POVNet: Image-Based Virtual Try-On Through Accurate Warping and Residual

Kedan Li, Jeffrey Zhang, and David Forsyth, *Fellow, IEEE*

***Abstract*— Virtual dressing room applications help online shoppers visualize outfits. Such a system, to be commercially viable, must satisfy a set of performance criteria. The system must produce high quality images that faithfully preserve garment properties, allow users to mix and match garments of various types and support human models varying in skin tone, hair color, body shape, and so on. This paper describes POVNet, a framework that meets all these requirements (except body shapes variations). Our system uses warping methods together with residual data to preserve garment texture at fine scales and high resolution. Our warping procedure adapts to a wide range of garments and allows swapping in and out of individual garments. A learned rendering procedure using an adversarial loss ensures that fine shading, etc. is accurately reflected. A distance transform representation ensures that hems, cuffs, stripes, and so on are correctly placed. We demonstrate improvements in garment rendering over state of the art resulting from these procedures. We demonstrate that the framework is scalable, responds in real-time, and works robustly with a variety of garment categories. Finally, we demonstrate that using this system as a virtual dressing room interface for fashion e-commerce websites has significantly boosted user-engagement rates.**

***Index Terms*—Virtual try-on, image generation, generative adversarial networks, positional encoding, warping, image inpainting, application.**

I. Introduction

THE fashion retail industry is going through a rapid transition from brick and mortar stores to e-commerce platforms [1]. Online fashion shops typically showcase products using neutral garment images and images of a single model wearing the garment. In this framework shoppers cannot mix & match garment combinations nor visualize outfits on themselves [2]. A virtual dressing room could restore this experience and significantly increase user-engagement and conversion rates [3]. However, traditional methods for enabling virtual try-on are expensive – often requiring 3D models, 3D garments, or special images for every item. A significant body of recent research investigates image-based virtual try-on [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21].

A virtual try-on system maps one or more *neutral garment images* – which show a garment in some neutral configuration, for example a shirt with arms laid flat – to an image of a model wearing those garments. Some specification of the model's pose might be used to obtain a good image. For a virtual try-on system to be *commercially viable* it should have important properties.

1) *Faithful Representation*: Images generated by the system must *faithfully represent all the attributes* of the garments. This includes the shape, the material, the prints and logos, the trims and borders, and every other detail of the specific garment. This is important, because a consumer misled by a faithless image will likely return a purchase, which is expensive.
2) *Image Quality*: Rendered images should appear photorealistic and should have high resolution. Aliasing effects, often caused by fine garment stripes, are intolerable. This is important, because users will ignore an interface that produces low quality images.
3) *Mix & Match*: The method must be able to render a complete outfit consisting of selected garments from different categories (for example, tops and outerwear and trousers). This is important, because users will be puzzled by an interface that allows them to see only one type of garment.
4) *Scalability*: The system must be scalable and interactive. This is important, because users will ignore a slow interface, and because vendors want to display a full catalog of garments.
5) *Garment Variety*: The system must support a wide variety of garment types.
6) *Model Control*: Users should be able to see garments on diverse models, varying by at least hair style, ethnicity, skin tone, and body shape. This is important, because different users will want to evaluate garments for quite different bodies.
7) *Garment Swapping*: Users should be able to swap one garment at a time, while maintaining the state of the model and the rest of the garments. This is important to support the metaphor that the interface is "like" a dressing room.

These properties are not entirely independent (for example, one can't have 3 without 5), but are distinct (for example, one can have 5 without 3). Some are much more demanding technically than others – for example, accurately representing the appearance of garments as body shape changes remains elusive – but all

Manuscript received 20 October 2022; revised 28 March 2023; accepted 1 June 2023. Date of publication 9 June 2023; date of current version 5 September 2023. This work was supported in part by the National Science Foundation under Grant 2106825, in part by the Office of Naval Research under Grant N000014-16-1-2007, and in part by gifts from Amazon and from Boeing. Recommended for acceptance by T.M. Hospedales. *(Corresponding author: Kedan Li.)*

Kedan Li and Jeffrey Zhang are with the Revery AI Inc., Champaign, IL 61820 USA (e-mail: kedan@revery.ai; jeff@revery.ai).

David Forsyth is with the Department of Computer Science, University of Illinois at Urbana-Champaign, Champaign, IL 61820 USA (e-mail: daf@illinois.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2023.3283302, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2023.3283302

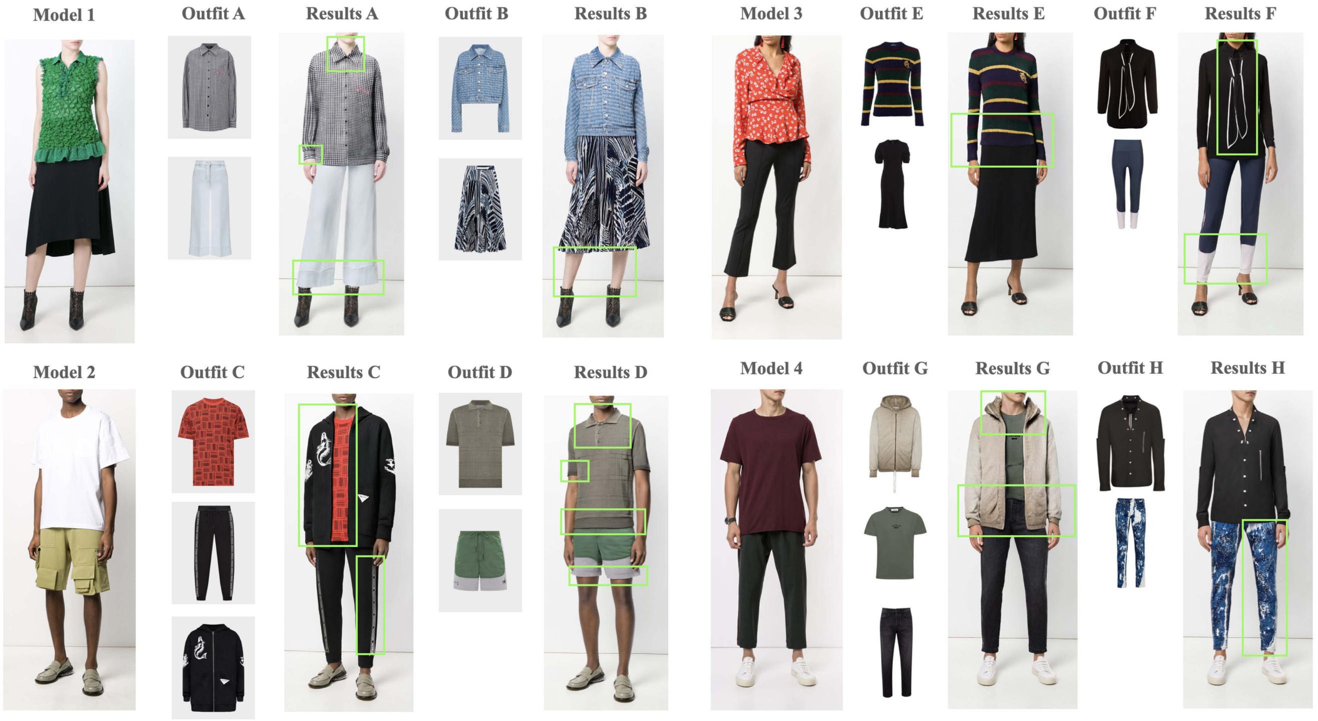


Fig. 1. Our method faithfully represents garments and renders high quality images. Faithful representation means the neutral garment and the garment rendered on the model *should be the same garment*. Boxes identify effects of particular interest (box-free images, see in supplemental materials for details). Notice: detailed garment textures are preserved (e.g., patterns on the denim jacket and the skirt in Outfit B; logos on the jacket and pattern on the shirt in C, ties on the shirt in F, the hood in G, prints on the jeans in H); structural properties are preserved (e.g., collar and cuff on shirt and wide trouser leg in A; collar, sleeve cuff and ribbon on the top and length of short in D, lines and ribbons in E, white trouser leg in F); natural interaction between the garment and the model's body (e.g., drape of the trouser leg in A; folds at the cuff caused by the hand A, B; shadow cast on the leg by the skirt B; shadow on inner leg A, D); the natural split and cast shadow in G and the realistic skin.

are important. We have listed these properties according to our rough estimate of their importance. These properties provide a sobering view of the state of current research (Section II). Much of the literature can achieve some, but not other, properties. GAN-based image generation methods [22], [23], [24], [25] can produce photo realistic try-on images at high resolution but cannot accurately represent specific garments. Image-based virtual try-on methods [4], [5], [6], [7], [8], [9], [11], [12], [13], [14], [16], [17], [19], [20], [21] preserve garment attributes, but typically operate on one garment of a single category per image (prior work mostly focused on tops). There are methods [10], [26], [27], [28] that support multi-garment interactions, but cannot preserve garment details.

A. Precise Outfit Visualization Net

We describe Precise Outfit Visualization Net (POVNet) – an extended version of OVNet [29] which can achieve all of the above properties to some extent. POVNet can produce accurate textures, necklines, and hemlines, and layers multiple garments with realistic overlay and shadows. The drapes adapt to the body pose and generate natural creases, folds, and shading. Skin and background are also synthesized with appropriate shadows casted from the garments. Our method significantly outperforms the state-of-the-art in multi-garment image synthesis (Figs. 9 and 12).

During inference, POVNet accepts a starting image y of a model wearing clothing and an optional desired skin tone t. POVNet produces a new image y' where the model is wearing a neutral garment of type c depicted in a neutral garment image x, as well as all clothes not of type c shown in y. The model in y has skin tone t if t is provided, but otherwise has the original skin tone. This architecture allows the user to replace clothes one at a time (take y' and feed it into POVNet with a different x).

POVNet consists of four main components. The *semantic layout generator* (SLG - G_{layout}) predicts a semantic layout (m'). This is a segmentation map indicating where garments lie in the image. G_{layout} accepts x, a map of the pose (p) of the model in y, and an initial layout (m_i). The resulting m' is used to guide image generation, particularly to resolve details of garment overlap. The *multi-warp garment generator* (MGG - $G_{garment}$) produces y'. This generator accepts m' from the semantic layout generator, x and y. It operates in two stages: first, a *warping module* produces several interacting warps of the neutral garment to the target image; second, an *inpainting module* (trained with an adversary) fuses these images together with partial layout information and t to produce a convincing image. Finally, the *residual enhancement* module maps patterns from the generated image back to the neutral garment image, computes a residual, then uses this to enhance resolution.

