Towards Scalable Unpaired Virtual Try-On via Patch-Routed Spatially-Adaptive GAN

Anonymous Author(s)
Affiliation
Address
email

Abstract

Image-based virtual try-on is one of the most promising applications of human-centric image generation due to its tremendous real-world potential. Yet, as most try-on approaches fit in-shop garments onto a target person, they require the laborious and restrictive construction of a paired training dataset, severely limiting their scalability. While a few recent works attempt to transfer garments directly from one person to another, alleviating the need to collect paired datasets, their performance is impacted by the lack of paired (supervised) information. In particular, disentangling style and spatial information of the garment becomes a challenge, which existing methods either address by requiring auxiliary data or extensive online optimization procedures, thereby still inhibiting their scalability. To achieve a *scalable* virtual try-on system that can transfer arbitrary garments between a source and a target person in an unsupervised manner, we thus propose a texture-preserving end-to-end network, the PAtch-routed SpaTially-Adaptive GAN (PASTA-GAN), that facilitates real-world unpaired virtual try-on. Specifically, to disentangle the style and spatial information of each garment, PASTA-GAN consists of an innovative patch-routed disentanglement module for successfully retaining garment texture and shape characteristics. Guided by the source person keypoints, the patch-routed disentanglement module first decouples garments into normalized patches, thus eliminating the inherent spatial information of the garment, and then reconstructs the normalized patches to the warped garment complying with the target person pose. Given the warped garment, PASTA-GAN further introduces novel spatially-adaptive residual blocks that guide the generator to synthesize more realistic garment details. Extensive comparisons with paired and unpaired approaches demonstrate the superiority of PASTA-GAN, highlighting its ability to generate high-quality try-on images when faced with a large variety of garments (e.g. vests, shirts, pants), taking a crucial step towards real-world scalable try-on.

1 Introduction

Image-based virtual try-on, the process of computationally transferring a garment onto a particular person in a query image, is one of the most promising applications of human-centric image generation with the potential to revolutionize shopping experiences and reduce purchase returns. However, to fully exploit its potential, scalable solutions are required that can leverage easily accessible training data, handle arbitrary garments, and provide efficient inference results. Unfortunately, to date, most existing virtual try-on methods [10, 29, 31, 30, 6, 7, 2] rely on *paired* training data, i.e., a person image and its corresponding in-shop garment, leading to laborious data-collection processes. Furthermore, these methods are unable to exchange garments directly between two person images, thus largely limiting their application scenarios and raising the need for *unpaired* solutions to ensure scalability.

While *unpaired* solutions have recently started to emerge, performing virtual try-on in an unsupervised setting is extremely challenging and tends to affect the visual quality of the try-on results. In particular, without access to the paired data, these models are usually trained by reconstructing the same person

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

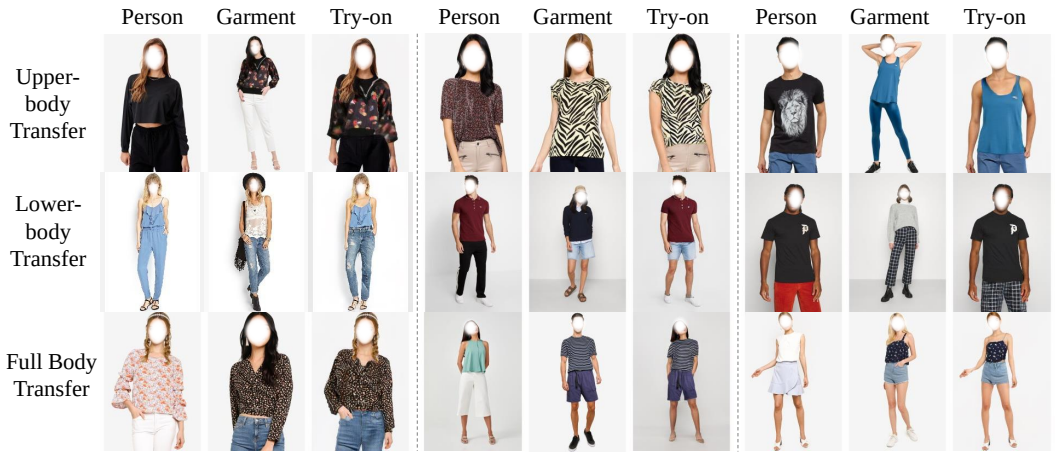

Figure 1: Example virtual try-on results by our PASTA-GAN which provides a scalable solution to the unpaired try-on task with the ability of transferring a large variety of garments, e.g. upper-body garments, lower-body garments and fully-body garments.

image, which is prone to over-fitting, and thus underperform when handling garment transfer during testing. The performance discrepancy is mainly reflected in the garment synthesis results, in particular the shape and texture, which we argue is caused by the entanglement of the garment style and spatial representations in the synthesis network during the reconstruction process.

