | 28.93 | 1.856 | 0.819 | 41.37 | 2.506 | 0.826 | 32.27 | 1.630 | 0.836 | 25.19 | 1.586 | 3.002 |
| CIT [32] | 0.860 | 26.41 | 1.496 | 0.834 | 31.77 | 1.753 | 0.810 | 35.58 | 1.734 | 0.835 | 21.99 | 1.313 | 3.022 |
| CP-VTON† [40] | 0.898 | 23.03 | 1.338 | 0.887 | 26.96 | 1.409 | 0.838 | 33.04 | 1.668 | 0.874 | 18.99 | 1.117 | **3.058** |
| CIT† [32] | 0.912 | 17.66 | 0.895 | 0.896 | 23.15 | 1.005 | 0.855 | 23.87 | 0.969 | 0.888 | 13.97 | 0.761 | 3.014 |
| VITON-GT [8] | 0.922 | 18.90 | 0.994 | 0.916 | 21.88 | 0.949 | 0.864 | 29.45 | 1.402 | 0.899 | 13.80 | 0.711 | 3.042 |
| WUTON [18] | 0.924 | 17.74 | 0.893 | 0.918 | 22.57 | 1.008 | 0.866 | 28.93 | 1.304 | 0.902 | 13.28 | 0.771 | 3.005 |
| ACGPN [42] | 0.889 | 19.03 | 1.028 | 0.874 | 24.46 | 1.208 | 0.845 | 22.42 | 0.944 | 0.868 | 13.79 | 0.818 | 2.924 |
| PF-AFN [10] | 0.918 | 19.03 | 1.237 | 0.907 | 23.43 | 1.018 | 0.869 | 21.94 | 0.723 | 0.902 | 14.36 | 0.756 | 3.023 |
| *Dense Pose* | | | | | | | | | | | | | |
| **Ours (Patch)** | 0.930 | 18.21 | 0.929 | 0.922 | 21.95 | 0.992 | 0.875 | 21.84 | 0.768 | 0.908 | 12.82 | 0.692 | 3.042 |
| **Ours (PSAD)** | 0.928 | 17.18 | 0.793 | 0.921 | 20.49 | 0.896 | 0.872 | **19.63** | **0.635** | 0.906 | 11.47 | 0.619 | 2.987 |
| *Pose Keypoints* | | | | | | | | | | | | | |
| **Ours (NoDisc)** | 0.926 | 18.84 | 0.943 | 0.915 | 22.48 | 0.943 | 0.873 | 23.71 | 0.937 | 0.907 | 13.51 | 0.704 | 3.041 |
| **Ours (Binary)** | 0.925 | 18.39 | 0.872 | 0.914 | 22.52 | 0.98 | 0.871 | 22.35 | 0.816 | 0.906 | 12.89 | 0.645 | 3.017 |
| **Ours (Patch)** | **0.931** | 18.40 | 0.841 | **0.923** | 21.46 | 0.955 | **0.876** | 21.94 | 0.814 | **0.909** | 12.53 | 0.666 | 3.043 |
| **Ours (PSAD)** | 0.928 | **17.04** | **0.762** | 0.921 | **20.04** | **0.795** | 0.872 | 20.98 | 0.672 | 0.906 | **11.40** | **0.570** | 3.036 |

dataset [13], composed of 16,253 image pairs. We employ this dataset to evaluate our solution in comparison with other state-of-the-art architectures on a widely-employed benchmark. In VITON, all images have a resolution of 256×192 and are divided into training and test set with 14,221 and 2,032 image pairs respectively.

**Evaluation metrics.** Following recent literature, we employ evaluation metrics that either compare the generated images with the corresponding ground-truths, *i.e.* Structural Similarity (SSIM), or measure the realism and the visual quality of the generation, *i.e.* Frechét Inception Distance (FID) [15], Kernel Inception Distance (KID) [1], and Inception Score (IS) [35].

**Training.** We train the three components of our model separately. Specifically, we first train the warping module and then the human parsing estimation module for 100k and 50k iterations respectively. Finally, we train the try-on module for other 150k iterations. We set the weight of the second-order difference constraint $\lambda_{const}$ to 0.01 and the weight of the adversarial loss $\lambda_{adv}$ to 0.1. All experiments are performed using Adam [22] as optimizer and a learning rate equal to $10^{-4}$. More details on the architecture and training stage are reported in the supplementary material.

### 5.2 Experiments on Dress Code

**Baselines and Competitors.** In this set of experiments, we compare with CP-VTON [40], CP-VTON+ [28], CIT [32], VITON-GT [8], WUTON [18], ACGPN [42], and PF-AFN [10], that we re-train from scratch on our dataset using source codes provided by the authors, when available, or our re-implementations. In addition to these methods, we implement an improved version of [40] (*i.e.* CP-VTON†) and of [32] (*i.e.* CIT†) in which we use, as an additional input to the model, the person representation $m$. To validate the effectiveness of the Pixel-level Semantic Aware Discriminator (PSAD), we also test a model trained with a patch-based discriminator [17] (Patch), a model trained


**Fig. 4.** Qualitative comparison between Patch and PSAD.


**Fig. 5.** Sample try-on results on the Dress Code test set.

by removing in our discriminator the $N$ semantic channels and only keeping the real/fake one [37] (Binary), and a baseline trained without the adversarial loss (NoDisc).

**Low-Resolution Results and Ablative Analysis.** In this experiment, we compare our complete model (PSAD) with state-of-the-art architectures for virtual try-on and with the Patch, Binary, and NoDisc baselines. For these comparisons, we consider the standard resolution for virtual try-on ($256 \times 192$). In Table 2, we report numerical results on the Dress Code test set. As it can be seen, our model obtains better results than competitors on all clothing categories in terms of almost all considered evaluation metrics. Quantitative results also confirm the effectiveness of PSAD in comparison with a pixel-level discriminator without semantics and with a standard patch-based discriminator, especially in terms of the realism of the generated images (*i.e.* FID and KID). PSAD is second to the Patch model only in terms of SSIM, and by a very limited margin. All discriminator-based configurations outperform the NoDisc baseline, thus showing the importance of incorporating a discriminator in a virtual try-on architecture. We also test the effect of using different representations for the body pose (human keypoints and dense pose). When comparing the two versions of our model, we find out that using dense pose helps to deal with full-body clothes