Although swapping an outfit during inference involves multiple applications of $G_{garment}$, the operations re-use m' so that the

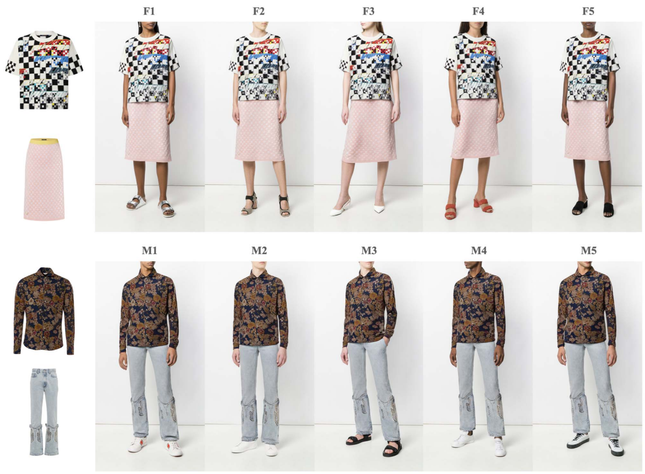


Fig. 2. *Faithful representation* means applying the same set of neutral garments to different models in different poses should result in images of different models in different poses *wearing the same garments*. This figure shows results for garments with rich design elements. Length, cut, pattern, texture, collar, cuff, and details like the fraying at the knees are preserved. Note how the bell-bottom jeans break at the foot in some poses, as they should. In some poses, the skirt binds against a forward leg; when it does, there is a natural shadow. The skirt casts shadows on the legs (as it should), and they are affected by pose. Models can have widely varied skin tones, which are rendered realistically.

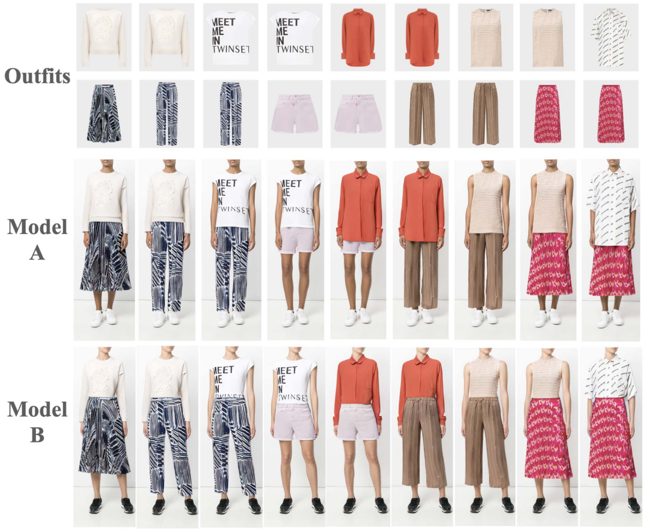


Fig. 3. A sequence of outfit visualizations produced by our method on two different models, representing the user experience of swapping in one garment at a time. Our method operates by applying a neutral garment to an existing image, producing a new image (which can have another new garment applied, etc.). Notice: faithful representation, as in Fig. 1 – while garment appearance can vary with model pose and attitude (tucked in; tucked out), garments *look the same* across models; existing garments are preserved, and new garments interact with them naturally; and garments drape naturally with pose.

process is relatively efficient and much of the generated image is unchanged when applying one new garment (in contrast to [10], which must change the whole layout each time one garment is changed). The sequential process also brings the benefit of allowing users to modify one garment at a time while the rest of the image remains untouched (Fig. 3). Users do not expect that swapping a garment will cause changes in the rest of the image – for example, changing a top should not cause the model's feet to move. Other benefits emerge from the details of each element.

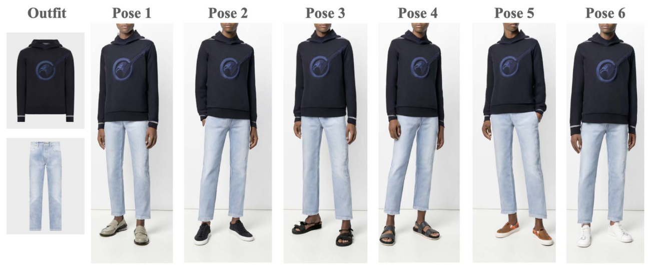


Fig. 4. Garments drape naturally as model pose changes. The figure shows renderings of the same outfit worn by the same model in different poses. Notice: faithful representation, as in Figs. 1 and 3; garments drape naturally with pose; inside leg shading and garment folds are consistent with pose; when a hand is a pocket of a garment, the garment plumps up and interacts naturally with the other garments (Poses 2 & 5).

TABLE I
This Table Compares SSIM [86], IS [87] and FID [88] Reported on the Original VITON Test Set. Numbers for Prior Works are Taken From the Original Work. POVNet+DensePose (Which Adapts Style-Based Flow [17] to Our Pipeline) Outperforms All Prior Works While POVNet Also Yields Strong Performance

Methods	SSIM↑	IS↑	FID↓
VITON [4]	.783	2.65	55.7
CP-VTON [5]	.745	2.76	24.5
GarmentGAN [63]	-	2.77	-
VTNFP [62]	.803	2.78	-
SieveNet [11]	.766	2.82	-
ClothFlow [8]	.841	-	23.68
ACGPN [9]	.845	2.83	16.6
OVNet [29]	.852	2.85	15.78
ZFlow [16]	.885	-	15.17
DCTON [13]	.838	2.85	14.82
Dress Code [18]	.890	2.84	13.71
RT-VTON [19]	-	-	11.66
SDAFN [20]	-	-	9.46
HR-VITON [21]	.864	-	9.38
Style-Based Flow [17]	.910	-	8.89
POVNet	.891	2.87	13.37
POVNet+DensePose	**.918**	**2.92**	**8.82**

POVNet contains a number of technical innovations which result in measurable improvements in accuracy. Because publicly available try-on datasets do not contain rich garment categories, we test on a dataset with all major garment categories from multiple fashion e-commerce websites. Evaluation on this new dataset shows that using multiple warps consistently outperforms single warp baselines in this new setting, demonstrated both quantitatively (Table III) and qualitatively (Fig. 8). Our try-on system also produces higher quality images compared to prior works on both single and multi-garment generation (Tables I and II, and Figs. 9 and 12). The residual enhancement yields clear improvement to garment details (Fig. 10) and super-resolution further increases the details captured by the generated image (as in Fig. 11).

TABLE II

This Table Compares SSIM [86] Score and the FID_∞ [89] Score on the Multi-Garment Dataset. POVNet Outperforms Dress Code [18] and the Original OVNet [29]. The Ablation Studies Suggest That Each Module Contributes. In Contrast to Table I, POVNet+DensePose Performs Significantly Worse Than POVNet on the Multi-Garment Dataset

Methods	SSIM ↑	FID_∞ ↓
Dress Code [18]	.821	.912
Original OVNet [29]	.840	.874
POVNet+DensePose	.834	.894
POVNet w/o Distance Transform	.846	.867
POVNet w/o Inpainter Formulation Change	.852	.854
POVNet w/o Residual Enhancement	.856	.849
Full POVNet	**.862**	**.846**

The purpose of POVNet is to solve practical commercial problems. To show the results have practical consequences, we have deployed our system on real e-commerce sites, where users can select garments and see generated visualizations in real-time. We show that user engagement is significantly improved when users can interact with POVNet. A live demo of a virtual try-on shopping interface powered by the latest version of our framework is publicly available.[1]

1) Warping, Inpainting and Residual: POVNet is designed around a warping process, because warping is known to be helpful in detail preservation [4], [5], [6], [7], [8], [9], [12], [13], [14], [16], [17], [19], [20], [21].

We warp one garment at a time onto the target image. Warping image features creates difficulties when one wants to apply more than one garment (for example, should the shirt be tucked in or out). An alternative is to encode garments into *feature vectors* and then broadcast the vectors onto a layout as O-VITON does [10]; this allows interactions between multiple garments, but makes it difficult to synthesize texture details precisely. In contrast to previous work, OVNet [29] uses multiple warpers for each garment (so dealing with the difficulties presented by, say, jackets, which often appear as two disjoint regions). Using multiple coordinated warps produces substantial quantitative and qualitative improvements over prior single-warp methods [4], [5], [7], [8], [9], [11].

We recognized that insufficiently controlled warps can result in images of a similar, but different, garment from the intended. For example, a warp might result in stripes that are too thick or too thin; a hemline that is too low or too high; or an oddly shaped collar. This could be controlled by allowing the warper to see a range of configuration features (binary masks or sparse body keypoints, as in [4], [5], [6], [7], [8], [9], [12], [13], [14], [19], [20], [21]). Alternately, one could use DensePose [30] style representations (as in [16], [17], [19]), but our experiments suggest that garments and poses are strongly correlated in training data, meaning that one encounters nasty generalization difficulties at run time (see supplemental materials for details). Unlike any prior work, POVNet uses a distance transform representation to ensure that warps do not distort the garment. Results in Fig. 9 shows a strong improvement in the warp coherence.

One advantage of warping is that one can compare the final rendered garment with the original neutral garment using the inverse of the warp. POVNet uses the resulting residual to correct minor errors, resulting in significant improvements in detail at high resolution.

In brief, POVNet made the following important improvements on top of the original OVNet [29]: (1) We use *distance transform* to significantly improve the coherence of the warp with respect to the body. (2) We recognize the difference between the warp garment and the generated image (the residual), and leveraging it to enhance the rendering quality. (3) We improve the formulation of the inpainting module.

II. Related Work

The general question of rendering people wearing prescribed garments admits a wide range of approaches. Key technologies are generating images of people and image warping. We review the literature through the lens of our properties. Faithful representation is a subtle and demanding property – we want any image generated from an example garment to be an image of a person wearing *that particular garment*, rather than some garment or a garment quite like it. Because current datasets usually contain at most one image of a person in a given garment, an adversarial loss can ensure that generated images look realistic, but cannot enforce the faithful representation requirement. There is a strong focus in the literature on image quality, but not all methods have mix&match properties. Scalability means that we focus on methods that use garment images, because alternatives – for example, obtain and use 3D models of person and garment [31], [32], [33] – are not at present scalable. Recent methods mostly study a variety of garments. Full model control is elusive, but some methods are able to vary models. Finally, recent methods tend to allow garment swapping.