While this is not a problem for traditional paired try-on approaches [30, 29, 31], which avoid this problem and preserve the garment characteristics by utilizing a supervised warping network or a 3D body model to obtain the warped garment, this is not possible in the unpaired setting due to the lack of warped ground truth. The few works that do attempt to achieve unpaired virtual try-on, therefore, choose to circumvent this problem by either relying on person images in various poses for feature disentanglement [19, 27, 26, 25], which again leads to a laborious data-collection process, or require extensive online optimization procedures [20, 14] to obtain fine-grain details of the original garments, harming the inference efficiency. However, none of the existing unpaired try-on methods consider the problem of coupled style and spatial garment information directly, which is crucial to obtain accurate garment transfer results in the unsupervised and unpaired virtual try-on scenario.

In this paper, to tackle the essential challenges mentioned above and resolve the deficiency of prior approaches, we propose a novel PAtch-routed SpaTially-Adaptive GAN, named PASTA-GAN, a scalable solution to the unpaired try-on task. Our PASTA-GAN can precisely synthesize garment shape and style (see Fig. 1) by introducing a patch-routed disentanglement module that decouples the garment style and spatial features, as well as a novel spatially-adaptive residual module to mitigate the problem of feature misalignment.

The innovation of our PASTA-GAN includes three aspects: First, by separating the garments into normalized patches with the inherent spatial information largely reduced, the patch-routed disentanglement module encourages the style encoder to learn spatial-agnostic garment features. These features enable the synthesis network to generate images with accurate garment style regardless of varying spatial garment information. Second, given the target human pose, the normalized patches can be easily reconstructed to the warped garment complying with the target shape, without requiring a warping network or a 3D model. Finally, the spatially-adaptive residual module extracts the warped garment feature and adaptively inpaints the region that is misaligned with the target garment shape. Thereafter, the inpainted warped garment features are embedded into the intermediate layer of the synthesis network, guiding the network to generate try-on results with realistic garment texture.

We conduct extensive experiments on the newly collected UnPaired Try-on (UPT) dataset and the two existing DeepFashion [17] and MPV [3] try-on benchmark datasets, showing that our unsupervised PASTA-GAN outperforms previous unpaired try-on approaches and even surpasses supervised methods under the paired try-on setting.[1]

[1]The source codes and dataset are provided in the Supplementary Materials.

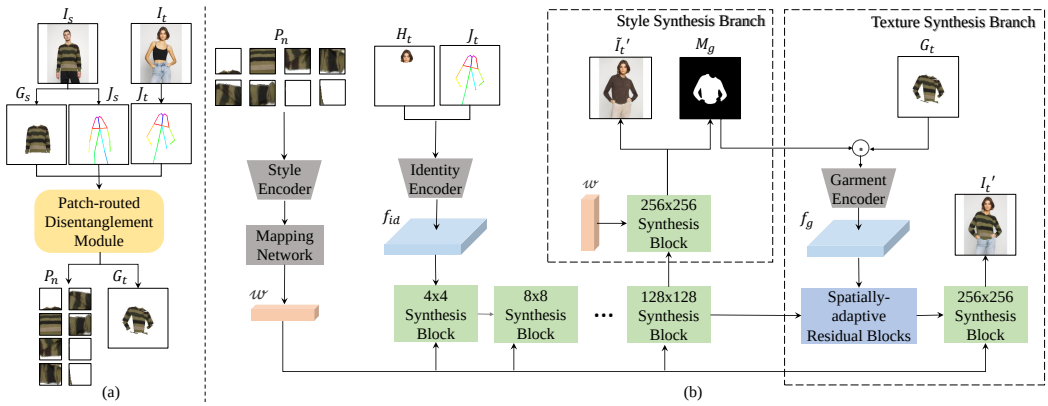

Figure 2: Overview of PASTA-GAN. (a) Given the source and target images of person (I_s, I_t), we can extract the source garment G_s, the source pose J_s, and the target pose J_t. The three are then sent to the patch-routed disentanglement module to yield the normalized garment patches P_n and the warped garment G_t. (b) The modified conditional StyleGAN first collaboratively exploits the disentangled style code w, projected from P_n, and the person identity feature f_{id}, encoded from target head and pose (H_t, J_t), to synthesize the coarse try-on result $\tilde{I}_t'$ in the style branch along with the target garment mask M_g. It then leverages the warped garment feature f_g in the texture branch to generate the final try-on result I_t'.

2 Related Work

Paired Virtual Try-on. Paired try-on methods [29, 3, 31, 9, 4, 30, 6, 7] transfer an in-shop clothing onto a reference person and have a common structure consisting of garment warping followed by inpainting of the try-on result to synthesize the final try-on image. VITON [10] first propose this pattern by utilizing a TPS warping module and a UNet inpainting module. CPVTON [29] replaces the time-consuming shape matching process in VITON with a trainable GMM network to improve the garment deformation accuracy. VTNFP [31] introduces human parsing to guide the generation in a semantically-aware way, while [9, 4] turn to flow prediction for pixel-level garments transfer. ACGPN [30] constructs a three stage model to incorporate garment warping, parsing prediction and image inpainting. Recently, VITON-HD [2] focuses on high-resolution virtual try-on and proposes an ALIAS normalization mechanism to normalize the misaligned area between the warped garment and the generated semantic map. [6, 7] improve the learning process further by adding cycle-consistency regularization and knowledge distillation, achieving state-of-the-art results. However, all these works require paired data that is cumbersome to collect. Furthermore, paired try-on is incapable of exchanging garments directly between two person images, which further limits its applications.