**Table 3.** High-resolution results on the Dress Code test set.

| | $512 \times 384$ | | | | $1024 \times 768$ | | | |
|---|---|---|---|---|---|---|---|---|
| **Model** | **SSIM ↑** | **FID ↓** | **KID ↓** | **IS ↑** | **SSIM ↑** | **FID ↓** | **KID ↓** | **IS ↑** |
| CP-VTON [40] | 0.831 | 29.24 | 1.671 | 3.096 | 0.853 | 36.68 | 2.379 | 3.155 |
| CP-VTON$^\dagger$ [40] | 0.896 | 10.08 | 0.425 | 3.277 | 0.912 | 9.96 | 0.338 | 3.300 |
| **Ours (Patch)** | **0.923** | 9.44 | **0.246** | 3.310 | **0.922** | 9.99 | 0.370 | 3.344 |
| **Ours (PSAD)** | 0.916 | **7.27** | 0.394 | **3.320** | 0.919 | **7.70** | **0.236** | **3.357** |

**Table 4.** Multi-garment try-on results on the Dress Code test set.

| **Model** | **Resolution** | **FID ↓** | **KID ↓** | **IS ↑** |
|---|---|---|---|---|
| CP-VTON$^\dagger$ [40] | $256 \times 192$ | 30.29 | 1.935 | **2.912** |
| VITON-GT [8] | $256 \times 192$ | 21.06 | 1.176 | 2.762 |
| WUTON [18] | $256 \times 192$ | 20.13 | 1.084 | 2.753 |
| **Ours (Patch)** | $256 \times 192$ | 19.86 | 1.006 | 2.784 |
| **Ours (PSAD)** | $256 \times 192$ | **17.52** | **0.749** | 2.832 |
| CP-VTON$^\dagger$ [40] | $512 \times 384$ | 22.96 | 1.327 | **3.273** |
| **Ours (Patch)** | $512 \times 384$ | 21.90 | 1.155 | 3.073 |
| **Ours (PSAD)** | $512 \times 384$ | **16.90** | **0.690** | 3.160 |
| CP-VTON$^\dagger$ [40] | $1024 \times 768$ | 23.30 | 1.393 | 3.261 |
| **Ours (Patch)** | $1024 \times 768$ | 20.26 | 0.841 | **3.498** |
| **Ours (PSAD)** | $1024 \times 768$ | **17.19** | **0.681** | 3.340 |

(*i.e.* dresses), but does not bring a consistent improvement over the use of human keypoints in our architecture. For this reason, we keep the latter model version for all the next experiments. In Fig. 4, we report a qualitative comparison between the results obtained with our Patch model and the PSAD version. In Fig. 5, we compare our results with those obtained by state-of-the-art competitors. Overall, our model with PSAD can better preserve the characteristics of the original clothes such as colors, textures, and shapes, and reduce artifacts and distortions, increasing the realism and visual quality of the generated images.

**High-Resolution Results.** For this experiment, we train and test our models and competitors using higher-resolution images ($512 \times 384$ and $1024 \times 768$). We compare with CP-VTON [40] and its improved version (CP-VTON$^\dagger$). Quantitative results for this setting are reported in Table 3 and refer to the entire test set of the Dress Code dataset. As it can be seen, our method outperforms the competitors. When generating images with resolution $1024 \times 768$, PSAD achieves the best results in terms of FID, KID, and IS with respect to the competitors and the Patch baseline.

**Multi-Garment Try-On Results.** As an additional experiment on the Dress Code dataset, we propose a novel setting in which the try-on is performed twice: first with an upper-body garment, and then with a lower-body item. This fully-unpaired setting aims to push further the difficulty of image-based virtual try-on, as it doubles the number of operations required to generate the resulting image. We remind that this experiment would have not been possible on the standard VITON dataset [13], as it contains only upper-body clothes. In Table 4, we report numerical results at varying image resolution. We can observe that PSAD outperforms the competitors and baselines on almost all the metrics for all the

**Table 5.** User study results. Our model is always preferred more than 50% of the time.

| | CP-VTON | VITON-GT | WUTON | ACGPN | PF-AFN | Ours (Patch) |
|---|---|---|---|---|---|---|
| **Realism** | 10.1 / **89.9** | 46.4 / **53.6** | 42.0 / **58.0** | 35.9 / **64.1** | 29.4 / **70.6** | 34.8 / **65.2** |
| **Coherency** | 11.5 / **88.5** | 32.1 / **67.9** | 41.6 / **58.4** | 23.1 / **76.9** | 25.0 / **75.0** | 36.9 / **63.1** |

**Fig. 6.** High-resolution results on the Dress Code test set in both single- and multi-garment try-on settings.

different image resolutions, with the only exception of the IS metric. Notably, the improvement of PSAD with respect to the Patch baseline ranges from 2.34 to 5.00 and from 0.16 to 0.46 in terms of FID and KID respectively.

**User Study.** While quantitative metrics used in the previous experiments can capture fine-grained variations in the generated images, the overall realism and visual quality of the results can be effectively assessed by human evaluation. To further evaluate the quality of generated images, we conduct a user study measuring both the realism of our results and their coherence with the input try-on garment and reference person. In the first test (Realism test), we show two generated images, one generated by our model and the other by a competitor, and ask to select the more realistic one. In the second test (Coherency test), in addition to the two generated images, we include the images of the try-on garment and the reference person used as input to the try-on network. In this case, we ask the user to select the image that is more coherent with the given inputs. All images are randomly selected from the Dress Code test set. Overall, this study involves a total of 30 participants, including researchers and non-expert people, and we collect more than 3,000 different evaluations (*i.e.* 1,500 for each test). Results are shown in Table 5. For each test, we report the percentage of votes obtained by the competitor / by our model. We also include a comparison with the Patch baseline. Our complete model is always selected more than 50% of the time against all considered competitors, thus further demonstrating the effectiveness of our solution.