A. Generating Images of People

Virtual try-on methods need to know some representation of pose and body shape of the target person. One might estimate the shape of the human body [34], [35], clothing items [36], [37] or both [38], [39] through 2D images. Tsiao et al. [40] learn a shape embedding to enable matching between human bodies and well-fitting clothing items. The DensePose [30] descriptor helps model the deformation and shading of clothes and has been adopted by recent work [41], [42], [43], [44], [45].

Zhu et al. [28] uses a conditional GAN to generate images based on pose skeletons and text descriptions of garments. SwapNet [46] learns to transfer clothes from person A to person B by disentangling clothing and pose features. Hsiao et al. [27] learn a fashion model synthesis network using per-garment encodings to enable minimal edits to specific items. Han et al. [44], [47] propose an inpainting method to complete missing clothing items on people. Dong et al. [48] introduce a framework that enables manipulation to person or garment attributes through sketch and color strokes using a novel conditional normalization method. Recently, Men et al. [49] propose a novel person image synthesis method, controllable through interpolating style and pose representations. Recently Cui et al. [26] proposed a method to iteratively dressing multiple garments on a model

[1] https://demo.revery.ai

with different styling option. However, the methods do not preserve structured spatial patterns (such as logos or prints) because they encode garment appearance into feature vectors to enable attributes manipulation. Chen et al. [50] extends try-on to different view points through sequence of poses, but the method have difficulties preserve garment attributes due to missing view points.

B. Image Warping

Image warping is the process of applying a parametric deformation to an image region; early methods are reviewed in [51], [52]. Spatial transformer networks estimate geometric transformations using neural networks [53]. Subsequent work learns networks to warp one object onto another. Warping works with images of rigid objects [54], [55] and non-rigid objects (e.g., clothing) [4], [5], [56]. While imputing a warp from a neutral garment image to a target image can be difficult, warping is extremely good at preserving details, and POVNet uses a warper for that reason.

At run-time, the warper must impute a warp from some representation of the garment image and the target. The choice of representation is important: there needs to be enough information so that the warper can tell when a garment is being distorted inappropriately. The standard choices [4], [5], [6], [7], [8], [9], [12], [13], [14], [19], [20], [21] are a combination of a binary layout mask and sparse body key points. This leads to problems, because the feature maps give relatively sparse information about the distortions in a warp.

Ayush et al. [16] improve the warping mechanism by incorporating 3D priors (through DensePose [30], which gives a dense representation of body configuration). He et al. [17] show that a flow-based architecture achieves better images. However, DensePose presents a challenge to the multi-garment settings: garments and poses are strongly correlated in training data and some garment-pose pairs are absent. For instance, if a user wants to render a missing garment-pose pair, there will be problems. For example, people wearing jackets are predicted to have significantly wider shoulders than people wearing shirts (see Supplementary Fig. 2). This effect is particularly significant when there are multiple garment categories. We discuss the effects in details in the Supplementary material, which is available online.

An alternative strategy is to provide a relatively rich representation of where points lie (roughly) on the source, so that the warper can avoid distortions that are unlikely. This is a form of positional encoding, now common in NLP [57], [58], [59] and shown to help convolution operators [60]. POVNet uses a distance transform procedure (after [61]) to control distortions produced by the warp and achieve faithful representation.

C. Image-Based Virtual Try-On

We review two classes of image-based methods – those that work on one garment (Section II-C1), and those that handle multiple garments (Section II-C2). Warping methods offer the best prospect of texture accuracy, but the specific details of how the warper is constructed are important. Obtaining high resolution synthesis presents particular challenges (Section II-C3).

TABLE III
This Table Reports the FID$_\infty$ [89] Score (Smaller is Better) of Our Method on the New Multi-Category Dataset. We Compare the Performance Using Different Numbers of Warps. Results Show That Using More Warps Yield Higher Quality Synthesis and the Quality are Mostly Consistent Across Category (Except for Full-Body Which is Under Presented in the Dataset)

warp	bottoms	full-body	tops	outerwear	overall
1	1.837	4.053	2.194	1.983	1.374
2	1.438	2.219	1.314	1.337	.904
4	1.412	**1.947**	**1.042**	1.309	.846
8	**1.407**	1.954	1.068	**1.277**	**.845**

1) Single-Garment Virtual Try-On: Single garment virtual try-on methods (SG-VITON) map a single garment onto a model image with emphasis on the faithful representation garments' identity. There is a strong emphasis on warping methods. VITON [4] first proposed using a thin plate spline (TPS) transformation to create a warp, followed by a generation network to synthesize the final output. CP-VTON [5] improves this method by using a differentiable component for TPS transformation. Other improvements are proposed to stabilize the TPS warper for the task [9], [15]. Han et al. [8] uses a flow estimation network to enable more degrees of freedom for the warp. Issenhuth et al. [12] propose a teacher-student training paradigm to warp without relying on human parsing and Ge et al. [14] further refine the method using a different formulation for knowledge distillation. To enable shape changes (e.g., short sleeve to long sleeve), a common procedure has been to predict a semantic layout of body segments and clothes to assist with image generation [8], [9], [11], [62], [63]. Other works propose architectural improvements toward better preservation of details [63], [64] and adding adversarial training during the refinement phase to improve image realism [7], [9], [62], [63]. Ge et al. [13] propose using cycle-consistency to improve try-on results. The virtual try-on task has also been extended to multi-view scenarios and videos [56], [65]. Others follow similar procedures [66], [67], [68].

Generally, these methods faithfully represent garment properties, can produce high quality images, and are scalable, at the cost of working with single garments of a single type (mostly tops). POVNet extends warping to preserve these properties, while working with multiple garments over multiple categories and allowing garment swapping.

2) Multi-Garment Virtual Try-On: Multi-garment virtual try-on is more challenging than SG-VTON, because one must ensure proper layering and accurate modeling of the interactions between garments. O-VITON [10] constructs a visual feature encoding which is broadcast into a layout using a learned procedure that allows interactions between multiple garments. The visual feature encoding causes a loss of texture detail, but an online optimization step (fine-tune a generator for every query) can repair this loss, at the expense of scalability. Model control is not available, nor is garment swapping. Morelli et al. [18] claimed to support multi-garments try-on, but the work lacks

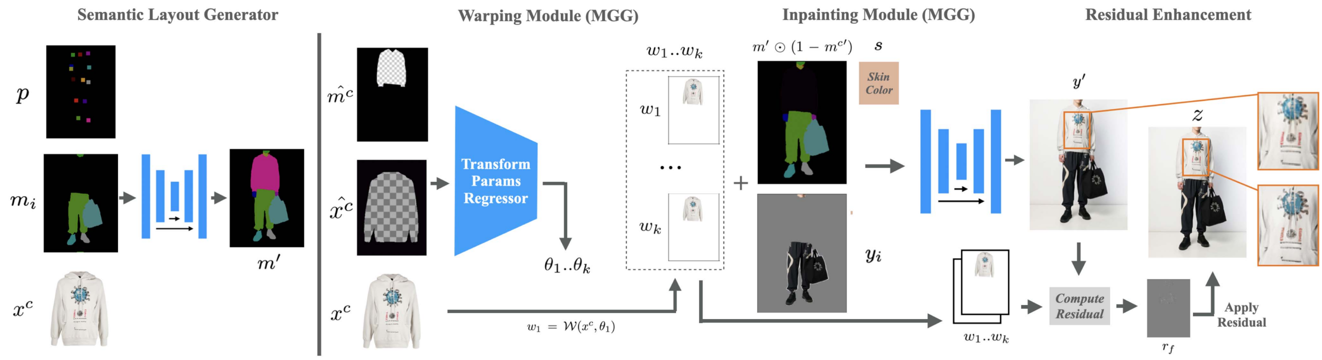


Fig. 5. POVNet accepts a starting image y of a model wearing clothing and an optional desired skin tone t. POVNet produces a new image y' where the model is wearing a neutral garment of type c depicted in a neutral garment image x, as well as all clothes not of type c shown in y. The model in y has skin tone t if t is provided, but otherwise has the original skin tone. There are two main components. The *semantic layout generator* (SLG - G_{layout}) predicts a semantic layout (m'), a segmentation map indicating where garments lie in the image. G_{layout} accepts x, a map of the pose (p) of the model in y, and an initial layout (m_i). The resulting m' is used to guide image generation, particularly to resolve details of garment overlap. The *multi-warp garment generator* (MGG - $G_{garment}$) produces y'. This generator accepts m' from the semantic layout generator, x and y. It operates in two stages: first, a *warping module* produces several interacting warps of the neutral garment to the target image; second, an *inpainting module* (trained with an adversary) fuses these images together with partial layout information and t to produce a convincing image. Finally, we apply a *residual enhancement* technique to improve rendering quality by restoring the missing details. We first compute the residual r – the difference between the warps $w_1 \ldots w_k$ and generated image y' conditioned on the garment region. Then, we retrieve the high frequency patterns r_f from the residual r and adding these details back to y', resulting in the enhanced image z (see Fig. 7 for details).

TABLE IV
THIS TABLE REPORTS THE INFERENCE TIME OF OUR METHOD FOR OUTFITS OF DIFFERENT NUMBERS OF ITEMS. THE RESULTS ARE MEASURED ACROSS 200 SAMPLES. THE RESULT SHOWS THAT OUR METHOD CAN DELIVER (4) REAL-TIME RESPONSE

Number of items in the outfit	Inference Time (Second)
1 item (Dress)	0.378 ± 0.023
2 items (Top + Bottom)	0.502 ± 0.024
3 items (Top + Bottom + Outerwear)	0.617 ± 0.032

descriptions of how it handles the interaction between multiple garments during inference.

In contrast, POVNet follows SG-VTON methods by adopting a warping based approach to achieve significantly better quality (Fig. 12) with high efficiency (see Table IV for running time analysis). POVNet uses a human parse to sort out the layering effects allow us to support interactive swapping.

3) High Resolution Virtual Try-On: Resolution is difficult. The industry demands high-resolution imagery to highlight garment details, but research methods operate at relatively low resolution [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [15]. One might apply single-image super-resolution methods to virtual try-on results [69], [70], [71], but such methods must be guessing at fine texture details. Choi et al. [72] demonstrate virtual try-on at 1 k resolution using a residual architecture. The method produces realistic 1 k resolution images, but cannot faithfully represent garment properties (e.g., [72], Fig. 1). VITON-HR [21] prevents occlusion through an improved architecture that predicts the warp and layout jointly. However, the method stretched the garment textures in undesired ways(e.g., [21], Fig. 1 bottom right, white flowers on the left sleeve and bottom torso went missing). In contrast, POVNet uses a residual to obtain high resolution images without loss of faithful representation.