Unpaired Virtual Try-on. Different from the aforementioned methods, some recent methods [19, 27, 26, 25, 20, 14] eliminate the need for in-shop garment images and directly transfer garments between two people. Among these methods, [19, 27, 26, 25] leverage pose transfer as a pretext task to learn disentangled pose and appearance features for human synthesis, but require images of the same person with different poses. In contrast, [20, 14] are more flexible and can be directly trained with unpaired person images. Nonetheless, to maintain the fine-grain details of the original garments, OVITON [20] requires online appearance optimization for each garment region during testing. VOGUE [14] utilizes the StyleGAN2 [12] framework and is capable of synthesizing high fidelity images. However, for each pair of person images, VOGUE needs to run an optimizer to obtain latent codes for each person image. Thereafter, VOGUE optimizes the interpolate coefficient for the latent codes and obtains the interpolated latent code for the try-on result. Thus, previous unpaired virtual approaches, either require the collection of additional data (i.e. multiple pose images) or require extensive optimization procedures, thus not presenting efficient and scalable solutions.

3 PASTA-GAN

Given a source image I_s, of a person wearing a garment G_s, and a target person image I_t, the unpaired virtual try-on task aims to synthesize the try-on result I_t' retaining the identity of I_t but

wearing the source garment G_s. To achieve this, our PASTA-GAN first utilizes the patch-routed disentanglement module (Sec. 3.1) to transform the garment G_s into normalized patches P_n that are mostly agnostic to the spatial features of the garment, and further deforms P_n to obtain the warped garment G_t complying with the target person pose. Then, an attribute-decoupled conditional StyleGAN2 (Sec. 3.2) is designed to synthesize try-on results in a coarse-to-fine manner, where we introduce novel spatially-adaptive residual blocks (Sec. 3.3) to inject the warped garment features into the generator network for more realistic texture synthesis. The training details and loss function will be described in Sec. 3.4. An overview of the model is provided in Fig. 2.

3.1 Patch-routed Disentanglement Module

Since the paired data for supervised training is unavailable for the unpaired virtual try-on task, the synthesis network has to be trained in an unsupervised manner via image reconstruction, and thus takes a person image as input and separately extracts the features of the intact garment and the features of the person representation to reconstruct the original person image. While such a training strategy retains the intact garment information, which is helpful for the garment reconstruction, the features of the intact garment entangle the garment style with the spatial information in the original image. This is detrimental to the garment transfer during testing. Note that the garment style here refers to the garment color and categories, i.e., long sleeve, short sleeve, etc., while the garment spatial information implies the orientation and size of each garment part in the person image, which are determined by the body pose and the camera distance for a particular image, respectively.

To address this issue, we explicitly divide the garment into normalized patches to remove the inherent spatial information of the garment. Taking the sleeve patch as an example, by using division and normalization, various sleeve regions from different person images can be deformed to normalized patches with the same orientation and scale. Without the guidance of the spatial information, the network is forced to leverage the garment style and learn style features to reconstruct the clothes in the synthesis image.

Fig. 3 illustrates the process of obtaining normalized garment patches, which includes two main steps: (1) pose-guided garment segmentation, and (2) perspective transformation-based patch normalization. Specifically, in the first step, the source garment G_s and human pose (joints) J_s are firstly obtained by applying [8] and [1] to the source person I_s, respectively. Given the body joints, we can segment the source garment into several patches P_s that will later be normalized, where the number of valid patches depends on the clothing category, e.g., 8 for long-sleeve shirts and 6 for T-shirts. Taking the upper-arm region for example, with the coordinates of the shoulder joint and the elbow joint in P_s, a quadrilateral crop (of which the four corner points are visualized in color in P_s of Fig. 3) covering the upper sleeve of G_s can be easily performed to produce an unnormalized garment patch. In the second step, all patches are normalized to remove their spatial information by perspective transformations. For this, we first define the same amount of template patches P_n with fixed 64×64 resolution as transformation targets for all unnormalized source patches, and then compute a homography matrix $\mathcal{H}_{s\to n} \in \mathbb{R}^{3\times 3}$ [32] between each pair of a P_s and a P_n, based on the four corresponding corner points of the two patches. Concretely, $\mathcal{H}_{s\to n}$ serves as a perspective transformation to relate the pixel coordinates in the two patches, formulated as:

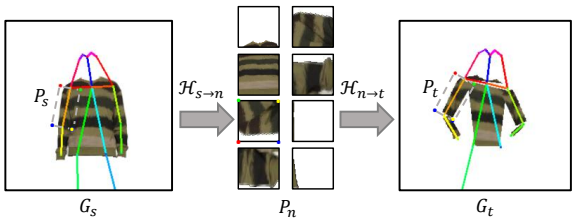

Figure 3: The process of the patch-routed deformation.

$$\begin{bmatrix} x^n \\ y^n \\ 1 \end{bmatrix} = \mathcal{H}_{s\to n} \begin{bmatrix} x^s \\ y^s \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x^s \\ y^s \\ 1 \end{bmatrix} \tag{1}$$

where (x^n, y^n) and (x^s, y^s) are the pixel coordinates in the normalized template patch and the unnormalized source patch, respectively. We first estimate an initial homography matrix with least-squares optimization to minimize the re-projection error:

$$\min_{\mathcal{H}_{s\to n}} \sum_i \left(x_i^n - \frac{h_{11}x_i^s + h_{12}y_i^s + h_{13}}{h_{31}x_i^s + h_{32}y_i^s + h_{33}} \right)^2 + \left(y_i^n - \frac{h_{21}x_i^s + h_{22}y_i^s + h_{23}}{h_{31}x_i^s + h_{32}y_i^s + h_{33}} \right)^2 \tag{2}$$

and then refine the initial $\mathcal{H}_{s\to n}$ further with the Levenberg-Marquardt method [5] to reduce the re-projection error even more. After that, we can transform the source patch with $\mathcal{H}_{s\to n}$ to the normalized patch P_n according to Eq. 1.