### 5.3 Experiments on VITON

To conclude, we train the try-on networks on the widely used VITON dataset [13]. For this experiment, we compare our PSAD and Patch models

**Table 6.** Try-on results on the VITON test set [13]. Note that all models are trained exclusively on VITON.

| Model | Resolution | SSIM ↑ | FID ↓ | KID ↓ | IS ↑ |
|---|---|---|---|---|---|
| CP-VTON [40] | 256 × 192 | 0.798 | 19.06 | 0.906 | 2.601 |
| CP-VTON+ [28] | 256 × 192 | 0.828 | 16.31 | 0.784 | 2.821 |
| SieveNet [20] | 256 × 192 | 0.766 | 14.65 | - | 2.820 |
| ACGPN [42] | 256 × 192 | 0.845 | - | - | 2.829 |
| DCTON [9] | 256 × 192 | 0.830 | 14.82 | - | **2.850** |
| **Ours (Patch)** | 256 × 192 | **0.893** | 14.76 | 0.495 | 2.733 |
| **Ours (PSAD)** | 256 × 192 | 0.885 | **13.71** | **0.412** | 2.840 |


**Fig. 7.** Sample generated results on the VITON test set.

with other state-of-the-art architectures. In particular, we report results from CP-VTON [40] and CP-VTON+ [28] using source codes and pre-trained models provided by the authors. For SieveNet [20], ACGPN [42], and DCTON [9], we use the results reported in the papers. Table 6 shows the quantitative results on the test set, while in Fig. 7 we report four examples of the generated try-on results. Also in this setting, PSAD contributes to increasing the realism and visual quality of synthesized images.

## 6 Conclusion

In this paper, we presented Dress Code: a new dataset for image-based virtual try-on. Dress Code, while being more than 3× larger than the most common dataset for virtual try-on, is the first publicly available dataset for this task featuring clothes of multiple macro-categories and high-resolution images. We also presented a comprehensive benchmark with up to nine state-of-the-art virtual try-on approaches and different baselines, and introduced a Pixel-level Semantic-Aware Discriminator (PSAD) that improves the generation of high-quality images and the realism of the results.

**Acknowledgments.** We thank CINECA, the Italian Supercomputing Center, for providing computational resources. This work has been supported by the PRIN project "CREATIVE: CRoss-modal understanding and gEnerATIon of Visual and tExtual content" (CUP B87G22000460001), co-funded by the Italian Ministry of University and Research.

## References


1. Bińkowski, M., Sutherland, D.J., Arbel, M., Gretton, A.: Demystifying MMD GANs. In: ICLR (2018) 10
2. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. In: CVPR (2017) 6, 18
3. Chen, C.Y., Lo, L., Huang, P.J., Shuai, H.H., Cheng, W.H.: FashionMirror: Co-Attention Feature-Remapping Virtual Try-On With Sequential Template Poses. In: ICCV (2021) 4
4. Choi, S., Park, S., Lee, M., Choo, J.: VITON-HD: High-Resolution Virtual Try-On via Misalignment-Aware Normalization. In: ICCV (2021) 1, 3, 5
5. Cui, A., McKee, D., Lazebnik, S.: Dressing in Order: Recurrent Person Image Generation for Pose Transfer, Virtual Try-On and Outfit Editing. In: ICCV (2021) 4
6. Dong, H., Liang, X., Shen, X., Wang, B., Lai, H., Zhu, J., Hu, Z., Yin, J.: Towards Multi-Pose Guided Virtual Try-on Network. In: ICCV (2019) 5
7. Fenocchi, E., Morelli, D., Cornia, M., Baraldi, L., Cesari, F., Cucchiara, R.: Dual-Branch Collaborative Transformer for Virtual Try-On. In: CVPR Workshops (2022) 3
8. Fincato, M., Landi, F., Cornia, M., Fabio, C., Cucchiara, R.: VITON-GT: An Image-based Virtual Try-On Model with Geometric Transformations. In: ICPR (2020) 3, 7, 10, 12
9. Ge, C., Song, Y., Ge, Y., Yang, H., Liu, W., Luo, P.: Disentangled Cycle Consistency for Highly-realistic Virtual Try-On. In: CVPR (2021) 3, 14
10. Ge, Y., Song, Y., Zhang, R., Ge, C., Liu, W., Luo, P.: Parser-Free Virtual Try-on via Distilling Appearance Flows. In: CVPR (2021) 3, 6, 10
11. Güler, R.A., Neverova, N., Kokkinos, I.: DensePose: Dense Human Pose Estimation In The Wild. In: CVPR (2018) 6, 7, 18
12. Han, X., Hu, X., Huang, W., Scott, M.R.: ClothFlow: A flow-based model for clothed person generation. In: ICCV (2019) 1, 4
13. Han, X., Wu, Z., Wu, Z., Yu, R., Davis, L.S.: VITON: An Image-based Virtual Try-On Network. In: CVPR (2018) 1, 2, 3, 4, 5, 6, 10, 12, 13, 14
14. He, S., Song, Y.Z., Xiang, T.: Style-Based Global Appearance Flow for Virtual Try-On. In: CVPR (2022) 3
15. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Klambauer, G., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a Nash equilibrium. NeurIPS (2017) 10
16. Hsieh, C.W., Chen, C.Y., Chou, C.L., Shuai, H.H., Liu, J., Cheng, W.H.: FashionOn: Semantic-guided image-based virtual try-on with detailed human and clothing information. In: ACM Multimedia (2019) 5
17. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-To-Image Translation With Conditional Adversarial Networks. In: CVPR (2017) 3, 9, 10, 19
18. Issenhuth, T., Mary, J., Calauzènes, C.: Do Not Mask What You Do Not Need to Mask: a Parser-Free Virtual Try-On. In: ECCV (2020) 3, 10, 11, 12, 25, 27, 29
19. Jae Lee, H., Lee, R., Kang, M., Cho, M., Park, G.: LA-VITON: A Network for Looking-Attractive Virtual Try-On. In: ICCV Workshops (2019) 3
20. Jandial, S., Chopra, A., Ayush, K., Hemani, M., Krishnamurthy, B., Halwai, A.: SieveNet: A Unified Framework for Robust Image-Based Virtual Try-On. In: WACV (2020) 4, 14

21. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: ECCV (2016) 8
22. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. In: ICLR (2015) 10
23. Lewis, K.M., Varadharajan, S., Kemelmacher-Shlizerman, I.: TryOnGAN: Body-Aware Try-On via Layered Interpolation. ACM Trans. Graphic. **40**(4) (2021) 2, 5, 6
24. Li, K., Chong, M.J., Zhang, J., Liu, J.: Toward Accurate and Realistic Outfits Visualization with Attention to Details. In: CVPR (2021) 2, 3, 4, 5, 6
25. Li, P., Xu, Y., Wei, Y., Yang, Y.: Self-Correction for Human Parsing. IEEE Trans. PAMI **44**(6), 3260–3271 (2022) 6
26. Liu, X., Yin, G., Shao, J., Wang, X., Li, H.: Learning to predict layout-to-image conditional convolutions for semantic image synthesis. In: NeurIPS (2019) 3, 9
27. Liu, Z., Luo, P., Qiu, S., Wang, X., Tang, X.: DeepFashion: Powering robust clothes recognition and retrieval with rich annotations. In: CVPR (2016) 3, 5
28. Minar, M.R., Tuan, T.T., Ahn, H., Rosin, P., Lai, Y.K.: CP-VTON+: Clothing Shape and Texture Preserving Image-Based Virtual Try-On. In: CVPR Workshops (2020) 3, 10, 14
29. Mir, A., Alldieck, T., Pons-Moll, G.: Learning to Transfer Texture From Clothing Images to 3D Humans. In: CVPR (2020) 1
30. Neuberger, A., Borenstein, E., Hilleli, B., Oks, E., Alpert, S.: Image Based Virtual Try-On Network From Unpaired Data. In: CVPR (2020) 2, 5
31. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: CVPR (2019) 3, 9
32. Ren, B., Tang, H., Meng, F., Ding, R., Shao, L., Torr, P.H., Sebe, N.: Cloth Interactive Transformer for Virtual Try-On. arXiv preprint arXiv:2104.05519 (2021) 10
33. Rocco, I., Arandjelovic, R., Sivic, J.: Convolutional neural network architecture for geometric matching. In: CVPR (2017) 7
34. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: MICCAI (2015) 8, 9
35. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved Techniques for Training GANs. In: NeurIPS (2016) 10
36. Santesteban, I., Thuerey, N., Otaduy, M.A., Casas, D.: Self-Supervised Collision Handling via Generative 3D Garment Models for Virtual Try-On. In: CVPR (2021) 1
37. Schonfeld, E., Schiele, B., Khoreva, A.: A U-Net Based Discriminator for Generative Adversarial Network. In: CVPR (2020) 11
38. Schönfeld, E., Sushko, V., Zhang, D., Gall, J., Schiele, B., Khoreva, A.: You only need adversarial supervision for semantic image synthesis. In: ICLR (2021) 3, 9
39. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015) 8
40. Wang, B., Zheng, H., Liang, X., Chen, Y., Lin, L., Yang, M.: Toward characteristic-preserving image-based virtual try-on network. In: ECCV (2018) 3, 6, 7, 10, 12, 14, 18, 20, 25, 27, 29
41. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-Resolution Image Synthesis and Semantic Manipulation With Conditional GANs. In: CVPR (2018) 3, 9
42. Yang, H., Zhang, R., Guo, X., Liu, W., Zuo, W., Luo, P.: Towards Photo-Realistic Virtual Try-On by Adaptively Generating-Preserving Image Content. In: CVPR (2020) 1, 3, 4, 6, 7, 8, 10, 11, 14, 25, 27, 29

43. Yildirim, G., Jetchev, N., Vollgraf, R., Bergmann, U.: Generating high-resolution fashion model images wearing custom outfits. In: ICCV Workshops (2019) 2, 4, 5, 6
44. Yoo, D., Kim, N., Park, S., Paek, A.S., Kweon, I.S.: Pixel-level domain transfer. In: ECCV (2016) 5
45. Yu, R., Wang, X., Xie, X.: VTNFP: An Image-based Virtual Try-on Network with Body and Clothing Feature Preservation. In: ICCV (2019) 4
46. Zhao, F., Xie, Z., Kampffmeyer, M., Dong, H., Han, S., Zheng, T., Zhang, T., Liang, X.: M3D-VTON: A Monocular-to-3D Virtual Try-On Network. In: ICCV (2021) 1
47. Zheng, N., Song, X., Chen, Z., Hu, L., Cao, D., Nie, L.: Virtually Trying on New Clothing with Arbitrary Poses. In: ACM Multimedia (2019) 5
48. Zhu, H., Cao, Y., Jin, H., Chen, W., Du, D., Wang, Z., Cui, S., Han, X.: Deep-Fashion3D: A Dataset and Benchmark for 3D Garment Reconstruction from Single Images. In: ECCV (2020) 1

## Supplementary Material

### Dress Code Dataset

In Table 7, we report the total number of images and the dimensions of the train/test splits in the Dress Code dataset. Additionally, we detail the dimension of each split with respect to the different macro-categories of the dataset (upper-body clothes, lower-body clothes, and dresses). In our experiments, we train our models on all training pairs of the dataset and test both on each category separately and on the entire test set. Additional sample image pairs from Dress Code are shown in Fig. 8, 9, and 10 with the corresponding pose keypoints, dense pose of the reference model, and segmentation mask of the human body. As it can be seen, images of our dataset have a great variety considering both the body pose of the reference models and category and textures of try-on garments. This can lead to virtual try-on architectures becoming more general and adapting to more challenging scenarios.

### Additional Implementation Details

**Data Pre-Processing.** To extract the person pose representation $p$, we employ either OpenPose [2] or DensePose [11]. Specifically, the keypoints of the human body extracted with OpenPose [2] are used to compute the 18-channel pose heatmap, where each channel corresponds to one body keypoint represented as an $11 \times 11$ white rectangle. While both the 25 channels label map and the 2 channels UV map estimated by DensePose [11] are concatenated and used with no further processing.

In order to create the masked person representation $m$, we remove the information regarding the target clothes and the interested part of the body from $I$. Hence, the model only sees the face, the hair, and the target person part of the body which do not contain ground-truth information. To produce such masked representation, we use both the target label map to extract the clothes area and the pose map to extract the area of the limbs. These areas are then merged to form the mask which is then dilated to avoid the model getting information about the target shape. Finally, all the non-modifiable areas in the image (*e.g.* face, hands, hairs, etc.) are subtracted from the generated mask. The final mask is then applied to the image $I$. Note that while dilating the mask introduces more complexity in the paired setting generation task, it is essential in the unpaired one, especially when trying to substitute a garment with another whose shape covers a much larger area of the image.