III. Precise Outfit Visualization Net

We have designed Precise Outfit Visualization Net (POVNet) to support the essential properties for VTON methods to be commercially successful. To achieve faithful representation and fast inference, our image generator adopts a warping-based approach instead of an encoding/embedding-based approach because of its obvious advantage in detail preservation and speed of inference. To support a variety of garments and obtain high quality images of open outerwear, we use a novel warping procedure that coordinates multiple warps. To enable mix & match of garments, layering effects and interactive edit of outfits, we use a novel framework for predicting partial layouts based on garments and poses, and allow manipulations of those layouts during inference. To ensure faithful representation across garment categories, we use a distance transform to produce a feature representation that enhances warping accuracy. Finally, to obtain high accuracy at high resolution, we use a residual enhancement strategy.

The semantic layout generator (SLG) G_{layout} (Fig. 5 left) must produce a detailed semantic segmentation of the image to be generated. Each pixel must be labelled with background or the type of the garment on that pixel. The layout must be realistic, and must be guided by the model's pose.

The multi-warp garment generator (MGG) $G_{garment}$ (Fig. 5 right) must produce a realistic image of a model wearing a specified garment. The MGG accepts a model image y, a layout m registered to the model image and produced by the SLG, and a garment image x^c of class c and predicts the realistic image. $G_{garment}$ consists of two modules – a warper, and an inpainting module.

A. The Semantic Layout Generator

The SLG is a U-Net which accepts a neutral garment image x^c, an initial (incomplete) layout m_i, and a pose p, and predicts a complete layout $m' = G_{layout}([x^c, m_i, p])$. Using an incomplete layout forces the U-Net to generalize.

The initial layout m_i is a layout that hides the details of one garment. For example, we choose to generate a layout associated with a top; then m_i is obtained by taking the ground truth layout m and setting the top, neckline, and arm classes to the background class. Further details of the procedure (see Supplementary material).

The SLG is trained with pairs of neutral garment images x^c and images y of models wearing that garment. The semantic layout m of y is recovered using an off-the-shelf human parsing model [73]. The pose map p is recovered using OpenPose [74], [75], [76], [77]. The initial layout m_i is obtained by setting the pixels in m labelled with the category of c to the background class. The network is trained using a pixel-wise cross-entropy loss and an LSGAN [78] loss to encourage the generated semantic layouts to resemble real semantic layouts. The total training loss for G_{layout} is then

$$\mathcal{L}_{layout} = \lambda_1 \mathcal{L}_{CE} + \lambda_2 \mathcal{L}_{GAN}, \tag{1}$$

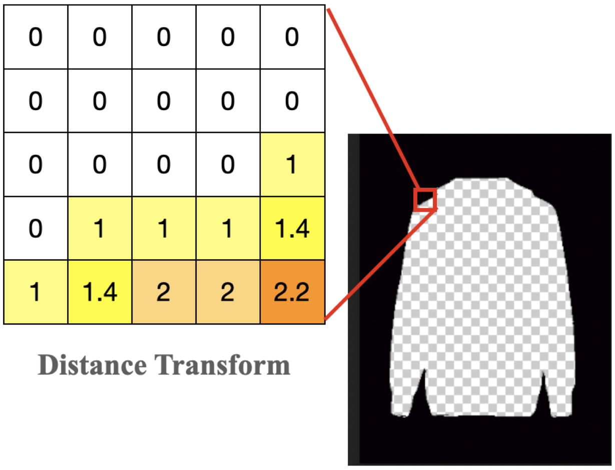
where λ_1 and λ_2 are the weights for each loss. Because the argmax function is non-differentiable, we adopt the Gumbel softmax trick et al. [79] to discretize the layout generator's output such that the gradient generated by the discriminator can flow back to the generator.

B. MGG: The Warping Module

The warper aligns the garment image x^c with the semantic layout of the garment class m^c (obtained as the c-labelled pixels in m). The warper uses distance transform features to control deformation of the garment. In contrast to prior work (which uses a single warp [4], [5], [8], [9]), the warper uses multiple coordinated warps, each of which has relatively few degrees of freedom, rather than a single warp with many degrees of freedom. Each warp is not required to fit perfectly as long as the other warps can make up for the misaligned regions. There are several advantages: warps with fewer parameters are easier to estimate; rigid warps are easier to regularize and more robust to non-regular shapes; multiple warps can naturally handle disconnected regions (e.g., split outerwear).

The inpainting module must generate the final image given all the warps, the predicted semantic layout m', the skin color of the model s (median color of the face), and the incomplete model image y_i where the target garment, skin, and background are masked out. The inpainting module is trained jointly with the warper, and so learns to combine warps. Other garments are kept to support garment swapping.

1) Distance Transform Features: Garments deform significantly, so the warp must deform the source significantly, but many kinds of deformation are not acceptable – for example, stretching a short skirt so that its hem lies at the ankle wholly misrepresents the garment. This means that learning to warp a neutral garment image to the position of a target mask accurately is very challenging, and in turn learning the warper may be simplified by providing features that help distinguish between acceptable and unacceptable warps. Prior work has used a binary mask of the target region(s) and human posture to as features to inform the warper [4], [5], [6], [7], [8], [9], [12], [13], [14], [16], [17], [18], but this approach denies the warper information about the details of deformation. For example, a binary mask feature cannot inform the warper when cloth is being overstretched in the interior of the garment.



0	0	0	0	0
0	0	0	0	0
0	0	0	0	1
0	1	1	1	1.4
1	1.4	2	2	2.2

Fig. 6. The figure shows an example of distance transform $\hat{x}$ for garment image x. The background region of the garment is filled with zeros. The distance transform of each foreground pixel is computed as the euclidian distance between itself and its closest background pixel. The distance transform $\hat{m^c}$ for the semantic layout is computed through the same procedure.

The distance transform for any pixel in the mask region is computed as the euclidean distance between itself and its nearest border pixel, as shown in Fig. 6. We replace the binary mask of the target region m^c with a distance transformed version $\hat{m^c}$, and provide a distance transform $\hat{x^c}$ for the source garment image x^c as well. The advantage of doing so is that a simple convolutional layer can determine the strain on the cloth implied by a particular warp by looking at derivatives of the distance transform. There is a scale issue: the garment layout on the person m^c is usually smaller than the garment image x^c. To avoid difficulties, we normalize the distance transform by the square root of the total number of non-zero pixels in the distance transform map.

2) The Warper: The warper network resembles a spatial transformer network [53]. A regressor takes in the garment image x^c, its distance transform $\hat{x^c}$ and the distance transform of the mask $\hat{m^c}$, and predicts k sets of spatial transformation parameters $\theta_1 \ldots \theta_k$. It then generates a grid for each set of transformation parameters, and samples grids from the garment image x^c to obtain k warps $w_1..w_k$ where $w_1 = \mathcal{W}(x^c, \theta_1)$. The warps are optimized to match the garment worn by the target model $m^c \odot y$ using per pixel $\mathcal{L}_1$ loss. Inspired by [8], we impose a structure loss to encourage the garment region z (a binary mask separating garment foreground and background as in Fig. 5) of x^c to overlap with the garment layout of the garment mask m^c on the model after warping. The warping loss is then

$$\mathcal{L}_{warp}(k) = |\mathcal{W}(x, \theta) - (m^c \odot y)| + \beta|\mathcal{W}(z, \theta_k) - m^c|, \tag{2}$$

where β controls the strength of the structure loss. This loss is sufficient to train a single warp baseline method. The choice of warper here is not crucial, but in our implementation, we use 2D affine transformations. Our choice of warps are more rigid, and thus easy to estimate.

3) The Cascade Loss: When $k > 1$, the j'th warp w_j is trained to address the mistakes made by previous warps w_i where $i < j$. For the jth warp, we compute the minimum loss among all the previous warps at every pixel location, written as

$$\mathcal{L}_{warp}(j) = \frac{\sum_{u=1,v=1}^{W,H} \min(\mathcal{L}_{warp}(1)_{(u,v)}..\mathcal{L}_{warp}(j)_{(u,v)})}{WH}, \tag{3}$$

where u, v are pixel locations; W, H are the image width and height; and $\mathcal{L}_{warp}(r)_{(u,v)}$ is the loss of the rth warp at pixel location u, v. The cascade loss computes the average loss across all warps. An additional regularization term is added to encourage the transformation parameters of all later warps to stay close to the first warp. Our final cascade loss is written as

$$\mathcal{L}_{casc}(j) = \frac{\sum_{i=1}^{j} \mathcal{L}_{warp}(i)}{k} + \alpha \frac{\sum_{i=2}^{j} \|\theta_j - \theta_1\|^2}{j-1}. \tag{4}$$

At each pixel location, the loss is the minimum among all warps, meaning if one warp does well in a region, other warps are not penalized for making mistakes in the same region. In practice, we observe 80% of the garment pixels are taken from the 1st warp. The subsequent warps often fill in the irregular regions caused by the complex interactions between the garment and the person (e.g., the gap between the first warp and the arm, the halves of open outerwear, the edge of neckline/hemline, etc.). The cascade loss also enforces a hierarchy among all warps, making it more costly for an earlier warp to make a mistake than for a later warp. This prevents oscillation during the training (multiple warps competing for the same objective).

The idea is comparable with boosting – using multiple simple warpers (weak learners), each with a small degree of freedom can handle complex geometric shapes when combined. Warpers interact with each other differently compared to classifiers – all the warps are predicted in parallel (not sequentially) in the forward pass of the network. But the *loss* for each warp is computed in a cascade (sequential) manner to coordinate multiple warps. This means that at training time the generator can reason about geometry, but at test time the warps are quickly computed. Training the warper and the image generator jointly allows the warps to adjust according to each other and the image generator to guide the warpers.