Moreover, the normalized patches can further be transformed to target garment patches P_t by utilizing the target pose J_t, which can be obtained from the target person I_t via [1]. The mechanism of that backward transformation is equivalent to the forward one in Eq. 1, i.e., computing the homography matrix $\mathcal{H}_{n\to t}$ based on the four point pairs extracted from the normalized patches P_n and the target pose J_t. The recovered target patches P_t can then be stitched to form the warped garment G_t that will be sent to the texture synthesis branch in Fig. 2 to generate more realistic garment transfer results. We can also regard $\mathcal{H}_{s\to t} = \mathcal{H}_{s\to n} \cdot \mathcal{H}_{n\to t}$ as the combined deformation matrix that warps the source garment to the target person pose, bridged by an intermediate normalized patch representation that is helpful for disentangling garment styles and spatial features.

3.2 Attribute-decoupled Conditional StyleGAN2

Motivated by the impressive performance of StyleGAN2 [12] in the field of image synthesis, our PASTA-GAN inherits the main architecture of StyleGAN2 and modifies it to the conditional version (see Fig. 2). In the synthesis network, the normalized patches are projected to the style code w through a style encoder followed by a mapping network, which is spatial-agnostic benefiting from the disentanglement module. In parallel the conditional information including the target head H_t and pose J_t is transferred into a feature map f_{id}, encoding the identity of the target person by the identity encoder. Thereafter the style code and the identity feature map are both injected to the StyleGAN2 synthesis blocks to respectively control the garment style and the person characteristics of the try-on result $\tilde{I}_t'$.

However, the standalone conditional StyleGAN is insufficient to generate compelling garment details especially in the presence of complex textures or logos. For example, although the illustrated $\tilde{I}_t'$ in Fig. 2 can recover accurate garment style (color and shape) given the disentangled style code w, it lacks the complete texture pattern. The reasons for this are twofold: First, the style encoder projects the normalized patches into a one-dimensional vector, resulting in loss of high frequency information. Second, due to the large variety of garment texture, learning the local distribution of the particular clothing details is highly challenging for the basic synthesis network.

To generate more accurate garment details, instead of only having a one-way synthesis network, we intentionally split PASTA-GAN into two branches after the 128×128 synthesis block, namely the Style Synthesis Branch (SSB) and the Texture Synthesis Branch (TSB). The SSB with normal StyleGAN2 synthesis blocks aims to generate intermediate try-on results $\widetilde{I_t}'$ with accurate garment style and predict a precise garment mask M_g that will be used by TSB. The purpose of TSB is to exploit the warped garment G_t, which has rich texture information to guide the synthesis path, and generate high-quality try-on results. We introduce a novel spatially-adaptive residual module specifically before the final synthesis block of the TSB, to embed the warped garment feature f_g (obtained by passing M_g and G_t through the garment encoder) into the intermediate features and then send them to the newly designed spatialy-apaptive residual blocks, which are beneficial for successfully synthesizing texture of the final try-on result I_t'. The detail of this module will be described in the following section.

3.3 Spatially-adaptive Residual Module

Given the style code that factors out the spatial information and only keeps the style information of the garment, the style synthesis branch in Fig. 2 can accurately predict the mean color and the shape mask of the target garment. However, its inability to model the complex texture raises the need to exploit the warped garment G_t to provide features that encode high-frequency texture patterns, which is in fact the motivation of the target garment reconstruction in Fig. 3.

However, as the raw feature map of G_t is directly obtained by stitching the target patches together, it certainly has its flaws and is coarser and more inaccurate than the predicted mask M_g. This shape misalignment in G_t will consequently reduce the quality of the extracted warped garment feature f_g.

To address this issue, we introduce the spatially-adaptive residual module between the last two synthesis blocks in the texture synthesis branch as shown in Fig. 2. This module is comprised of a garment encoder and three spatially-adaptive residual blocks with feature inpainting mechanism to modulate intermediate features by leveraging the inpainted warped garment feature.

Figure 4: Illustration of misalignment between the warped garment and target garment shape. The orange and green region represent the region to be inpainted and to be removed, respectively.