**Warping Module.** Two feature extraction networks are included in the warping module, with four 2-strided down-sampling convolutional layers with a kernel size of 4 plus two 1-strided ones with a kernel size of 3. The first extraction network takes as input the try-on clothing item $c$, while the second one works on the concatenation between the person representation $m$ and the pose of the reference person $p$. Following [40], a correlation map is then computed between the outputs of the two feature extraction networks and then fed to a convolutional

**Table 7.** Number of train and test pairs for each category of the Dress Code dataset.

| | **Images** | **Training Pairs** | **Test Pairs** |
|---|---|---|---|
| **Upper-body Clothes** | 30,726 | 13,563 | 1,800 |
| **Lower-body Clothes** | 17,902 | 7,151 | 1,800 |
| **Dresses** | 58,956 | 27,678 | 1,800 |
| **All** | 107,584 | 48,392 | 5,400 |

network, consisting of two 2-strided convolutional layers with a kernel size of 4 and two 1-strided convolutional layers with a kernel size of 3. The output is forwarded through a fully connected layer that predicts the parameters of the geometric transformation. In particular, these parameters are the TPS anchor point coordinate offsets having a size of $2 \times 5 \times 5 = 50$. Batch normalization is applied to all convolutional layers. For the high-resolution versions of our model, we add an additional 2-strided down-sampling convolutional layer with a kernel size of 4 to both feature extraction networks.

**Human Parsing Estimation Module.** It is based on the U-Net architecture with four blocks in both encoder and decoder. Each block is composed of two sequences of a convolutional layer with a kernel size of 3, instance normalization, and a ReLU activation function. Each encoding block is followed by a 2-strided max pooling layer with a kernel size of 2, while each decoding block is preceded by a 2-strided transposed convolutional layer with a kernel size of 2 to upsample feature maps. Each encoding block is connected to the corresponding decoding block using skip connections. When training with high-resolution images, we add a U-Net block in both encoder and decoder.

**Try-On Module.** The encoder has four U-Net blocks, each having two convolutional layers with a kernel size of 3 and a 2-strided max pooling layer with a kernel size of 2. The decoder is symmetric but, instead of max pooling, the feature maps are up-sampled using a 2-strided transposed convolutional layer with a kernel size of 2. Also in this case, when training with high-resolution images, we add a U-Net block in both encoder and decoder.

**Discriminator.** PSAD works at pixel-level, classifying each pixel as one of the $N$ semantic classes of the human parser or as fake. The architecture is composed of 6 downsampling and 6 upsampling blocks arranged according to the U-Net architecture. The last layer is a $1 \times 1$ spatial convolution that brings the feature dimensionality to $N + 1$.

When training the Patch-based baseline, we instead employ PatchGAN [17] as our discriminator, which does not operate at pixel-level but instead classifies square image patches as real or fake, averaging all predictions to get the final result. It consists of three 2-strided down-sampling convolutional layers and one 1-strided down-sampling convolutional layer, all having a kernel size of 4. We use a convolutional layer to generate a scalar output in the last layer. Except for the first, we utilize batch normalization and apply Leaky ReLU with a 0.2 slope after each layer.

**Table 8.** High-resolution results on the Dress Code test set

| | Upper-body | | | Lower-body | | | Dresses | | | All | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Model** ($512 \times 384$) | **SSIM** ↑ | **FID** ↓ | **KID** ↓ | **SSIM** ↑ | **FID** ↓ | **KID** ↓ | **SSIM** ↑ | **FID** ↓ | **KID** ↓ | **SSIM** ↑ | **FID** ↓ | **KID** ↓ | **IS** ↑ |
| CP-VTON [40] | 0.850 | 48.24 | 3.365 | 0.826 | 57.38 | 4.00 | 0.845 | 24.04 | 0.891 | 0.831 | 29.24 | 1.671 | 3.096 |
| CP-VTON$^{\dagger}$ [40] | 0.916 | 14.37 | 0.442 | 0.910 | **12.54** | **0.432** | 0.861 | 21.82 | 0.720 | 0.896 | 10.08 | 0.425 | 3.277 |
| **Ours (Patch)** | **0.943** | 13.48 | 0.346 | **0.936** | 19.86 | 0.717 | **0.893** | 20.35 | 0.623 | **0.923** | 9.44 | **0.246** | 3.310 |
| **Ours (PSAD)** | 0.936 | **11.65** | **0.180** | 0.931 | 17.83 | 0.643 | 0.884 | **15.99** | **0.324** | 0.916 | **7.27** | 0.394 | **3.320** |
| | **Upper-body** | | | **Lower-body** | | | **Dresses** | | | **All** | | | |
| **Model** ($1024 \times 768$) | **SSIM** ↑ | **FID** ↓ | **KID** ↓ | **SSIM** ↑ | **FID** ↓ | **KID** ↓ | **SSIM** ↑ | **FID** ↓ | **KID** ↓ | **SSIM** ↑ | **FID** ↓ | **KID** ↓ | **IS** ↑ |
| CP-VTON [40] | 0.862 | 60.40 | 4.730 | 0.840 | 60.35 | 4.236 | 0.858 | 24.44 | 0.873 | 0.853 | 36.68 | 2.379 | 3.155 |
| CP-VTON$^{\dagger}$ [40] | 0.931 | 14.63 | 0.387 | 0.930 | **16.46** | **0.393** | 0.877 | 23.80 | 0.832 | 0.912 | 9.96 | 0.338 | 3.300 |
| **Ours (Patch)** | **0.944** | 13.38 | 0.273 | 0.933 | 19.97 | 0.654 | **0.890** | 24.14 | 0.807 | **0.922** | 9.99 | 0.370 | 3.34 |
| **Ours (PSAD)** | 0.941 | **12.10** | **0.171** | **0.935** | 19.02 | 0.641 | 0.882 | **17.93** | **0.425** | 0.919 | **7.70** | **0.236** | **3.357** |

**Training.** The experiments with low-resolution images are performed using a batch size of 32, while we use a batch size of 16 when training with high-resolution images for both $512 \times 384$ and $1024 \times 768$ resolutions. All experiments with $256 \times 192$ and $512 \times 384$ images are performed on 4 NVIDIA V100 GPUs, taking 10 hours to train the human parsing estimation module, one day for the warping module training stage, and around two days to train the try-on module. When instead training with full-resolution images, we split the batch size on 16 GPUs.

## Additional Results

**Low-Resolution Results.** In Fig. 11, we show some failure cases, while some additional qualitative comparisons between PSAD and the corresponding Patch-based baseline are shown in Fig. 12. In Fig. 13, 14, and 15, we report further try-on results on sample image pairs respectively extracted from upper-body clothes, lower-body clothes, and dresses by comparing our model with previously proposed try-on architectures re-trained on our newly collected dataset.

**High-Resolution Results.** Table 8 shows the complete try-on performances when generating high-resolution images while, in Fig. 16, we report some qualitative results.