C. MGG: The Inpainting Module

The inpainting Module accepts all warps $w_1..w_k$ applied to the neutral garment image, the semantic layout without the garment mask $m \odot (1 - m^c)$ (or $m' \odot (1 - m^{c'})$ during inference), and the incomplete image y_i, and produces the final image y' of the model wearing the neutral garment x^c and all other garments shown in y. This is not a standard inpainting task because the exact content to inpaint is provided through the input channels, but may be in the wrong place (in contrast to filling in missing portions of an image [80], [81], [82], [83]). We use a U-Net architecture to encourage copying information from the input. In addition to the cascade loss $\mathcal{L}_{casc}$, we train the network to reconstruct the ground truth image using a per-pixel $\mathcal{L}_1$ loss, a perceptual loss $\mathcal{L}_{perc}$ [84], and a Spectral Norm GAN with hinge loss $\mathcal{L}_{GAN}$ [85]. The total loss for training $G_{garment}$ with k warps is written as

$$\mathcal{L}_{garm}(k) = \gamma_1 \mathcal{L}_{casc}(k) + \gamma_2 \mathcal{L}_1 + \gamma_3 \mathcal{L}_{perc} + \gamma_4 \mathcal{L}_{GAN}, \tag{5}$$

where γ_1, γ_2, γ_3 and γ_4 are the weights for each loss.

Note an important modification over the original OVNet [29] – the inpainter does not see m' or $m^{c'}$. Following He et al. [17], the generated garment layout m' is often smoothed and inaccurate, and so providing the $m^{c'}$ as input encourages the network to adopt the $m^{c'}$, often resulting in a loss of details in the garment shape. In contrast, we knock out $m^{c'}$ and provide $m' \odot (1 - m^{c'})$ as the input – so the type of garment that must be placed does not appear in the mask. Because the garment warps computed through the distance transform already suggest the position of the garment, the inpainting network does not require the garment layout to generate the final output image. These modifications yield significant improvements both qualitatively (Fig. 9) and quantitatively (Fig. 1).

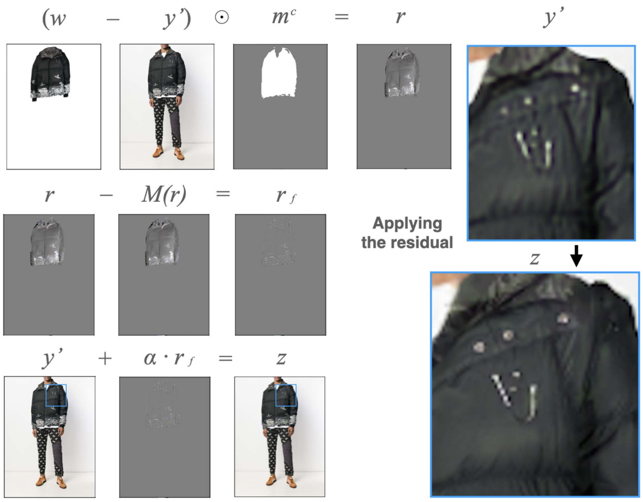


Fig. 7. The residual enhancement procedure computes the difference between the warped garment image w and the generated image y'; this is masked using the binary mask for the garment type (obtained from m) to produce the residual r. This is median filtered to yield $r_f = r - M(r)$, an estimate of missing high spatial frequencies. Finally, we add r_f back to the generated image y', and multiply by a hyper-parameter α to control the magnitude. As the figure shows, both the letter "VJ" and the buttons become visible after the augmentation.

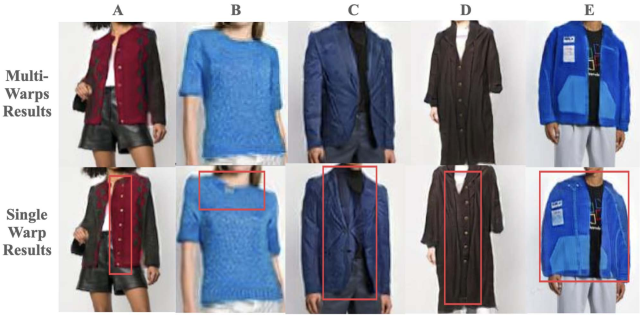


Fig. 8. Two or more warpers are better than one. When using a single warp: the buttons are in the wrong place in A and D; there are problems with sleeve boundaries in E; there is a severe misalignment in C; and there is a misplaced tag in B. All problems are fixed in multi-warp results.

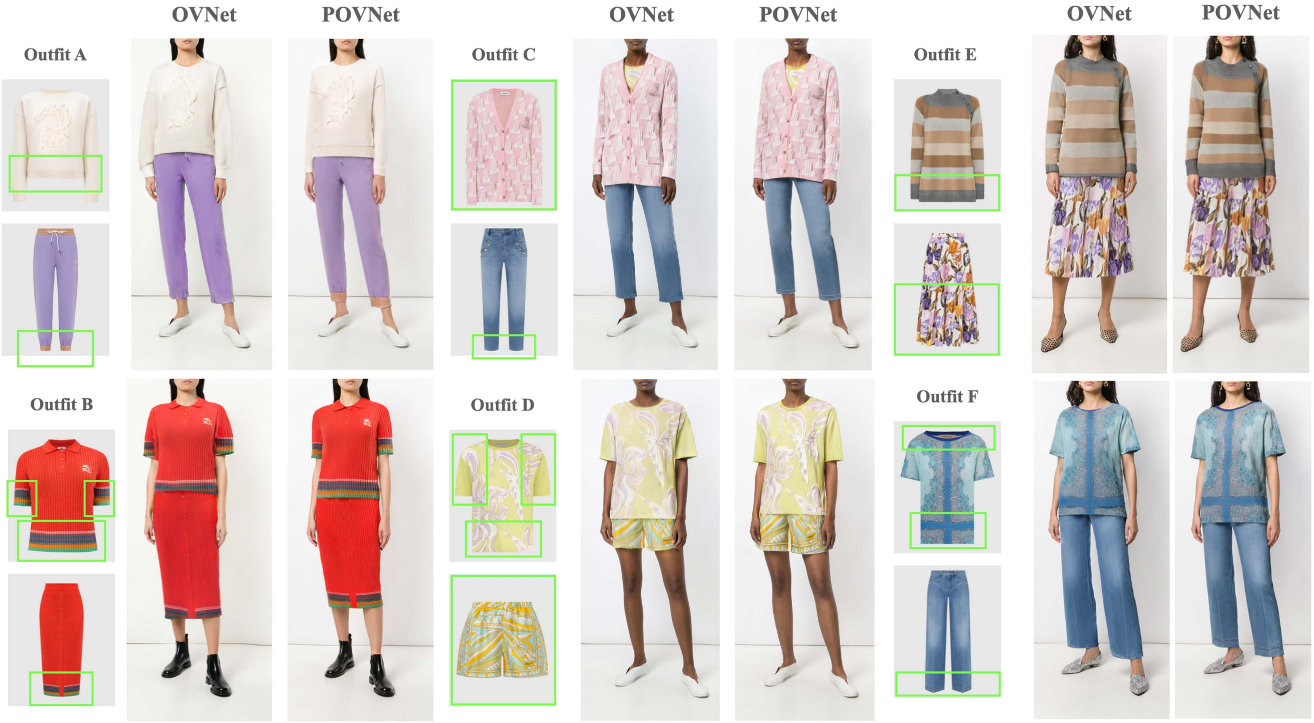


Fig. 9. Qualitative comparison between POVNet and an earlier version (OVNet - no residual enhancement and no distance transform) significantly favors POVNet. While differences may appear small, POVNet faithfully represents garments in ways that OVNet does not. For example, in outfit A OVNet crops the trouser cuffs and POVNet does not; in E, OVNet crops the hem – so misstating the pattern of the garment, and POVNet does not. Similar effects on trouser cuffs/hems in B, C, F. As another example, the layout of texture on the blouse in F is wrong for OVNet (crossing too close to hem) and right for POVNet. Similar effects in C, D, E. Garments tend to follow the body pose better with POVNet. For example, the blue cross is slightly off center in POVNet F (as it should be), but not in OVNet F. Similar effects in B, D. Improvements to fine detail like this are difficult to capture with FID, but are crucial to shoppers who rely on the images to understand the appearance of the garments. Errors can result in expensive returns; for example, a shopper who bought the jersey in E based on the OVNet image might return it for having too thick a gray stripe at the bottom.

D. Residual Enhancement

Our residual enhancement is a scheme to ensure generated images show garment detail at the highest available resolution. Consider pixels lying on the generated garment in the generated image. The pipeline has produced these pixels by applying a set of warps to the neutral garment image, then combining these pixels with other information in the inpainting module. This means that the generated garment is different from the warped neutral garment, and comparing the two provides a residual. This residual is informative because the inpainting module tends to supply shading details, which are at relatively low spatial frequency. In turn, any high spatial frequency residual components are a sign of missing texture detail.

Residual enhancement allows us to produce generated images at a resolution higher than the training resolution, when neutral garment images are available at that resolution. While the quantitive improvement is small (Table II), there is good qualitative evidence that details are better preserved (Figs. 7, 10, and 11). The procedure is simple and quick, and helps ensure faithful representation.

1) Simple Residual Enhancement: For a single warped version w of a neutral garment image, the residual is straightforward to compute. We compute $r = (w - y') \odot m^c$ (recall y' is the generated image and m^c is a mask that preserves all pixels of garment type c). Here m^c is recovered from the SLG. We now compute $r_f = r - M(r)$, where M is a median filter. Finally,

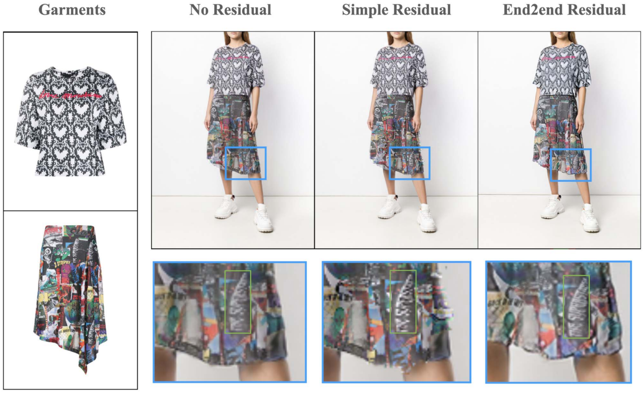


Fig. 10. A qualitative comparison between not using residual enhancement, using simple heuristic-based residual enhancement, and using residual enhancement with end-to-end training shows that residual enhancement can produce better quality images. Note that letters and the thin lines become more visible with even the simple enhancement. When the residual is applied using a heuristic, there are minor artifacts on the boundary (the edge of the skirt) that appear unnatural in certain regions. Using end-to-end training can produce smooth edges for the residuals. Note the vertical words "THE SUPERSTARS" become legible (greenbox).