To be specific on the feature inpainting process, we first remove the part of G_t that falls outside of M_g, and explicitly inpaint the misaligned regions of the feature map within M_g with average feature values (see Fig. 4). The inpainted feature map can then help the final synthesis block infer reasonable texture in the misaligned parts. Therefore given the predicted garment mask M_g, the warped garment G_t and its mask M_t, the process of feature inpainting can be formulated as:

$$M_{align} = M_g \cap M_t, \tag{3}$$

$$M_{misalign} = M_g - M_{align}, \tag{4}$$

$$f'_g = \mathcal{E}_g(G_t \odot M_g), \tag{5}$$

$$f_g = F_{c'} \odot (1 - M_{misalign}) + \mathcal{A}(F_{c'} \odot M_{align}) \odot M_{misalign}, \tag{6}$$

where $\mathcal{E}_g(\cdot)$ represents the garment encoder and f'_g denotes the raw feature map of G_t masked by M_g. $\mathcal{A}(\cdot)$ calculates the average garment features and f_g is the final inpainted feature map.

Subsequently, inspired by the SPADE ResBlk from SPADE [21], the inpainted garment features are used to calculate a set of affine transformation parameters that efficiently modulate the normalized feature map within each spatially-adaptive residual block. The normalization and modulation process for a particular sample $h_{z,y,x}$ at location ($z \in C, y \in H, x \in W$) in a feature map can then be formulated as:

$$\gamma_{z,y,x}(f_g)\frac{h_{z,y,x} - \mu_z}{\sigma_z} + \beta_{z,y,x}(f_g), \tag{7}$$

where $\mu_z = \frac{1}{HW}\sum_{y,x} h_{z,y,x}$ and $\sigma_z = \sqrt{\frac{1}{HW}\sum_{y,x}(h_{z,y,x} - \mu_z)^2}$ are the mean and standard deviation of the feature map along channel z. $\gamma_{z,y,x}(\cdot)$ and $\beta_{z,y,x}(\cdot)$ are the convolution operations that convert the inpainted feature to affine parameters. Eq. 7 serves as a learnable normalization layer for the spatially-adaptive residual block to better capture the statistical information of the garment feature map, thus helping the synthesis network to generate more realistic garment texture.

With the modulated intermediate feature maps produced by the spatially-adaptive residual module, the texture synthesis branch can effectively utilize the reconstructed warped garment and generate the final compelling try-on result with high-frequency texture patterns.

3.4 Training Details and Loss Functions

As paired training data is unavailable, our PASTA-GAN is trained unsupervised via image reconstruction. During training, we utilize the reconstruction loss $\mathcal{L}_{rec}$ and the perceptual loss [11] $\mathcal{L}_{perc}$ for both the coarse try-on result $\widetilde{I}'$ and the final try-on result I':

$$\mathcal{L}_{rec} = \sum_{I \in \{\widetilde{I}', I'\}} \|I - I_s\|_1 \quad and \quad \mathcal{L}_{perc} = \sum_{I \in \{\widetilde{I}', I'\}} \sum_{k=1}^{5} \lambda_k \left\|\phi_k(I) - \phi_k(I_s)\right\|_1, \tag{8}$$

where $\phi_k(I)$ denotes the k-th feature map in a VGG-19 network [28] pre-trained on the ImageNet [24] dataset. We also use the L_1 loss between the predicted garment mask M_g and the real mask M_{gt} which is obtained via human parsing [8]:

$$\mathcal{L}_{mask} = \|M_g - M_{gt}\|_1. \tag{9}$$

Besides, for both $\widetilde{I}'$ and I', we calculate the adversarial loss $\mathcal{L}_{GAN}$ which is the same as in StyleGAN2 [12]. The total loss can be formulated as

$$\mathcal{L} = \mathcal{L}_{GAN} + \lambda_{rec}\mathcal{L}_{rec} + \lambda_{perc}\mathcal{L}_{perc} + \lambda_{mask}\mathcal{L}_{mask}, \tag{10}$$

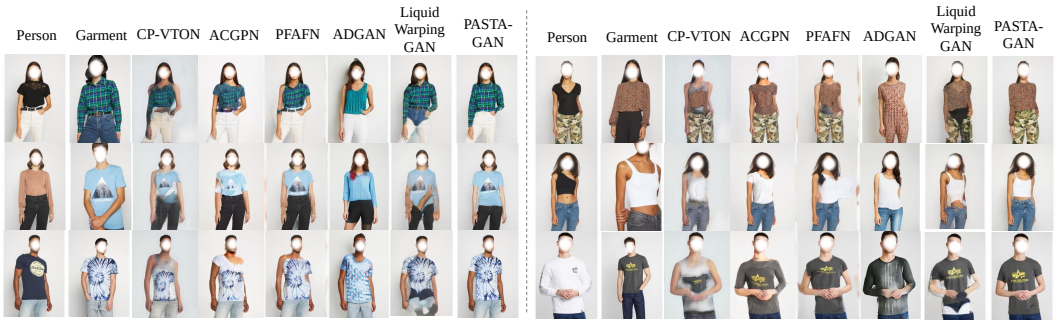

Figure 5: Visual comparison among PASTA-GAN and the baseline methods under the unpaired setting on the UPT dataset. Please zoom in for more details.

Figure 6: Visual comparison among PASTA-GAN and the paired baseline methods under their corresponding test setting on the MPV dataset [3]. Please zoom in for more details.

where λ_{rec}, λ_{perc}, and λ_{mask} are the trade-off hyper-parameters.