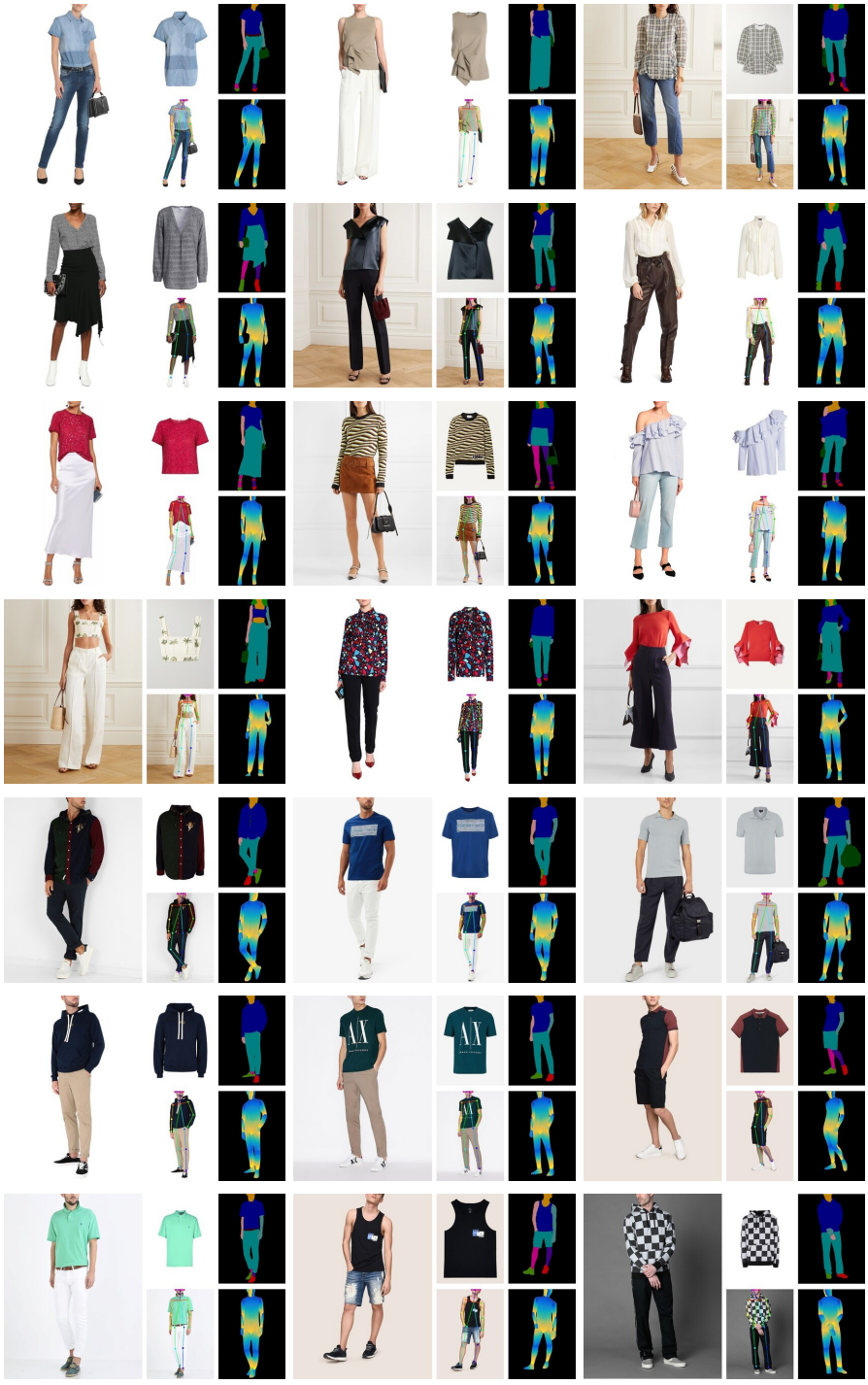

**Fig. 8.** Sample images of upper-body clothes and reference models from Dress Code.

**Fig. 9.** Sample images of lower-body clothes and reference models from Dress Code.

**Fig. 10.** Sample images of dresses and reference models from Dress Code.


**Fig. 11.** Failure cases on the Dress Code test set.


**Fig. 12.** Qualitative comparison between PSAD and the Patch-based baseline.


CP-VTON† [40]
WUTON [18]
ACGPN [42]
Ours (PSAD)