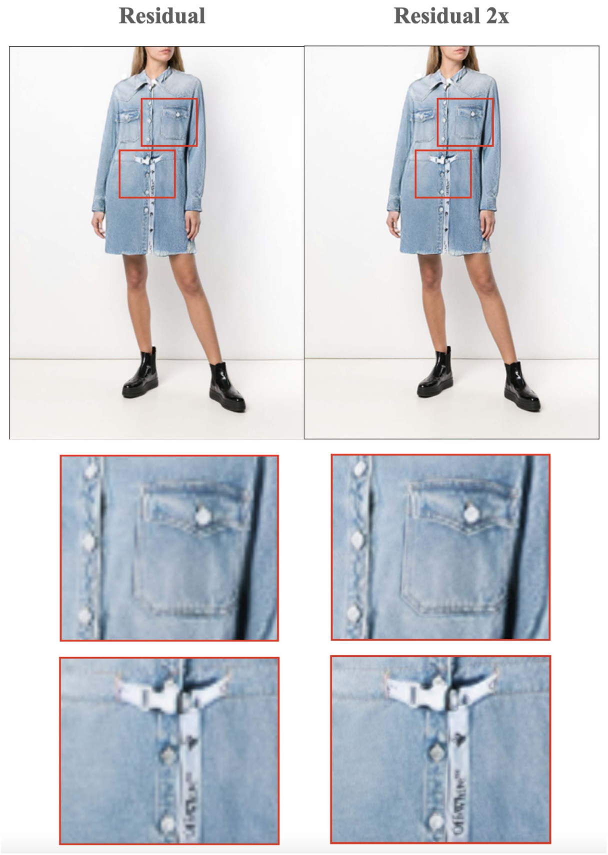


Fig. 11. Residual enhancement allows our pipeline to produce images at resolutions greater than those it was trained for (super-resolution). Given a high-resolution x, we can generate a y where the top stitches around the pocket became clearly visible (here x is 2 x training resolution). The "Off White" text on the waistline belt also becomes more readable (zoom in to see). Details on the skin are not augmented because there is no residual to copy from, but garment details are more important than skin details for a virtual dressing room.

the augmented image is $z = y' + \alpha r_f$. The full procedure is illustrated in Fig. 7 and Algorithm 1 (supplementary material).

When there are multiple warped versions w_i of the neutral garment, computing the residual requires care. One must decide which warp accounts for which pixel in the generated image. A reasonable procedure is to rank the warps by the magnitude of the residual in ascending order, then apply the residual from warps in order. When applying the residual for a given warp, we exclude any pixels that already have a non-zero residual applied from a previous warp as illustrated in Algorithm 2 available online. This means that, in general, at each pixel the most accurate warp with a high frequency residual component will be used to correct the pixel. This heuristic procedure has drawbacks. It produces residual artifacts at the boundary of the garment (Fig. 10), likely because the boundaries of the semantic layout do not always exactly match those of the generated image. Furthermore some regions can have unnatural looking high brightness (Fig. 10), likely because the filter is not perfect.

2) End-to-End Training of Residual Enhancement: A more sophisticated use of the residual is to learn a pixel classifier network to that predict the residual from which warp should be applied at each pixel location. The residual classifier network consists of two convolution layers with ReLU activation following the intermediate layer and Softmax activation following the final layer. The network takes the features from the last hidden layer of the garment generator $G_{garment}$ as input and outputs a 4D tensor u of shape (B, $k+1$, W, H) where B is the batch size and k is the total number of warps. We then compute a softmax of the $k+1$ channels. Write s_i for the resulting softmax value for the ith channel; $s_1 \ldots s_k$ correspond to each of the warps, and s_{k+1} allows the residual to be zero. We now obtain the residual $r = \sum_{i=1}^{j} s_i \odot r_{f\,i}$ by multiplying the softmax for each channel with the residuals computed from the corresponding warp and summing (Algorithm 3 in supplementary material). All this is differentiable, so we can train the multi-warp residual enhancement jointly with POVNet using identical training losses. The adversarial loss is able to capture and suppress the artifacts and smoothen the residual as shown in Fig. 10.

3) Garment Texture Super Resolution: Now assume we have a neutral garment image x_h at resolution higher than that used in training. Residual enhancement allows us to use this resolution without retraining the whole network. At run-time, we downsample x_h to get x at the training resolution. We apply the pipeline to obtain appropriate spatial transform parameters for each warp, then apply these spatial transforms to x_h to obtain high resolution warps $w_{1\,h}, \ldots w_{kh}$. We then upsample the generated image y' to obtain y'_h, and apply residual enhancement to y'_h using the high resolution warps. While this procedure yields accurate high resolution garment images without needing to be trained on high-resolution images, it cannot produce high resolution skin or background. Users focused on garments may not be bothered by lower resolution skin.

IV. Experiments

A. Datasets and Experiment Setup

To demonstrate that our method works with various garment type, we experiment on a dataset of 321 k fashion products obtained from several fashion e-commerce websites through affiliate marketing program. Revery.AI has the right to use these images through an affiliate partnership with the source websites. The dataset contains all the available garment categories. Each product includes a neutral garment image (front-view, laying flat, plain background), and a model image (single person, front-view). Garments are grouped into four types (top, bottoms, outerwear, or full-body). We randomly split the data into 80% for training, 5% for validation, and 15% for testing. Because the model images do not come with body parsing annotation, we use off-the-shelf human parsing models [73] to generate semantic layouts as training labels.

We also compare with single garment methods by training and testing our method on the VITON dataset [4]. Following prior work, we report SSIM [86], Inception Score (IS) [87] and FID [88] score on the original VITON test set [4].

We compare with Dress Code [18] and the original OVNet [29] on the multi-category dataset. We report Frechet Inception Distance Infinity [89] (FID$_\infty$) as Chong et al. [89] shown that it is a more reliable metric than FID [88]. Other details about network architectures, training procedures, and

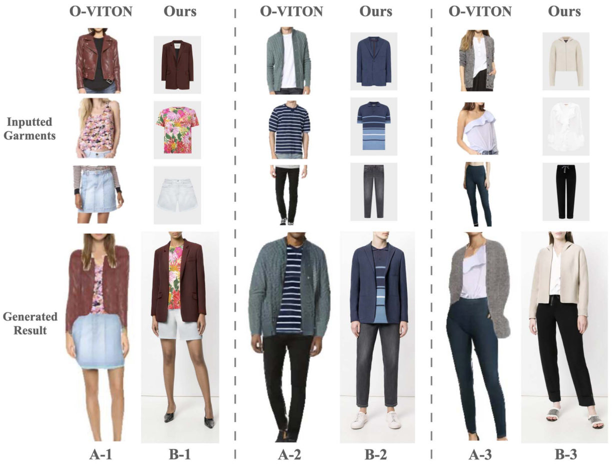


Fig. 12. Qualitative comparison between O-VITON [10] and POVNet favors POVNet. Particularly significant are improvements in faithful representation and image quality. The top rows show the garments in the outfit and the bottom row shows the generated try-on results. For a fair comparison, we found garment images that most closely resemble the garments chosen in [10] in terms of style, color, and texture. Image results for O-VITON are directly taken from their paper as no code is available. There is a substantial difference in quality between results. The unnaturally flat torso and uneven shoulders of A-1 are not present in B-1. In A-2, the buttons on the jacket are distorted/missing, whereas B-2 represents them accurately. In A-3, the jacket and top lack realism due to missing creases, folds, and bumps compared to B-3. Properties of the arms are also kept intact in B-3.

hyper parameters are provided in the Supplementary material. Quantitative comparison against Neuberger et al.'s [10] is impossible because their implementation is not released. Thus, we compare with them qualitatively on Fig. 12 and more extensively in the Supplementary, available online.

We implement a variation of our method – POVNet+DensePose – by adapting the warper and the image generator of Style-Based Flow [17] (which consumes DensePose) to our pipeline (see Supplementary for implementation details, available online). We evaluate POVNet+DensePose on both datasets to understand how using DensePose impacts the performance on single-garment versus multi-garment setting.

B. Results

Faithful Representation. Detail preservation is crucial but difficult to evaluate through quantitative metrics. Thus, we rely mostly on qualitative figures. Evaluation can be quite subtle. One should check to see that in all the figures, the exact details (such as the prints, the patterns, the logos, the color, the trims, the ribbons, the collars, the pockets, the buttons, etc.) are all preserved. Fig. 9, demonstrates POVNet improvements over the original OVNet, largely obtained by the distance transform. The ribbons and the cuffs are often cut off or slightly misaligned in OVNet, but are perfectly aligned in POVNet (for example, outfits A, B, E and F); the internal patterns of the garment are sometimes stretched in OVNet but are perfectly aligned in POVNet (for example outfits C, D and E). Having multiple warpers is helpful and Fig. 8 shows that having multiple warpers can significantly improve and correct the details for split outerwear. The buttons and the zips in the center are often cut-off or misaligned in the single warp example but are addressed with multiple warps. Fig. 2 demonstrates that we preserve garment attributes consistently across different poses and models. Pay attention to the coherence of the length, shape, textures and details of the garment.

Image Quality. Using standard image quality metrics, our method outperforms prior work in single and multi-garment try-on. (Tables I and II). Qualitative comparison against O-VITON [10] (Fig. 12) shows that our synthesis appears much more realistic: the natural split of the blazer, the shape edge, and the casted shade on each side of the short (A-1 versus B-1); the natural overlay of the blazer, the smooth interaction between the neck and the collar and convincing fabric properties (A-2 versus B-2); the natural weight shift of the pose and details of the hands and skin (A-3 versus B-3).

An ablation study performed against the original OVNet [29] reveals how the different components improve the generation quality. As shown in Table II: adding distance transform yields significant enhancement as the warper gains access to more meaningful feature representations; removing the garment layout from the inpainting module yields better results because the network can rely on the warp to synthesize the garment shape rather than relying on the predicted mask. Residual enhancement also yields improvements but is less prominent on the quantitative metrics. Note that using DensePose yield worse results (even though it outperforms on VITON dataset). This confirms our hypothesis that the bias in human body representation can negatively impact the generalization of multi-garment try-on methods. See Supplementary for more evidence, which is available online.

Qualitatively, residual enhancement is helpful to preserve fine-grained details of the garment (shown in Fig. 10). The heuristic procedure yields minor artifacts, but these are fixed by the end-to-end trained version. Fig. 11 shows an example of super-resolution. By applying the residual using a higher resolution garment image, we recover details that are previously unidentifiable.