To imitate the self-occlusion in the source person, we also randomly remove one of the four arm patches in the warped garment with a probability of α_1. To further boost the texture synthesis in the misalignment region, we use random mask from [15] to additionally erase part of the warped garment with a probability of α_2. During training, α_1 and α_2 are set to 0.2 and 0.9, respectively.

4 Experiments

Datasets. We conduct experiments on two existing virtual try-on benchmarks (MPV [3] and DeepFashion [17]) and our newly collected large-scale benchmark dataset for unpaired try-on, named UPT. UPT contains 33,254 half- and full-body front-view images of persons wearing a large variety of garments, e.g., long/short sleeve, vest, sling, pants, etc. UPT is further split into a training set of 27,139 images and a testing set of 6,115 images. In addition, we also pick out the front view images from MPV [3] and DeepFashion [17] to expand the size of our training and testing set to 54,714 and 10,493 respectively, which forms the final UPT dataset. Personally identifiable information (i.e. face information) has been masked out. Additional details for the UPT dataset are provided in the supplementary material.

Metrics. We apply the Fréchet Inception Distance (FID) [22] to measure the similarity between real and synthesized images, and perform human evaluation to quantitatively evaluate the synthesis quality of different methods. For the human evaluation, we design three questionnaires corresponding to the three used datasets. In each questionnaire, we randomly select 40 try-on results generated by our PASTA-GAN and the other compared methods. Then, we invite 30 volunteers to complete the 40 tasks by choosing the most realistic try-on results. Finally, the human evaluation score is calculated as the chosen percentage for a particular method.

Implementation Details. Our PASTA-GAN is implemented using PyTorch [23] and is trained on 8 Tesla V100 GPUs. During training, the batch size is set to 96 and the model is trained for 4 million

Table 1: The FID score [22] and human evaluation score among different methods under the unpaired setting on the DeepFashion dataset [17] and our UPT dataset.

Method	DeepFashion		UPT	
	FID ↓	Human Evaluation ↑	FID ↓	Human Evaluation ↑
CP-VTON [29]	69.46	2.177%	70.76	1.551%
ACGPN [30]	44.41	4.597%	37.99	3.448%
PFAFN [7]	46.19	4.677%	36.69	4.224%
ADGAN [19]	37.36	21.29%	39.60	7.241%
Liquid Warping GAN [16]	42.18	12.98%	33.18	9.310%
PASTA-GAN (Ours)	**21.58**	**54.27%**	**7.852**	**74.22%**

Table 2: The FID score [22] and human evaluation score among different methods under their corresponding test setting on the MPV dataset [3].

Method	CP-VTON [29]	ACGPN [30]	PFAFN [7]	PASTA-GAN(Ours)
FID ↓	37.72	23.20	17.40	**16.48**
Human Evaluation ↑	8.071%	12.64%	28.71%	**50.57%**

iterations with a learning rate of 0.002 using the Adam optimizer [13] with $\beta_1 = 0$ and $\beta_2 = 0.99$. The loss hyper-parameters λ_{rec}, λ_{perc}, and λ_{mask} are set to 40, 40, and 100, respectively.[2]

Baselines. To validate the effectiveness of our PASTA-GAN, we compare it with the state-of-the-art methods, including three paired virtual try-on methods, CP-VTON [29], ACGPN [30], PFAFN [7], and two unpaired methods Liquid Warping GAN [16] and ADGAN [19], which have released the official code and pre-trained weights.[3] We directly use the pre-trained model of these methods as their training procedure depends on the paired data of garment-person or person-person image pairs, which are unavailable in our dataset. When testing paired methods under the unpaired try-on setting, we extract the desired garment from the person image and regard it as the in-shop garment to meet the need of paired approaches. To fairly compare with the paired methods, we further conduct another experiment on the paired MPV dataset [3], in which the paired methods take an in-shop garment and a person image as inputs, while our PASTA-GAN still directly receives two person images. See the following two subsections for detailed comparisons on both paired and unpaired settings.

4.1 Comparison with the state-of-the-art methods on unpaired benchmark

Quantitative: As reported in Table 3.4, when testing on the DeepFashion [17] and the UPT dataset under the unpaired setting, our PASTA-GAN outperforms both the paired methods [29, 30, 7] and the unpaired methods [19, 16] by a large margin, obtaining the lowest FID score and the highest human evaluation score, demonstrating that PASTA-GAN can generate more photo-realistic images. Note that, although ADGAN [19] is trained on the DeepFashion dataset, our PASTA-GAN still surpasses it. Since the data in the DeepFashion dataset is more complicated than the data in UPT, the FID scores for the DeepFashion dataset are generally higher than the FID scores for the UPT dataset.
Qualitative: As shown in Fig. 5, under the unpaired setting, PASTA-GAN is capable of generating more realistic and accurate try-on results. On the one hand, paired methods [29, 30, 7] tend to fail in deforming the cropped garment to the target shape, resulting in the distorted warped garment that is largely misaligned with the target body part. On the other hand, unpaired method ADGAN [19] cannot preserve the garment texture and the person identity well due to its severe overfitting on the DeepFashion dataset. Liquid Warping GAN [16], another publicly available appearance transfer model, heavily relies on the 3D body model named SMPL [18] to obtain the appearance transfer flow. It is sensitive to the prediction accuracy of SMPL parameters, and thus prone to incorrectly transfer the appearance from other body parts (e.g., hand, lower body) into the garment region in case of inaccurate SMPL predictions. In comparison, benefited by the patch-routed mechanism, PASTA-GAN can learn appropriate garment features and predict precise garment shape. Further, the spatially-adaptive residual module can leverage the warped garment feature to guide the network to synthesize try-on results with realistic garment textures.