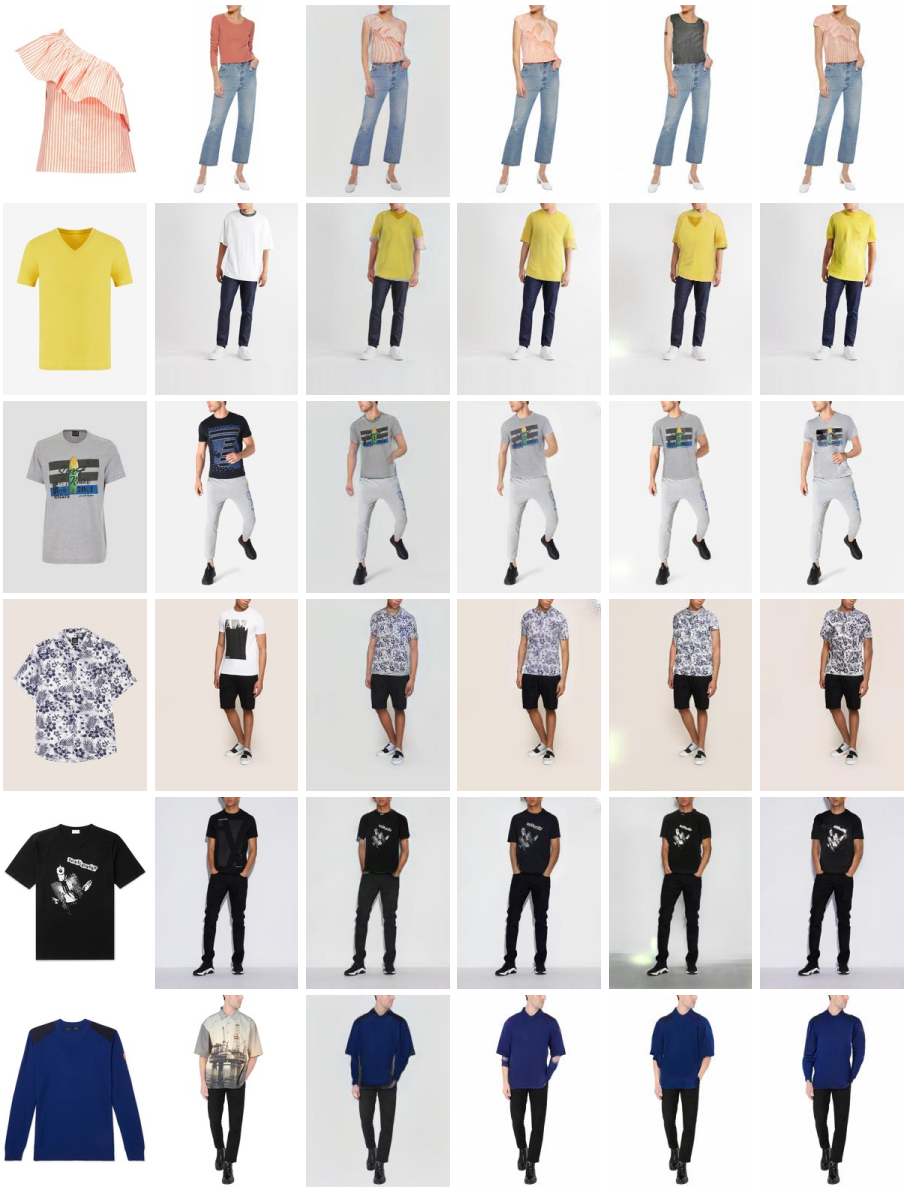

**Fig. 13.** Sample try-on results using upper-body clothes and reference models from the Dress Code test set.

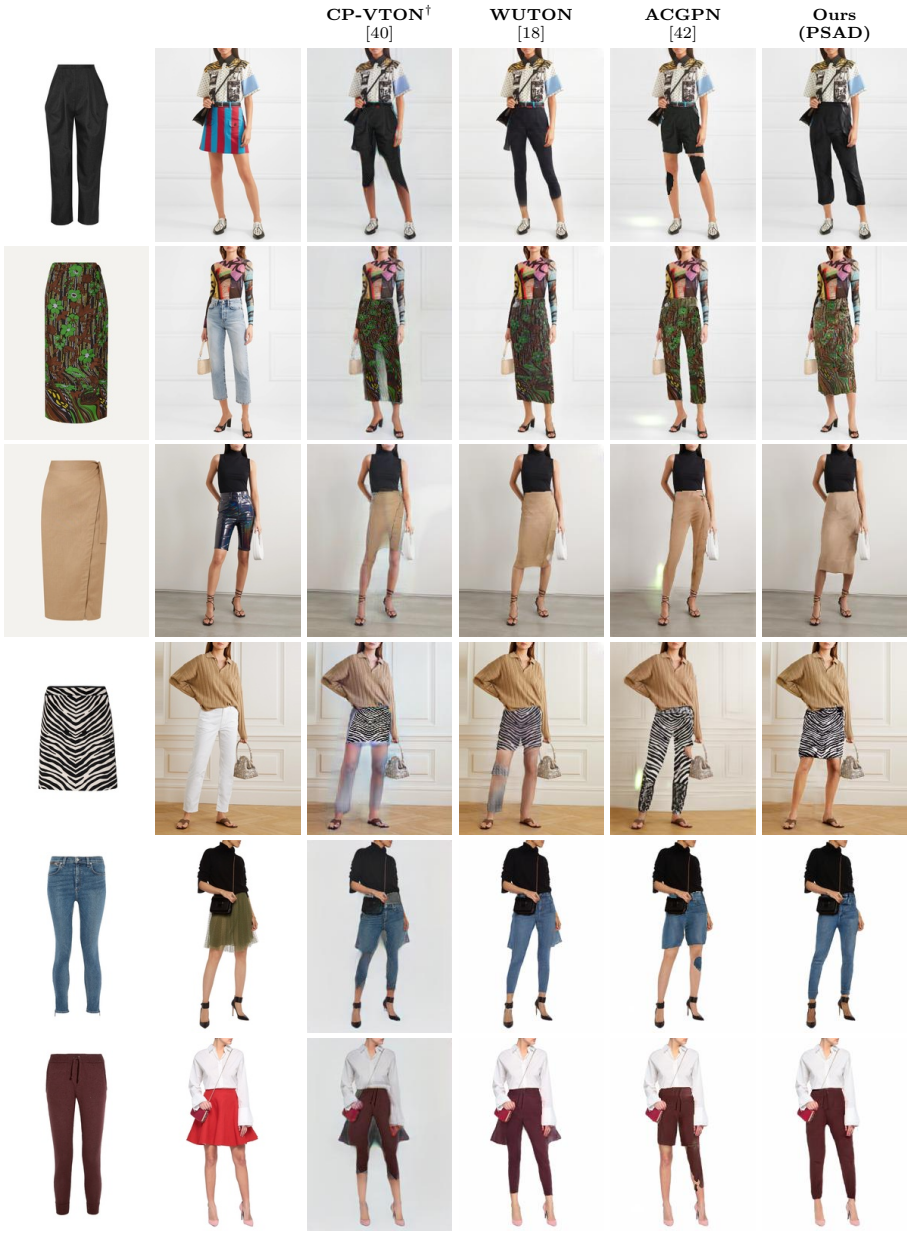
CP-VTON† [40]
WUTON [18]
ACGPN [42]
Ours (PSAD)

**Fig. 14.** Sample try-on results using lower-body clothes and reference models from the Dress Code test set.


CP-VTON† [40]
WUTON [18]
ACGPN [42]
Ours (PSAD)