Mix & Match. Figs. 1, 2, 3, 9, 12 and other examples in the Supplementary, which is available online, show a large number of outfits in neutral garment images paired with on-model images synthesized by our method. These examples demonstrate that our method can render arbitrary combination of garments.

Scalability. Our system is efficient enough to power an interactive interface, as Table IV shows. The latency of our inference is around half a second on a 1 NVIDIA Tesla T4 (AWS's slowest GPU). As our inference is iterative, the inference time is longer when there are more garments in the outfit, with 3 garments outfits being 0.61 s. Each T4 machine can process 4 inference jobs in parallel.

Garment Variety. In our qualitative examples, we showed examples of various garment types (blouse, t-shirt, shirt, tank tops, sweaters, hoodies, blazers, jackets, coats, trousers, shorts, skirts, full-body garments, etc.) across both genders to demonstrate the wide range of garments we support. Table III shows that our method performs mostly consistently on tops, bottoms, outerwear, and less well with full-body garments (which are rare in the dataset). Fig. 8 shows that the multi-warp method improves rendering of split outerwear, which single warp methods find difficult.

Model Control. Fig. 4 shows that our method can render an outfit selection on a chosen model with a diverse set of poses. The garments' properties are consistent across all poses (e.g., changed stances and hand/arm positions), suggesting that the network has learned a robust garment representation. The skin-tone, body ratio and appearance also appear consistent across different images. Pay attention to Pose 2 & 5 when the hands are in the pockets; the sleeve interacts seamlessly with the pocket. Fig. 2 shows that our system can render on models of different skin tone and hair styles. Full model control remains elusive because managing the effects of body shape is so hard.

Garment Swapping. Fig. 3 shows our method supports interactive swapping of garments. Note when one garment changes, the other garments and the model's pose remains consistent. The interaction between garments appears natural, with a shadow cast by the garment when it is untucked and a clear waistline when the garment is tucked. Note the inpainting module changes the shaded area when the model swaps from a skirt to slacks.

Bias in Body Pose Representation. Results show that POVNet+DensePose outperforms POVNet on the single garment VITON dataset (Table I) but performs worse than POVNet on the multi-garment dataset (Table II). This validates our assumption that using DensePose is suboptimal on multi-garment try-on settings because DensePose representation is biased by the garment worn on the person (Supplementary, Fig. 2, available online). Although OpenPose also has such bias (Supplementary, Fig. 3, available online), Openpose's bias is more subtle than DensePose because its representation is simpler. Thus, we recommend using OpenPose over DensePose for multi-garment try-on. See Supplementary Section 2 for more discussions, which is available online.

C. A Case Study

We performed a live study to demonstrate that our method is ready to power a live virtual dressing room interface for commercial use. We partnered with Zalora, one of the largest fashion e-commerce platforms in South East Asia. We deployed a virtual dressing room interface that allows users to mix & match any garment combination and visualize the outfit on a model (with an example in Fig. 13).

During a 3-month pilot, about 103 k users (3.5% of the site's traffic) interacted with the experience during the pilot. The small percentage of adoptions makes an A/B test difficult to show significance. Instead, we identified the cohort of users who engaged with the dressing room and compare it with their engagement and conversion rate before and after they adopted the dressing room. Results in Table V shows that the interface powered by the proposed method significantly increases the user engagement and conversion rate. The results strongly indicate that the proposed method can support commercial applications by satisfying most of the requirements defined in the Introduction.

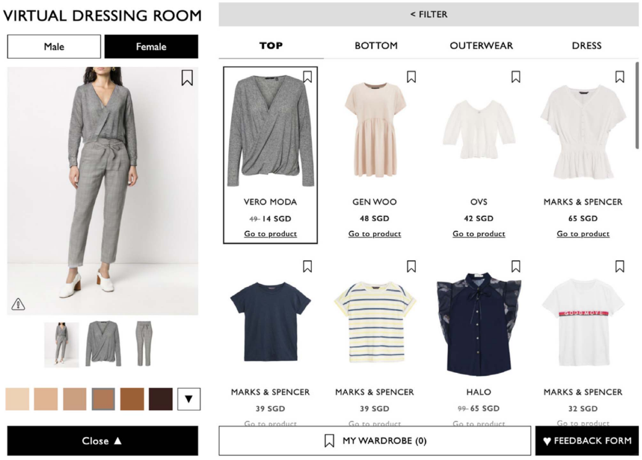


Fig. 13. A virtual dressing room interface powered by our method, deployed on a fashion e-commerce website (Zalora). Shoppers can click on any garment on the right side and the system instantly renders an image of the selected model wearing the chosen garment. Shoppers can also choose between models of different skin tones and ethnicities. A live demo is available at https://demo.revery.ai.

TABLE V
THE TABLE SHOWS USER ENGAGEMENT AND CONVERSION STATISTICS COLLECTED DURING A 3-MONTH PILOT OF A VIRTUAL DRESSING ROOM (VDR) INTERFACE POWERED BY OUR METHOD. RESULTS SHOW THAT THE SAME COHORT OF USERS SPEND ABOUT 1.4X TIMES LONGER ON THE SITE AND HAVE A CONVERSION RATE INCREASE OF 21% AFTER THEY ADOPT THE VIRTUAL DRESSING ROOM

	Session Length (MM:SS)	Conversion Rate
Without VDR	04:03	-
With VDR	**05:29**	**+21.3%**

V. CONCLUSION

In this work, we outlined the 7 important characteristics to enable commercially viable virtual dressing room experience and propose a framework that meets all the major requirements (with an emphasis on detail preservation and multi garment try-on). Several design choices are crucial: combining a warping based generation method and using human parsing to allow layering, coordinating multiple warps, leveraging distance transform to improve the accuracy of warping, using the residuals further enhance the details, etc.

Despite the success, our method can be improved in many aspects. Our method can handle variations in body pose, hair style and skin tone, but not body shape, variations. The other aspect we do not address yet are facial expressions, shoes, bags and accessories. Enabling these would get us one step closer to a full virtual fitting room experience that can directly work with consumers' photos (a very challenging task). To solve this problem, other challenges involves a the main challenge lies in handling out of distribution user-uploaded photos. Additionally, enabling try-on for shoes, bags, and other accessories would make the outfit generation complete.

References

[1] A. Balchandani et al., "State of fashion 2022: An uneven recovery and new frontiers," 2022. [Online]. Available: https://www.mckinsey.com/~/media/mckinsey/industries/retail/our%20insights/state%20of%20fashion/2022/the-state-of-fashion-2022.pdf

[2] K. Vaccaro, T. Agarwalla, S. Shivakumar, and R. Kumar, "Designing the future of personal fashion," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2018, Art. no. 627.

[3] T. Zhang, W. Y. C. Wang, L. Cao, and Y. Wang, "The role of virtual try-on technology in online purchase decision from consumers' aspect," *Internet Res.*, vol. 29, p. 5, Feb. 2019.

[4] X. Han, Z. Wu, Z. Wu, R. Yu, and L. S. Davis, "VITON: An image-based virtual try-on network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7543–7552.

[5] B. Wang, H. Zheng, X. Liang, Y. Chen, and L. Lin, "Toward characteristic-preserving image-based virtual try-on network," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 607–623.

[6] I. Rocco, R. Arandjelović, and J. Sivic, "Convolutional neural network architecture for geometric matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 39–48.

[7] H. Dong, X. Liang, K. Gong, H. Lai, J. Zhu, and J. Yin, "Soft-gated warping-GAN for pose-guided person image synthesis," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 472–482.

[8] X. Han, X. Hu, W. Huang, and M. R. Scott, "ClothFlow: A flow-based model for clothed person generation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 10470–10479.

[9] H. Yang, R. Zhang, X. Guo, W. Liu, W. Zuo, and P. Luo, "Towards photo-realistic virtual try-on by adaptively generating ↔ preserving image content," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7847–7856.

[10] A. Neuberger, E. Borenstein, B. Hilleli, E. Oks, and S. Alpert, "Image based virtual try-on network from unpaired data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5183–5192.

[11] S. Jandial, A. Chopra, K. Ayush, M. Hemani, A. Kumar, and B. Krishnamurthy, "SieveNet: A unified framework for robust image-based virtual try-on," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2020, pp. 2171–2179.

[12] T. Issenhuth, J. Mary, and C. Calauzènes, "Do not mask what you do not need to mask: A parser-free virtual try-on," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 619–635.

[13] C. Ge, Y. Song, Y. Ge, H. Yang, W. Liu, and P. Luo, "Disentangled cycle consistency for highly-realistic virtual try-on," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 16923–16932.

[14] Y. Ge, Y. Song, R. Zhang, C. Ge, W. Liu, and P. Luo, "Parser-free virtual try-on via distilling appearance flows," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8481–8489.

[15] G. Liu, D. Song, R. Tong, and M. Tang, "Toward realistic virtual try-on through landmark-guided shape matching," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 2118–2126.

[16] A. Chopra, R. Jain, M. Hemani, and B. Krishnamurthy, "ZFlow: Gated appearance flow-based virtual try-on with 3D priors," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 5413–5422.

[17] S. He, Y.-Z. Song, and T. Xiang, "Style-based global appearance flow for virtual try-on," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 3460–3469.

[18] D. Morelli, M. Fincato, M. Cornia, F. Landi, F. Cesari, and R. Cucchiara, "Dress code: High-resolution multi-category virtual try-on," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 345–362.

[19] H. Yang, X. Yu, and Z. Liu, "Full-range virtual try-on with recurrent tri-level transform," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 3460–3469.

[20] S. Bai, H. Zhou, Z. Li, C. Zhou, and H. Yang, "Single stage virtual try-on via deformable attention flows," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 409–425.

[21] S. Lee, G. Gu, S. Park, S. Choi, and J. Choo, "High-resolution virtual try-on with misalignment and occlusion-handled conditions," in *Proc. Eur. Conf. Comput. Vis.*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., 2022, pp. 204–219.

[22] I. J. Goodfellow et al., "Generative adversarial nets," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[23] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2242–2251.

[24] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," 2018, *arXiv:1809.11096*.

[25] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8107–8116.

[26] A. Cui, D. McKee, and S. Lazebnik, "Dressing in order: Recurrent person image generation for pose transfer, virtual try-on and outfit editing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2021, pp. 3935–3940.