[2]Please refer to more details, hyperparameter analyses and code in the supplementary materials.

[3]For all these prior approaches, research use is permitted according to the respective licenses. Note, we are unable to compare with [14] and [20] as they have not released their code or pre-trained model.

Method	SSB	TSB	GP	NRB	SRB	FID ↓
PASTA-GAN⋆	✓	✗	✗	✗	✗	21.99
PASTA-GAN∗	✓	✗	✓	✗	✗	17.69
PASTA-GAN†	✗	✓	✓	✗	✓	11.27
PASTA-GAN‡	✓	✓	✓	✓	✗	10.08
PASTA-GAN	✓	✓	✓	✗	✓	**7.851**

Figure 7: Qualitative results and quantitative results of the ablation study with different configurations, in which SSB, TSB, GP, NRB, SRB refer to style synthesis branch, texture synthesis branch, garment patches, normal residual blocks, and spatially-adaptive residual blocks, respectively.

4.2 Comparison with the state-of-the-art methods on paired benchmark

Quantitative: Tab. 2 illustrates the quantitative comparison on the MPV dataset [3], in which the paired methods are tested under the classical paired setting, i.e., transferring an in-shop garment onto a reference person. Our unpaired PASTA-GAN, nevertheless, can surpass the paired methods especially the state-of-the-art PFAFN [7] in both FID and human evaluation score, further evidencing the superiority of our PASTA-GAN. **Qualitative:** Under the paired setting, the visual quality of the paired methods improves considerably, as shown in Fig. 6. However, the paired methods depend on TPS-based or flow-based warping architectures to deform the whole garment, which may lead to the distortion of texture and shape since the global interpolation or pixel-level correspondence is error-prone in case of large pose variation. Our PASTA-GAN, instead, warps semantic garment patches separately to alleviate the distortion and preserve the original garment texture to a larger extent. Besides, the paired methods are unable to handle garments like sling that are rarely presented in the dataset, and perform poorly on full-body images. Our PASTA-GAN instead generates compelling results even in these challenging scenarios.

4.3 Ablation Studies

Patch-routed Disentanglement Module: To validate its effectiveness, we train two PASTA-GANs without texture synthesis branch, denoted as PASTA-GAN⋆ and PASTA-GAN∗, which take the intact garment and the garment patches as input of the style encoder, respectively. As shown in Fig. 7, PASTA-GAN⋆ fails to generate accurate garment shape. In contrast, the PASTA-GAN∗ which factors out spatial information of the garment, can focus more on the garment style information, leading to the accurate synthesis of garment shape. However, without the texture synthesis branch, both of them are unable to synthesize the detailed garment texture. The models with the texture synthesis branch can preserve the garment texture well as illustrated in Fig 7.

Spatially-adaptive Residual Module To validate the effectiveness of this module, we further train two PASTA-GANs with texture synthesis branch, denoted as PASTA-GAN† and PASTA-GAN‡, each of which respectively excludes the style synthesis branch and replaces the spatially-adaptive residual blocks with normal residual blocks. Without the support of the corresponding components, both PASTA-GAN† and PASTA-GAN‡ fail to fix the garment misalignment problem, leading to artifacts outside the target shape and blurred texture synthesis results. The full PASTA-GAN instead can generate try-on results with precise garment shape and texture details. The quantitative comparison results in Fig. 7 further validate the effectiveness of our designed modules.

5 Conclusion

We propose the PAtch-routed SpaTially-Adaptive GAN (PASTA-GAN) towards facilitating scalable unpaired virtual try-on. By utilizing the novel patch-routed disentanglement module and the spatially-adaptive residual module, PASTA-GAN effectively disentangles garment style and spatial information and generates realistic and accurate virtual-try on results without requiring auxiliary data or extensive online optimization procedures. Experiments highlight PASTA-GAN's ability to handle a large variety of garments, outperforming previous methods both in the paired and the unpaired setting.

We believe that this work will inspire new scalable approaches, facilitating the use of the large amount of available unlabeled data. However, as with most generative applications, misuse of these techniques is possible in the form of image forgeries, i.e. warping of unwanted garments with malicious intent.

References

[1] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. In *IEEE TPAMI*, 2019.

[2] Seunghwan Choi, Sunghyun Park, Minsoo Lee, and Jaegul Choo. Viton-hd: High-resolution virtual try-on via misalignment-aware normalization. In *CVPR*, 2021.

[3] Haoye Dong, Xiaodan Liang, Xiaohui Shen, Bochao Wang, Hanjiang Lai, Jia Zhu, Zhiting Hu, and Jian Yin. Towards multi-pose guided virtual try-on network. In *ICCV*, pages 9026–9035, 2019.

[4] Haoye Dong, Xiaodan Liang, Xiaohui Shen, Bowen Wu, Bing-Cheng Chen, and Jian Yin. Fw-gan: Flow-navigated warping gan for video virtual try-on. In *ICCV*, October 2019.