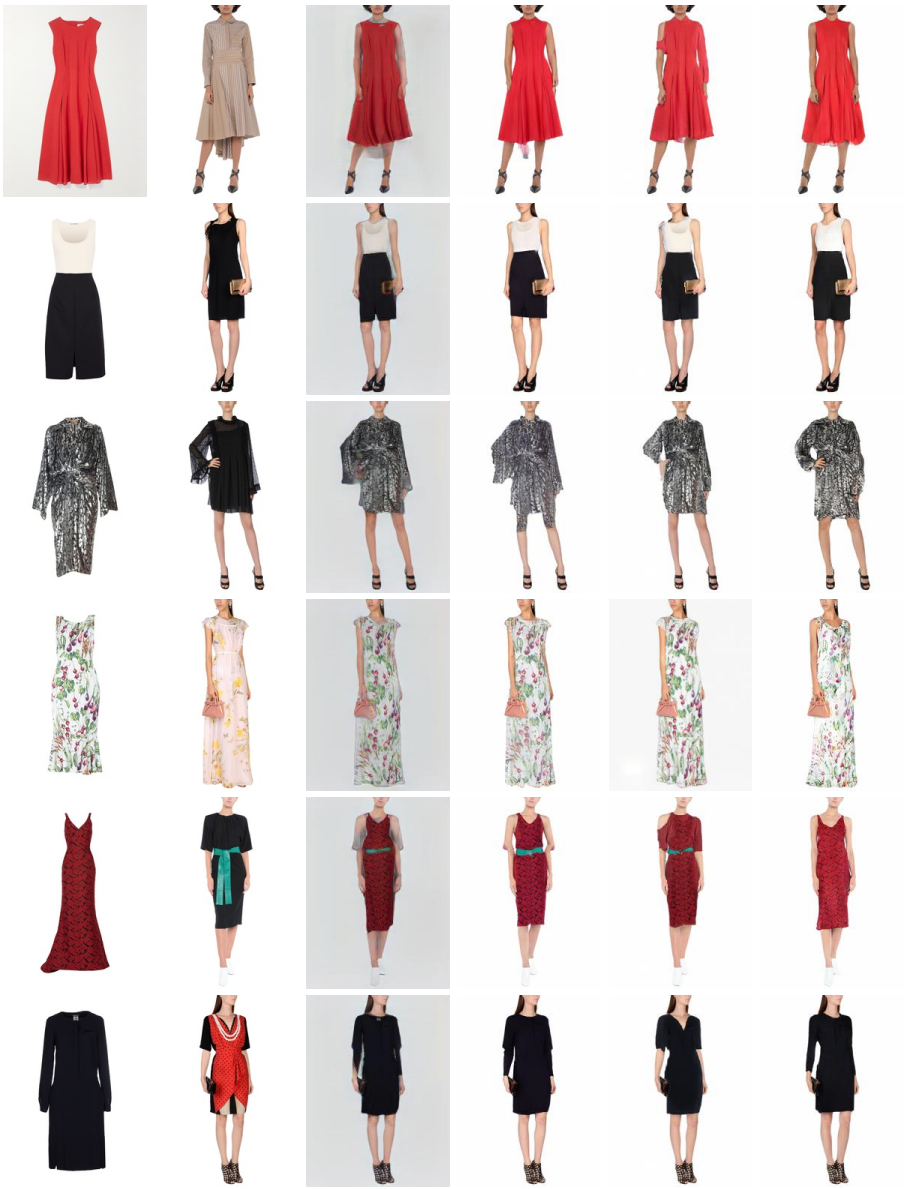

**Fig. 15.** Sample try-on results using dresses and reference models from the Dress Code test set.

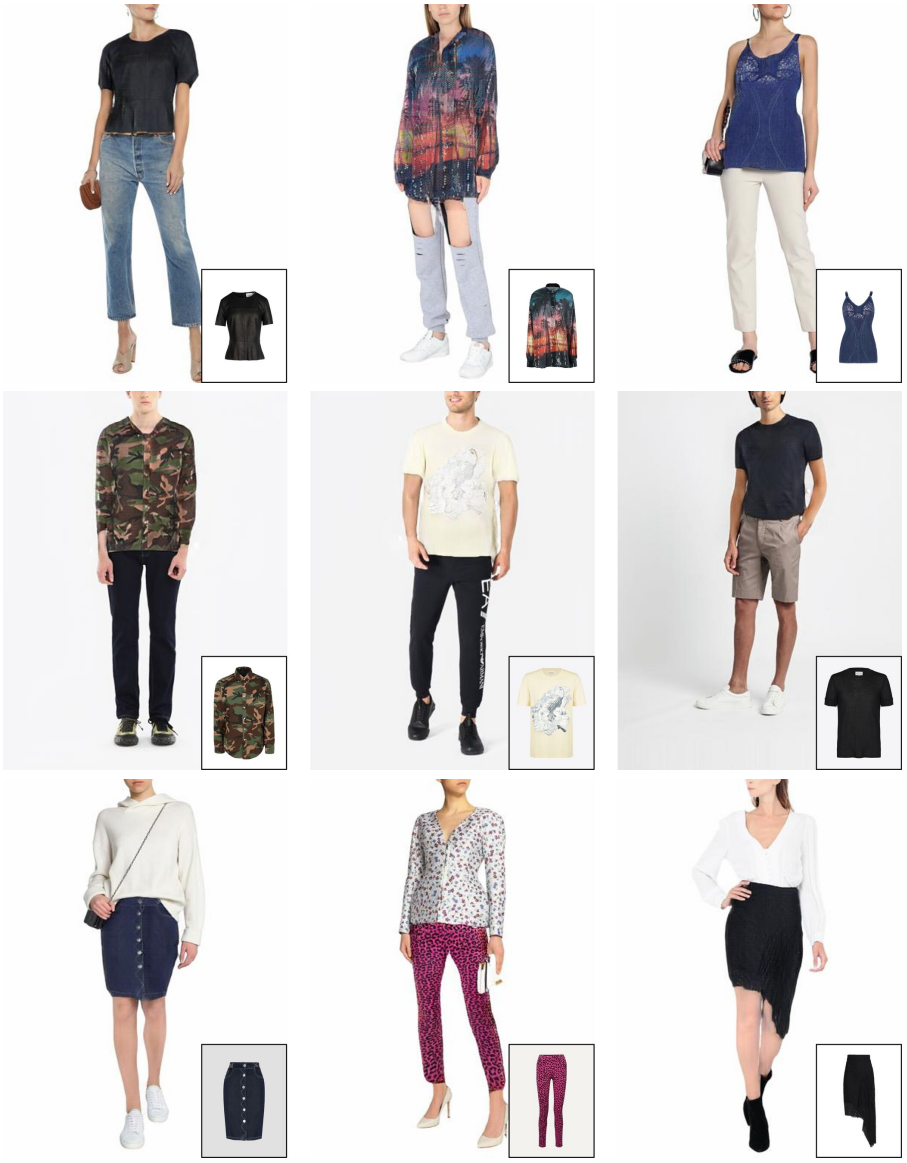

**Fig. 16.** Sample high-resolution results on the Dress Code test set.