[27] W.-L. Hsiao, I. Katsman, C.-Y. Wu, D. Parikh, and K. Grauman, "Fashion: Minimal edits for outfit improvement," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 5046–5055.

[28] S. Zhu, S. Fidler, R. Urtasun, D. Lin, and C. C. Loy, "Be your own prada: Fashion synthesis with structural coherence," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1689–1697.

[29] K. Li, M. J. Chong, J. Zhang, and J. Liu, "Toward accurate and realistic outfits visualization with attention to details," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15541–15550.

[30] R. Alp Guler, N. Neverova, and I. Kokkinos, "DensePose: Dense human pose estimation in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7297–7306.

[31] W. Chen et al., "Synthesizing training images for boosting human 3D pose estimation," 2015, *arXiv:1604.02703*.

[32] P. Guan, L. Reiss, D. Hirshberg, A. Weiss, and M. Black, "DRAPE: DRessing any PErson," *ACM Trans. Graph.*, vol. 31, 2012, Art. no. 35.

[33] C. Patel, Z. Liao, and G. Pons-Moll, "TailorNet: Predicting clothing in 3D as a function of human pose, shape and garment style," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7363–7373.

[34] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, "Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 561–578.

[35] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7122–7131.

[36] R. Danerek, E. Dibra, A. C. Oztireli, R. Ziegler, and M. H. Gross, "DeepGarment: 3D garment shape estimation from a single image," *Comput. Graph. Forum*, vol. 36, pp. 269–280, 2017.

[37] M.-H. Jeong, D.-H. Han, and H.-S. Ko, "Garment capture from a photograph," *J. Visual. Comput. Animation*, vol. 26, pp. 291–300, 2015.

[38] R. Natsume et al., "SiCloPe: Silhouette-based clothed people–supplementary materials," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4475–4485.

[39] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li, "PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2304–2314.

[40] W.-L. Hsiao and K. Grauman, "Dressing for diverse body shapes," 2019, *arXiv:1912.06697*.

[41] N. Neverova, R. A. Güler, and I. Kokkinos, "Dense pose transfer," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 128–143.

[42] A. K. Grigor'ev, A. Sevastopolsky, A. Vakhitov, and V. S. Lempitsky, "Coordinate-based texture inpainting for pose-guided human image generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12127–12136.

[43] Z. Wu, G. Lin, Q. Tao, and J. Cai, "M2E-try on net: Fashion from model to everyone," in *Proc. 27th ACM Int. Conf. Multimedia*, 2018, pp. 293–301.

[44] L. Yu, Y. Zhong, and X. Wang, "Inpainting-based virtual try-on network for selective garment transfer," *IEEE Access*, vol. 7, pp. 134125–134136, 2019.

[45] M. Chen, Y. Qin, L. Qi, and Y. Sun, "Improving fashion landmark detection by dual attention feature enhancement," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop*, 2019, pp. 3101–3104.

[46] A. Raj, P. Sangkloy, H. Chang, J. Hays, D. Ceylan, and J. Lu, "SwapNet: Image based garment transfer," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 679–695.

[47] X. Han, Z. Wu, W. Huang, M. R. Scott, and L. S. Davis, "Compatible and diverse fashion image inpainting," 2019, *arXiv:1902.01096*.

[48] H. Dong et al., "Fashion editing with adversarial parsing learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8117–8125.

[49] Y. Men, Y. Mao, Y. Jiang, W.-Y. Ma, and Z. Lian, "Controllable person image synthesis with attribute-decomposed GAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5083–5092.

[50] C.-Y. Chen, L. Lo, P.-J. Huang, H.-H. Shuai, and W.-H. Cheng, "FashionMirror: Co-attention feature-remapping virtual try-on with sequential template poses," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 13789–13798.

[51] P. S. Heckbert, “Fundamentals of texture mapping and image warping,” Jun. 1989. [Online]. Available: http://www2.eecs.berkeley.edu/Pubs/TechRpts/1989/5504.html

[52] F. Bookstein, “Principal warps: Thin-plate splines and the decomposition of deformations,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 6, pp. 567–585, Jun. 1989.

[53] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, “Spatial transformer networks,” in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.

[54] D. Ji, J. Kwon, M. McFarland, and S. Savarese, “Deep view morphing,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7092–7100.

[55] C.-H. Lin, E. Yumer, O. Wang, E. Shechtman, and S. Lucey, “ST-GAN: Spatial transformer generative adversarial networks for image compositing,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9455–9464.

[56] H. Dong, X. Liang, B. Wang, H. Lai, J. Zhu, and J. Yin, “Towards multi-pose guided virtual try-on network,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9025–9034.

[57] A. Vaswani et al., “Attention is all you need,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[58] P. Shaw, J. Uszkoreit, and A. Vaswani, “Self-attention with relative position representations,” in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2018, pp. 464–468.

[59] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov, “Transformer-XL: Attentive language models beyond a fixed-length context,” in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 2978–2988.

[60] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, “Convolutional sequence to sequence learning,” in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 1243–1252.

[61] G. Borgefors, “Distance transformations in digital images,” *Comput. Vis. Graph. Image Process.*, vol. 34, pp. 344–371, 1986.

[62] R. Yu, X. Wang, and X. Xie, “VTNFP: An image-based virtual try-on network with body and clothing feature preservation,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 10510–10519.

[63] A. H. Raffiee and M. Sollami, “GarmentGAN: Photo-realistic adversarial fashion transfer,” 2020, *arXiv:2003.01894*.

[64] J. Wang, W. Zhang, W.-H. Liu, and T. Mei, “Down to the last detail: Virtual try-on with detail carving,” 2019, *arXiv:1912.06324*.

[65] H. Dong, X. Liang, X. Shen, B. Wu, B.-C. Chen, and J. Yin, “FW-GAN: Flow-navigated warping GAN for video virtual try-on,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1161–1170.

[66] D. Song, T. Li, Z. Mao, and A. Liu, “SP-VITON: Shape-preserving image-based virtual try-on network,” *Multimedia Tools Appl.*, vol. 79, pp. 33757–33769, 2020.

[67] R. HyugJae, M. Lee, M. KangCho, and G. Park, “LA-VITON: A network for looking-attractive virtual try-on,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop*, 2019, pp. 3129–3132.

[68] K. Ayush, S. Jandial, A. Chopra, and B. Krishnamurthy, “Powering virtual try-on via auxiliary human segmentation learning,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, 2019, pp. 3193–3196.

[69] W. Sun and Z. Chen, “Learned image downscaling for upscaling using content adaptive resampler,” *IEEE Trans. Image Process.*, vol. 29, pp. 4027–4040, 2019.

[70] B. Niu et al., “Single image super-resolution via a holistic attention network,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 191–207.

[71] S. Anwar and N. Barnes, “Densely residual laplacian super-resolution,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1192–1204, Mar. 2022.

[72] S. Choi, S. Park, M. Lee, and J. Choo, “VITON-HD: High-resolution virtual try-on via misalignment-aware normalization,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 14126–14135.

[73] P. Li, Y. Xu, Y. Wei, and Y. Yang, “Self-correction for human parsing,” 2019, *arXiv:1910.09777*.

[74] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, “Convolutional pose machines,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4724–4732.

[75] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, “OpenPose: Realtime multi-person 2D pose estimation using part affinity fields,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 172–186, Jan. 2021.

[76] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, “Hand keypoint detection in single images using multiview bootstrapping,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4645–4653.

[77] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, “Realtime multi-person 2D pose estimation using part affinity fields,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1302–1310.

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

[79] E. Jang, S. Gu, and B. Poole, “Categorical reparameterization with gumbel-softmax,” in *Proc. Int. Conf. Learn. Representations*, 2017.

[80] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, “High-resolution image inpainting using multi-scale neural patch synthesis,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4076–4084.

[81] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, “Image inpainting for irregular holes using partial convolutions,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 89–105.

[82] J. Yu, Z. L. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, “Generative image inpainting with contextual attention,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5505–5514.

[83] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, “Free-form image inpainting with gated convolution,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 4470–4479.

[84] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 694–711.

[85] C. Xiaopeng, C. Jiangzhong, L. Yuqin, and D. Qingyun, “Improved training of spectral normalization generative adversarial networks,” in *Proc. 2nd World Symp. Artif. Intell.*, 2020, pp. 24–28, doi: 10.1109/WSAI49636.2020.9143310.

[86] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

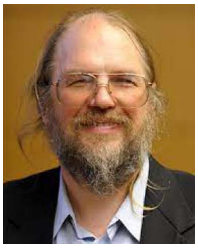
[87] T. Salimans et al., “Improved techniques for training GANs,” in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.

[88] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs trained by a two time-scale update rule converge to a local nash equilibrium,” in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6629–6640.

[89] M. J. Chong and D. Forsyth, “Effectively unbiased fid and inception score and where to find them,” 2019, *arXiv:1911.07023*.

Kedan Li received the BS degree in computer science from the University of Illinois at Urbana-Champaign. He is currently working toward the PhD degree with the University of Illinois at Urbana-Champaign. He is the founder / CEO of Revery AI Inc. He is an expert in Artificial Intelligence applications in fashion and spent the last four years performing research in relevant domains. Before that, he is a serial entrepreneur and worked as a software engineer with Fitbit.

Jeffrey Zhang received the bachelor’s degree in computer science from the University of California, Berkeley. He is currently working toward the PhD degree with the University of Illinois at Urbana-Champaign studying computer vision. He is the co-founder/COO of Revery AI Inc. His research interests lie in virtual try-on, representation learning, continual learning, and applications of machine learning (specifically medical applications of machine learning).



David Forsyth (Fellow, IEEE) received the BSc and MSc degrees in electrical engineering from the University of the Witwatersrand, Johannesburg, and the MA and DPhil degrees from Oxford University. He is Fulton-Watson-Copp chair in computer science with U. Illinois at Urbana-Champaign. He has published more than 170 papers on computer vision, computer graphics and machine learning. He has served as program co-chair or as general co-chair for numerous major computer vision conferences. He became an ACM fellow, in 2014. His textbook, “Computer Vision: A Modern Approach” (joint with J. Ponce and published by Prentice Hall) is widely adopted as a course text. A further textbook, “Probability and Statistics for Computer Science”, came out two years ago; yet another (“Applied Machine Learning”) has just appeared. He has served two terms as editor in chief, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, serves on a number of scientific advisory boards, and has an active practice as an expert witness.