[5] Henri P Gavin. The levenberg-marquardt algorithm for nonlinear least squares curve-fitting problems. In *Department of Civil and Environmental Engineering, Duke University*, pages 1–19, 2019.

[6] Chongjian Ge, Yibing Song, Yuying Ge, Han Yang, Wei Liu, and Ping Luo. Disentangled cycle consistency for highly-realistic virtual try-on. In *CVPR*, 2021.

[7] Yuying Ge, Yibing Song, Ruimao Zhang, Chongjian Ge, Wei Liu, and Ping Luo. Parser-free virtual try-on via distilling appearance flows. In *CVPR*, 2021.

[8] Ke Gong, Yiming Gao, Xiaodan Liang, Xiaohui Shen, Meng Wang, and Liang Lin. Graphonomy: Universal human parsing via graph transfer learning. In *CVPR*, 2019.

[9] Xintong Han, Xiaojun Hu, Weilin Huang, and Matthew R. Scott. Clothflow: A flow-based model for clothed person generation. In *ICCV*, October 2019.

[10] Xintong Han, Zuxuan Wu, Zhe Wu, Ruichi Yu, and Larry S Davis. Viton: An image-based virtual try-on network. In *CVPR*, pages 7543–7552, 2018.

[11] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711, 2016.

[12] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *ICCV*, pages 8110–8119, 2020.

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[14] Kathleen M Lewis, Srivatsan Varadharajan, and Ira Kemelmacher-Shlizerman. Vogue: Try-on by stylegan interpolation optimization. In *arXiv preprint arXiv:2101.02285*, 2021.

[15] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *ECCV*, page 85–100, 2018.

[16] Wen Liu, Wenhan Luo Lin Ma Zhixin Piao, Min Jie, , and Shenghua Gao. Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In *ICCV*, 2019.

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

[18] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: a skinned multi-person linear model. In *ACM Transactions on Graphics*, volume 34, 2015.

[19] Yifang Men, Yiming Mao, Yuning Jiang, Wei-Ying Ma, and Zhouhui Lian. Controllable person image synthesis with attribute-decomposed gan. In *ICCV*, pages 5084–5093, 2020.

[20] Assaf Neuberger, Eran Borenstein, Bar Hilleli, Eduard Oks, and Sharon Alpert. Image based virtual try-on network from unpaired data. In *ECCV*, pages 5184–5193, 2020.

[21] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *ICCV*, pages 2337–2346, 2019.

[22] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On buggy resizing libraries and surprising subtleties in fid calculation. In *arXiv preprint arXiv:2104.11222*, 2021.

[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *arXiv preprint arXiv:1912.01703v1*, 2019.

[24] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. In *IJCV*, volume 115, pages 211–252, 2015.

[25] Kripasindhu Sarkar, Vladislav Golyanik, Lingjie Liu, and Christian Theobalt. Style and pose control for image synthesis of humans from a single monocular view. In *arXiv preprint arXiv:2102.11263*, 2021.

[26] Kripasindhu Sarkar, Lingjie Liu, Vladislav Golyanik, and Christian Theobalt. Humangan: A generative model of humans images. In *CVPR*, 2021.

[27] Kripasindhu Sarkar, Dushyant Mehta, Weipeng Xu, Vladislav Golyanik, and Christian Theobalt. Neural re-rendering of humans from a single image. In *ECCV*, pages 596–613, 2020.

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015.

[29] Bochao Wang, Huabin Zheng, Xiaodan Liang, Yimin Chen, Liang Lin, and Meng Yang. Toward characteristic-preserving image-based virtual try-on network. In *ECCV*, pages 589–604, 2018.

[30] Han Yang, Ruimao Zhang, Xiaobao Guo, Wei Liu, Wangmeng Zuo, and Ping Luo. Towards photo-realistic virtual try-on by adaptively generating preserving image content. In *CVPR*, pages 7850–7859, 2020.

[31] Ruiyun Yu, Xiaoqi Wang, and Xiaohui Xie. Vtnfp: An image-based virtual try-on network with body and clothing feature preservation. In *ECCV*, pages 10511–10520, 2019.

[32] Qiang Zhou and Xin Li. Stn-homography: estimate homography parameters directly. In *arXiv preprint arXiv:1906.02539*, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Section Abstract and Introduction.
   (b) Did you describe the limitations of your work? [Yes] Limitations are discussed with help of failure examples, which are provided in the supplementary material.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We discuss misuse of the proposed method in the Conclusion.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The code and data links are provided in the supplementary material.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] More details are included in the supplementary material.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] A fixed seed was used for all experiments and we provide a sensitivity analysis in the supplementary material.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] It is included in the implementation details in Sec. 4.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes]

(b) Did you mention the license of the assets? [Yes] All the models used in our work are publicly released under an MIT License or a license restricted to non-commercial research and educational purposes. For the UPT dataset details are provided in the supplementary material.

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We include the source code in the supplementary material.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] The data and models used in this work are publicly released.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] The face information in the source and the target person images have been masked out (see Sec. 4.1 Experiment Setup).

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [Yes] The instruction with an example is provided in the supplementary material.

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] The participants consisted of volunteers.